Effects of Foveation on Early Visual Representations

by

William F. Broderick

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Center for Neural Science

New York University

May, 2022

Dr. Eero P. Simoncelli

Dr. Jonathan Winawer

…In that Empire, the Art of Cartography attained such Perfection that the map of a single Province occupied the entirety of a City, and the map of the Empire, the entirety of a Province. In time, those Unconscionable Maps no longer satisfied, and the Cartographers Guilds struck a Map of the Empire whose size was that of the Empire, and which coincided point for point with it. The following Generations, who were not so fond of the Study of Cartography as their Forebears had been, saw that that vast map was Useless, and not without some Pitilessness was it, that they delivered it up to the Inclemencies of Sun and Winters. In the Deserts of the West, still today, there are Tattered Ruins of that Map, inhabited by Animals and Beggars; in all the Land there is no other Relic of the Disciplines of Geography.

—Suarez Miranda, Viajes de varones prudentes, Libro IV, Cap. XLV, Lerida, 1658

— Jorge Luis Borges, "Del rigor en la ciencia"

Dedication

To Anna, you truly are an amalgam.

Acknowledgements

A PhD is a long, strange, and difficult process, especially during a global pandemic. I am very thankful for those who helped make it as enjoyable and fulfilling as it was. First and foremost, thank you to my advisors, Jon Winawer and Eero Simoncelli, for teaching me to think critically and communicate clearly, for supporting my interest in open science, and providing a supportive environment to learn. I would also like to thank my manager during my internship at Facebook Reality Labs, Gizem Rufo, for constantly pulling me back from the minutiae to keep the important questions in view, even as the time required for the project far exceeded our predictions; her mentorship was the best part of that internship. I am grateful to my committee, Tony Movshon and David Heeger, for their feedback over the years. The work presented here is far stronger for their input.

Thank you to everyone at Facebook Reality Labs: Trisha Lian, Scott Murdison, Minjung Kim, Ian Erkelens, Romain Bachy, Kevin Rio, the late Kevin Mackenzie, Todd Bell, Anjul Patney, and especially the other interns, Alex Göttker, Nilsu Atilgan, Lili Zhang, and Angela Rădulescu. While at NYU, I have a tendency to not think about the fact that eyes move or people see in color, so being part of a broader team of vision scientists, with a variety of expertise, was a great learning opportunity.

Thank you to the members of the Winawer lab over the years: Noah Benson, Eline Kupers, Iris Groen, Stephanie Montenegro, Serra Favila, Marc Himmelberg, Jan Kurzawski, Ilona Bloem, Rania Ezzo, Ekin Tünçok, Rob Woodry, and Jiyeong Ha. And to the members of the Lab for Computational

Vision: Alex Berardino, Olivier Hénaff, Jimmy Wang, Manu Rhagavan, Tim Oleskiw, Jing Yang Zhou, Caroline Haimerl, Colin Bredenberg, Sreyas Mohan, Zahra Kadkhodaie, and Hope Lutwak. I'd like to especially thank Corey Ziemba, Paul Levy, and Kate Bonnen, for always answering my questions about physiology and psychophysics — the mingling of theory-minded vision scientists with a variety of focuses is one of the reasons LCV was such a great place to do a PhD. I'd like to further thank the members of LCV who worked on plenoptic: Kate, Lyndon Duong, Pierre-Étienne Fiquet, Nikhil Parthasarathy, Teddy Yerxa, and Xinyuan Zhao. Working together on a shared code base was a humbling learning experience, and what we have created is better for everyone's involvement.

My time at NYU was made much more meaningful for the unofficial and semiofficial groups I took part in, especially ScAAN and philosophy journal club. Thank you to Will Adler, Jenn Lee, Ionatan Kuperwajs, and Weiji Ma, for keeping ScAAN running, and to Maija Honig, Julian Saliani, Colin Bredenberg, Jenn, and Kate Bonnen for the work we did together on the waterfront map and Maryland transit project. To the members of philosophy journal club, Jenn, Colin, Zhiwei Li, David Barack, and Jess Thompson, struggling through philosophy of science books and articles with you was some of my favorite extra-curricular reading. I am also thankful to Open Life Science and Software Carpentry, which taught me a lot about how to run open science projects and teach technical skills.

Thank you to all the grad students and postdocs who took part in my experiments: I am forever indebted to your willingness to stare at blinking grayscale images and press some buttons.

Thank you to the administrative staff at CNS for keeping everything working, especially Amala Ankolekar-Hinge, Erik Cruz, Jess Holman, and Heather McKellar.

To those who directly assisted in my research: thank you to Noah Benson for his assistance with the retinotopy analysis in the second chapter, and Eline Kupers for returning to the scanner too many times to help me measure the projector's modulation transfer function. Thank you to Kendrick Kay for including my logpolar grating stimuli in the Natural Scenes Dataset, allowing us

to see how reliable our results are and how well they account for responses to the other images, and to Jiyeong Ha for analyzing that data. Thank you to David Brainard for the use of his photos in the third chapter, both from the published UPenn Natural Image Database ([209]) and the unpublished set of images from around Philadelphia. Thank you to Pablo Velasco, Keith Sanzenbach, Valerio Luccio, and the rest of NYU's Center for Brain Imaging for training and help with the fMRI data collection. All of the work presented here would have been literally impossible without the use of a computing cluster, so I would like to thank the NYU High Performance Computing team, especially Shenglong Wang, as well as the Flatiron Institute's Scientific Computing Core, especially Nick Carriero and Robert Blackwell, for their often thankless work maintaining their clusters, as well as the direct help they gave me in how to use their respective clusters effectively. Thank you also to everyone who read and provided feedback on this thesis: Caroline, Kate, Nikhil, Vicky, Talya Cooper, and Corey. Finally, I got assistance from a lot of people when it came time to share the data, code, and computational environments for the work presented here, a topic dear to my heart, and I am incredibly grateful for their help. They include Shenglong Wang, Vicky Rampin, Rémi Rampin, Pablo Velasco, Deb Verhoff, Kate Pechekhonova, Nick Wolf, Liz Lovero, Aaron Watters, and Dylan Simon.

To everyone outside of science: thank you for keeping me going. My family, for supporting me on this journey from the before I knew what doing a PhD entailed and providing me with plenty of good food and drink whenever we're together. The Dardicks, for welcoming me into the family. All the friends I've played board games and tabletop RPGs throughout the years, especially Somya Chhajlani, Patrick Monari, Nathaniel Culver, Nick Thompson, Michael Jigo, Antonio Fernandez, A. Licata, Zhiwei Li, Bas van Opheusden, Maija Honig, Henry Towbin, and Natalie Kaplan. Getting together to play make believe for a couple of hours every week was a great way to recover from sciencing all day. And finally, my wife Anna Dardick — this would not have been possible without you.

Preface

Chapter 2 has been published as [29]. Chapter 3 presents a project done in collaboration with Gizem Rufo, which I started while working as an intern at Facebook Reality Labs with her. Chapter 4 describes a software package written in collaboration with Kate Bonnen, Lyndon Duong, Pierre-Étienne Fiquet, Nikhil Parthasarathy, Teddy Yerxa, and Xinyuan Zhao, all graduate students or postdocs in Eero Simoncelli's lab.

Abstract

Human vision is far from uniform across the visual field. At fixation, we have a region of high acuity known as the fovea, and acuity decreases with distance from the fovea. However, it is not true that peripheral vision is just a blurrier version of foveal vision, and finding a precise description of how exactly they differ has been challenging. This thesis presents two investigations into how the processing of visual information changes with location in the visual field, both focused on the early visual system, as well as a description of a software package developed to support studies of the type found in the second study. In the first study, we use functional magnetic resonace imaging (fMRI) to measure how spatial frequency tuning changes with orientation and visual field location in human primary visual cortex (V1). V1 is among the best-characterized regions of the primate brain, and we know that nearly every neuron in V1 is selective for spatial frequency and orientation. We also know that V1 neurons' preferred spatial frequencies decrease with eccentricity, which aligns with the decrease in peak spatial frequency sensitivity found in perception. However, precise descriptions of this relationship have been elusive, due to the difficulty of characterizing tuning properties across the whole field. By utilizing fMRI's ability to measure responses across the entire cortex at once to a set of stimuli designed to efficiently map spatial frequency preferences, along with a novel analysis method which fits the responses of all voxels simultaneously, we present a compact description of this property, providing an important building block for future work.

In the second study, we build perceptual pooling models of the entire visual field from simple

filter models inspired by retinal ganglion cells and V1 neurons. We then synthesize a large number of images to investigate how the sensitivities and invariances of these models align with those of the human visual system. This allows us to investigate to what extent the change in perception across the visual field can be accounted for by well-understood models of low-level visual processing, rather than requiring more cognitive phenomena or models with millions of parameters. Finally, I describe an open-source software package developed by members of the Simoncelli lab that provides four image synthesis methods in a shared, general framework. These methods were all developed in the lab over the past several decades and have been described in the literature, but their widespread use has been limited by the difficulty of applying them to new models. By leveraging the automatic differentiation built into a popular deep learning library, our package allows for the use of synthesis method with arbitrary models, providing an important resource for the vision science community. Altogether, this thesis presents a step forward in understanding how visual processing differs across the visual field and, with the effort to share the code, data, and computational environment of the projects, provides resources for future scientists to build on.

Contents

List of Figures

List of Tables

1 Introduction

Human vision is far from uniform across the visual field. Humans have a central region of high acuity, the fovea, which we move constantly and rapidly around visual scenes in order to acquire information. This is a feature we share with other primates, carnivores, many birds (some species of which, such as hawks, have two foveas, [132]), and jumping spiders (though the anatomical implementation is vastly different, [146]). Not all animals, however, share this feature. Many mammals, such as rabbits and ungulates, have a visual streak: their region of highest retinal ganglion cell density is a horizontal strip in the center of the retina [132]. Rodents, despite being closely related to humans, lack a fovea (though they have an area centralis, with slightly increased retinal ganglion cell density, [132]). This diversity suggests that the fovea has evolved multiple times over the course of evolution, and that the spatial sampling of the light that enters the eye has important consequences for behavior.

This thesis presents two investigations into how the processing of visual information changes across the visual field, focusing on the early visual system, that is, the stages after the photoreceptors and before the secondary visual cortex (V2). These stages, including retinal ganglion cells (RGCs) and primary visual cortex (V1), are among the best characterized of the primate central nervous system, with decades of study into how they respond to visual input using a variety of techniques. However, much of the foundational work comes from single-unit electrophysiology, the use of which for characterizing visual field maps was described by David Hubel as "a dismaying exercise in tedium, like trying to cut the back lawn with a pair of nail scissors" [103]. We thus have

a much less complete picture of how processing in these areas changes across the visual field than one might hope. In the second chapter of this thesis, I use functional magnetic resonance imaging (fMRI) to investigate how spatial frequency tuning varies with population receptive field (pRF) location, making use of the fact that both these properties vary smoothly across the cortical map and are thus well suited for fMRI, with which we can gather the responses of the full cortical map to a full-field stimulus. In the third chapter, I extend upon earlier work which uses computational models and behavioral experiments to estimate the spatial scale of pooling of image statistics [7, 70, 219], investigating in more detail the information discarded in the beginning of the visual system. The final chapter discusses an open-source software package, `plenoptic`, that has been developed to support studies like those found in the third chapter, as well as the importance of open-source software to science more generally.

1.1 A brief history of models of the early visual system

The work presented in this thesis builds off a long literature in the visual sciences, which built up the classic models of neurons in the early visual system from linear receptive fields, adding non-linearities to account for additional phenomena. The concept of receptive fields begins with Sherrington [194], who used the term to describe the patch of skin that elicits muscle reflex in dogs when touched. Hartline [87] brought this concept into visual neuroscience with his studies of the frog retina, defining it as "the region of the retina which must be illuminated in order to obtain a response in any given [optic] fiber". By 1953, Kuffler [129] was using an expanded version of the concept, which "include[d] all areas in *functional* connection with a ganglion cell" (emphasis original). Kuffler [129] emphasized that, while the anatomical configuration of a receptive field (the actual receptors connected to a ganglion cell) are fixed, this functional receptive field depends upon the experiment which defines it, especially the properties of the stimulus and the state of the cell's light adaptation. This conceptual shift demonstrates the field's focus on a computational

or functional account of neural activity, rather than an anatomical or biophysical one

This functional focus can also be seen by the field's long investigation into receptive field linearity. A linear shift-invariant (LSI) system can be completely characterized by its impulse response, making it exceedingly computationally tractable: if neuronal responses were both linear and shift-invariant, the response of a given neuron to a single stimulus would allow neuroscientists to predict its response to any stimulus they could present. There's a variety of work through the 1950s and 1960s on these questions, culminating in the investigation by Enroth-Cugell and Robson [64], which split cat RGCs into two classes: approximately linear (X) and very non-linear (Y; in particular, their response to drifting gratings did not vary with phase in a linear manner). These studies do not investigate *how* linear responses are achieved, and linearity's mathematic simplicity belies biological complexity in its implementation. But by this point, a simple standard model of RGCs was forming: a linear receptive field consisting of concentric rings of excitatory and inhibitory regions, referred to as the "on" center and "off" periphery, or vice versa. This linear model fails to capture some nonlinearities [193, 216], the most obvious of which is the spike threshold: neuronal firing rate cannot go negative, and thus a simple static nonlinearity (rectification) is applied to clip negative responses [45].

Similar investigations into linearity were also carried out in the lateral geniculate nucleus (LGN) of the thalamus and the primary visual cortex, the next two steps in the visual pathway. Hubel and Wiesel [104] first investigated the receptive fields of cat V1, mapping their separate inhibitory and excitatory areas, explicitly referencing the earlier work of Kuffler [129] in the cat retina. They classified the cells they characterized into two categories, "simple" and "complex": "these fields were termed 'simple' because like retinal and geniculate fields (1) they were subdivided into distinct excitatory and inhibitory regions; (2) there was summation within the separate excitatory and inhibitory parts; (3) there was antagonism between excitatory and inhibitory regions; and (4) it was possible to predict responses to stationary or moving spots of various shapes from a map of the excitatory and inhibitory areas." [105]. A neuron which failed any of these four parts

was classified as "complex". The last point, though it does not say so explicitly, is about the cell's linearity: if a cell's response is linear, a complete map of the receptive field is sufficient to predict its response to arbitrary stimuli. These papers from Hubel and Wiesel also demonstrated that, unlike RGCs and LGN neurons, both simple and complex cell receptive fields are oriented: in Hubel and Wiesel's Nobel prize-winning experiments, V1 neurons responded most strongly to oriented bars of light, demonstrating an orientation tuning that would be further characerized in later studies. By 1978, Movshon and colleagues were characterizing the primary difference between simple and complex cells with reference to phase sensitivity, the now standard method for classifying cells as simple or complex (e.g., [81]): complex cell responses "vary little in amplitude or wave form as the spatial phase of the grating is varied" [152], while simple cells "chang[e] their response amplitude sinusoidally as the spatial phase of the grating is changed" [153].

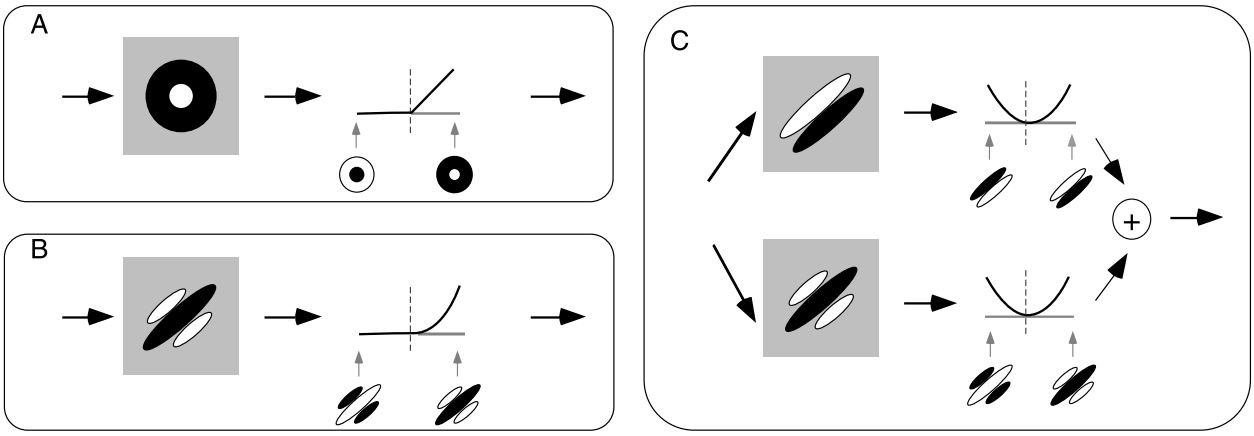


Figure 1.1: Schematic showing basic models of neurons in the early visual system. Each model starts by taking the dot product of a linear filter with an input image, followed by a static nonlinearity. (A) Retinal ganglion cell and lateral geniculate nucleus model. The receptive field is unoriented, with an excitatory center and an inhibitory surround (or vice versa). The static nonlinearity is a rectification: all negative outputs are set to 0. (B) V1 simple cell model. The receptive field is oriented and bandpass, tuned for both orientation and spatial frequency. The static nonlinearity is also a rectification, though the displayed one is a nonlinear rectification. (C) V1 complex cell model. This model squares and sums the output of two simple cell-like subunits whose phases are offset by 90 degrees to compute the local energy, selective for orientation and spatial frequency, but insensitive to phase. Unlike the RGC and simple cell models, the rectification here is "full-wave" (rather than discarding the negative outputs of the linear filter), which is equivalent to taking the half-wave rectified outputs of the sign-flipped version of the subunit filter, as shown underneath the nonlinearity schematic. Adapted from [37]

These two papers from Movshon and colleagues also appear to be the first to put models of V1

simple and complex cells to quantitative test: "None of these [previous] groups used the powerful technique of Fourier analysis to interpret their data, or to relate neuronal responses to gratings with those to simpler types of geometric stimuli." [153]. The figures in these papers therefore include not only summaries of neuronal responses, but also lines showing the predictions of implementations of the verbal models informing this line of research, showing clearly where neural data matched or diverged from model predictions. This line of work, along with the work on RGCs described earlier, led to the crystallization of basic models of neurons in the early visual system, shown in figure 1.1. All these models start by passing the input image through a linear filter, then applying a static nonlinearity. These are the only steps for models for RGCs (A, [64]) and V1 simple cells (B, [152]), which differ primarily in the structure of their receptive fields (unoriented vs. oriented), while the complex cell model (C) sums the squared outputs of two subunits, represented by a quadrature pair of simple cell-like filters (with matched orientation and spatial frequency selectivity) to get the "energy model", which is phase-insensitive [1].

1.2 On the importance of computational models

Having computational models is invaluable for scientific research: they are tools to help scientists think better [185], require us to think more deeply about the system that we are modeling [84], and force us to make our assumptions explicit [200]. As Smaldino [200] put it, "models are, by and large, stupid…[yet] stupid models are extremely useful. They are useful because humans are boundedly rational and because language is imprecise." This "stupidity" is a feature, not a bug: implementation removes the ambiguities found in verbal descriptions and provides the scientist with specific predictions to test against. In models of the early visual system, early work focused on whether the system was linear and, because linear systems are well-defined and well-understood, this was fairly straightforward to do even without an explicit model. Pretty quickly, however, researchers noticed nonlinearities in the V1 neurons beyond

those accounted for by rectification or the energy model, including response saturation [4] and nonspecific suppression [21]. Computational models allow researchers to ensure that these phenomena are not accounted for by existing models and, when an improved model comes along, to see how many such phenomena it can account for. The divisive normalization model proposed by Heeger [88] accounted for many such nonlinear behaviors, unifying disparate-seeming phenomena. Such an improvement requires implementing and testing a model: by reasoning alone, it is far from obvious that adding division to the basic models outlined in figure 1.1 would predict such behavior. This speaks to a broader point: even relatively simple models have surprising behaviors. Sometimes, as in this case, these are encouraging, and sometimes they are disappointing, but in either case we understand the models better for having encountered them.

Despite their effectiveness in the study of perception, the theory-first approach described above, exemplified by the field's focus on linearity and computational models, is not universally valued in neuroscience and psychology. There are many who espouse a naive empiricism, which holds that "scientific 'facts' can be derived from observation or empirical tests, independent of theoretical commitments" [185]. These include such luminaries as Santiago Ramon y Cajal, who advised young scientists that: "A scholar's positive contribution is measured by the sum of the original data that he contributes. Hypotheses come and go but data remain. Theories desert us, while data defend us" [181] (this position seems somewhat ironic given Ramon y Cajal's position as a key figure in the debate over the neuron doctrine, one of the earliest theoretical debates in neuroscience), and György Buzsáki, who suggests that neuroscience should "start with the brain (independent variable) and define descriptors of behavior (dependent variables) that are free from philosophical connotations and can be communicated across laboratories, languages, and cultures." [34]. Gershman [76] (from whence the previous two examples are drawn) characterizes this viewpoint as a belief in the scientist's "innocent eye", wherein "if one *just looks* at the data, then facts can be documented and progress can be made."

This view seems particularly misplaced in the context of the line of research presented earlier

in this section. An open question hanging around the edge of these early results is how we should then understand visual neurons: what is it that they *do*? The two main schools of thought can be summarized as "features" vs. "filters": are visual neurons detectors of certain features in the world (features which can be understood and simply described by humans) or are they better understood as filters with some extra non-linearities? As De Valois, Albrecht, and Thorell [55] summarize in their aptly-titled paper: "[Hubel and Wiesel [105]] describe the optimal stimulus for cortical cells as being an elongated light or dark bar of a particular width and orientation, or a sharp, correctly oriented edge between light and dark. Their descriptions of optimal stimuli in such semi-naturalistic terms…led others to develop theories of pattern perception in which a complex pattern would be dissected into simple units such as bars of particular widths, and edges; these units would then be combined into more complex combinations of bars and edges. Such an approach seems intuitively reasonable and was readily accepted." On the other hand, "[Campbell and Robson [35]] proposed that the visual system analyzed spatial variation in light in terms of spatial frequency content of the pattern, rather than in terms of more naturalistic features such as edges…In essence it was proposed, the visual cortex would behave in a manner similar to the cochlea in the auditory system — as a crude Fourier analyzer." Adelson and Bergen described this distinction as between "stuff" and "things", when they assert their preference for the filter view by saying "we are interested in how early vision measures 'stuff' rather than in how it labels 'things'." [2], though allegiance is rarely stated so explicitly. Most often, researchers' preference for one of these views lurks in the background, informing the questions they find interesting, the stimuli they present, and their interpretation of the results.

These differing views demonstrate, at the very least, that different researchers, trained in the same discipline and working at the same time, can look at the same data and arrive at very different interpretations. That is, they demonstrate that the facts do not emerge from simple examination of the data. And these differing interpretations led researchers to conceptualize of the visual system in different ways. As De Valois, Albrecht, and Thorell [55] note above, the feature view led to

hypotheses that had edges serving as the most basic unit of visual perception, combining them into more and more complicated forms until the emergence of objects represented in cortical areas at the end of the ventral stream (Marr [143] provides one version of this). The filter view, on the other hand, invites researchers to think of subsequent stages performing similar filtering operations, taking differences (i.e., derivatives) across the dimensions of earlier filter selectivities (e.g., orientation or scale), as presented in Adelson and Bergen [2].

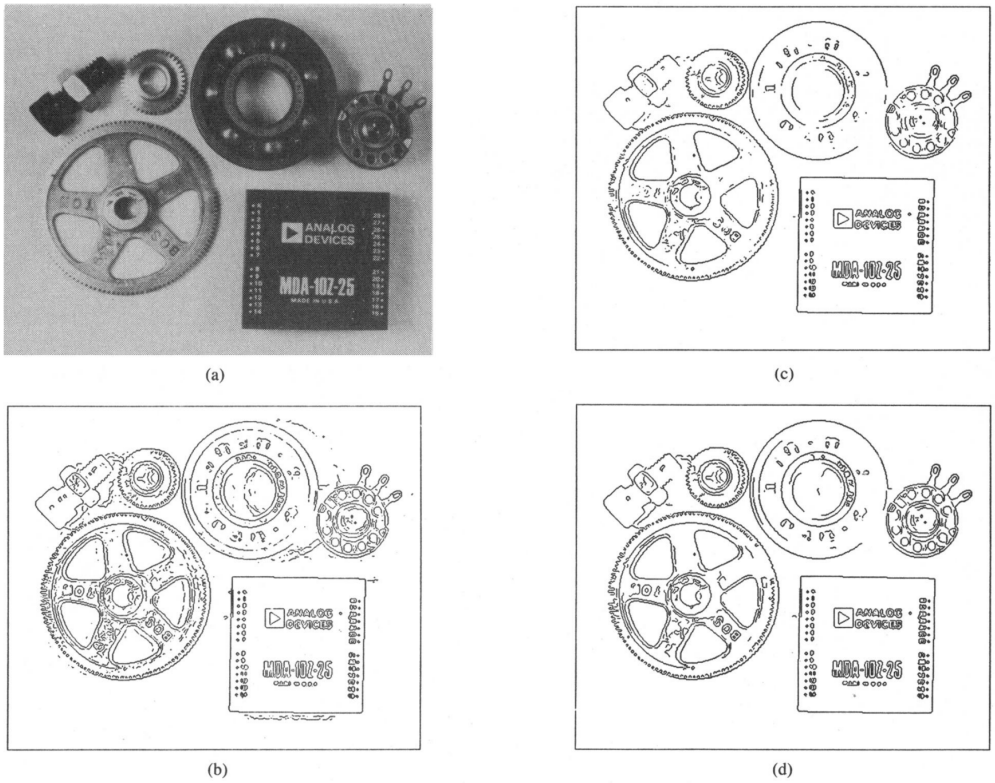


Figure 1.2: Illustration of edge detector performance from Canny [36], figure 7. (a) shows the original image, (b) and (c) the outputs of the proposed edge detection algorithm thresholded at two different values, and (d) the outputs thresholded with hysteresis using both the thresholds in (b) and (c). All three proposed edges maps include edges in the reflection of the metal and shadow while missing boundaries between the objects and the background (nut and bolt on the top left). No human comparison is present, but it seems unlikely that humans would assign edges in the same way. I posit that "edges" is more likely to be a higher-order property, such as object boundaries, rather than a low-level image property. The computer vision field's difficulty in developing a reliable algorithm for edge detection supports this possibility.

Even if we restrict ourselves to V1, the filter interpretation is more useful. The outputs of a filter or filter with some nonlinearities are well-understood, even to stimuli far from the test set,

with a long literature in signal processing investigating their properties. It is less clear what a "feature detector" should do in more general situations. How should an edge detector respond to a curved line, which contains multiple orientations? How should it respond to a two-dimensional sine wave grating, which has no hard edges? Calling a V1 neuron an edge detector does not clarify matters. Additionally, no one has an implementation of an edge detector as such. There is a long history of edge detection in computer vision, and most algorithms use some bandpass filter selective to high frequencies. That is, the "edge detectors" are actually just the sort of filters described for V1 neurons above, and their outputs do not align with the edges that humans assign to images. Human-assigned edges seem to be hierarchical and a property of objects, rather than a low-level image property one could use to build objects: as can be seen in figure 1.2, edge detectors often highlight a variety of edges that humans consider unimportant, such as reflection patterns and shadows, while occasionally missing those considered important, like edges that separate an object from the background, often in areas of low contrast (a psychophysical experiment to validate this is necessary). If we conceptualize V1 neurons as filters, rather than feature detectors, we thus have a more useful frame for understanding the early visual system.

The distinction between the two perspectives may seem rather small, especially when compared to the rest of neuroscience. After all, one could say that both give rise to models that are far too simple: they are feedforward, with no recurrence or feedback, they often ignore or assume the linearity of the temporal dimension (though not always), and they ignore completely the question of biological implementation. As Heeger, Simoncelli, and Movshon [89] point out, our knowledge of the circuitry is nowhere near the level of detail required to build any sort of "biologically accurate" model without a large number of assumptions. But these models do ignore a good deal of the knowledge we do have about the brain, and, perhaps unintuitively, the fact that they do so is a strength. These simplifications are what the philosopher of science Angela Potochnik calls **idealizations**: "assumptions made without regard for whether they are true and often with full knowledge they are false…we artificially simplify the parts of accounts that we aren't interested

in to improve our access in a variety of ways to the parts we are interested in" [179]. These filter-based models typically ignore many components of cortical processing, such as feedback, to focus on what can be accomplished with simple filtering and nonlinearities. As neuroscientists and psychologists have a tendency to say a surprising result requires some poorly-understood high-level process, when, in actuality, a low-level process suffices, this is an important benefit. Bergen and Adelson [18] provides a simple example with texture perception, demonstrating that knowledge of features such as terminators, corners, and intersections are unnecessary to explain human performance; a linear center-surround filter followed by full-wave rectification accounts for human discriminability on patterns of randomly oriented Xs, Ts, and Ls. The use of idealized models pushes back against this tendency among researchers to assume something higher-level, more "cognitive", or more "human" must be required to explain patterns of behavior.

This thesis takes the position that such simple models can be used fruitfully and that, despite their long history, their usefulness has not been expended. We should take them seriously, push them as far as they can go, and seek to understand their predictions and their foibles. The third chapter takes the energy model of V1 discussed above and uses it as the basis for a psychophysical model of the whole visual field. How well does this model align with human perception? In what ways does the model's sensitivites and invariances align with those of the human visual system and in what ways do they diverge? What does this imply about our understanding of the early visual system? This simple model provides a handle onto these questions and thus proves a useful tool for understanding the visual system.

1.3 The change of visual processing with eccentricity is poorly characterized

Another major theme of this thesis is how the processing of visual information changes across the visual field. The work described in the first section of this introduction, which serves as the

basis for the models discussed in the following chapters, was all performed using single unit electrophysiology. While this method has many benefits, a broad sampling of eccentricities is not among them. Studies that look across many eccentricities do exist (e.g., [74]), but owing to their increased difficulty, they are not as common and thus not all properties have been characterized.

In particular, spatial frequency selectivity has not been characterized across the visual field. In the basic V1 models discussed in the previous section, the first step involves a bandpass, oriented linear filter, which models the fact that the majority of V1 simple and complex cells are tuned for both orientation and spatial frequency. Thus, characterizing V1 neurons' spatial frequency selectivity is important. We know neuronal receptive field size increases with eccentricity, and there are a variety of behavioral performance measures whose performance "gets larger" with eccentricity (e.g., acuity decreases, crowding distance increases), which suggest that spatial frequency preferences shift to lower spatial frequencies as eccentricity grows larger. There is some evidence of this from physiology (e.g., De Valois, Albrecht, and Thorell [54] show lower peak spatial frequency for parafoveal than foveal cells), but with too limitated a range of eccentricities to get a sense for what this relationship looks like. As this property likely changes smoothly, fMRI, which measures responses from the entire brain at once, is well-suited to the task, but existing studies mostly viewed it as an aside to their main research question and found wildly different values.

If we would like to create cumulative knowledge, able to build on each other's results to improve our understanding, a vague sense that "spatial frequency preferences decrease" is not enough. Indeed, the work presented in my second chapter was inspired by an attempt to build a model of fMRI responses in V1, which would require knowing what this relationship looked like, and our resulting surprise that we could find no such data in the literature. The second chapter of my dissertation examines how spatial frequency tuning changes as a function of visual field location and stimulus orientation, hopefully providing a jumping off point for future models or for comparisons with electrophysiological data of the same.

1.4 The incentives do not promote cumulative science

The final chapter of this thesis is not an empirical study. Instead, it presents a software package, `plenoptic`, that I have worked on over the past several years with other members of the Laboratory for Computational Vision (LCV), discussing how it can be used for experiments and why it is useful, as well as arguing for the importance and under-appreciation of open software in academic science. The package facilitates the use of several stimulus synthesis methods that have been developed by the lab over the years. These methods enable researchers to better understand their computational models, helping them explore stimulus space to get a sense for the model's sensitivities and invariances. This approach has a long history in vision science, dating back to the color-matching experiments of the 19th century, but is less common in the field today. The goal of `plenoptic` is to provide reliable, general implementations of these methods, allowing more researchers to apply them to their own models and to carry out experiments like the one presented in the third chapter.

Importantly, nothing present in the package (or in that chapter) is scientifically novel: the methods and models have all been described in the literature before, and we make use of an existing python package (`pytorch`, [168]) to perform the automatic differentiation that enables our general implementation. This was a conscious decision on the part of the developers, which we stand by for several reasons. A particularly relevant one is that, like most academic labs, LCV has trouble with knowledge loss, "the frequent turnover of researchers…[ensures] capturing and retaining knowledge is a continuous struggle" [147]. Each of the synthesis methods included in the package were the result of a former graduate student or postdoc's research, and the software that supported the original publication was not intended or suitable for broader use. `plenoptic` serves as a central repository of knowledge about these methods, enabling future students and postdocs to not only use the software, but also to benefit from our hard-won expertise through the tests, documentation, and comments that accompany the code.

However, this decision stands in contrast with the focus of academic science (and those of the industrial research groups that have become increasingly involved in the machine learning community), where the incentives are to publish as many novel findings as possible (see [160] for a discussion of this in psychology). These incentives have begun to be questioned among the scientific community more broadly with the increasing awareness of how they have contributed to an alarmingly large number irreproducible results across the sciences (e.g., [46]), but it is worth stepping back to consider these how these misaligned incentives affect science beyond the single issue of reproducibility.

Most scientists view their actions as increasing humanity's store of knowledge, as increasing our understanding of the world, an endeavor which requires, in Newton's classic turn of phrase, "standing on the shoulders of giants" [144]. That is, we wish to be part of a cumulative science. Yet the incentives for individual researchers do not promote this. Publications are not sufficient for reproducibility, regardless of whether reproduction is the end goal or is the starting point for an extension or comparison. In an anecdote which will sound familiar to any researcher who has tried to implement a model or a novel method based solely on the description found in a paper, Topalidou et al. [210] describe the difficulty they had in reproducing a model of the basal ganglia from the literature, which eventually required three months of work and collaboration with the original authors. This is clearly insufficient if we wish to build on the work of other researchers, yet publications are the currency of academic science. Van Dijk, Manor, and Carey [213] found that the most predictive factors for whether a researcher becomes a principal investigator (PI) are their number of publications, the impact factor of the journals in which they published, and the number of papers that received more citations than average for the journal. In Tregoning and McDermott [211], the authors' number two rule for how to become a PI is to publish papers (only "have ideas" ranks as more important), and they advise early career researchers not to "start any work unless you can see the route to publication." Nowhere do they discuss steps a researcher can take to ensure their colleagues can make use of their results. This aligns with the sense among

junior researchers I have talked to that the most important criterion on which a prospective PI will be judged is their publication record, and the importance many scholars place on the idea of finding "the best journal" (i.e., the one with the highest impact factor) that will accept their paper.

If one wishes to go beyond publication, whether that's sharing data or ensuring their code can be run by others, this will take time which the researcher could have used to work on other publications. Prof. Russ Poldrack, well-known in the neuroimaging field for his advocacy of open practices and his support of a variety of open-source tools, says "in my discussions with ECRs, I also try not to sugarcoat the fact that some of the remedies [to increase reproducibility] we advocate are likely to make them less competitive on the job market in the short term... the current incentives toward a large quantity of high-impact publications cut directly against this kind of integrity" [174]. We have not, as a field, decided that these efforts are worth rewarding, even if many scientists support them in principal.

Even if one wishes take these steps, this unsupported state makes it difficult to learn how to do so and to decide how much effort should be spent on them. Let us focus on computational reproducibility, which should be relatively low-hanging fruit: ensuring that some other researcher can rerun your analysis and get the same results (without requiring your consultation). Software engineering has developed a suite of tools that can facilitate this goal, and, due to the open source movement, most are freely-available with plenty of documentation and examples. However, most neuroscientists are not software engineers, and so translating them to our use cases is not trivial. Furthermore, we quickly run into the question of longevity: how long should we ensure that results are computationally reproducible? The scientific result is generally understood as being timeless (until overturned by some later result), but technology moves fast and code written fifty years ago will not run today without serious effort. So should we expect scientific analyses to be computationally reproducible for 1 year? 5 years? 10? The more future-proof you wish your analysis to be, the more time and effort will be required. Perhaps we should think of this in a tiered manner, where all analyses should be reproducible for a few years, and then the field should

maintain the most important results for as long as they are considered relevant. However, we have not begun to have these conversations as a field. Since the scientific community has not viewed computational reproducibility as an important goal for the field, researchers have no guidance on what to share or how much effort to spend sharing it.

My intent here is not to critique the behaviors of any individual researcher attempting to find a job (given the scarcity of tenure-track positions, focusing on publications is understandable), but to highlight how the incentives with which we have structured our scientific communities do not serve us well. What alternatives are there? The simplest would be to change how academic hiring and tenure decisions are made: departments could explicitly reward or require data sharing or software development and they could only consider an applicant's N self-selected best publications rather than all of them, with N increasing with seniority [68]. These two proposals could be combined in some way, for example, considering N papers or $N/2$ papers all with data sharing or $N/2$ papers and one widely-used software package, etc. Other proposals have targeted the system of academic journals themselves, questioning how we could build a system of scholarly communication that better serves our interests (see [27] for a representative example). Finally, we could include a broader array of long term scientific academic jobs beyond tenure-track faculty. NYU's Center for Brain Imaging provides one example of this, and the US Research Software Engineer Association lobbies for another, the university support of research software engineers (note my emphasis on long term jobs: one of the problems with the proliferation and lengthening of postdoc positions is the instability they engender in researchers' lives, and we should seek to combat this trend).

The community could also re-evaluate how we think about individual research projects. In most of neuroscience and psychology, the emphasis is placed on being the first author of a publication, which incentivizes smaller author lists and thus, smaller projects. Furthermore, in most graduate student projects the data is collected either by the student themselves or by a close collaborator. But neither of these are necessary: high-energy physics, for example, frequently has author lists

extending into the hundreds, with large-scale experiments at particle accelerators generating data used by the entire field. Machine learning regularly utilizes what Mark Liberman refers to as the "common task method" or common task framework (CTF) [139], wherein a dataset is created for the evaluation of a specific task that members of the community attempt to solve, ImageNet being perhaps the most well-known to vision scientists [56]. Both of these provide alternatives that depend upon the research community collectively agreeing on what problems are worth solving, what dataset would be helpful for doing so, and how to evaluate success. BrainScore [190] is one laudable attempt at providing a CTF for evaluating computational models of the macaque visual system, but it is largely the work of a single lab and, for V1 at least, consists largely of data *summaries* (distribution of response properties) rather than complete datasets. Effort must be taken when constructing CTFs, especially in a field unused to them, to get the buy-in of a wide number of investigators, to think deeply about the goals and how to evaluate success, and to avoid a variant of what Box [23] calls "mathematistry": defining a technical problem with a well-defined data set and objective and conflating it with the broader scientific question, such that when one solves the technical problem, one can pretend they have solved the scientific one (e.g., assuming that near-perfect performance on ImageNet means that we understand human object recognition). That is, we must be willing to regularly question the link between our technical and scientific measures of success [155], just as I am proposing that we must reconsider what kind of scientific community our incentive structure helps construct, and how we may change the incentives to encourage the science we wish to see.

We could also take computational models, not just the papers that describe them, more seriously as objects of the scientific record. As described earlier in this introduction, having common models that multiple researchers can investigate from different angles is enormously beneficial for increasing understanding of a system. But, in vision science, these models have gotten more complex over the years (e.g., [81, 173, 217]), and so the chance that future researchers will be able to reimplement them from only the description provided in the paper has become

more and more remote. This makes it difficult to achieve the cumulative success found in the early days of the research program: each lab is working with slightly different variants of the model, wasting time reinventing the wheel, unsure exactly how their models differ from those of other labs and to what extent it matters. This situation makes it difficult to tell to what extent a new phenomena can be accounted for by existing models or whether a new tweak is necessary. And the difficulty for those who specialize in early vision is nothing compared to that of researchers who study the rest of the brain or computer vision: researchers who study face perception are unable to build models that have a "V1 frontend" without essentially being an expert in V1 as well, and the machine learning community would love to compare their models against the predictions of a "classic neuroscience model", but often end up with one that looks like a strawman. If, instead, we considered the models themselves important scientific objects, we could provide standard implementations in multiple languages, with notes on variants that are preferred by some investigators and phenomena that are well-accounted or unaccounted for. We could regularly test the models to ensure the implementations continue to be usable and that the analyses that use them are reproducible. Such a system would mesh nicely with a common task framework for evaluation of such models and would serve as a source of institutional knowledge for the field, but accomplishing it would require a radical shift from how we currently evaluate and reward scientific activities.

All of the above proposals would involve reimagining what our scientific system looks like and cannot be achieved quickly. However, smaller steps can be taken by individual researchers, labs, and departments. Providing official recognition for some of these activities, such as allowing them to count for a chapter of a thesis like found here, is one such step. Explicitly seeking applicants for faculty positions who support such practices is another. PIs can also build a culture of data sharing, providing support themselves and encouraging students to help each other. The work presented in this thesis is my attempt to take a tiny step in this direction, presenting empirical studies that can be built upon in later work through their shared data and models (with parameter values),

along with a software package providing usable implementations of existing methods. Hopefully it does so, moving us towards the ultimate goal of a scientific culture wherein researchers produce fewer papers, but better science.

2 Mapping Spatial Frequency Preferences Across Human Primary Visual Cortex

2.1 Abstract

Neurons in primate visual cortex (area V1) are tuned for spatial frequency, in a manner that depends on their position in the visual field. Several studies have examined this dependency using fMRI, reporting preferred spatial frequencies (tuning curve peaks) of V1 voxels as a function of eccentricity, but their results differ by as much as two octaves, presumably due to differences in stimuli, measurements, and analysis methodology. Here, we characterize spatial frequency tuning at a millimeter resolution within human primary visual cortex, across stimulus orientation and visual field locations. We measured fMRI responses to a novel set of stimuli, constructed as sinusoidal gratings in log-polar coordinates, which include circular, radial, and spiral geometries. For each individual stimulus, the local spatial frequency varies inversely with eccentricity, and for any given location in the visual field, the full set of stimuli span a broad range of spatial frequencies and orientations. Over the measured range of eccentricities, the preferred spatial frequency is well-fit by a function that varies as the inverse of the eccentricity plus a small constant. We also find small but systematic effects of local stimulus orientation, defined in both absolute coordinates

and relative to visual field location. Specifically, peak spatial frequency is higher for pinwheel than annular stimuli and for horizontal than vertical stimuli.

2.2 Introduction

A fundamental goal of visual neuroscience is to quantify the relationship between stimulus properties and neural responses, across the visual field and across visual areas. Studies of primary visual cortex (V1) have been especially fruitful in this regard, with electrophysiological measurements providing good characterizations of the responses of individual neurons to a variety of stimulus attributes [43, 54, 105, 184]. Nearly every neuron in V1 is selective for the local orientation and spatial frequency of visual input, and this has been captured with simple computational models built from oriented bandpass filters [53, 88, 115, 175, 188, 217].

The characterization of individual neural responses provides only a partial picture of the representation of visual information in V1. In particular, we know that the representation is not homogeneous – receptive field sizes grow and spatial frequency preferences decrease with distance from the fovea (eccentricity, [54]) – but we do not have a general quantitative description of the relationship between these response properties and location in the visual field. There are hundreds of millions of neurons in V1 [221], and thus, single-unit electrophysiology is unappealing as a methodology for addressing this question.[1] Functional magnetic resonance imaging (fMRI) offers complementary strengths and weaknesses, allowing simultaneous measurement of responses across all of visual cortex, but at a resolution in which each measurement represents the combined responses of thousands of neurons, limiting the characterization to properties that change smoothly across the cortical surface. Fortuitously, core properties of V1 such as position and spatial frequency tuning *do* vary smoothly across the cortical map [105, 107], and so are well suited for summary measures with fMRI. This has led to successful characterization of "population receptive fields"

[1] David Hubel described the process of characterizing visual field maps using single-unit electrophysiology as "a dismaying exercise in tedium, like trying to cut the back lawn with a pair of nail scissors" [103].

(pRFs), which specify the location and size in visual space of voxel responses [222]. A recent study [3] characterized voxel-wise spatial frequency tuning in early visual cortex, but did not provide an overall description of the dependence of this tuning on retinotopic location or stimulus orientation.

Here, we provide a compact parametric characterization of the spatial frequency and orientation preferences of population receptive fields in area V1, across the visual field. How compact a description can one expect? The information processing of a cortical area such as V1 would be simplest to study and describe if each location in the map analyzed the image with the same computations. This assumption of homogeneous processing is central to signal and image processing, and underlies recent developments in computer vision based on Convolutional Neural Networks [134]. But this assumption can be immediately rejected for primate visual systems, since we know that resolution declines precipitously with eccentricity. At the other extreme, if each part of the map analyzed the image in an entirely unique way, the prospect of understanding its function would be hopeless. Fortunately, many properties, such as receptive field size, vary smoothly and systematically with receptive field position, and similar types of models are able to successfully describe neural data across species, individuals, and map locations (e.g., [37]).

An attractive intermediate possibility is that cortical processing is conserved across the visual field, up to a dilational scale factor. One hypothesis is that eccentricity-dependent RF scaling emerges first in the RGCs, and then all subsequent stages simply perform a homogeneous (convolutional) transform on their afferents, thus inheriting the eccentricity-scaling of RF sizes. This would result in all neuronal tuning across the cortex being scaled versions of each other. For example, if V1 neurons were tuned such that their preferred spatial frequency was always p periods per receptive field, and their receptive fields grew linearly as they moved away from the fovea, such that $s = ar$ (where s is the diameter of the receptive field and r is the eccentricity), then neuronal peak spatial frequency would equal $f = p/s = p/ar$. If this approximates the true relationship between spatial frequency tuning and eccentricity, then sinusoidal gratings, which have a constant

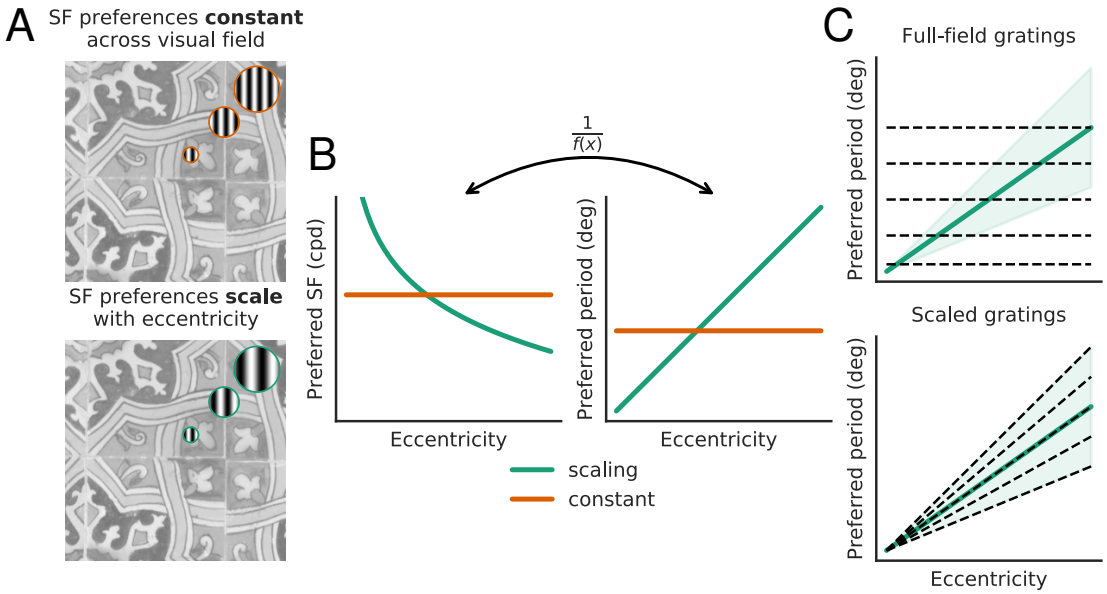


Figure 2.1: (A) Illustration of two extremal models for spatial frequency preferences across the visual field. Top: preferences are conserved across the visual field (despite changes in receptive field size). Bottom: preferred spatial period (inverse of spatial frequency) is proportional to eccentricity (along with receptive field size). Tile image is an original photograph from author's collection. (B) Preferred SF (left) and period (right) as a function of eccentricity, for the two models (red and green curves). (C) Efficiency of stimuli (dashed lines) for probing the scaling model. Top: If preferences scale with eccentricity, conventional full-field two-dimensional sine gratings are an inefficient way to measure spatial frequency tuning: gratings with a large period will be ineffective at driving responses in the fovea and those with a low period will be ineffective for the periphery. Bottom: Oscillating stimuli whose period grows linearly with eccentricity provide a more efficient choice .

frequency everywhere in the image, are an inefficient choice of stimulus to measure this, as high frequencies will be shown at the periphery and low frequencies at the fovea, neither of which will drive responses effectively.

To enable efficient characterization of local spatial frequency preferences, we develop a novel set of global stimuli in which local frequency scales inversely with eccentricity, and which span a variety of orientations. We use these stimuli to probe the dependency of spatial frequency preferences on orientation and retinal location, and summarize this using a compact functional description that is jointly fit to data over the whole visual field. The model parameterization allows spatial frequency tuning to vary with eccentricity, and allows both spatial frequency tuning and BOLD amplitude to vary with retinotopic angle and stimulus orientation. This modeling approach allows flexibility for our parameters of interest, but is not arbitrarily flexible. This is necessary in

order to be able to concisely describe how spatial frequency is encoded across the whole visual field and to enable extrapolation to stimuli or visual field positions not included in the study.

2.3 Methods

All experimental materials, data, and code for this project can be found online under the MIT or similarly permissive license. Specifically, minimally pre-processed data are found on OpenNeuro [141], code on GitHub, and other materials on OSF and the NYU Faculty Digital Archive (view README in the software repository for download and usage instructions).

2.3.1 Stimulus design

To efficiently estimate preferred spatial frequency across the visual field, we use a novel set of grating stimuli with spatially-varying frequency and orientation. Figure 2.1 illustrates the logic of the stimulus construction, which is designed for efficient characterization of a system whose preferred spatial frequency falls with eccentricity. Conventional large-field two-dimensional sine gratings will be inefficient for such a system, since the stimulus set will include low-frequency stimuli which are ineffective for the fovea, and high-frequency stimuli which are ineffective for the periphery. Instead, we construct "scaled" log-polar stimuli, such that local spatial frequency decreases in inverse proportion to eccentricity (figure 2.2B). Specifically, all stimuli are of the form

$$f(r, \theta) = \cos(\omega_r \ln(r) + \omega_a \theta + \phi), \tag{2.1}$$

where coordinates (r, θ) specify the eccentricity and polar angle of a retinal position, relative to the fovea. The angular frequency ω_a is an integer specifying the number of grating cycles per revolution around the image, while the radial frequency ω_r specifies the number of radians per unit increase in $\ln(r)$. The parameter ϕ specifies the phase, in radians. The local spatial frequency

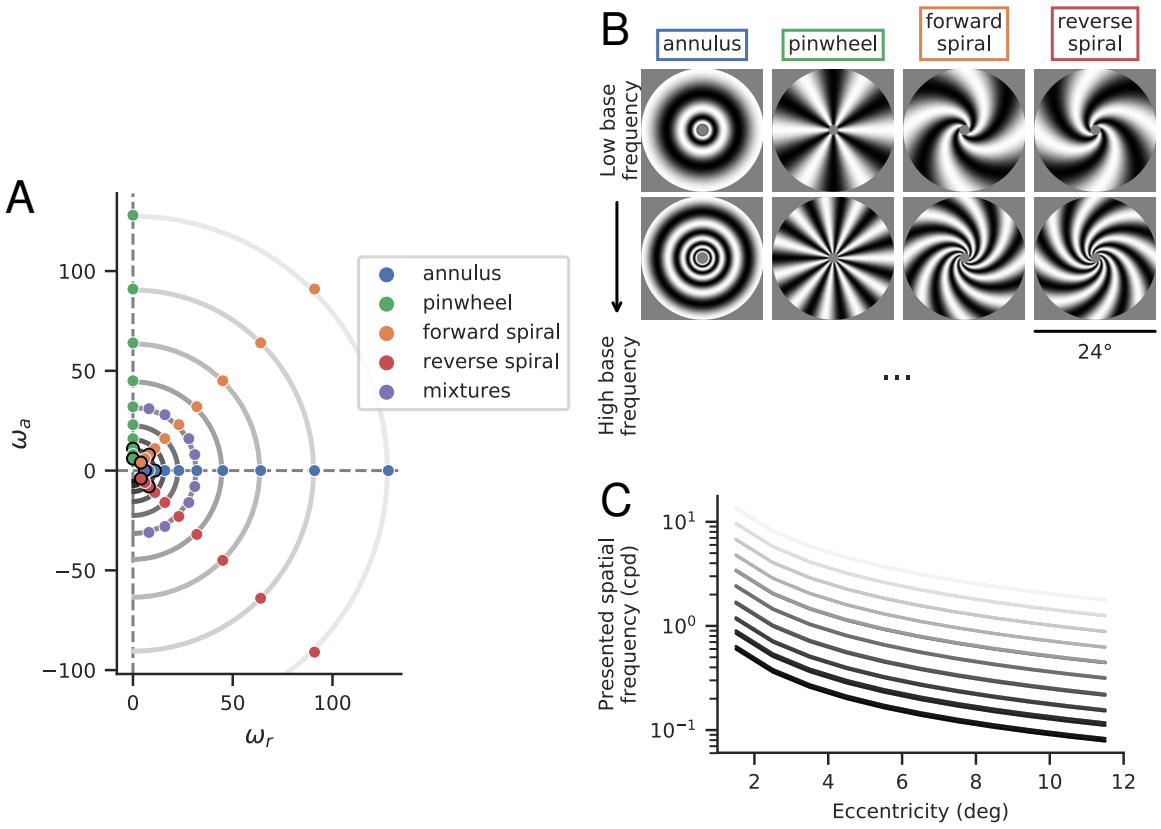


Figure 2.2: Stimuli. (A) Base frequencies (ω_r, ω_a) of experimental stimuli. Stimulus category is determined by the relationship between ω_a and ω_r, which determines local orientation information (Eq. 2.3). (B) Example stimuli from four primary classes, at two different base frequencies. These stimuli correspond to the dots outlined in black in panel A. (C) Local spatial frequencies (in cycles per degree) as a function of eccentricity. Each curve represents stimuli with a specific base frequency, $\sqrt{\omega_r^2 + \omega_a^2}$, corresponding to one of the semi-circular contours in panel A. The two rows of stimuli in panel B correspond to the bottom and 3rd-from-bottom curves.

is equal to the magnitude of the gradient of the argument of cos(·) with respect to retinal position (see Supplement 2.7.1):

$$\omega_l(r,\theta) = \frac{\sqrt{\omega_r^2 + \omega_a^2}}{r}. \tag{2.2}$$

That is, local frequency is equal to Euclidean norm of the frequency vector (ω_r, ω_a) divided by eccentricity (in units of radians per pixel or radians per degree, depending on the units of r), which implies that the local spatial period of the stimuli grows linearly with eccentricity. Similarly, the local orientation can be obtained by taking the angle of the gradient of the argument of cos(·) with respect to retinal position (see Supplement 2.7.1):

$$\theta_l(r,\theta) = \theta + \tan^{-1}\left(\frac{\omega_a}{\omega_r}\right). \tag{2.3}$$

That is, the local grating orientation is the angular position relative to the fovea, plus the angle of the two-dimensional frequency vector (ω_r, ω_a). Note that θ_l is in absolute units (e.g., $\theta_l = 0$ indicates local orientation is vertical, regardless of location). For our stimuli, this depends on the polar angle, but a uniform grating has the same θ_l value everywhere in the image (its orientation thus does not depend on polar angle).

We generated stimuli corresponding to 48 different frequency vectors (see Fig. 2.2), at 8 different phases $\phi \in \{0, \pi/4, \pi/2, \ldots, 7\pi/4\}$. The frequency vectors were organized into five different categories:

1. Pinwheels:

   $\omega_r = 0$, $\omega_a \in \{6, 8, 11, 16, 23, 32, 45, 64, 91, 128\}$

2. Annuli:

   $\omega_a = 0$, $\omega_r \in \{6, 8, 11, 16, 23, 32, 45, 64, 91, 128\}$

3. Forward spirals:

   $\omega_r = \omega_a \in \{4, 6, 8, 11, 16, 23, 32, 45, 64, 91\}$

4. Reverse spirals:

   $\omega_r = -\omega_a \in \{4, 6, 8, 11, 16, 23, 32, 45, 64, 91\}$

5. Fixed-frequency mixtures:

   $(\omega_r, \omega_a) \in \{(8, 31), (16, 28), (28, 16), (31, 8), (31, -8), (28, -16), (16, -28), (8, -31)\}$

Note that ω_a values must be integers (since they specify cycles per revolution around the image), and we chose matching integer values for ω_r. Because of this constraint, the pinwheel/annulus and the forward/reverse spiral stimuli have slightly different local spatial frequencies. For the same reason, the local spatial frequency of the mixture stimuli is only approximately matched across stimuli ($\sqrt{\omega_a^2 + \omega_r^2} \approx 32$). Across all stimuli, the spatial frequencies presented at any given eccentricity span a 20-fold range (figure 2.2C). For example, at the most foveal portion of the stimuli (from 1 to 2 deg) the frequencies are log-spaced from 0.6 to 13.65 cpd. In the most peripheral region (11 to 12 deg)), the range is 0.078 to 1.78 cpd.

2.3.2 Display Calibration

The projector used to display stimuli in our experiments was calibrated to produce light intensities proportional to luminance. In addition, we wanted to compensate for spatial blur (due to a combination of display electronics or optics) that could systematically alter the frequency content of our stimuli. We estimated the modulation transfer function (MTF) of the projector (i.e., the Michelson contrast as a function of spatial frequency), shown in figure 2.3. We used a calibrated camera and developed custom software to process and analyze photographs of full-contrast square-wave gratings. We found that the contrast of the projected image decreased by roughly 50% as it approached the Nyquist frequency of 0.5 cycles per display pixel. We compensated for these effects by rescaling the amplitude of low frequency content in our stimuli, by an amount proportional to the inverse MTF (note that the more natural procedure of increasing the high frequency content is not practical, as it could exceed the maximum contrast that can be displayed).

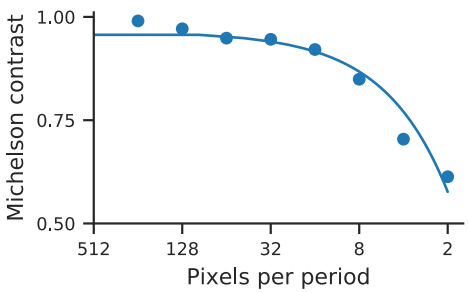


Figure 2.3: Estimated modulation transfer function (MTF) of the projector used in our experiments. Michelson contrast was measured for periods from 2 to 256 pixels (blue points) and then fit with a univariate spline (blue curve) with smoothing degree 1 [218]. The fitted spline was used for calibration.

2.3.3 Participants

Twelve participants (7 women and 5 men, aged 22 to 35), including an author (W.F.B.), participated in the study and were recruited from New York University. All subjects had normal or corrected-to-normal vision. Each subject completed 12 runs, except for sub-04, who only completed 7 of the 12 runs due to technical issues. The quality of their GLMdenoise fits and their final model fits do not vary much from those of the other subjects. All subjects provided informed consent before participating in the study. The experiment was conducted in accordance with the Declaration of Helsinki and was approved by the New York University ethics committee on activities involving human subjects.

2.3.4 Experimental Design

The experiment was run on an Apple MacIntosh computer, using custom scripts with PsychoPy [171], presented on a luminance-calibrated MTF-corrected VPixx ProPixx projector. Images were projected onto a screen, which the subject viewed through a mirror. The screen was 36.2 cm high and 83.5 cm from the subject's eyes (73.5 cm from screen to mirror, and approximately 10 cm from mirror to eyes). Stimuli were constrained to a circular aperture filling the height of the display (12 deg radius), with an anti-aliasing mask at the center (0.96 deg radius). Each stimulus

class was presented in a 4-s trial, during which the 8 images with different phases were shown in randomized order. Each of the 8 images was presented once, cycled on and off (300-msec on, 200-msec off) in order to minimize adaptation. A movie of a single run can be viewed online. Each of the 48 stimulus classes was presented once in each of 12 runs, with the presentation order of the stimulus classes and of the phases randomized across runs. Subjects viewed these stimuli while performing a one-back task on a stream of alternating black and white digits (1-sec on, 1-sec off) at the center of the screen in order to ensure accurate fixation, minimize attentional effects, and maintain a constant cognitive state. Thus, the central one degree of vision always contained either a blank midgrey screen or a black or white digit. This lessens the possibility of differences in fixational eye movements that might arise from differences in stimulus structure near the fovea. Behavioral responses were recorded using a button box (see supplement 2.7.2 for behavioral analysis).

2.3.5 fMRI Scanning Protocol

All MRI data for the spatial frequency experiment were acquired at the NYU Center for Brain Imaging using a 3T Siemens Prisma scanner with a Siemens 64 channel head/neck coil. For fMRI scans, we used the CMRR MultiBand Accelerated EPI Pulse Sequence (Release R015a) (TR, 1000 ms; TE, 37 ms; voxel size, $2mm^3$; flip angle, 68 degrees; multiband acceleration factor, 6; phase-encoding, posterior-anterior) [67, 148, 234]. High resolution whole-brain anatomical T1-weighted images (1 mm^3 isotropic voxels) were acquired from each subject for registration and segmentation using a 3D rapid gradient echo sequence (MPRAGE). Two additional scans were collected with reversed phase-encoded blips, resulting in spatial distortions in opposite directions. These scans were used to estimate and correct for spatial distortions in the EPI runs using a method similar to Andersson, Skare, and Ashburner [6], as implemented in FSL [201].

2.3.6 Preprocessing

fMRI data were minimally preprocessed using a custom script (available from the Winawer lab) which builds a Nipype [79, 80] pipeline. Brain surfaces were reconstructed using `recon-all` from FreeSurfer v6.0.0 [51]. Functional images were motion corrected using `mcflirt` (FSL v5.0.10 [113]) to the single-band reference image gathered for each scan. Each single-band reference image was then registered to the distortion scan with the same phase-encoding direction using `flirt` (FSL v5.0.10 [82, 112, 113]). Distortion correction was performed using an implementation of the TOPUP technique [6] using `TOPUP` and `ApplyTOPUP` (FSL v5.0.10 [201]). The unwarped distortion scan was co-registered to the corresponding T1w using boundary-based registration [82] with 9 degrees of freedom, using `bbregister` (FreeSurfer v6.0.0). The motion correcting transformations and BOLD-to-T1w transformation were concatenated using `ConvertXFM` (FSL v5.0.10) and then were applied to the functional runs in a single step along with the unwarping warpfields using `ApplyWarp` (FSL v5.0.10). Applying the corrections in a single step minimizes blurring from the multiple interpolations.

2.3.7 Retinotopy

A separate retinotopy experiment was used to obtain the population receptive field (pRF) location and size for V1 voxels in each subject [222]. This experiment consisted of six standard pRF mapping runs, with sweeping bar contrast apertures filled with a variety of colorful objects, faces and textures. This stimulus has been shown to be effective in evoking BOLD responses across many of the retinotopic maps in visual cortices [14, 16, 98]. The results of this pRF mapping were combined with a retinotopic atlas [12] in order to improve the accuracy of the retinotopic map (see Benson and Winawer [14] for a description of this method). The stimulus, fMRI acquisition parameters, and fMRI pre-processing for the retinotopy experiments are described in detail in Benson and Winawer [14] and Himmelberg et al. [98].

2.3.8 Stimulus response estimation

Response amplitudes were estimated using the GLMdenoise MATLAB toolbox [118]. The algorithm fits an observer-specific hemodynamic response function (HRF), estimating response amplitudes (in units of percent BOLD signal change) for each voxel and for each stimulus, with 100 bootstraps across runs. Thus for each voxel we estimate 48 responses (one for each unique pair (ω_a, ω_r), averaged over the 8 phases shown within the trials). The algorithm also includes three polynomial regressors (degrees 0 through 2) to capture the signal mean and slow drift, and noise regressors derived from brain voxels that are not well fit by the GLM.

The combined retinotopy and GLMdenoise measurements consist of (for each voxel): the visual area, population receptive field location and size, and 100 bootstrapped response amplitudes to each of the 48 stimuli.

2.3.9 One-Dimensional Tuning Curves

We fit one-dimensional log-normal tuning curves to the responses of groups of voxels at different eccentricities (lying within one-degree eccentricity bins):

$$\hat{\beta}_b(\omega_l) = A_b \cdot \exp\left(\frac{-\left(\log_2(\omega_l) + \log_2(p_b)\right)^2}{2\sigma_b^2}\right) \tag{2.4}$$

where $\hat{\beta}_b(\omega_l)$ is the average BOLD response in eccentricity bin b at spatial frequency ω_l (in cycles per degree), A_b is the response gain, p_b is the preferred period (the reciprocal of the peak spatial frequency, ω_b, which is the mode of the tuning curve), and σ_b is the bandwidth, in octaves. Fits were obtained separately for the four primary stimulus classes (pinwheel, annulus, forward spiral, and reverse spiral).

We fit these tuning curves 100 times per subject, per stimulus class, and per eccentricity, bootstrapping across the fMRI runs (12 per subject).

2.3.10 Two-Dimensional Tuning Curves

Our one-dimensional tuning curves are averaged over stimulus orientation and retinotopic angle. To capture the effect of these additional stimulus attributes, we developed a two-dimensional model for individual voxel responses as a function of stimulus local spatial frequency (in cycles per degree), ω_l, stimulus local orientation, θ_l, voxel eccentricity (in degrees), r_v, and voxel retinotopic angle, θ_v (figure 2.4A). Responses are again assumed to be log-normal with respect to spatial frequency:

$$\hat{\beta}_v(\omega_l, \theta_l) = A_v \cdot \exp\left(\frac{-\left(\log_2(\omega_l) + \log_2(p_v)\right)^2}{2\sigma^2}\right) \tag{2.5}$$

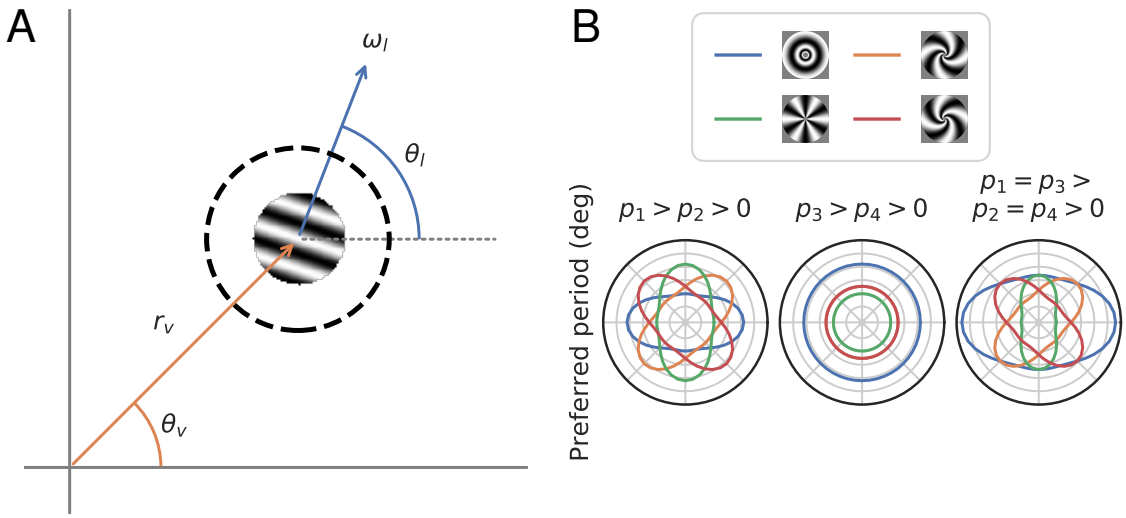


Figure 2.4: (A) Local stimulus parameterization for the two-dimensional model. The model is a function of four variables, two related to voxel population receptive field location and two related to stimulus properties. r_v and θ_v specify the eccentricity (in degrees) and the retinotopic angle of the location of the center of the voxel's population receptive field, relative to the fovea. ω_l and θ_l, specify the local spatial frequency (in cycles per degree) and the local orientation (in radians, counter-clockwise relative to horizontal) of the stimulus, at the center of that voxel's population receptive field (dashed line). (B) Schematic showing the effects of p_i parameters on preferred period as a function of retinotopic angle at a single eccentricity for the four main stimulus types used in this experiment. When $p_1 > p_2 > 0$ (and $p_3 = p_4 = 0$), the effect of orientation on preferred period is in the absolute reference frame only, i.e., preferred period only depends on absolute orientation (e.g., vertical or horizontal). In this plot, preferred period varies with retinotopic angle because the absolute orientation of our stimuli vary with retinotopic angle (for another example, see fig 2.10, where the relative amplitude effect is also only in the absolute reference frame; thus the relative amplitude is always higher for vertical than horizontal stimuli). When $p_3 > p_4 > 0$ (and $p_1 = p_2 = 0$), the effect of orientation is in the relative reference frame only, and annulus stimuli will always have the highest preferred period. Finally, when all $p_i \neq 0$, the effects are mixed.

In our one-dimensional analysis, we fit parameters $\{p, A, \sigma\}$ separately to each eccentricity band and stimulus class. Based on the results of that analysis (see 2.4), we assume σ is constant across eccentricities, retinal position, and local stimulus spatial frequency (while others have found some variation in bandwidth with respect to these variables, this study focuses on peak spatial frequency tuning and we do not include extra flexibilty in model bandwidth, in order to avoid overfitting). We assume functional forms for the dependencies of parameters p and A on retinal position, local stimulus spatial frequency, and local stimulus orientation. First, we parameterize the effect of eccentricity, fitting the preferred period as an affine function of a voxel's eccentricity r_v: $p_v = ar_v + b$. We assume that this baseline dependency is modulated by effects of retinotopic

angle and stimulus orientation, both of which are known to affect visual perception [10, 90, 231]. Specifically, we express preferred period as:

$$\begin{aligned} p_v = [ar_v + b][1 &+ p_1 \cos(2\theta_l) + p_2 \cos(4\theta_l) \\ &+ p_3 \cos(2(\theta_l - \theta_v)) \\ &+ p_4 \cos(4(\theta_l - \theta_v))]. \end{aligned} \tag{2.6}$$

The parameters p_i have the following interpretations:

p_1: absolute cardinal effect, horizontal vs. vertical. A positive p_1 means that voxels have a higher preferred period for vertical than for horizontal stimuli.

p_2: absolute cardinals vs. obliques effect, horizontal/vertical vs. diagonals. A positive p_2 means that voxels have a higher preferred period for cardinal than for oblique stimuli.

p_3: relative cardinal effect, annuli vs. pinwheels. A positive p_3 means that voxels have a higher preferred period for annular than for pinwheel stimuli.

p_4: relative cardinals vs. obliques effect, annuli/pinwheels vs. spirals. A positive p_4 means that voxels have a higher preferred period for annuli and pinwheels than for spirals.

p_1 and p_2 have effects in the absolute reference frame because they only depend on θ_l, the orientation in absolute terms, whereas p_3 and p_4 additionally depend on θ_v and thus have effects in the relative reference frame.

To illustrate these effects, we show tuning functions for several stimulus classes given a few possible parameter combinations (figure 2.4B). We also provide an interactive tool that enables the user to set arbitrary values for all parameters and to probe how the parameter settings influence the pattern of responses to various stimulus types.

We also express the gain of the BOLD responses as a function of voxel retinotopic angle and stimulus orientation (without the eccentricity-dependent base term):

$$
\begin{aligned}
A_v = (1 &+ A_1 \cos(2\theta_l) + A_2 \cos(4\theta_l) \\
&+ A_3 \cos(2(\theta_l - \theta_v)) \\
&+ A_4 \cos(4(\theta_l - \theta_v))),
\end{aligned}
\tag{2.7}
$$

This parameterization allows the amplitude to vary depending on both absolute stimulus orientation (θ_l), and stimulus orientation relative to retinotopic angle ($\theta_l - \theta_v$), but not on absolute retinotopic location. This choice is premised on the fact that voxel-to-voxel variation in the amplitude of the BOLD signal depends in part on factors that are not neural. For example, BOLD amplitude is influenced by draining veins [121, 135] and the orientation of the gray matter surface relative to the instrument magnetic field [72], as well as other factors not directly related to neural responses.

In addition, the model cannot capture categorical differences across the visual field, e.g., between upper and lower, or foveal and parafoveal visual field, except insofar as the parametric forms allow (linear function of eccentricity, harmonics of stimulus orientation).

2.3.11 Model fitting

We fit the 2D model to all V1 voxels simultaneously, excluding voxels whose population receptive field (pRF) center lies outside the stimulus, those whose pRF center lies within one standard deviation of the stimulus border, and those with an average negative response to our stimuli. Voxels with negative responses but whose pRFs are centered within the stimulus extent are likely dominated by artifacts such as those arising from draining veins [135, 233].

The remaining voxels vary widely in their signal to noise ratio. Typically in fMRI analyses, all voxels whose noise level lies above some threshold are excluded from the analysis. Here, we instead weight each voxels' loss by its precision, so that noisier voxels will contribute less to the parameter estimates. Specifically, we use a normalized mean-squared error loss over voxels:

$$L_v(\beta_v, \hat{\beta}_v) = \frac{1}{\sigma_v^2} \sum_{i=1}^{n} \frac{1}{n} \left(\frac{\beta_{iv}}{||\beta_v||_2} - \frac{\hat{\beta}_{iv}}{||\hat{\beta}_v||_2} \right)^2 \tag{2.8}$$

where i indexes the n different stimulus classes, β_{iv} is the response of voxel v (estimated using GLMdenoise) to stimulus class i, $\hat{\beta}_{iv}$ is the response to stimulus class i predicted by our model, $||\beta_v||_2$ is the L2-norm of β_v (across all stimulus classes), and σ_v^2 is the variance of voxel v's response (that is, $\sigma_v^2 = \frac{1}{n} \sum_{i=1}^{n} \sigma_{vi}^2$, where σ_{vi} is half of the 68 percentile range of the response of voxel v to stimulus class i, as estimated by GLMdenoise). This loss function is equivalent to the cosine between response vectors β_v and $\hat{\beta}_v$ multiplied by $\frac{2}{n\sigma_v^2}$. Normalization of the β_v and $\hat{\beta}_v$ vectors allows the fitting to be agnostic to variations in absolute response amplitude, capturing the response dependency on stimulus and retinal location.

We minimize the average of this loss across all appropriate voxels, using custom code written in PyTorch [169] and using the AMSGrad variant of the Adam optimization algorithm [124, 183]. To assess model accuracy, we use 12-fold cross-validation (see 2.5.0.1). Specifically, we fit the model to 44 of the 48 stimulus classes, then get predictions for the 4 held-out classes. We do this for each of the 12 subsets, which get us a complete $\hat{\beta}_v$ that we can compare against β_v.

2.3.12 Software

Data analysis, modeling, and figure creation were done using a variety of custom scripts written in Python 3.6.3 [215], all found in the software repository associated with this paper. The following packages were used: snakemake [149], Jupyter Lab [125], numpy [86], matplotlib [106], scipy [218], seaborn [227], pandas [145, 205], nipype [79, 80], nibabel [28], scikit-learn [170], neuropythy [14], pytorch [169], psychopy [171], FSL [201], freesurfer [51], vistasoft, and GLMdenoise [118].

2.4 One-Dimensional Analysis

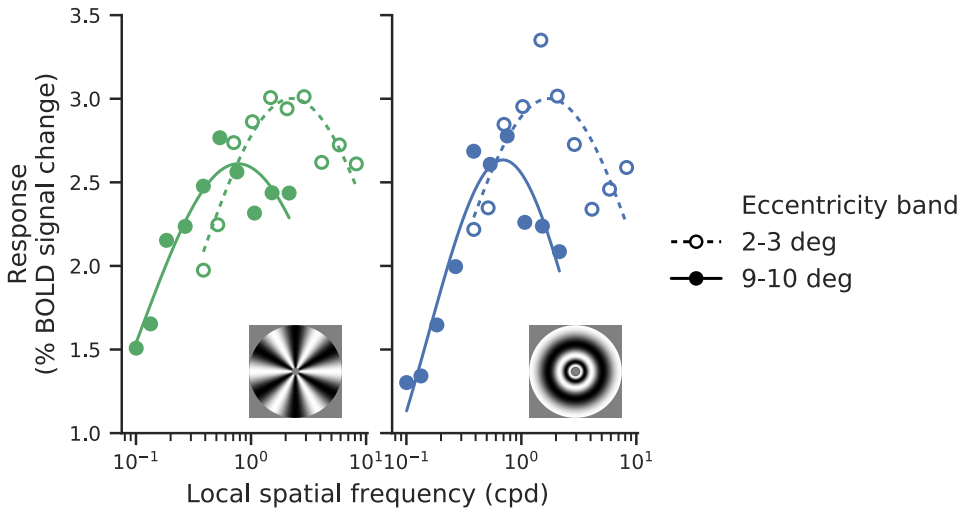


Figure 2.5: Example data and best-fitting log-normal tuning curves for responses of one subject (sub-01) to pinwheel (left) and annular (right) stimuli. The solid line and filled circles correspond to 9-10deg eccentricity, while dashed line and empty circles correspond to 2-3deg.

We start by analyzing the data as a function of spatial frequency alone (i.e., averaging over orientation), which requires fewer assumptions and is easier to visualize. We fit log-normal tuning curves to averaged voxel responses at each eccentricity for each of the four main stimulus classes. The log-normal function provides a reasonably good fit to the data (see figure 2.5).

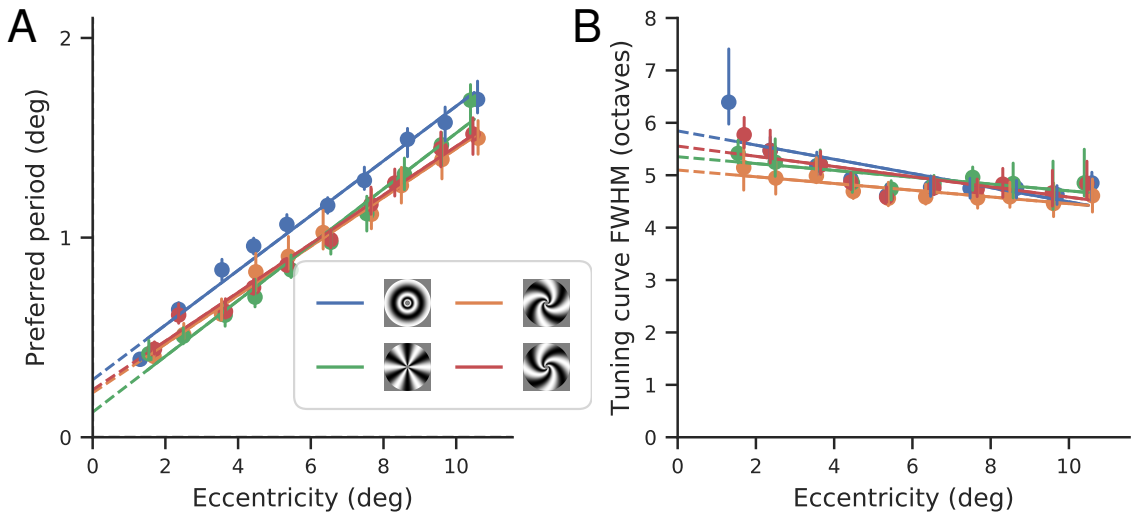


Figure 2.6: Spatial frequency tuning. (A) Preferred period of tuning curves (parameter p_b in equation (2.9), $n = 12$), as functions of eccentricity, fit separately for the four different stimulus classes. Points and vertical bars indicate the median and 68% confidence intervals obtained from bootstraps combining subjects using a precision-weighted average (see text). Lines are the best linear fits. (B) Full-width half-maximum (in octaves) of tuning curves, as functions of eccentricity, fit separately for the four different stimulus classes. Points and vertical bars indicate the median and 68% confidence intervals obtained from bootstraps combining subjects using a precision-weighted average (see text). Lines are the best linear fits.

We then combined the preferred periods across subjects by bootstrapping a precision-weighted mean: for each eccentricity and stimulus class, we selected 12 subjects at random with replacement, multiplied each subject's median preferred period by the precision of that estimate, and averaged the resulting values:

$$p = \frac{\sum_{s=1}^{12} \frac{\tilde{p}_s}{\sigma_s^2}}{\sum_{s=1}^{12} \frac{1}{\sigma_s^2}} \tag{2.9}$$

where $\tilde{p}_s$ is the median preferred period value for subject s and σ_s is the difference between the 16th and 84th percentile for that subject. This bootstrapping is done 100 times to obtain median values and 68% confidence intervals displayed in figure 2.6A. The precision weighted average has the virtue of giving more weight to better parameter estimates while not fully discarding data.

The preferred period for each stimulus class is well-described as an affine function of eccentricity, with a positive offset. Thus, the spatial frequency preferences of V1 do not scale perfectly with eccentricity (e.g., the preferred frequency at 4 degrees is not half that of 2 degrees). There is

also a noticeable dependence on stimulus orientation, with the annular stimuli exhibiting a larger preferred period than the other three stimuli at each eccentricity. Differences between the other stimulus types are more subtle, but perhaps indicate a slightly reduced slope for the two spiral stimuli relative to the pinwheel.

We do the same precision-weighted bootstrapping process for the full-width half-maximum (in octaves) of the tuning curves shown in 2.6B. We can see that the FWHM is mostly constant across eccentricities, except for some larger, noisier values for the most foveal voxels. We believe this apparent dip is due to how the fits are constrained, rather than a real decline in tuning curve width: as can be seen in figure 2.8A, the presented frequencies shift from the right of the tuning curve to the left for more peripheral voxels. In the periphery and the fovea, where most of the presented frequencies fall on one side of the curve, the width is unlikely to be well-constrained, resulting in the higher error bars seen in figure 2.6B. FWHM additionally appears to be consistent across stimulus types.

2.5 Two-Dimensional Model Results

The one-dimensional model provides a useful but limited overview of spatial frequency selectivity. In particular, we've treated the four stimulus classes as discrete categories, rather than members of a continuum over relative orientation. Moreover, this analysis conflates the effects of absolute orientation (relative to a global vertical/horizontal coordinate system) and orientation relative to a voxel's retinotopic angle. These might be systematically different, and because there are more voxels at some retinotopic angles than others (e.g., [15, 197]), the averaging might cause systematic biases in the summary measures. Finally, the analysis examines peak spatial frequency tuning but does not examine possible differences in BOLD amplitude for different stimulus orientations.

The two-dimensional model described in section 2.3.10 allows us to more directly and compre-

hensively assess how spatial frequency tuning varies across the visual field. Instead of binning voxels by eccentricity, we fit all voxels simultaneously, with each voxel's contribution to the loss function weighted by the precision of its responses. By fitting each voxel, we can tease apart the effects of absolute and relative orientation (figure 2.4B). We are able to parameterize these effects on both preferred period and gain. Finally, the fitted model will generate predictions for the response of any voxel in the visual field to any spatial frequency and orientation (though its predictions will likely decrease in accuracy the farther the voxel's retinotopic location and stimulus properties move from the those included in this study).

2.5.0.1 Model selection

The full 2D model has 11 parameters, and we used cross-validation in order to determine which are necessary to explain the data in V1. Omitting or including all combinations of parameters would yield 2^{11} possible models. To reduce this, we grouped the parameters into several small sets, based on whether they affect the preferred period or gain and whether their effect is determined by eccentricity, relative orientation, or absolute orientation. For example, p_1 and p_2 both affect preferred period as a function of absolute orientation and so are always both present or both absent. Moreover, we only tested parameter combinations that we considered plausible; for example, we do not test relative preferred period and absolute gain. Figure 2.7A shows the 14 candidate submodels considered. When fitting model 8, for example, the parameters $\sigma, a, b, p_1, p_2, A_1, A_2$ are all fit, while p_3, p_4, A_3, A_4 are set to 0; this corresponds to modeling the preferred period as a linear function of eccentricity, modulated by absolute orientation, and modeling the gain as also modulated by absolute orientation.

Submodels are fit per subject, with 12-fold cross-validation, withholding four random stimuli from fitting on each fold, using the same partitions across models and subjects. After training, predictions are generated for these 4 stimuli, and the subject's cross-validation loss for the model is computed across all of the held-out data (12 folds). Cross-validation loss varies greatly across

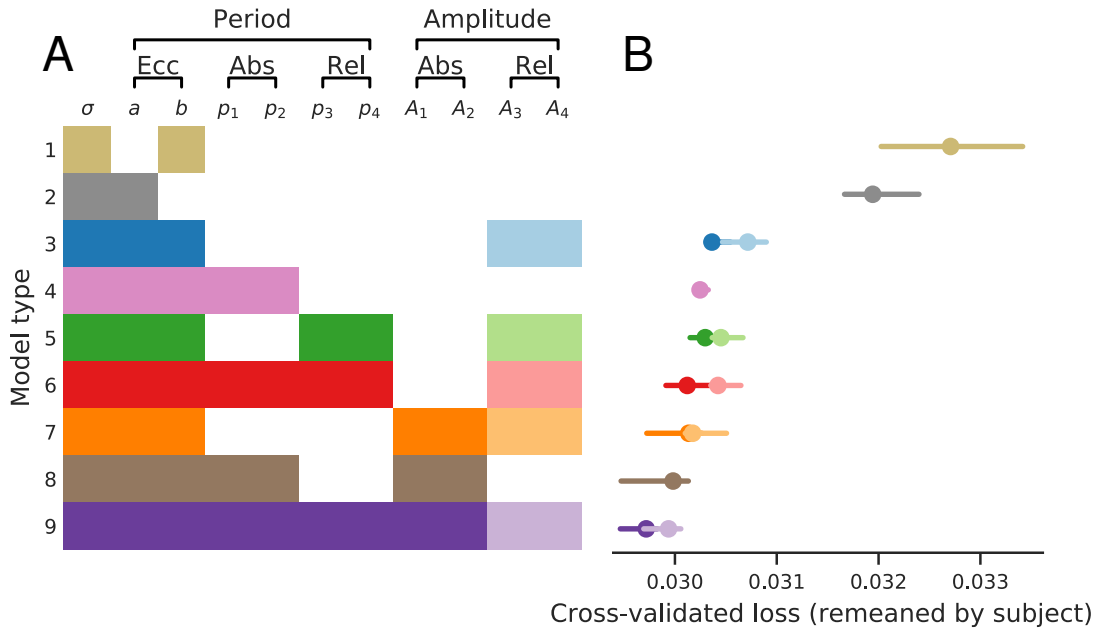


Figure 2.7: Nested model comparison via cross-validation. (A) 14 different submodels are compared to determine which of the 11 parameters, as defined in Equations (2.4), (2.6), and (2.7), are necessary. Model parameters are grouped by whether they affect the period or the gain, and whether their effect relates to eccentricity, absolute orientation, or relative orientation. Filled color boxes indicate parameter subset used for each submodel. (B) Cross-validated loss for each submodel. Models are fit to each subject separately, using 12-fold cross-validation (each fold leaves out 4 random stimuli). Quality of fit varies across subjects, so to combine subjects and view the effect of model, we subtract each subject's mean loss across models, then add back the average loss across subjects and models. Bars show the 68% confidence intervals from bootstrapped mean across subjects.

subjects, dependent on the subject's signal to noise ratio. To combine across subjects, we normalize the data by subtracting each subject's mean cross-validation loss across models. For visualization, we then add back the average loss across subjects. Figure 2.7B shows the median cross-validation loss and 68% confidence intervals of these losses. For some rows, two models are shown: the model with and without fitting parameters A_3, A_4. (The variant that fits those parameters is shown in the desaturated color.) The results indicate that 9 of the 11 parameters contribute to accurately predicting responses. By fitting each of the 14 candidate models to each subject individually, we find that all parameter groupings improve performance except for A_3 and A_4: the loss is greater whenever those two are included.

Comparing the losses of models 1, 2, and 3 reveals the importance of the two parameters relating eccentricity to preferred period: while a line through the origin (model 2) captures the data better than a constant value (model 1), the performance increases substantially with an affine model using both terms (model 3). In sum, both parameters a and b are required to accurately explain the data, and preferred period increases linearly with eccentricity with a non-zero intercept.

Beyond eccentricity, the effect of orientation on preferred period does not change performance much unless one also adds the effect on gain (models 4 through 6 all have similar performance). The effect of relative orientation on gain by itself has a negative effect on performance, as can be seen by comparing the saturated and desaturated points for models 3, 5, 6, 7, and 9. Absolute orientation, on the other hand, improves performance, as can be seen by comparing 6 and 9, 4 and 8, or 3 and 7. Therefore, for the remainder of this paper, we use the saturated point of model 9, which has the lowest cross-validation loss and fits all preferred period parameters, p_k, as well as those that capture the effect of absolute orientation on gain.

2.5.0.2 Spatial frequency tuning across stimulus orientation and visual field positions

Having selected model 9, we then re-fit it to each subject without cross-validation. Specifically, we fit model 9 to each of 100 bootstraps from each subject separately, giving us 100 estimates of

each model parameter per subject. Figure 2.8A shows three example voxels' median responses and model 9's median predictions, as a function of local spatial frequency, from one subject. As expected, the peak of the spatial frequency tuning function decreases with increasing eccentricity. The bandwidth (in octaves) is comparable across eccentricities, and the plots indicate that the stimuli sampled the local spatial frequencies appropriately at each eccentricity.

Overall, the log-Gaussian tuning function provided a good fit to the complete dataset. Figure 2.8B shows the responses of all voxels, across all subjects, as a two-dimensional histogram, aligned to the peak spatial frequency per voxel, plotted together with the model's predictions. We can see the responses are symmetric about the peak, demonstrating that a log-Gaussian (as opposed to a linear Gaussian) function is the better choice. The responses do appear to deviate slightly from the model tuning curve: slightly flatter at the peak and falling faster away from it. A larger exponent could potentially improve the fit, e.g., $\exp(-\log_2(x)^4)$ instead of $\exp(-\log_2(x)^2)$. However, such a change will not have a large effect on the estimates of preferred spatial frequency, which is the primary focus of this paper.

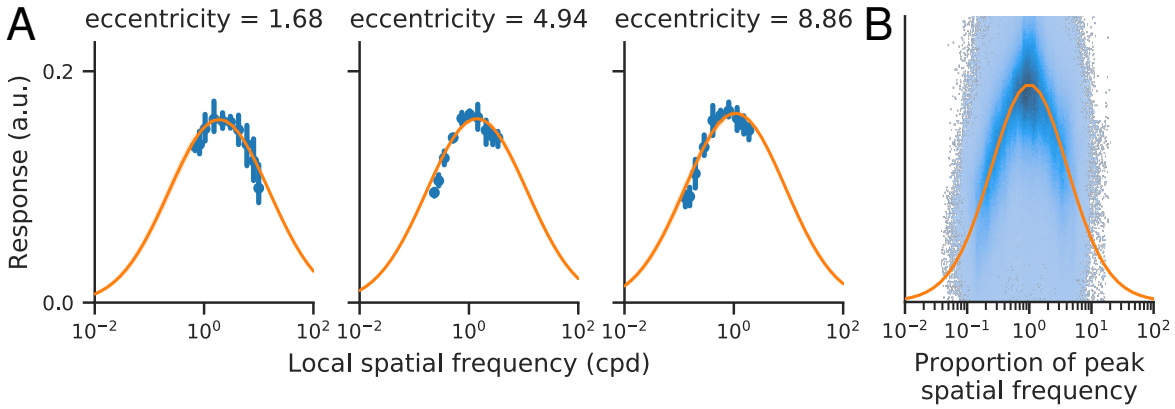


Figure 2.8: (A) Three example voxels from a single subject (sub-01). Blue points indicate median voxel responses across bootstraps. Error bars indicate variation as a function of orientation. Orange line shows model 9's predictions, in both cases as a function of the local spatial frequency at the center of each voxel's pRF. (B) Responses of all voxels across all subjects as two-dimensional histogram. For each voxel and stimulus orientation, responses are plotted as a function of spatial frequency, relative to peak spatial frequency. Orange line shows model 9's predictions.

To consolidate our findings, we combine the model parameters across subjects by bootstrapping a precision-weighted mean. For each parameter, we select 12 subjects with replacement, multiply

each subject's median parameter estimate by the precision of their response amplitudes (as estimated by GLMdenoise) averaged over all fit voxels, and average the resulting values. We then take this set of parameters and generate a set of predictions for the preferred period and gain across eccentricities and retinotopic angles, as well as for different stimulus classes (which determine the orientation seen by each voxel). We do this 100 times, plot the resulting median and 68% CI predictions in Figure 2.10, and plot the resulting median and 68% CI for the parameter values in Figure 2.9. We observe five distinct properties of the fitted functions:

Preferred period is an affine function of eccentricity. Specifically, the preferred period, as a function of eccentricity, is well approximated by a line with a significantly non-zero intercept. As discussed in the introduction, preferred period cannot decrease to zero at the fovea, since this would imply an infinite preferred spatial frequency. However, our stimuli do not include the region around the fovea, and thus our data do not constrain frequency tuning in that region. As such, the fitting procedure could potentially have arrived at an intercept of zero, supporting a "hinged line" model in which the preferred period decreases linearly with decreasing eccentricity and levels out at some minimal value, as proposed in Freeman and Simoncelli [70].

Preferred period is largest for annular stimuli. As also seen in the 1D analysis (section 2.4), the annular stimuli have the highest preferred period at each eccentricity (figure 2.10A, left). Unlike in the 1D analysis, we can now see that the difference between the annuli and pinwheel stimuli varies as a function of retinotopic angle, with the largest difference at the horizontal meridian, decreasing to almost 0 by the vertical meridian (figure 2.10A, top right). At the horizontal meridian, the median preferred period is 1.06 for annuli and 0.80 for pinwheels. This difference as a function of stimulus angle is equivalent to about 2 degrees of eccentricity at a constant stimulus orientation.

Preferred period is largest for vertical stimuli. A similar pattern is seen for the model predictions for horizontal and vertical stimuli, in which there is an overall difference, modulated

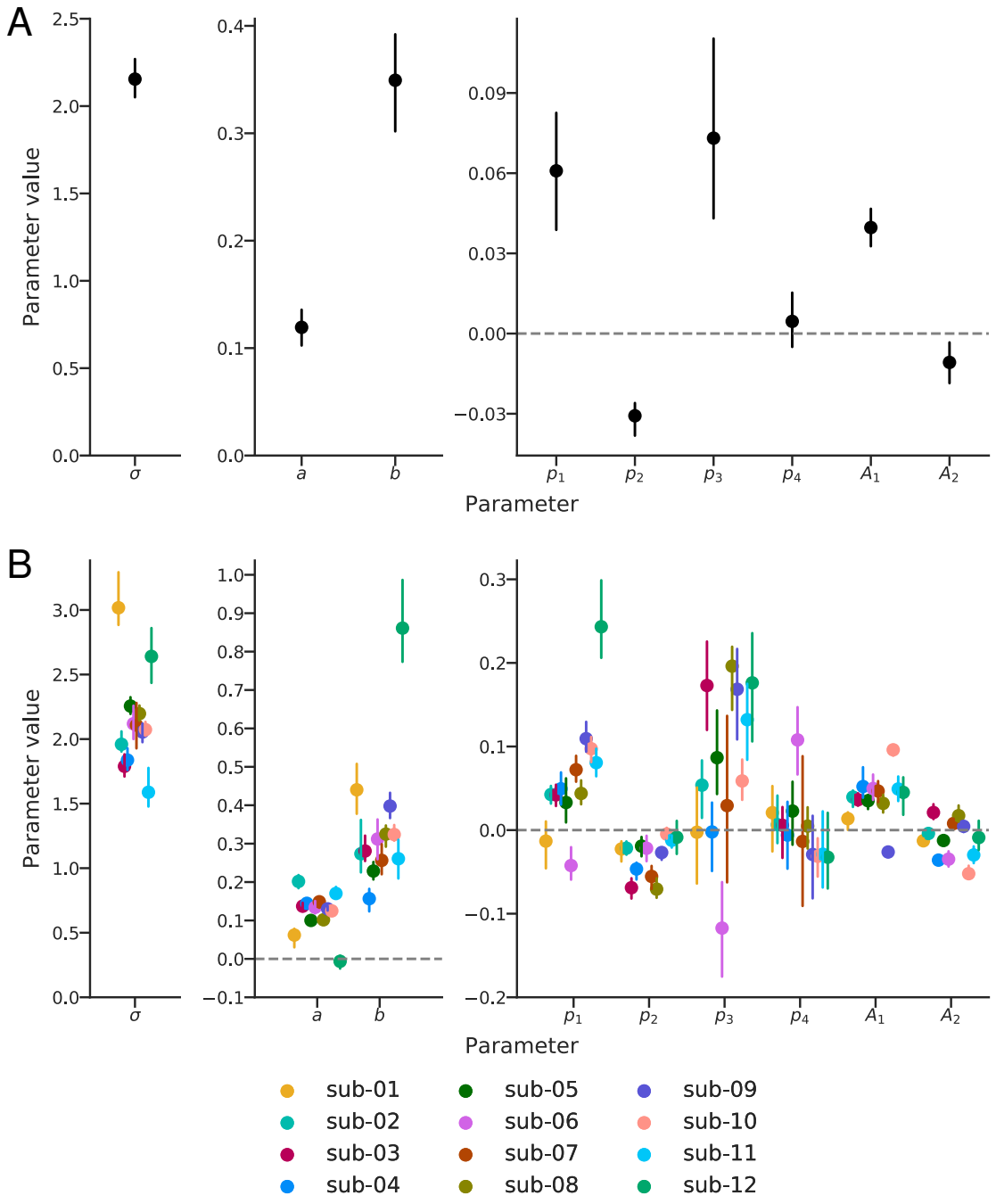


Figure 2.9: Parameter values (A) combined across all subjects and (B) in individual subjects. In both panels, median values ± 68% bootstrapped confidence intervals are plotted (note that A_3 and A_4 have been omitted, as determined from the previous model-selection analysis). (A) Parameter values obtained by bootstrapping parameter values across subjects from fits to the individual subject. A precision-weighted average is computed from each bootstrap. (B) Individual subject parameter values, bootstrapped across scans (as computed by GLMdenoise). A csv file containing these values (and instructions for use) can be found in the project software repository.

by retinotopic angle (figure 2.10B): The difference between their preferred periods reaches its maximal value at the horizontal meridian and decreases to almost 0 by the vertical meridian. This dependency between the preferred period effect and retinotopic angle comes from the combination of the vertical and annular biases: at the horizontal meridian, both go in the same direction (i.e., a vertical stimulus is an annular stimulus) and thus the gap in preferred period between vertical / annulus stimuli and horizontal / pinwheel stimuli is large. At the vertical meridian, on the other hand, they oppose each other (i.e., a vertical stimulus is a pinwheel stimulus), and, since the size of the two effects is roughly equal, the gap in preferred period between vertical / pinwheel stimuli and horizontal / annulus stimuli is small.

Gain is largest for vertical stimuli. The effect of stimulus orientation on gain is smaller than the effect on preferred period, but more consistent across subjects. According to the model fits, vertical orientations evoke the largest BOLD signal (highest gain) and horizontal orientations the lowest. The two diagonal orientations are intermediate. The forward and reverse diagonal stimuli do not differ in gain because model 9 does not fit parameters A_3 or A_4, which would differentiate them. The gain for annuli and pinwheels varies as a function of retinotopic angle, based on where they align with the absolute orientation. Thus, the annuli have the highest gain on the horizontal meridian (where their absolute orientation is vertical), the pinwheels have the highest gain on the vertical meridian (where their absolute orientation is vertical), and the spirals have the highest gain on their respective diagonals.

Spatial frequency tuning is broad. By examining Figure 2.9A, we see that the standard deviation (σ) of our model is about 2.2 octaves, equivalent to a full-width half-max of 5.1 octaves. (The variability in the estimate comes from bootstrapping across subjects and across runs, not from variation across voxels or stimulus orientation, neither of which we modeled.) The 2.2 octave standard deviation of the tuning function is large relative to the variation in peak tuning across the V1 map. For example, the difference in preferred period between a foveal voxel (0 deg eccentricity,

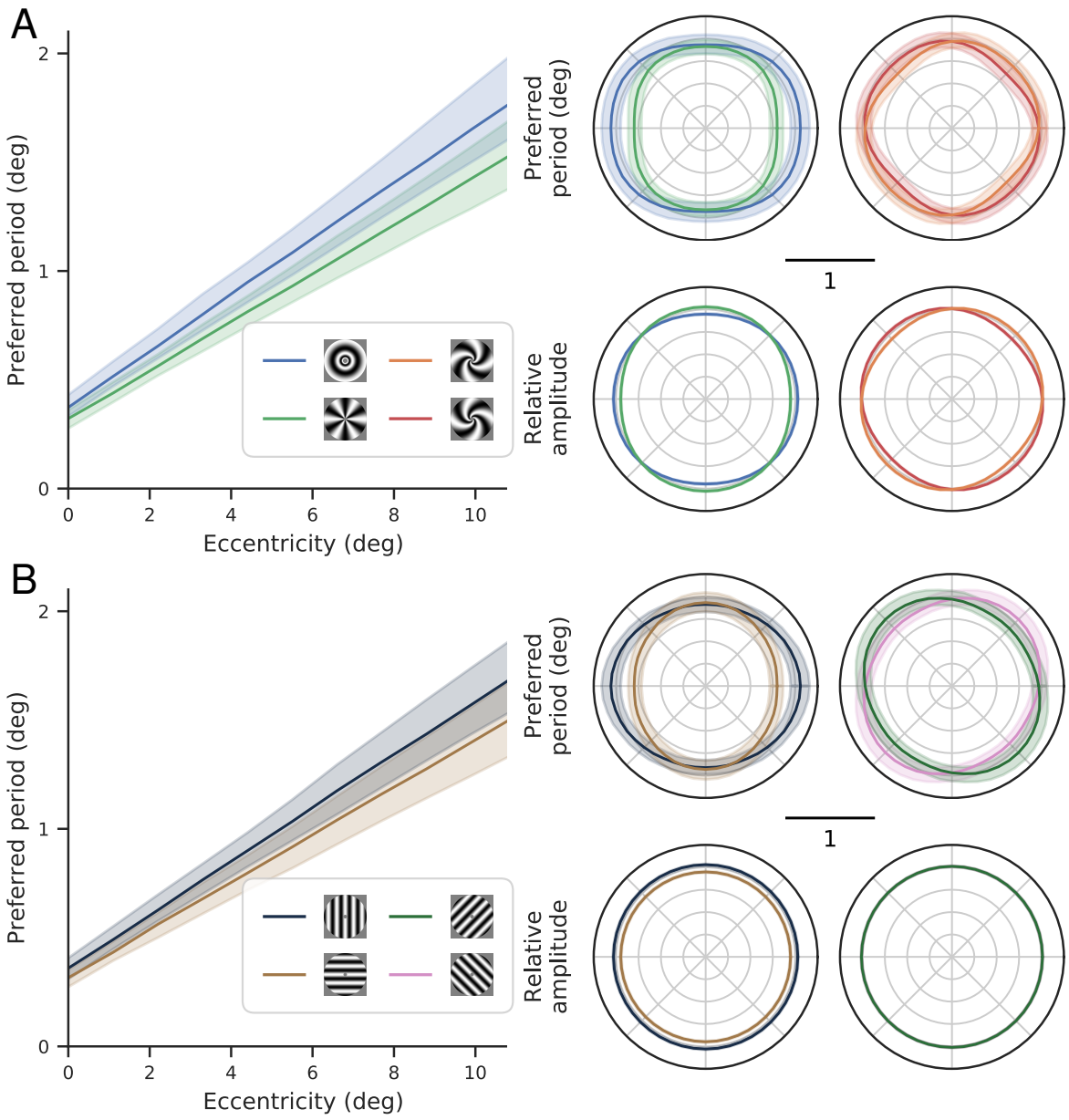


Figure 2.10: Spatial frequency preferences across the visual field in (A) relative and (B) absolute reference frames. In both panels, the left shows the preferred period as a function of eccentricity, top right shows the preferred period as a function of retinotopic angle at an eccentricity of 5 degrees, and bottom right shows the relative gain as a function of retinotopic angle (which does not depend on eccentricity; note that this relative gain does not change across voxels, only within a given voxel for different orientations). Only the extremal periods are shown in the left plot, for clarity (the others lie between the two plotted lines), and the cardinals and obliques are similarly plotted separately in the right plot for clarity. The predictions come from the model with parameter values shown in figure 2.9A, with the lines showing predictions from the median parameter and shaded region covering the 68% CI. Those parameters result from bootstrapping a precision-weighted average to combine the parameters from each subject's individual fit with this model. Compare left plot in panel (A) to figure 2.6B.

0.35 deg period) and a 10 deg voxel (about 1.6 deg period) is equivalent to 1 standard deviation of the foveal voxel's tuning function (2.2 octaves).

2.5.0.3 Preferred period is uncorrelated with V1 surface area

We observed substantial differences in preferred period across subjects. For example, at 6 deg eccentricity, preferred period ranges from 0.78 to 1.49 deg across our 12 subjects. A natural question is whether our measured preferred period is related to other functional or anatomical measures in V1. We motivated our initial scaling hypothesis by presenting the idea that the preferred spatial frequency may be a constant number of periods per population receptive field, and thus should drop as pRF size increases. Could the variability in pRF size across subjects account for the variability we see in preferred period? Estimated pRF size is far less reliable than pRF location, and so instead we compare preferred period to V1 surface area, which gives more robust estimates [98, 136]. The results can be seen in supplement figure 2.12, comparing the preferred period at 6 degrees eccentricity with the total V1 surface area across participants. Both values span a range of 2:1, but they are essentially uncorrelated with each other (R^2: median -3.42×10^{-3}, 68% CI: $[-2.84 \times 10^{-1},\ 9.75 \times 10^{-2}]$).

2.5.0.4 Effect of retinotopic angle

To keep the model parameterization tractable, we excluded effects of retinotopic angle on preferred spatial frequency (except as mediated via relative stimulus orientation). To get a sense for whether retinotopic angle alone has additional explanatory power in our dataset, we fit model 3 ($p = ar_v + b$, no effect of stimulus orientation and no modulation of gain) to the median BOLD response estimates on the quarters of the visual field around the two horizontal meridians ($\theta_v \in [0, \pi/4] \cup (3\pi/4, 5\pi/4] \cup (7\pi/4, 2\pi]$), and the quarters of the visual field around the two vertical meridians ($\theta_v \in (\pi/4, 3\pi/4] \cup (5\pi/4, 7\pi/4]$). The bootstrapped average across subjects of the preferred period as a function of eccentricity for these two variants is shown in figure 2.13A.

We can see that the model fit to voxels near the horizontal meridians has a higher preferred period near the fovea and a lower preferred period in the periphery, with the two meridian-only variants crossing at around 3 degrees. The error bars in that figure represent both the within-subject difference between the two variants and the between-subject differences in preferred period; figure 2.13B shows the difference between the two variants, calculated within subjects and then bootstrapped across them. This effect is clearly reliable across subjects, and the difference of approximately -.27 at 11 degrees is about 16% of the average preferred period there. Since our stimuli are balanced across relative stimulus orientations, this suggests that there is an effect of retinotopic angle alone on spatial frequency tuning, though further characterization is needed.

2.6 Discussion

We've used a set of log-polar grating stimuli to efficiently estimate spatial frequency preference in fMRI voxels of human V1. We quantified the effects of eccentricity, retinotopic angle, and stimulus orientation on voxel preferred period and response gain. As expected, the strongest relationship is the dependency on eccentricity: on average–across stimulus orientation, retinotopic angle, and subject–the preferred period is an affine function of eccentricity, which grows with a slope of about 0.12 degrees per degree of eccentricity and an intercept of about 0.35 degrees at the fovea. Preferred period is also modulated systematically by both stimulus orientation and retinotopic angle. Along the horizontal meridian, the increase in preferred period from horizontal to vertical stimuli (or, equivalently, from annular to pinwheel stimuli) is roughly equivalent to that seen when increasing eccentricity by 2 deg. On the vertical meridian, preferred periods of horizontal/vertical stimuli are indistinguishable. The response gain also exhibited small but systematic variations with stimulus orientation. Horizontal stimuli have an approximately 8% smaller response gain than vertical stimuli throughout the visual field.

2.6.1 Strengths

Our results are obtained using a multivariate, stimulus-referred model. Typically, stimulus-referred modeling of fMRI signals either fits each voxel independently (voxel-wise modeling) or fits average responses across regions. Voxel-wise modeling (e.g., [119]) has the flexibility of allowing researchers to place few or no constraints on the relationship of models across voxels. This flexibility comes with high parameter dimensionality: even a single visual area like V1, would typically require thousands of parameters, which can result in high noise sensitivity and lack of interpretability. Fitting models to regions of interest rather than voxels (e.g,. [24]) reduces dimensionality, but loses cortical (and thus, retinotopic) resolution. Our method combines positive aspects of both approaches: it is sensitive to variability in the response properties across voxels, while placing constraints on how the parameters relate to each other across voxels to generate a useful and interpretable summary.

An advantage of the stimulus-referenced modeling approach is generalization. A 2D model of spatial frequency tuning is likely to simplify development of a more complete image-computable model of the visual cortex. Some image-computable models fit to fMRI or electrocorticography responses operate only on band-pass filtered images because they do not incorporate spatial frequency tuning (e.g., [94, 117, 120]). The stimuli were band-passed in these experiments to reduce the complexity of the image space. There have been some attempts to generalize models across scale but these have not been informed by a comprehensive set of measurements or models of spatial frequency tuning [13, 165]. A further advantage of the multivariate parametric approach is that it helps reduce bias from skewed voxel sampling. For example, there are fewer voxels near the vertical than near the horizontal meridian [15]. In a voxel-wise fitting approach, preferences of voxels near the vertical meridian might be poorly fit or not fit at all (if no voxels have pRF centers along the meridian). Here, the parametric approach uses all the data to estimate each parameter, allowing better estimates for locations with limited data. Finally, a parametric model facilitates

comparison across studies, as other measurements of spatial frequency tuning might not sample the identical orientations, spatial frequencies, and visual field locations.

2.6.2 Limitations

Our modeling approach has at least two important limitations. First, the characterization of the V1 maps is based on fMRI measurements, which combine the integration of the fMRI measurement (blood oxygenation within voxels) and the selectivity of those neurons linked to the changes in blood oxygenation. Some aspects of our results, such as the substantial additive offset at the fovea, may be particularly affected by these additional sources of integration. Comprehensive measures of spatial tuning across the entire map at the level of individual neurons do not exist. Models that explicitly account for both tuning of individual neurons and measurement pooling functions, such as Haak, Cornelissen, and Morland [85] and Keliris et al. [122] will be important for clarifying the relative contributions of these sources.

Second, our analysis assumes that for each voxel and each stimulus, there is a single spatial frequency and orientation driving the response. Because both stimulus properties varied continuously across our images, this use of the instantaneous frequency approximation is only valid locally. We think the effects are likely small since V1 receptive fields are relatively small and our stimulus properties varied gradually. In later stages of the visual system, where receptive fields are substantially larger, this use of instantaneous frequency will become an increasingly worse approximation to the range of spatial frequencies within the receptive fields.

Finally, residual eye movements (microsaccades) could affect our results by increasing the positional uncertainty of the stimuli, or by effectively blurring them due to temporal integration. We think these effects are likely to be small (see supplement section 2.7.2 for more discussion), but we cannot entirely rule them out.

2.6.3 Related fMRI studies

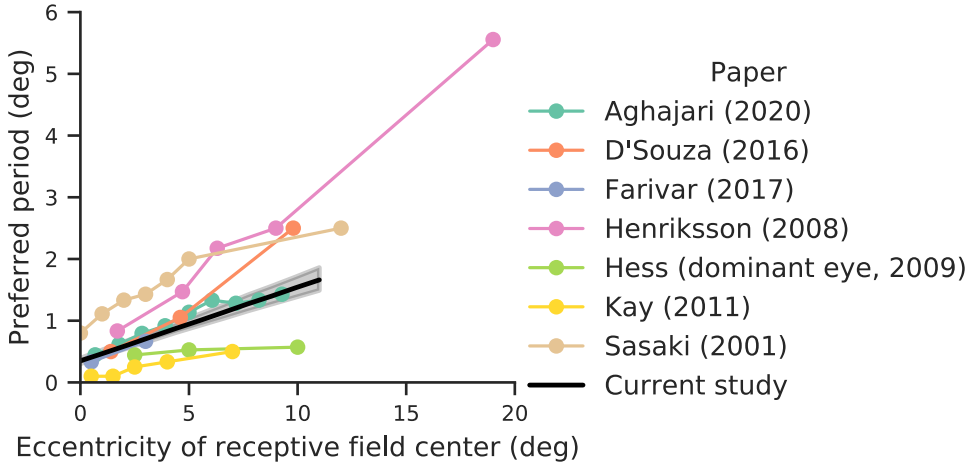


Figure 2.11: Comparison to previously reported eccentricity-dependence of spatial frequency measured with fMRI. All results show the preferred period at that eccentricity in V1 (all papers reported preferred spatial frequency; the reciprocal of that is shown). All values were estimated from published figures and are thus approximate. Black line represents our result, averaged across stimulus orientation and retinotopic angle, with line showing the median and shaded region the 68% confidence interval from precision-weighted bootstrap across subjects, as in figure 2.10.

A number of previous studies have reported spatial frequency preferences at multiple eccentricities in human V1 using fMRI. A comparison of those findings shows a wide range of estimates (figure 2.11; [3, 49, 65, 93, 189, 212]). With the exception of Aghajari, Vinke, and Ling [3] and Henriksson et al. [93], these studies did not pursue the question of V1 spatial frequency tuning as their main question. All studies agree that preferred period grows as an affine function of eccentricity, but the exact values for the slope and intercept vary widely. Overall, our results are most consistent with those of Aghajari, Vinke, and Ling [3]. These studies fit tuning curves to different voxels or bands of voxels and plotted the peak as a function of eccentricity (sometimes, as in Aghajari, Vinke, and Ling [3], also separately plotting this for different quadrants of the visual field), similar to our 1D fits shown in figure 2.6A. The variability across studies could be due to many factors, including display calibration, analysis methods, temporal frequency of stimulus presentation, and the wide variety of spatial patterns used, from natural images in "Understanding Visual Representation by Developing Receptive-Field Models" [212] to phase-scrambled noise

in Farivar et al. [65] to plaids in Hess et al. [95]. Resolving the discrepancies may require use of multiple stimulus classes and analysis methods in the same study.

Our 2D model assumes constant bandwidth in octaves. Aghajari, Vinke, and Ling [3] investigate the bandwidth of voxel spatial frequency tuning in more detail, concluding that it grows at a constant rate from approximately 3 octaves near the fovea to about 4.3 octaves at 9 degrees. Our model, like theirs, assumes a log-Gaussian tuning curve (figure 2.8B), but our bandwidth estimate of 5 octaves is larger than any of the values they observe in V1. We see no obvious explanation for the discrepancy. De Valois, Albrecht, and Thorell [54] measure spatial frequency bandwidth in macaque V1 simple and complex cells at multiple retinotopic locations and find a median bandwidth of approximately 1.5 octaves (similar across cell types and locations); they also show a negative correlation between peak spatial frequency tuning and spatial frequency bandwidth, with some low-pass neurons having a bandwidth of up to 3.25 octaves. Since neurons with a variety of spatial frequency tunings are found at any given retinotopic location, it is expected that V1 voxels would exhibit broader tuning than individual V1 neurons. This parallels findings in spatial receptive fields, which show larger sizes when measured for voxels with fMRI than measured in single units [59, 122].

2.6.4 Orientation tuning

There has been a long debate in the literature about whether orientation tuning is detectable in the BOLD signal on the spatial scale of voxels and, if so, what that means [39, 69, 116, 187]. Our model is recovering some degree of orientation tuning: with non-zero A_1 and A_2 values, response varies sinusoidally as a function of orientation. More specifically, we find an overall bias for vertical gratings. Freeman, Heeger, and Merriam [69] found a mix of vertical bias near the fovea and a radial bias (e.g., voxels along the horizontal meridian preferring horizontal gratings) in the periphery. While our model agrees with the first finding, we find no evidence for the second (though our model does not allow for categorically distinct responses in the fovea and periphery,

and fitting them separately may find some evidence for this, similar to the issue of retinotopic angle, see 2.5.0.4 and figure 2.13). However, we hesitate to interpret these results too strongly; as Carlson [39] and Roth, Heeger, and Merriam [187] point out, orientation biases can be induced in the BOLD response by the stimulus presentation even with unbiased underlying neuronal responses. Further work is needed to tease out the source and implication of orientation tuning.

2.6.5 Scale and rotation invariance

An idealized model of visual system organization is that spatial frequency tuning (preferred period) is proportional to eccentricity, while being independent of polar angle and stimulus orientation. For example, the log-polar model of the warping of the visual field onto the V1 cortical surface by Schwartz [192] has these properties. The scaling with eccentricity has been proposed by Schwartz and others [214] to endow the system with invariance to dilation and rotation (for transformations centered at the fovea), enabling perceptual generalization (but see Cavanagh [41] for a different interpretation). Our model fits show systematic deviations from each of these three properties.

First, we find that preferred period grows as an *affine* function of eccentricity, with a non-zero intercept. Independent of any measurements, one would not expect basic properties such as receptive field size to grow proportional to eccentricity due to limits at the fovea (the optics and cone apertures set upper bounds on resolution.) One simple correction to the idealized scaling model is adding an offset, or affine transform, as we have done here. This is consistent with some models of cortical magnification in V1 [14, 100]. An alternative model form is piece-wise linear (e.g., a "hinged line"), that is flat in the vicinity of the fovea, and grows proportional to eccentricity beyond that (as used by Freeman and Simoncelli [70] to describe ventral stream receptive fields). This allows scale invariance outside the flat, foveal region. Our data are better fit by an affine function than a hinged line. The effect is relatively large: the offset at the fovea (preferred period of 0.35 deg) is equivalent to the difference in preferred period between 0 and 3 deg eccentricity. A

substantial offset implies that the human V1 representation in the center of the visual field does not approximate a scaling rule, as also noted by Cavanagh [41]. Given the importance of foveal vision for object recognition, the deviation from an idealized scaling rule at the fovea may have important implications for perception. Size judgments are in fact not invariant to eccentricity [156] and have been shown to track individual differences in the topography of V1 [151].

Second, we show spatial frequency tuning depends on orientation at the horizontal meridian, but not at the vertical meridian (see Figure 2.10, right panel). This is because the preferred period tuning for absolute orientation (vertical > horizontal) and for relative orientation (annuli > pinwheels) add for locations on the horizontal meridian, but cancel for locations at the vertical. In separate analyses, we also observed an overall higher peak spatial frequency for visual field quadrants near the horizontal meridian than the vertical outside of the central 3 deg, consistent with Aghajari, Vinke, and Ling [3]. These results suggest that the quality of spatial representation will depend on polar angle. This is consistent with a large body of psychophysical results showing that performance on various tasks, including spatial resolution and contrast sensitivity, depend on stimulus polar angle, with better performance along the horizontal meridian than the vertical meridian and better performance along the lower vertical meridian than the upper vertical meridian (see Himmelberg, Winawer, and Carrasco [97] and the citations therein).

Finally, we show an overall annular bias in preferred spatial frequency: for any location in the visual field, an annular stimulus will have the lowest preferred spatial frequency, this bias varies across retinotopic angle, and increases with eccentricity. Few studies have examined the combination of stimulus orientation and retinotopic angle with sufficient resolution to determine whether an orientation effect is relative or absolute. An exception is Wilkinson et al. [230], who used interference fringes to examine sinusoidal grating acuity changes across the visual field, and found that it is proportional to the sampling of retinal ganglion cells everywhere in the retina. Consistent with our study, they show that radial acuity is always higher than tangential acuity, that this effect is largest along the nasal horizontal meridian, and that the minimum angle of

resolution (.5 / cutoff spatial frequency) grows roughly linearly with eccentricity. All told, this suggests that many, but not all, of the effects observed in the current study originate with the sampling of the midget retinal ganglion cell lattice.

2.7 Appendix

2.7.1 Stimulus properties

The local spatial frequency of our stimuli is equal to the magnitude of the gradient of the argument of cos(·) in Equation 2.1. Writing that argument as $g(r, \theta) = \omega_r \ln(r) + \omega_a \theta + \phi$, we differentiate to obtain the horizontal/vertical spatial frequency:

$$\omega_x = \frac{\partial g}{\partial x} = \frac{\partial g}{\partial r}\frac{\partial r}{\partial x} + \frac{\partial g}{\partial \theta}\frac{\partial \theta}{\partial x} = \frac{x\omega_r - y\omega_a}{x^2 + y^2} \tag{2.10}$$

$$\omega_y = \frac{\partial g}{\partial y} = \frac{\partial g}{\partial r}\frac{\partial r}{\partial y} + \frac{\partial g}{\partial \theta}\frac{\partial \theta}{\partial y} = \frac{y\omega_r + x\omega_a}{x^2 + y^2}. \tag{2.11}$$

The local spatial frequency is then

$$\begin{aligned} \omega_l &= \sqrt{\omega_x^2 + \omega_y^2} \\ &= \frac{\sqrt{(x^2 + y^2)(\omega_r^2 + \omega_a^2)}}{x^2 + y^2} \\ &= \frac{\sqrt{\omega_r^2 + \omega_a^2}}{r}. \end{aligned} \tag{2.12}$$

Thus, the local spatial frequency is proportional to the magnitude of the base frequency vector (ω_a, ω_r), decreasing as the inverse of eccentricity (r). To convert this from radians per pixel to cycles per degree, we multiply by a conversion factor, $c = \frac{1}{2\pi}\frac{\text{size}_{\text{pix}}}{\text{size}_{\text{deg}}}$. We use this measure of local spatial frequency when fitting tuning curves.

We can similarly find the local stimulus orientation, θ_l, by computing the angle of the frequency vector (ω_x, ω_y):

$$\begin{aligned}\theta_l &= \arctan\frac{\omega_y}{\omega_x} \\ &= \arctan\frac{y\omega_r + x\omega_a}{x\omega_r - y\omega_a} \\ &= \theta + \arctan\frac{\omega_a}{\omega_r}\end{aligned} \tag{2.13}$$

where $\theta = \arctan\frac{y}{x}$. The local stimulus orientation is thus the sum of the angular location θ and the angle of the base frequency vector (ω_r, ω_a).

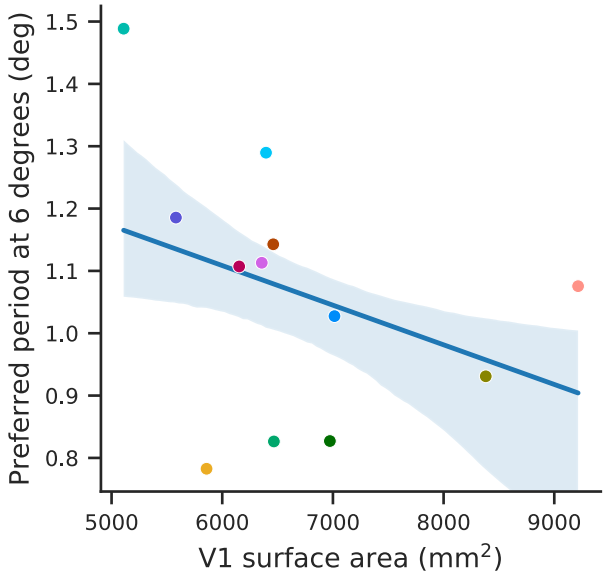


Figure 2.12: Scatter plot showing the preferred period at 6 degrees eccentricity against the total V1 surface area (both hemispheres) for all subjects. A middling eccentricity value was chosen so that the effects of both parameters a and b are visible. The two variables are essentially uncorrelated (line shows the median and the shaded region the 68% CI linear regression, bootstrapped across subjects). Subject colors are as in figure 2.9B.

2.7.2 Behavior

Behavioral data is plotted in figures 2.14 and 2.15, combining across subjects and plotting them separately, respectively. When combining across subjects, there is no consistent pattern between performance and stimulus type: all stimulus types show similar behavior. When looking on a subject-by-subject basis, about half of the subjects show some differences across stimulus types.

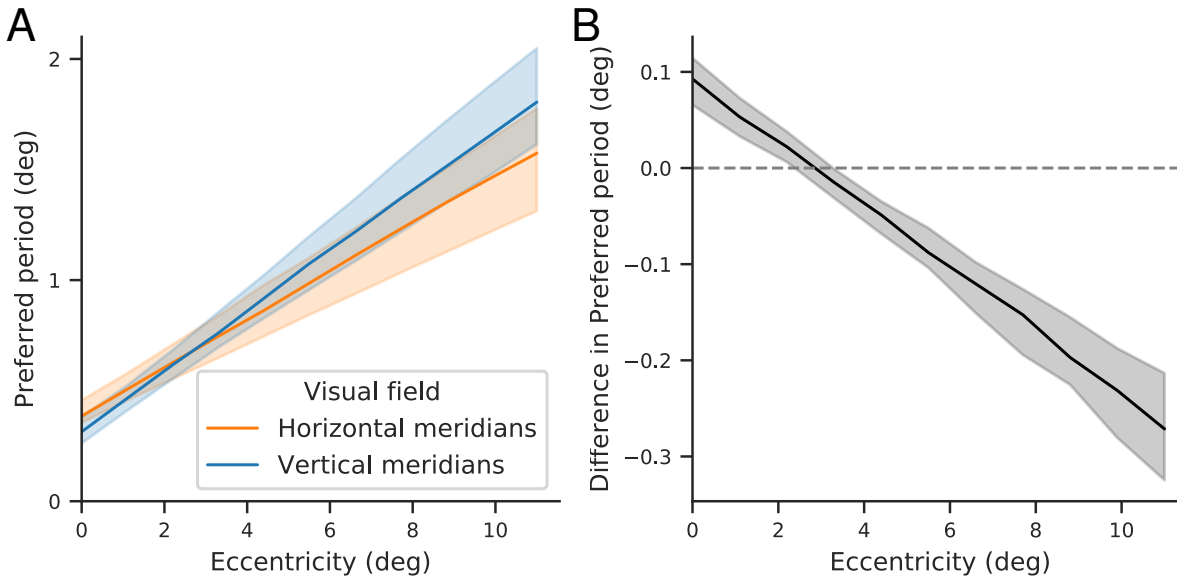


Figure 2.13: Voxels near the vertical meridians have a higher preferred period in the periphery and a lower preferred period near the fovea. Model 3 ($p = ar_v + b$, no effect of stimulus orientation or modulation of gain) was fit to each subject's median estimates of BOLD response. (A) Preferred period as a function of eccentricity for the two portions of the visual field. Lines and shaded region show the median and 68% CI from combining the preferred period across subjects by bootstrapping a precision-weighted average; uncertainty thus reflects both between-subject variance of preferred period and within-subject variance of the two visual field segments. (B) Difference between the preferred period in the voxels near the horizontal and vertical meridians, calculated within subjects, and then combined by bootstrapping a precision-weighted average.

We might then worry that differences in eye movements or fixation stability may affect our results, such that any differences between spatial frequency tuning for annuli and pinwheels, say, are actually the result in differences in eye movements between those conditions. However, there appears to be no relationship between these behavioral patterns and the parameter values plotted in figure 2.9B. For example, sub-01 and sub-08 show similar behavioral patterns, with the highest miss rates for pinwheels, followed by forward spirals, then annuli and reverse spirals. However, their parameter values are not more similar to each other than to any other subject's, making it unlikely that our parameter fits reflect differences in eye movement across stimuli.

Similarly, one might worry that differences in stimulus-independent eye movements might affect our results: fixational eye movements are known to be more common along the horizontal than vertical meridian, occuring at a rate of about 6 per minute [207], with microsaccades in both directions having a median amplitude of 20 arcmin when there is a fixation target [44]. These increase the uncertainty of the spatial frequency within a voxel's pRF, but, whenever voxels whose pRFs are on the right horizontal meridian see an increase in eccentricity due to a horizontal eye movement, voxels on the left horizontal meridian will see a decrease in eccentricity (and vice versa). Since our model does not allow left/right asymmetries in the fits, these two effects would approximately cancel. Moreover, even if they did not fully cancel, microsaccades are relatively small compared to the observed effects of orientation (equivalent to about a 2 deg shift in eccentricity).

In addition to the uncertainty in voxel location discussed above, microsaccades combined with temporal averaging may blur the stimulus slightly, suppressing the high frequencies in our stimuli, which may shift our measured tuning curves to slightly lower frequencies. This effect would be most pronounced for those stimuli whose period is the same magnitude as the eye movements, which are present in our stimuli. This would increase the preferred periods on our plots and would have a larger effect on voxels at lower eccentricities. Eliminating or fully accounting for this effect is impossible given our setup, and future studies are necessary to account for its magnitude. Given

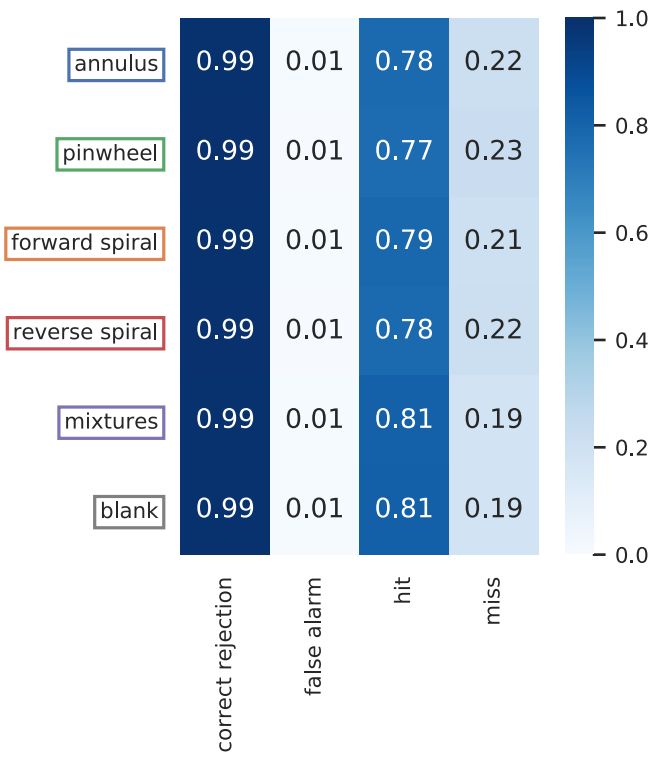


Figure 2.14: Summary of behavior combined across subjects. Stimulus type is indicated along the vertical axis ("blank" means there was no stimulus on the screen; these trials were interleaved throughout the scan as well as present at the very beginning and end), and outcome is indicated on the horizontal axis, with numbers giving the percentage of trials that fall into that category. Percentages and color are normalized so that the sum of correct rejections and false alarms is 1, as is the sum of hits and misses. During scans, subjects viewed a pseudo-random stream of digits at fixation and their task was to press a button whenever the digit repeated, which it did on one-sixth of the trials (the same digit was never shown three trials in a row). Behavior was consistent across stimulus types.

the small size of fixational eye movements, we think their effects are likely to be small, but we cannot entirely rule them out.

2.7.3 Individual fits

Figures 2.16 through 2.22 show the individual subject fits for preferred period as a function of eccentricity from the 1d analysis (figure 2.16); preferred period as a function of eccentricity from the 2d model for relative (figure 2.17) and absolute (figure 2.18) reference frames; preferred period as a function of retinotopic angle (at 5 degrees eccentricity) for relative (figure 2.19) and absolute (figure 2.20) reference frames; and the relative gain as a function of retinotopic angle for

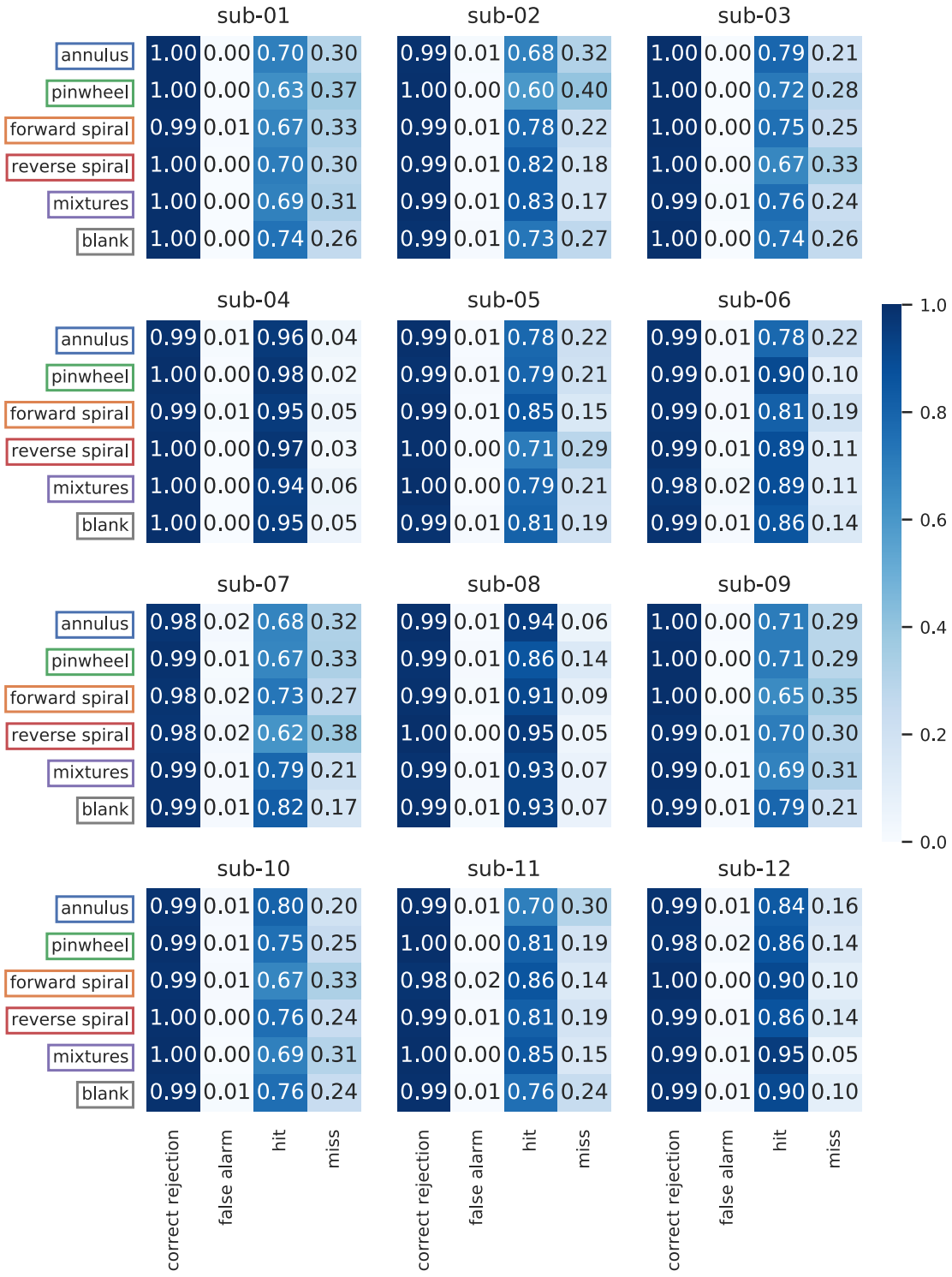


Figure 2.15: Summary of behavior on a per-subject basis. For details, see caption of 2.14. Performance varies across subjects (though false alarm rates are consistently low), but as in 2.14, there is no consistent difference in behavior across stimulus types.

relative (figure 2.21) and absolute (figure 2.22) reference frames. In all, we show the median and 68% confidence intervals obtained from bootstrapping across that subject's fMRI runs.

Note that sub-12's results are an outlier: their preferred period does not change as a function of eccentricity (also visible in the parameter plots in figure 2.9B; their $a = 0$). The noise in their GLMdenoise fits does not suggest any problems with the quality of this data, and the quality of their retinopic maps is also consistent with the other subjects. Therefore, they have been included in the analyses presented in this paper.

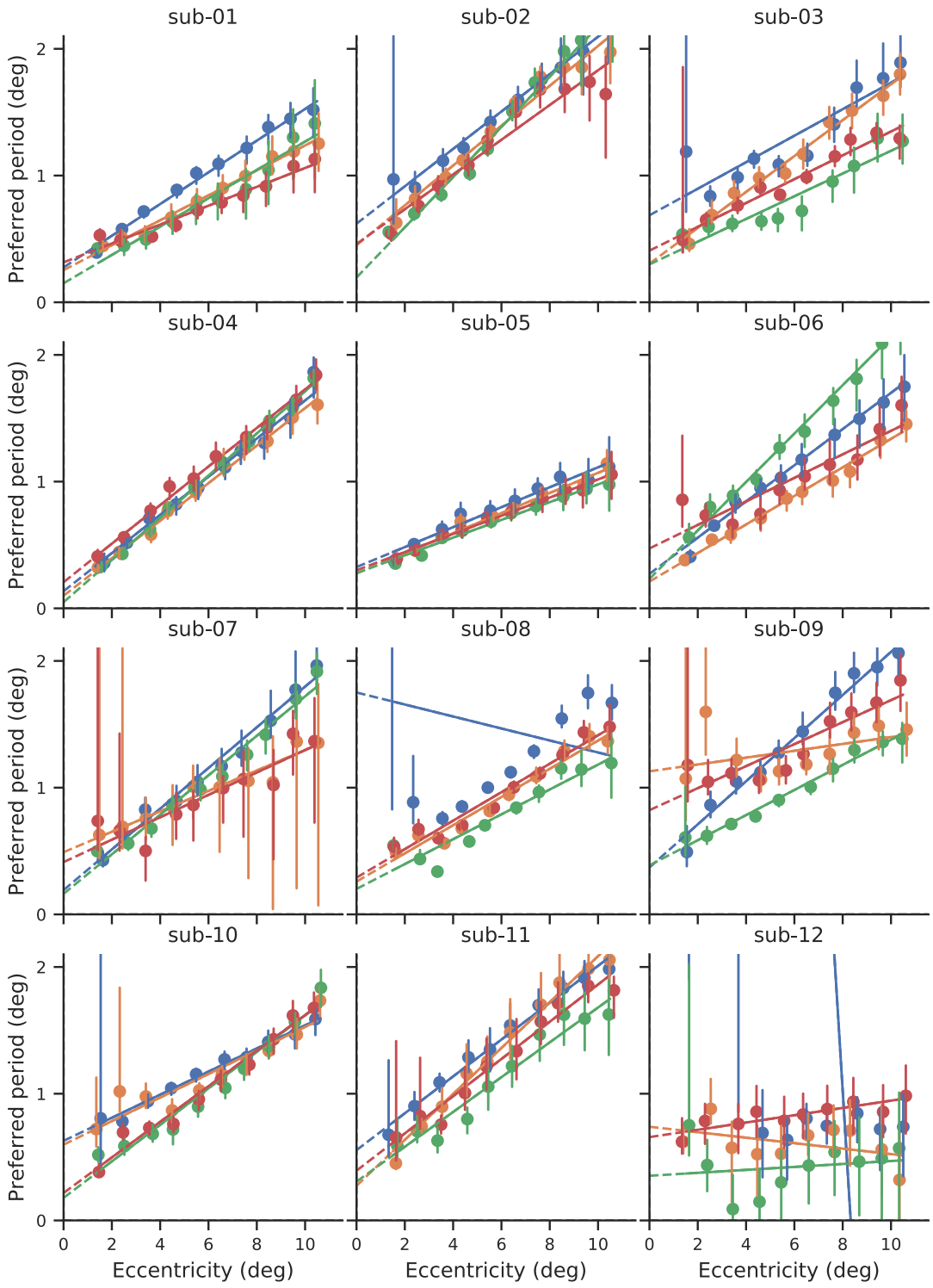


Figure 2.16: Individual subjects' preferred period as function of eccentricity from 1d fits (as in figure 2.6A), for different stimulus classes. Points and vertical bars indicate the median and 68% confidence interval obtained from bootstrapping across fMRI runs.

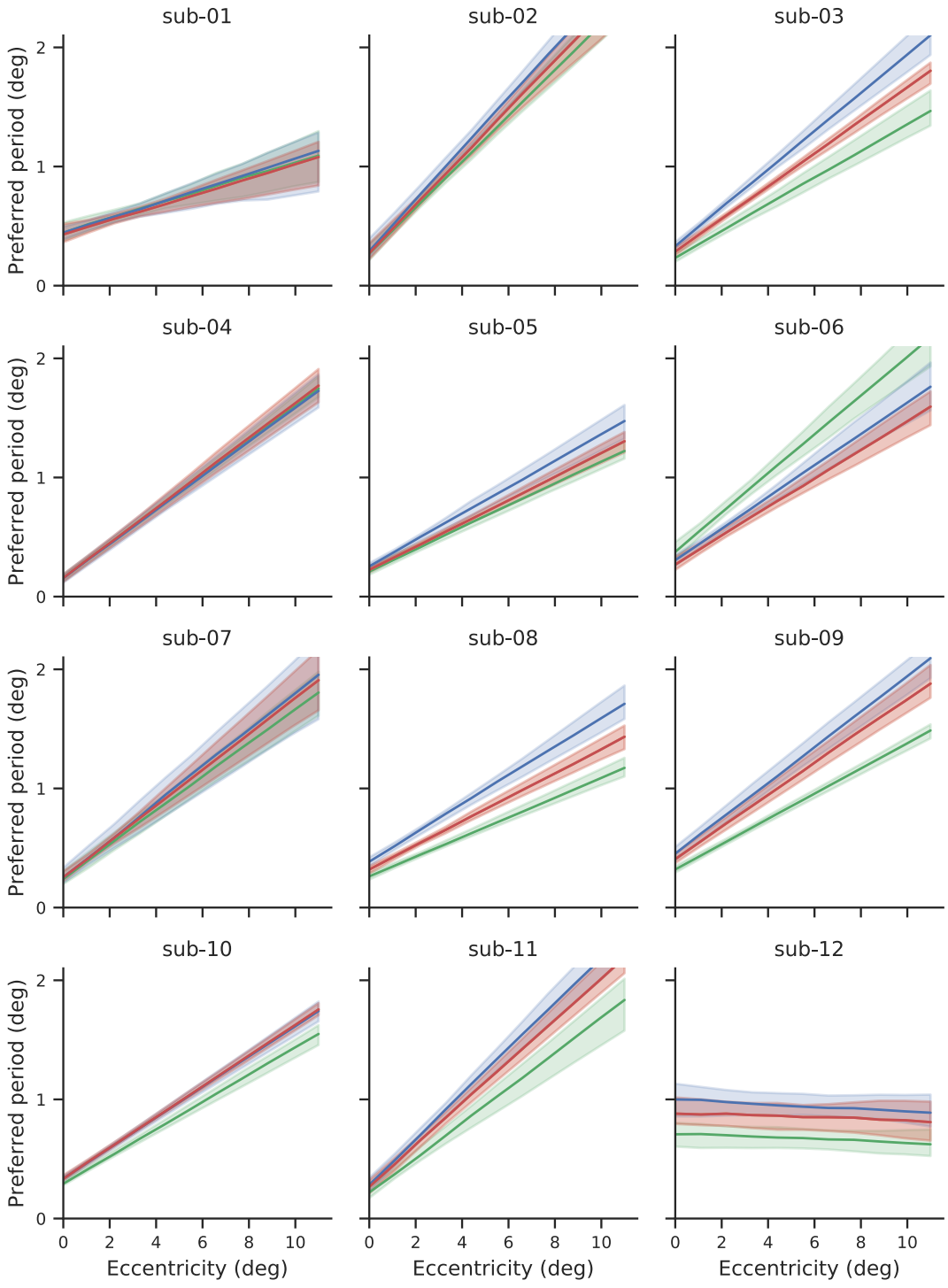


Figure 2.17: Individual subjects' preferred period as function of eccentricity from 2d model for relative reference frame (as in left panel of figure 2.10A). Averaged across all angles, lines show the median parameter and shaded regions cover the 68% confidence intervals obtained from bootstrapping across fMRI runs.

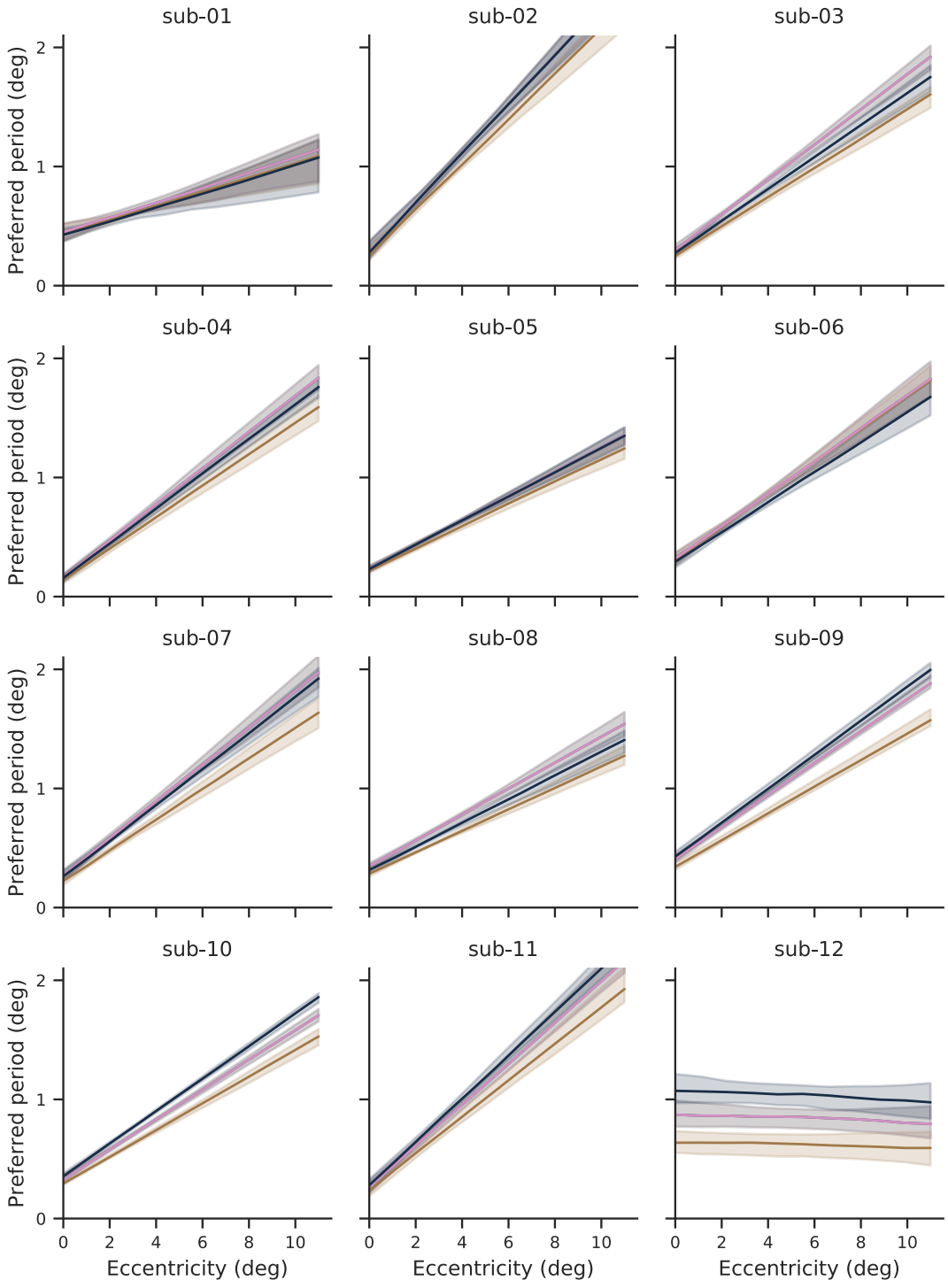


Figure 2.18: Individual subjects' preferred period as function of eccentricity from 2d model for absolute reference frame (as in left panel of figure 2.10B). Averaged across all angles, lines show the median parameter and shaded regions cover the 68% confidence intervals obtained from bootstrapping across fMRI runs.

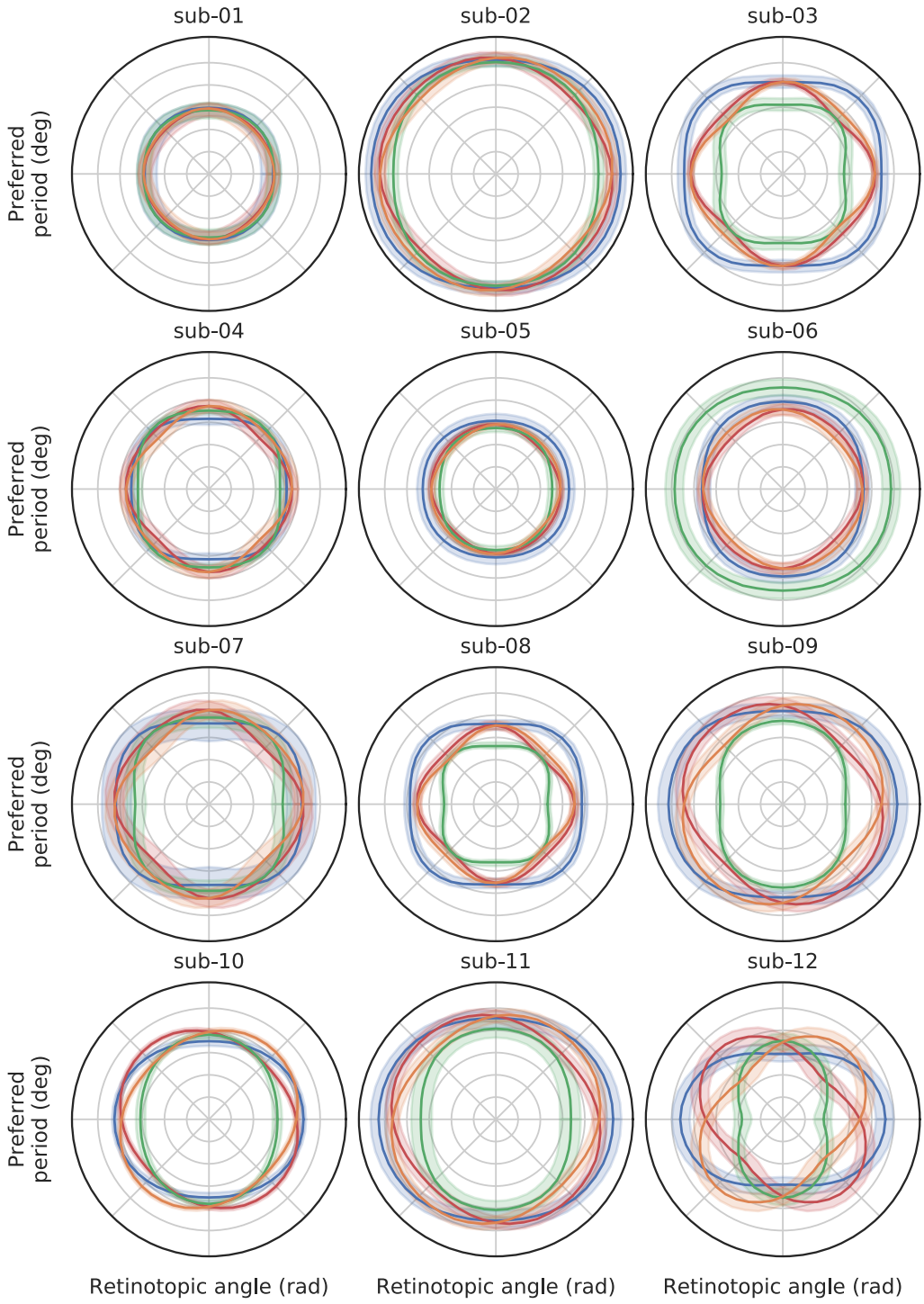


Figure 2.19: Individual subjects' preferred period as a function of retinotopic angle at an eccentricity of 5 degrees for relative reference frame (as in top right panel of figure 2.10A). Lines show the median parameter and shaded regions cover the 68% confidence intervals obtained from bootstrapping across fMRI runs.

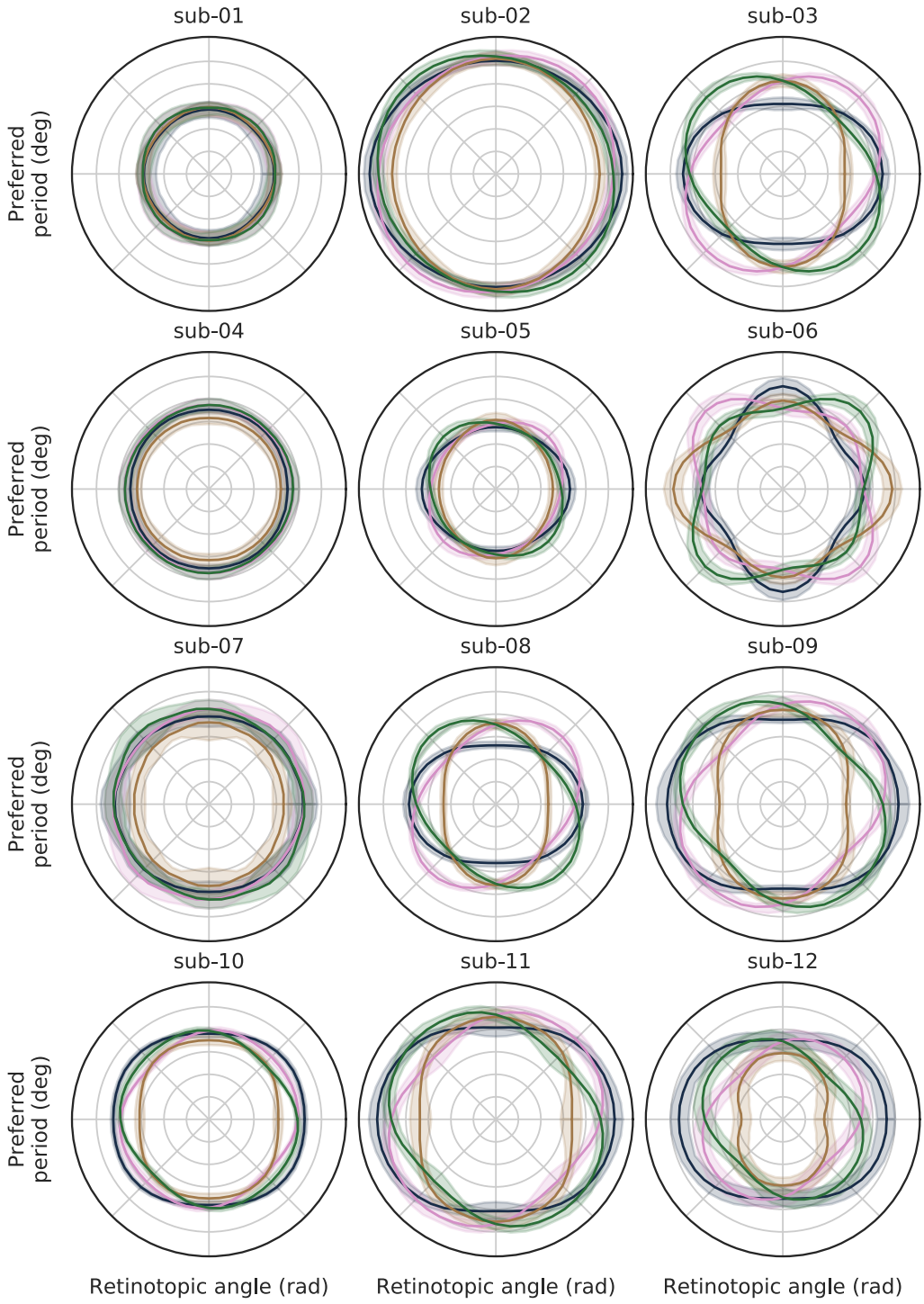


Figure 2.20: Individual subjects' preferred period as a function of retinotopic angle at an eccentricity of 5 degrees for absolute reference frame (as in top right panel of figure 2.10B). Lines show the median parameter and shaded regions cover the 68% confidence intervals obtained from bootstrapping across fMRI runs.

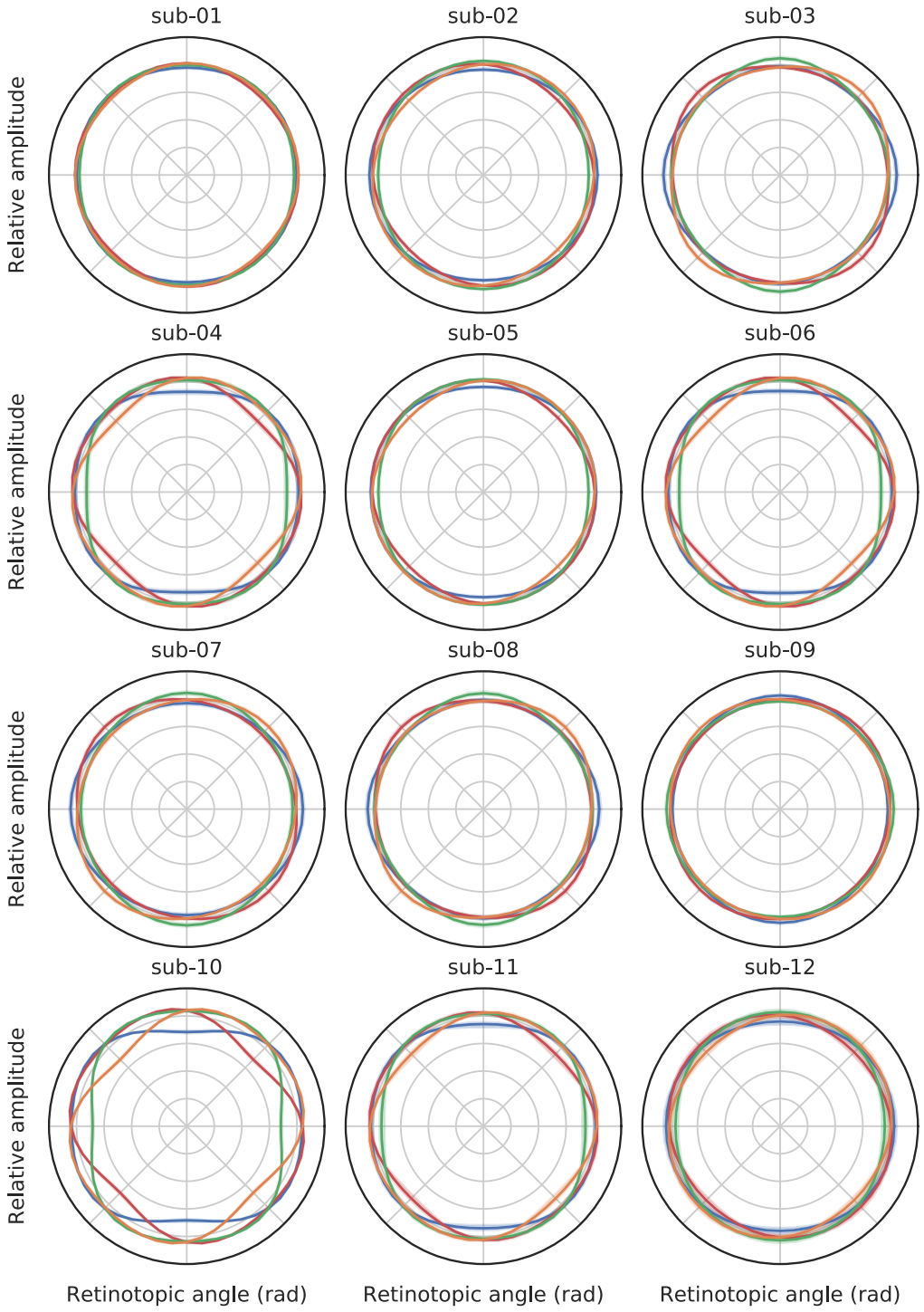


Figure 2.21: Individual subjects' relative gain as a function of retinotopic angle for relative reference frame (as in bottom right panel of figure 2.10A). Lines show the median parameter and shaded regions cover the 68% confidence intervals obtained from bootstrapping across fMRI runs.

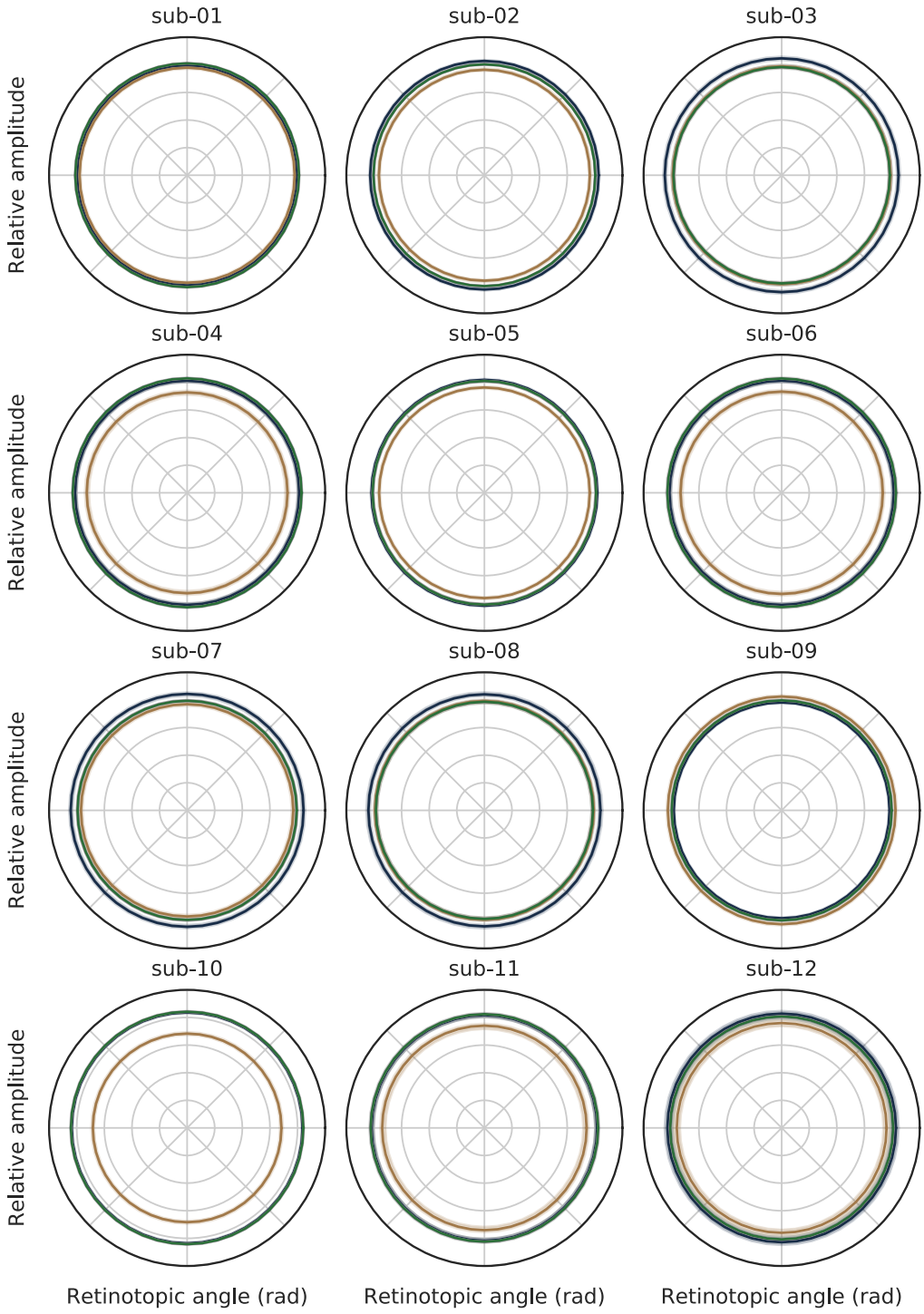


Figure 2.22: Individual subjects' relative gain as a function of retinotopic angle for absolute reference frame (as in bottom right panel of figure 2.10B). Lines show the median parameter and shaded regions cover the 68% confidence intervals obtained from bootstrapping across fMRI runs.

3 Foveated metamers of the early visual system

3.1 Abstract

Human abilities to discriminate and identify many visual attributes vary across the visual field, and are notably worse in the periphery compared to the fovea. Much work investigating these phenomena focuses on either the earliest stages of vision (optics and receptor sampling) or high-level cortical stages, but two important intermediate processes, spatial pooling of luminance and the computation of local spectral energy, likely contribute to differences in performance across the visual field as well. We built luminance and energy pooling models, which average the relevant statistic in pooling windows whose diameters grow linearly with eccentricity, used psychophysical experiments to measure the window size where human and model abilities to discriminate match, and compared to physiological values from retinal ganglion cells (RGCs) and primary visual cortex (V1), the brain areas most often associated with luminance and energy pooling. We do so using much larger stimuli than previously used, subtending 53.6 by 42.2 degrees of visual angle. We found the critical scaling for the luminance model was about four times smaller than that of the energy model, and, consistent with earlier studies, that a smaller critical scaling value was required when discriminating a synthesized image from a natural image than when discriminating two synthesized images. Our results quantify the link between image statistics

and the spatial scale of pooling, and raise questions about what makes some image pairs easy to discriminate and some hard, when both sets are synthesized with the same image statistics and scaling size.

3.2 Introduction

Vision science is often concerned with what things look like (appearance), but a long and fruitful thread of research has investigated what humans cannot see, that is, the information they are insensitive to. Perceptual metamers, images that are physically distinct but perceptually identical, are used to demonstrate information that is lost in visual processing. Their use dates back to the mid-19th century, when they were instrumental in the development of the Young-Helmholtz theory of trichromacy [91]. These color metamers clarified human sensitivity to light wavelengths, demonstrating that the human visual system projects the infinite dimensionality of light to three dimensions, though it would be more than a century before the physiological basis for this was discovered: when the outputs of the three cone classes are matched, differences in the wavelength composition of stimuli are invisible.

The visual system also discards a great deal of information about the spatial properties of images. In this paper, we will examine human insensitivity to spatial information, which has a strong eccentricity component: human ability to precisely resolve spatial features decreases with retinal eccentricity, as demonstrated by decreasing acuity and increasing crowding distance, among other measures. The loss of spatial information likely arises from pooling mechanisms that are downstream from the photoreceptors, including retinal and cortical circuitry. To model this decreasing sensitivity to spatial information with eccentricity, we built two "pooling models" of the early visual system, which average image statistics in radially-oriented windows that grow larger with eccentricity [70, 123]: one that approximates retinal ganglion cell spatial pooling by averaging local luminance (**luminance model**) and one that approximates primary visual

cortical neurons by averaging local spectral energy and luminance (**energy model**). These models represent the periphery as qualitatively similar to the fovea: the same computations are performed, just over a larger area. We test these models using **model metamers** (images that are physically distinct but with identical model representations) in a psychophysical experiment to determine the largest parameter value for which model metamers are also perceptual metamers. This way of model testing allows us determine whether the models' sensitivities and insensitivities match that of the human visual system, a stricter test than only checking sensitivities.

This procedure rests on the assumption that information processing in the visual system is approximately hierarchical: information discarded at an early stage cannot be recovered later. Color metamers produce identical cone responses and are thus indistinguishable. Analogously, if two images generate identical responses in all retinal ganglion cells or primary visual cortical neurons, they will be perceptually identical. Much work has been done to create perceptual metamers by matching models of neural activity at a high level of the visual processing hierarchy (extra-striate ventral areas, [66, 70, 109, 219]) and at a low level (photoreceptors). However, there has been much less work creating perceptual metamers based on models of the levels in between.

Even for a given model, previous studies [31, 57, 219] have shown that the extent of maximal spatial pooling able to generate perceptual metamers depends on the two images being compared. First, if an image of a natural scene is being compared to a synthesized image, the pooling windows must be smaller than if two synthesized images are being compared. Second, even within these comparisons, the performance depends on the content of the natural scene. Neither of these effects are captured by the models.

Here, we synthesize model metamers and measure their perceptual discriminability. We build models that pool two types of image statistics, luminance and spectral energy, to capture some of the processing between photoreceptor outputs and ventral stream representations. Moreover, we test discriminability between different types of image pairs, including images of natural scenes vs. synthesized images and pairs of synthesized images. The two types of models and multiple types of

comparisons shed light on and raise new questions about what makes such images distinguishable.

3.3 Methods

All experimental materials, data, and code for this project can be found online under the MIT or similarly permissive licenses. Code is found on GitHub and a browser of synthesized metamers online. View GitHub README for instructions on how to download and use data.

3.3.1 Metamers

In order to investigate the extent to which our foveated models capture human perception, we synthesized sets of **model metamers**, physically distinct images with identical model representations [70]. Such images allow us to better understand model representation and facilitate comparisons with human perception [66]. While many model testing frameworks probe model representations, model metamers highlight the model's null space as well, as the resulting metamers combine the features of the target image that the model is *sensitive* to with features of the initialization image that the model is *insensitive* to. A useful model of human perception will share not just human sensitivities, but human invariances as well, and to effectively evaluate this our test images must not just throw out such information, but randomize it as well. This way of approaching the problem has the opposite feel of Bayesian inference or the use of a natural image prior, which seek to find the most likely images that gave rise to a model representation; for metamer synthesis, we want to explore the space of possible images much more widely and pay particular attention to the *least* natural images, as these provide a more stringent test of the model.

We use these model metamers in a psychophysics experiment (3.3.5) in order to determine, for each model, the largest parameter value at which these model metamers are perceptual metamers; that is, we find the images that are the most physically distinct while being perceptually identical under our experimental conditions.

Throughout this paper, we will refer to the model metamers as such when discussing synthesis or model details, and as "synthesized images" when discussing psychophysics and human perception, in order to avoid confusion with perceptual metamers, which are a subset of those synthesized images.

3.3.2 Models

We built foveated models of human perception that approximate the computations performed in primary visual cortex (V1) and retinal ganglion cells (RGC), figure 3.1. Both models are "pooling models" [7, 70, 123, 219], which compute statistics across the image and then take the weighted average in pooling windows. These pooling windows model neuronal receptive fields: they are laid out in a polar fashion and grow as you move away from the fovea. Like previous studies [70], our pooling windows are overlapping and radially-elongated (twice as long in the radial direction), but unlike previous studies, we use Gaussians for our pooling windows, which overlap more and thus give a smoother representation, which resulted in higher-quality synthesized images. As with previous pooling models, the statistics the models compute are chosen to correspond to the representation of a visual area and are thus held fixed, while the rate at which the pooling windows grow in size is the model's only free parameter. This is the model's **scaling** and gives the slope between the pooling window diameter at full-width half-max in the radial direction and the location of its center, both in degrees. Thus, the pooling windows of a model with scaling .1 will have a diameter of 1 at 10 degrees eccentricity, 2 at 20 degrees, etc.

Scaling can also be used to describe how neuronal receptive fields [70] and voxel population receptive fields [222] grow with eccentricity. Pooling models have been proposed to identify visual areas when their scaling values and computed statistics roughly match those of the corresponding area, with the logic that, if a model and a visual area are computing the same summary statistics of an image in the same size pooling regions, pairs of model metamers should be perceptual metamers as well, because the feedforward outputs of the region should be identical. Thus, Freeman and

Simoncelli [70] proposed that their mid-ventral model, which pooled Portilla-Simoncelli texture statistics [178], provided evidence of texture representation in V2, as their scaling value of 0.5 matched the phsyiological scaling found in V2 (but see Wallis et al. [219] for additional nuance).

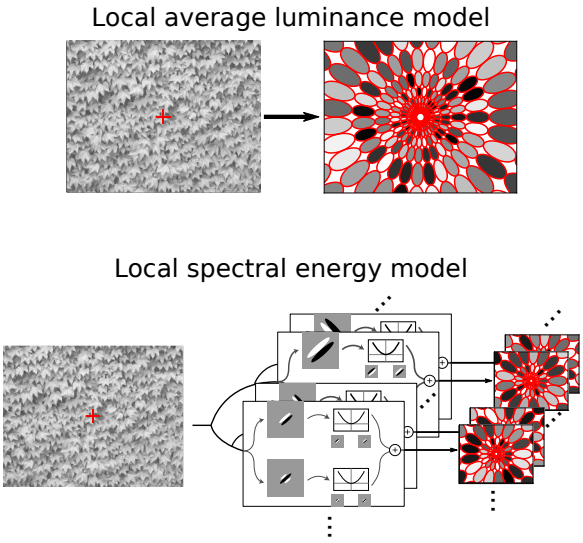


Figure 3.1: Models pool image statistics in radially-oriented Gaussians whose width grows linearly with eccentricity, the rate of which, **scaling**, is the models' only free parameter. The luminance model (top) pools luminance, and so has one statistic per window, approximating the spatial pooling performed by retinal ganglion cells. The spectral energy model (bottom) pools spectral energy at 4 orientation and 6 scales, as well as luminance, for a total of 25 statistics per window. This approximates the calculations performed by primary visual cortex (V1). Spectral energy is computed using the complex steerable pyramid constructed in the Fourier domain, squaring and summing across the real and imaginary components [198]. Full resolution version of this figure can found on the OSF.

What differentiates these models from each other is which statistics are averaged in the pooling windows. The luminance model, which approximates retinal computation, simply pools pixel values, which, in our setup, are proportional to luminance. Thus, a pair of luminance model metamers have the same average luminance within each of the pooling windows. This is a relatively loose constraint and, in particular, the Gaussian pooling windows are low-pass filters and thus insensitive to the highest frequencies. Luminance model metamers thus maintain high frequency information from their initialization, as can be seen in figure 3.4. This may seem counter-intuitive, as one may expect luminance metamers to look like blurred versions of their target image. In fact, such an image would be one possible metamer, but as the model is insensitive to high frequencies, *any* modulation can be done to those statistics: removing them completely,

as done by blurring the image, but also vastly increasing the power in those frequencies across phases, as seen in figure 3.4. The middle row of figure 3.4 shows two example model metamers, one for the model with a high scaling value and one with a low scaling value. While the high-scaling model metamer is obviously perceptually distinct from the target image for humans, regardless of where they fixate, the low-scaling image appears confusable when fixating at the center of the image, i.e., when the human fovea is aligned with the model fovea. We can thus see that, for a given model, it is the interaction between the scaling value and the pooled statistics that determine possible perceptual metamers.

While the luminance model only pools pixel values, the spectral energy model pools spectral energy at multiple scales and orientations, as well as pixel values. The energy model is built directly on the image, not on the outputs of the luminance model. A complex steerable pyramid with 6 scales and 4 orientations (constructed in the Fourier domain, [198]) is constructed on the input image, then the energy is computed by squaring and summing across the real and imaginary components (which correspond to even and odd filters) for each sub-band. This energy is then averaged in the pooling windows (a separate set of windows is constructed for each scale, as coarser scales are constructed on progressively down-sampled and filtered versions of the image). Thus, a pair of energy model metamers have the same average oriented energy and luminance within each of these windows. The bottom row of figure 3.4 shows two model metamers, one with a high scaling value and one with a low scaling value. The low scaling value is approximately that of the high scaling value for the luminance model metamer shown in the middle row, while the high scaling value is approximately the physiological value of V1 [70]. The high-scaling model metamer is perceptually distinct from the target image, but its appearance is completely different than that of the high-scaling luminance model metamer – there is none of the high-frequency speckled pattern and instead the far periphery appears somewhat like pink noise (as the energy model is matching spectral energy, that is approximately what's going on). The low-scaling model metamer, on the other hand, again appears as a potential human metamer when fixating but,

when the observer moves their eyes to the periphery, is clearly distinct from the target image. As the pooling windows get larger towards the periphery, the model metamer does a poorer job of matching the hard lines found in the target image, which require a precise alignment of phases across scales, a feature that is not directly represented by our model but constrained when the pooling windows are small enough.

As can be seen in figure 3.4, both models can generate potential perceptual metamers and, in fact, all pooling models should be able to generate perceptal metamers for some scaling value, though for a poor model, that value may need to be so small as to result in pixel-to-pixel matching. As the pooled statistics get more perceptually-relevant, larger windows are sufficient for generating perceptual metamers, as we see with the example model metamers above, where the energy model potential human metamer is generated with a scaling value about six times larger than the luminance model one, and about five times smaller than those synthesized in Freeman and Simoncelli [70] using texture statistics. The goal of the present study is to use psychophysics to find the largest scaling value where the above two models generate perceptual metamers, a value we call the **critical scaling**, and compare this value to the physiological one for the visual areas we are attempting to model.

The central circle with a radius of .5 degree was not visible by the model and was matched pixel-by-pixel in our synthesized images, approximating the fovea, where no pooling occurs. Additionally, if the scaling value was small enough, windows for some distance beyond this region would be smaller than a pixel and so the only solution is to match the pixel values in that region directly. For our image resolution of 2048 by 2600 and display size of 53.6 by 42.2 degrees, models with scaling value of 0.063 have windows whose area at FWHM is smaller than a pixel out to 0.52 degrees, with this number increasing quadratically as scaling decreases, reaching 3.29 degrees for scaling 0.01 (see 3.6.3 for more discussion).

3.3.3 Synthesis

We synthesized model metamers matching 20 different natural images (the **target images**) collected from the authors' (W.F.B and E.P.S) personal collections, as well as from the UPenn Natural Image Database [209] and from an unpublished collection by David Brainard. The photos were chosen from these collections so that they were large, that pixel intensities were proportional to luminance, that they were 16-bit images, and that they had not undergone compression, which could result in artifacts. They were converted to grayscale using scikit-image's `color.rgb2gray` function [220], cropped to 2048 by 2600 pixels (the photos from Prof. Brainard were 2014 pixels tall, so a small amount of reflection padding was used to reach 2048 pixels), and had their pixel values rescaled to lie between .05 and .95 (synthesized images were still allowed to have pixel values between 0 and 1; without rescaling the target images, synthesis resulted in strange artifacts with pixels near 0, as this was the minimum allowed value). The images were chosen to span a variety of natural image content types, including buildings, animals, and natural textures (see figure 3.2).

We synthesized the model metamers using custom code written in PyTorch [169], using the AMSGrad variant of the Adam optimization algorithm [124, 183], with learning rate 0.01. Slightly different approaches were used for the luminance and energy model metamers. For the luminance model metamers, the objective function was to minimize the mean-squared error between the model representation of the target and synthesized images, $L(x, \hat{x}) = (M(x) - M(\hat{x}))^2$, and synthesis was run for 5000 iterations. For the energy model metamers, the objective function also contained a quadratic range penalty term, which penalized any pixel values outside of $[0, 1]$, $L(x, \hat{x}) = .5(M(x) - M(\hat{x}))^2 + .5\mathcal{B}(\hat{x})$, and synthesis was run for 15000 iterations. Additionally, energy model metamer synthesis used stochastic weight averaging [108], which helped avoid local optima by averaging over pixel values as synthesis neared convergence, and used coarse-to-fine optimization [178]. Additionally, each statistic (in both models) was z-scored using the average

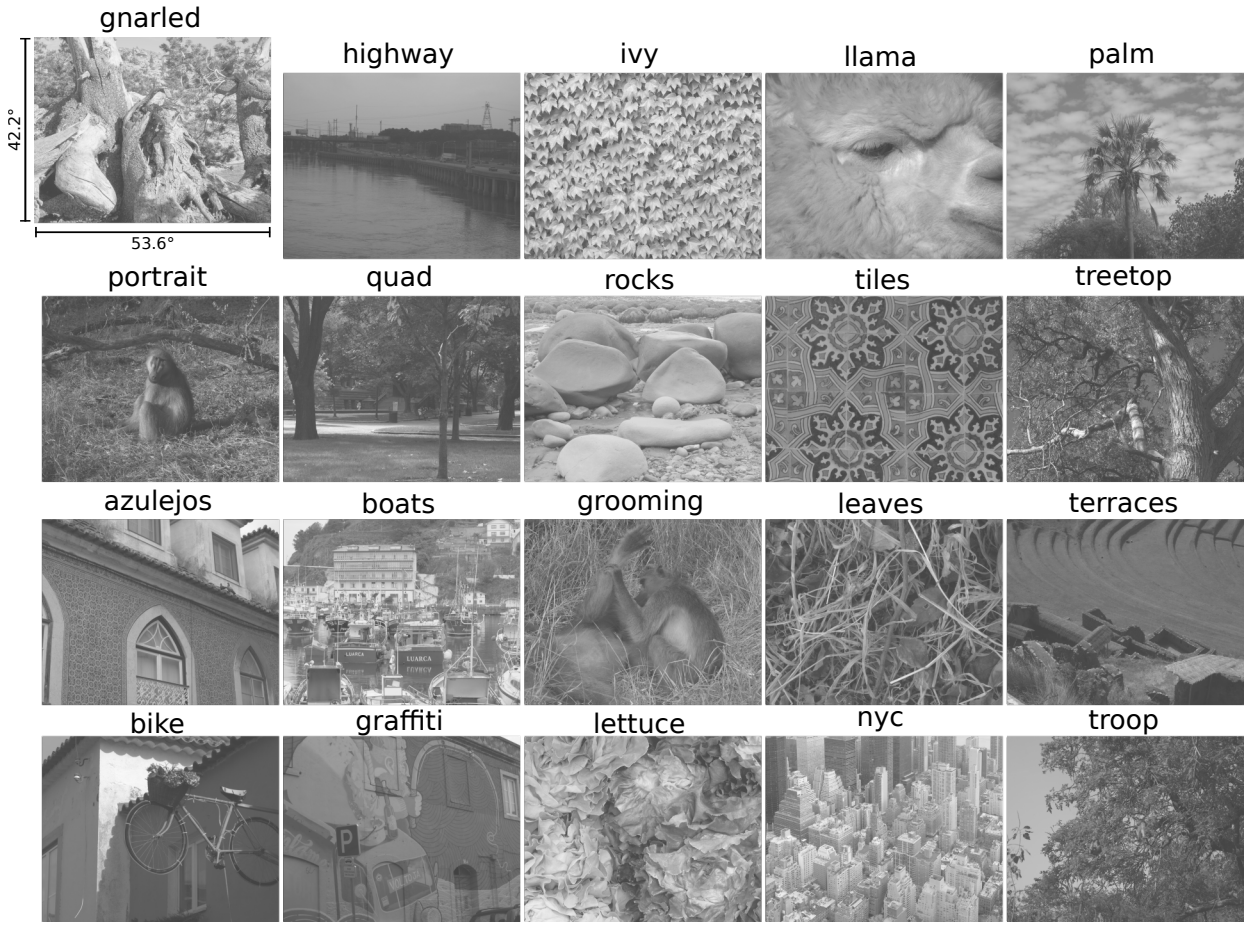


Figure 3.2: Target images covered a variety of image content, including textures, objects, and scenes. The images were selected because they are large 16-bit images, with pixel intensities proportional to luminance and no compression artifacts. Images were converted to grayscale, cropped to 2048 by 2600 pixels, were displayed at 53.6 by 42.2 degrees, and had their pixel values rescaled to lie between .05 and .95. Synthesized images discussed in this paper were all synthesized so that their model representations matched that of one of these images. Full resolution version of this figure can be found on the OSF.

statistic value computed across the entire image on a selection of grayscale texture images, which improved synthesis performance. For both models, synthesis terminated early if the loss had not decreased by more than $1e-9$ over the past 50 iterations. While not all model metamers achieved the same loss values, with differences in synthesis loss across target images, there was no relationship between the remaining loss and behavioral performance.

In order to avoid possible synthesis artifacts resulting from approximation errors, we performed the pooling operation in an inefficient manner: for each model, its windows were represented as

two tensors, one representing the angular slices and one representing the radial annuli, which, when multiplied together, would give the individual windows, with separate sets of windows for each scale in the energy model. This required a large amount of memory, and so for scaling values below 0.09, models were too large to perform synthesis on the available NVIDIA GPUs with 32GB of memory. Thus, all luminance model metamers were computed on the CPU, and synthesis of a single image took from about an hour for scaling 1.5 to 2 days for scaling 0.058 to 14 days for scaling 0.01. For the energy model metamers, the lowest two scaling values were computed on the CPU, with synthesis taking about a week. For those energy model metamers which were able to be computed on the GPU, synthesis took from 5 hours for scaling 0.095 to 1.5 hours for scaling 0.27 and above.

Synthesized images for original vs. synth and synth vs. synth white noise comparisons (see 3.3.5) were initialized with full-field patches of white noise (each pixel sampled from a uniform distribution between 0 and 1). For each model, scaling value, and target image, three different initialization seeds were used. A unique set of three seeds was used for each scaling value and target image, except for the following, which all used $\{0, 1, 2\}$

- Luminance model: azulejos, bike, graffiti, llama, terraces, tiles; scaling 0.01, 0.013, 0.017, 0.021, 0.027, 0.035, 0.045, 0.058, 0.075 and 0.5.
- Energy model: azulejos, bike, graffiti, llama, terraces, tiles; scaling 0.095, 0.12, 0.14, 0.18, 0.22, 0.27, 0.33, 0.4, and 0.5.

For original vs. synth and synth vs. synth natural image comparison, synthesized images for each model, scaling value, and target image were initialized with three random choices from among the rest of the target images.

3.3.4 Observers

Eight participants (5 women and 3 men, aged 24 to 33), including an author (W.F.B.), participated in the study and were recruited from New York University. All subjects had normal or corrected-to-normal vision. Each subject completed nine one-hour sessions. One subject (sub-00) also performed seven additional sessions. All subjects provided informed consent before participating in the study. The experiment was conducted in accordance with the Declaration of Helsinki and was approved by the New York University ethics committee on activities involving human subjects.

3.3.5 Psychophysics experiment

A psychophysics experiment was run in order to determine which of the synthesized model metamers were also perceptual metamers, see figure 3.3 for schematic. During the experiment, observers viewed a series of grayscale 8-bit images on a monitor, at a size of 53.6 by 42.2 degrees. An image would appear, divided in half by a vertical midgray bar 2 degrees wide, for 200 msecs, before being replaced by a midgray screen for 500 msecs, followed by a second image for another 200 msecs, also divided by a vertical midgray bar. Images were presented for 200 msecs to minimize the possibility of eye movements and have behavior depend on feedforward processes, and the dividing bar prevented participants' use of spatial edges to perform the task. After the second image, in which one half was identical to the first image and one half had changed, a midgray screen appeared with text prompting a response, and the observer's task was to report which half had changed; the observer had as much time as necessary to respond. The two compared images were either two synthesized images (synthesized for identical models with the same scaling value and target image, but different initializations) or one synthesized image and its target image. Either image could be presented first.

The midgray blank screen presented between images serves as a mask to prevent participants

from using motion cues to discriminate the two images. Our models aim to capture the steady state response, not the transient response, which would be necessary to predict whether changes to an image would be visible. This mask forces the participants to use the image content to discriminate between the two images, rather than relying on temporal edges (analogous to our use of the vertical bar to prevent the use of spatial edges). This introduces a memory component to the task (participants must remember the first image in order to compare it to the second image), which is also present in previous metamer discrimination experiments [57, 70, 219], and also prevents participants from exploiting any small calibration imperfections in our experimental set up. We believe the precise duration of this mask is unimportant for our results: first, Bennett and Cortese [11] found the duration of a blank screen did not affect thresholds in a spatial frequency discrimination task over a range from 200 to 10,000 msec, and second, mask duration is likely to have a similar effect on performance as image presentation duration, which Freeman and Simoncelli [70] found affected asymptotic performance but not critical scaling.

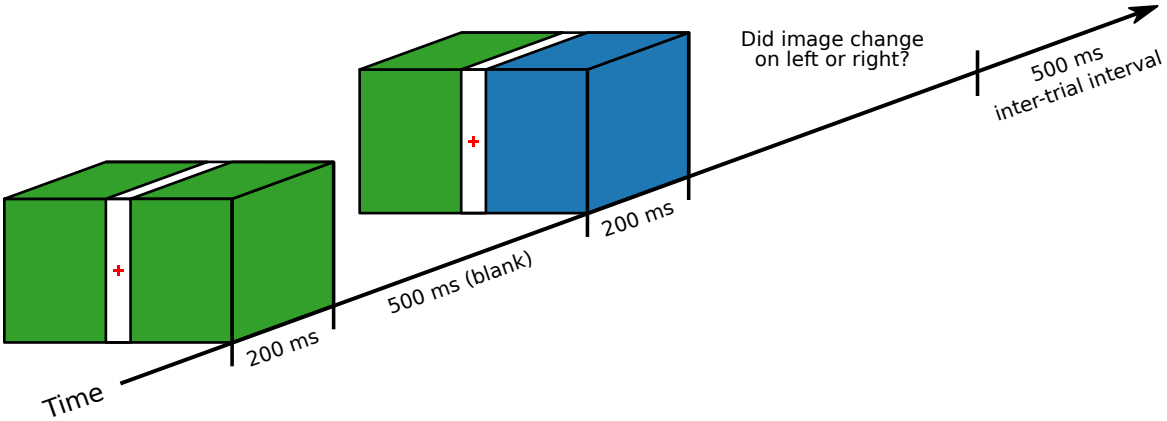
We performed four different comparisons:

1. Original vs. synth, white noise: the two images being compared were always one synthesized image and its target image, and the synthesized image was initialized with a sample of white noise.

2. Synth vs. synth, white noise: both images were synthesized, with the same model, scaling value, and target image, but different white noise seeds as synthesis initialization.

3. Original vs synth, natural image: the two images being compared were always one synthesized image and its target image, and the synthesized image was initialized with a different natural image from our data set.

4. Synth vs synth, natural image: both images were synthesized, with the same model, scaling value, and target image, but initialized with different natural images from our data set.



	First image	Second image left	Second image right
Original vs. Synth	Original	Original	Synth
		Synth	Original
	Synth	Original	Synth
		Synth	Original
Synth vs. Synth	Synth *i*	Synth *i*	Synth *j*
		Synth *j*	Synth *i*
	Synth *j*	Synth *i*	Synth *j*
		Synth *j*	Synth *i*

Figure 3.3: Schematic of psychophysics task. Top shows the structure for a single trial: a single image is presented for 200 msec, bisected in the center by a gray bar, followed by a blank screen for 500 msec. The image returns, with a random half of the image changed to the comparison image, for 200 msec. The participants then have as long as needed to say which half of the image changed, followed by a 500 msec intertrial interval. Bottom table shows possible comparisons. In original vs. synth, one image was the target image whose model representation the synthesized images match (see figure 3.2), and the other was one such synthesized image. In synth vs. synth, both were synthesized images targeting the same original image, with the same model scaling, but different initialization. In experiments, dividing bar, blanks, and backgrounds were all midgray. For more details see text.

For a given model and comparison, the full data set consisted of three sessions, each of which contained 5 runs of approximately 8 to 12 minutes each. Subjects were instructed to take a brief rest between runs. Each session contained the synthesized images across all scaling values for five target images, and each run contained all synthesized images for three target images, with a single image rotating in and out on consecutive runs. For a given comparison (between either two synthesized image or a synthesized image and its target), there are four possible stimulus configurations, and each of these showed up once per run, so that each comparison was made 12

times over the course of the session. Each subject saw synthesized images for 15 of the 20 target images; all subjects saw the first ten, and the remaining sets of five were balanced across subjects.

Subjects completed several training runs. Before their first session, to demonstrate task structure, they completed the task as described above, comparing two natural images and two noise samples (one white, one pink). Then, before their first session of each comparison type including a natural image, they completed a training run showing two natural images and two synthesized images of the type included in the session, one with the largest scaling included in the task and one with the smallest. Before their session of each comparison type comparing two synthesized images, they similarly completed a training run comparing four synthesized images, two with a low scaling value and two with a high scaling value, for each of two target images. Each training run took one to two minutes and was repeated if performance on the high scaling synthesized images were below 90% or subjects expressed uncertainty about their ability to perform the task (participants were expected to perform close to chance for the low scaling synthesized images). A video of a single energy model training run, original vs. Synth: white noise comparison, can be found on the OSF.

All observers took part in original vs. synth and synth vs. synth white noise comparisons for the energy model and original vs. synth white noise comparison for the luminance model. Energy model synth vs. synth white noise comparison was always the last comparison, as it was the most challenging. One observer, sub-00, also did a single session of luminance model synth vs. synth white noise comparison, as well as three sessions of energy model original vs. synth and synth vs. synth natural image comparisons. Before each session which included a natural image (the original vs. synth comparisons), subjects were shown the five natural images that would be part of that session, as well as two example synthesized images per target image, one with a low scaling value, one with a high scaling value. Before each session comparing two synthesized images (the synth vs. synth comparison), subjects were shown four example synthesized images per target image, two with a low scaling value and two with a high scaling value. This was done

so observers had some sense of what the images looked like and how they differed.

3.3.6 Apparatus

The stimuli were displayed on an Eizo CS2740 LED flat monitor running at 60 Hz with resolution 3840x2160. The monitor was gamma-corrected to yield a linear relationship between luminance and pixel value. The maximum, minimum, and mean luminances were 147.73, .3939, and 77.31 cd/m^2, respectively.

The experiment was run using custom code written in Python 3.7.0 using PsychoPy 3.1.5 [171], run on an Ubuntu 20.04 LTS desktop. A button box was used to record the psychophysical response data. All stimuli were presented as 8-bit grayscale images.

The experiment was run with a viewing distance of 40 cm, giving 48.5 pixels per degree of visual angle. A chin and forehead rest was used to maintain head position. No eyetracking was used.

3.3.7 Data analysis

All trials were analyzed, a total of 4320 trials per subject per model per comparison (across 15 images and 8 scaling values) for all energy model comparisons and for luminance model original vs. synth white noise comparison. Luminance model synth vs. synth, white noise comparison had 1440 trials (across 5 images and 8 scaling values) for a single subject. The luminance model natural image comparisons were not run. Where behavioral data is plotted in this paper, the proportion correct is the average across all relevant trials.

To quantify performance as a function of model scaling, we used the same 2-parameter function

for discriminability d' as Freeman and Simoncelli [70]:

$$d'(s;\alpha,s_c) = \begin{cases} \alpha(1-\frac{s_c^2}{s^2}), & s > s_c \\ 0, & s \leqslant s_c \end{cases}$$

where s_c is the critical scaling value (below which participants cannot discriminate the stimuli) and α is the max d' value (called the "proportionality factor" in Freeman and Simoncelli [70]).

We transform d' into the probability correct using the same function as in Freeman and Simoncelli [70]:

$$P(s;\alpha,s_c) = \Phi\left(\frac{d'(s;\alpha,s_c)}{\sqrt{2}}\right)\Phi\left(\frac{d'(s;\alpha,s_c)}{2}\right) + \Phi\left(\frac{-d'(s;\alpha,s_c)}{\sqrt{2}}\right)\Phi\left(\frac{-d'(s;\alpha,s_c)}{2}\right)$$

where Φ is the cumulative of the normal distribution. The probability correct is 50% when $d' = 0$ (and thus when scaling is at or below the critical scaling), reaches about 79% when $d' = 2$ and 98% when $d' = 4$. As the α parameter above gives the maximum d' value, it has a monotonic relationship with the asymptotic performance.

The posterior distribution over parameters s_c and α was estimated using a hierarchical, partial-pooling model, with independent subject- and image-level effects for both s_c and α, with each model and comparison estimated separately. Subject responses were modeled as samples from a Bernoulli distribution with probability $(1-\pi)P(s) + .5\pi$, where π is the lapse rate, estimated independently for each subject. Estimates were obtained using a Markov Chain Monte Carlo (MCMC) procedure written in Python 3.7.10 [215] using the `numpyro` package, version 0.8.0 [20, 172]. MCMC sampling was conducted using the No U-Turn Sampler algorithm ([99], step size 1, target acceptance probability 0.8, and max tree depth 10), with 4 chains, each with 20,000 samples (10,000 of which were discarded as warmup). Convergence was assessed using the $\hat{R}$ statistics ([30], looking for $\hat{R} < 1.1$) and by examining traceplots. Parameters were given weakly-informative priors and both s_c and α were estimated on natural logarithmic scales.

In sum, for model $m \in \{E, L\}$, comparison t, subject x, image i, and scaling s:

$$y_1, \ldots, y_n \sim \text{Bernoulli}((1 - \pi_{mtx})P(s; \alpha_{mtxi}, s_{c,mtxi}) + .5\pi_{mtx})$$

$$\log \alpha_{mtxi} = \alpha_{mt} + \alpha_{mti} + \alpha_{mtx}$$

$$\log s_{c,mtxi} = s_{c,mt} + s_{c,mti} + s_{c,mtx}$$

with the following priors:

$$\alpha_{mt} \sim \mathcal{N}(1.6, 1)$$

$$s_{c,Et} \sim \mathcal{N}(-1.38, 1)$$

$$s_{c,Lt} \sim \mathcal{N}(-4, 1)$$

$$\pi_{mtx} \sim \text{Beta}(2, 50)$$

$$\alpha_{mtx} \sim \mathcal{N}(0, \sigma_{\alpha,mtx})$$

$$\alpha_{mti} \sim \mathcal{N}(0, \sigma_{\alpha,mti})$$

$$s_{c,mtx} \sim \mathcal{N}(0, \sigma_{s_c,mtx})$$

$$s_{c,mti} \sim \mathcal{N}(0, \sigma_{s_c,mti})$$

$$\sigma_{\alpha,mtx} \sim \text{HalfCauchy}(.1)$$

$$\sigma_{\alpha,mti} \sim \text{HalfCauchy}(.1)$$

$$\sigma_{s_c,mtx} \sim \text{HalfCauchy}(.1)$$

$$\sigma_{s_c,mti} \sim \text{HalfCauchy}(.1)$$

The priors for $s_{c,mt}$ of the energy and luminance models correspond to critical scales of .25 and .018, respectively, which are derived from the center of the V1 physiological range plotted in Freeman and Simoncelli [70] figure 5 and from the slope of a line fit to the dendritic field diameter vs eccentricity of midget retinal ganglion cells in Dacey and Petersen [50] figure 2B (see section 3.6.1). This captures our prediction that the models' critical scaling values should match that of

the physiological scaling in the corresponding brain area, should be independent of comparison type and consistent across images and subjects, while not placing too much of a constraint on the parameters.

The posterior distribution represents the model's beliefs about the parameters given the priors and data and is summarized throughout this paper as the posterior mean and 95% high density intervals (the 95% HDI represents the 95% of a probability distribution with the highest probability density, as opposed to the more common 95% equal-tailed interval, which has 2.5% of its density on either side of its limits; for symmetric distributions, these will be identical, but can diverge markedly if the distribution is highly skewed, [128]).

3.3.8 Software

This project was done using a variety of custom scripts written in Python 3.7.10 [215], all found in the GitHub repository associated with this paper. The following packages were used: `snakemake` [149], `JAX` [25], `matplotlib` [106], `psychopy` [171], `scipy` [218], `scikit-image` [220], `pytorch` [169], `arviz` [130], `numpyro` [20, 172], `pandas` [145, 182], `seaborn` [227], `jupyterlab` [125], and `xarray` [102].

3.4 Results

For a given model and comparison, performance increases with scaling in a monotonic fashion, and is fit well by our choice of psychophysical curve (figure 3.5(A)). The exception is the synth vs. synth comparison for the luminance model, which we will return to later. First, let us focus on the original vs. synth comparisons. For both models, performance is at or near chance for the smallest tested scaling values tested and exceeds 90% for the largest. The critical scaling values, as seen in figure 3.5(B) are approximately 0.016 and 0.06, respectively. While this value for the luminance model falls between the physiological scaling of midget and parasol retinal ganglion

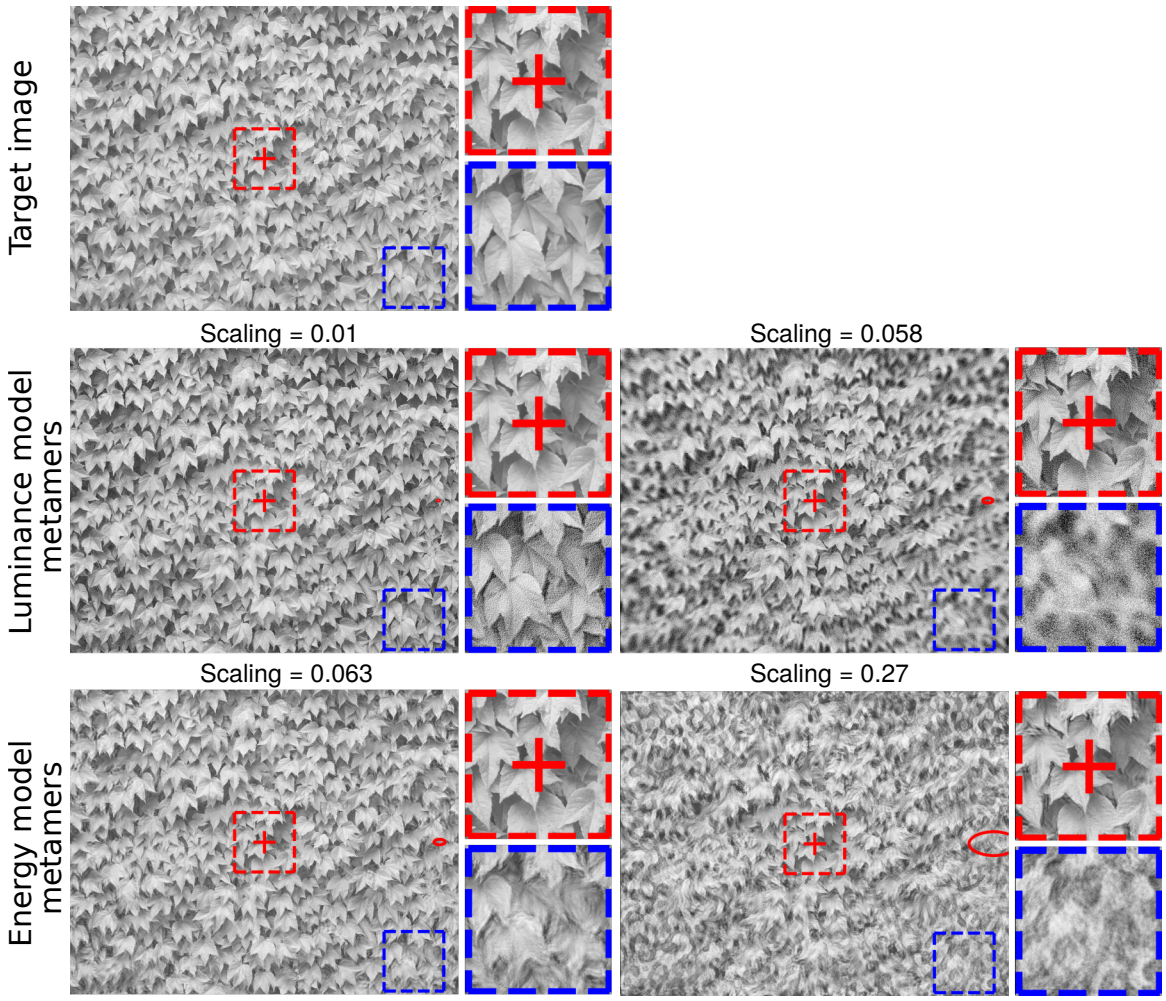


Figure 3.4: Synthesized model metamers are perceptual metamers with their target image at low scaling values, but easily discriminable at high values. The top row shows the target image whose representation the four model metamers have been synthesized to match; the difference between these four images is which model was used, as well as that model's scaling parameter (the ellipse on each image shows the pooling window contour at half-max at that eccentricity). The middle row shows model metamers for the luminance model, while the bottom shows those for the energy model. In both, the left image comes from the smallest scaling value tested in the original vs. synth comparison, while the right comes from the highest. For both, the left image is a perceptual metamer: when fixating at the cross in the center of the image, the two images are perceptually identical to the target image at the top. However, when looking at the cutout of the periphery in the blue box, we can clearly see that all three images are different. The luminance model metamer has small amounts of high-frequency noise remaining from its initialization with a patch of uniform noise, since its windows effectively act as low-pass filters, they become increasingly insensitive to high frequencies towards the periphery; however, this level of noise at this frequency is imperceptible when fixating. Similarly, the energy model metamer's periphery contains more complicated distortions owing to its phase insensitivity. For both models, these patterns are exaggerated in the high-scaling model metamer, to the level where they are easily detectable when fixating. Full resolution version of this figure can be found on the OSF.

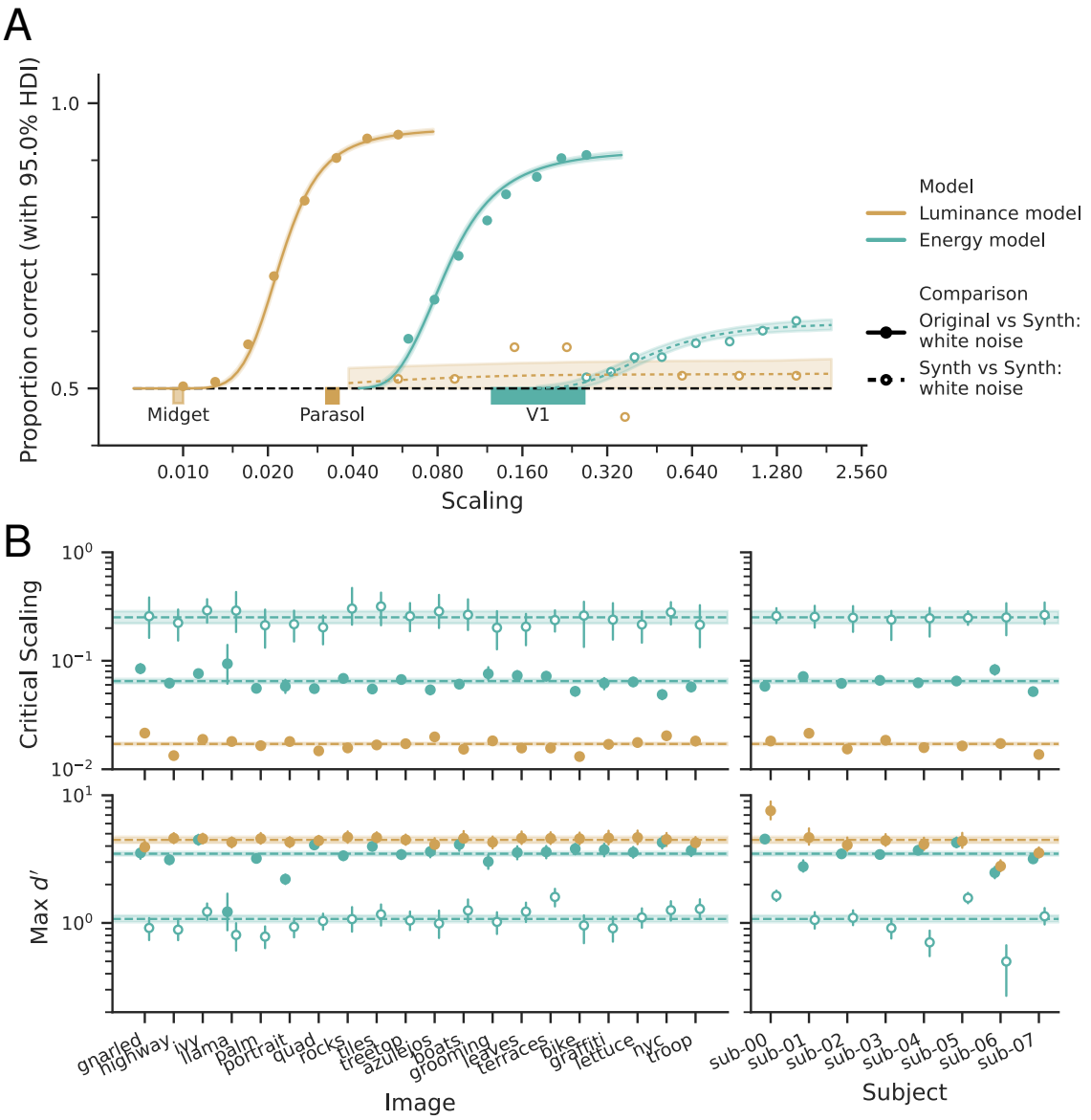


Figure 3.5: Original vs. synth comparisons have smaller critical scaling values than synth vs. synth comparisons, and the luminance model has smaller critical scaling than the energy model. (A) Probability correct as a function of scaling for energy and luminance models, original vs. synth (solid line) and synth vs. synth (dashed line) comparisons. Data points represent the average across subjects and images, 4320 trials per data point except for luminance model synth vs. synth comparison, which have 180 per data point (one subject, five images). Lines represent the posterior predictive means of psychophysics curves across subjects and images, with the shaded region giving the 95% HDI. Labeled horizontal bars give the range of physiological scaling values for the associated retinal ganglion cell type or cortical area (Freeman and Simoncelli [70] for V1, 3.6.1 for RGCs). (B) Parameter values for these comparisons. Top row shows the critical scaling value and the bottom the value of the max d' parameter. Left column presents the values for each image, averaged across subjects, while the right presents the values for each subject, averaged across images. Points represent the posterior means, shaded regions the 95% HDI, and horizontal dashed lines and shaded regions the global means and 95% HDI. Note that the luminance model, synth vs synth: white noise comparison is not shown in this panel, because the data was poorly fit by this curve – as can be seen in panel A, the psychophysical curve is essentially flat at chance and thus the fit had low max d' and low critical scaling), with high uncertainty.

cells, the energy model's critical scaling is approximately half the lower end of V1's range.

3.4.1 Performance differs between original vs. synth and synth vs. synth comparisons

We can also see that the difference between original vs. synth and synth vs. synth is large for both models: comparing two synthesized images is much more difficult than comparing a synthesized image to its target image. For the luminance model, in fact, this discrimination is impossible (preventing us from estimating either psychophysical curve parameter), regardless of the scaling value. This discrimination is possible, though difficult, for the energy model, with performance reaching about 60%, on average (there are substantial differences across subjects in this asymptotic performance, see figure 3.5(B) and figure 3.16). While the critical scaling value we find for this comparison is comparable to the value found in Freeman and Simoncelli [70] (and thus the physiological scaling value of V1), the asymptotic performance is much lower. We believe this to be primarily the result of experimental differences, see section 3.6.2 for details.

The difficulty of differentiating between two synthesized images, for either model, is striking. Figure 3.6 may help explain why: as the pooling windows grow very large for the luminance model, identical model representations serve as a very lax constraint: there are many possible images with matched average pixel values in regions that large. As our synthesized images were initialized with white noise, the completed model metamers appear to be two different samples of white noise with matched large-scale pattern of dark and light splotches. Humans are bad at differentiating between samples of white noise and thus this task is impossible, no matter how large the windows grow. In the limit, if the scaling value was so large that the model had only one pooling window, two model metamers would be two patches of white noise with the same mean, and differentiating them would still be impossible (Wallis et al. [219] made a similar point when discussing their preference for the original vs. synth task). Analogously, synthesis with the

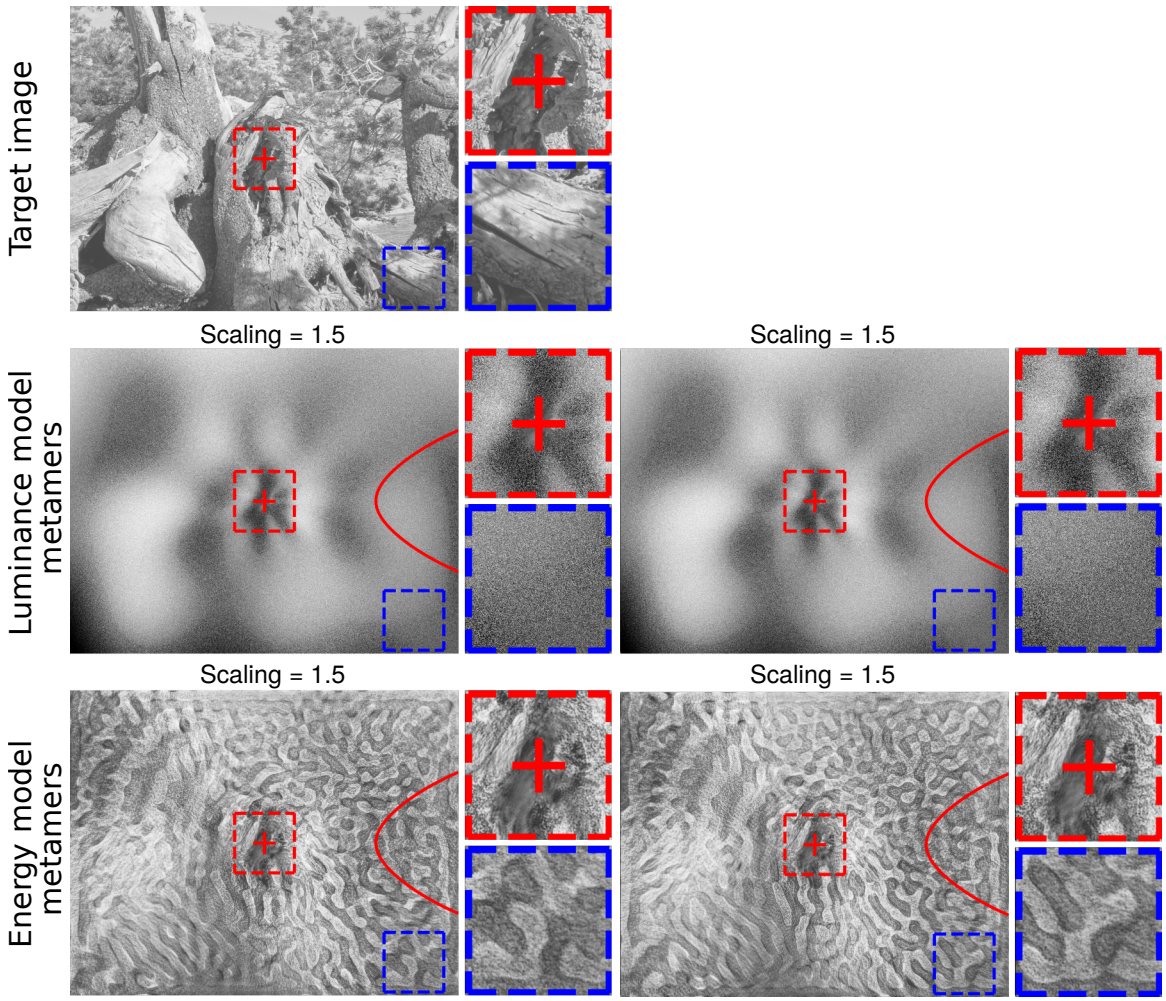


Figure 3.6: With the highest tested scaling value, 1.5, the original vs. synth comparison is trivial while the synth vs. synth comparison is difficult (energy model) or impossible (luminance model). This figure has the same arrangement as figure 3.4, except the two model metamers both have the same scaling value, but different uniform noise initializations. All four of the model metamers can be easily distinguished from the natural image at top (original vs. synth), but are difficult to distinguish from each other, despite the fact that their pooling windows have grown very large (synth vs. synth). Luminance model metamers with such a large scaling value are essentially patches of white noise, which are impossible from humans to distinguish from each other, even when free-viewing with unlimited time, let alone when fixating for 200 msec. Energy model metamers, on the other hand, have snake-like patterns, such that comparing two of them is similar to comparing two gratings with the same orientation and spatial frequency but different phases, which is difficult but not impossible. Full resolution version of this figure can be found on the OSF.

energy model forces local orientated spectral energy to match, without explicitly constraining the phase, and so as the windows grow larger, the resulting synthesized images look more and more like samples of pink noise with oriented bands running through them. Comparing two samples of pink noise is also extremely difficult for humans, but here the task is possible, if just barely, by comparing the exact positions of the bands; this is akin to comparing the phase of two gratings with identical orientation and spatial frequency (see peripheral inset in bottom row of figure 3.6).

3.4.2 The interaction between model sensitivities and image content affects performance

The critical scaling, which is the focus of this project, does not vary much across images for a given model and comparison. However, performance at super-threshold scaling values does vary substantially across images, as quantified by the max d' parameter and demonstrated in figure 3.7. When looking at the energy model data for original vs. synth, we can see that the llama and nyc images are two ends of the continuum: the llama image (red) is the most difficult (with the lowest max d'), while the nyc image (purple) is the least (with the highest max d'). By examining their respective target images in panel B, we can see why that might be the case: the llama image looks pink-noise-like, almost cloud-like, while the nyc image is full of hard edges in the cardinal directions, which require precise alignment of phases across scales to capture. As discussed above, synthesizing energy model metamers involves matching local oriented spectral energy, which discards phase information; in order to well-approximate the buildings of nyc, the windows must be very small. Conversely, the fluffiness of llama is easy for the model to capture. Thus, we can see that the difficulty of the task depends on the interaction between the synthesizing model and the target image.

This point is emphasized by viewing the data in figure 3.15, which shows data in the same format as figure 3.7A, but for both models and both comparisons. With the exception of the top

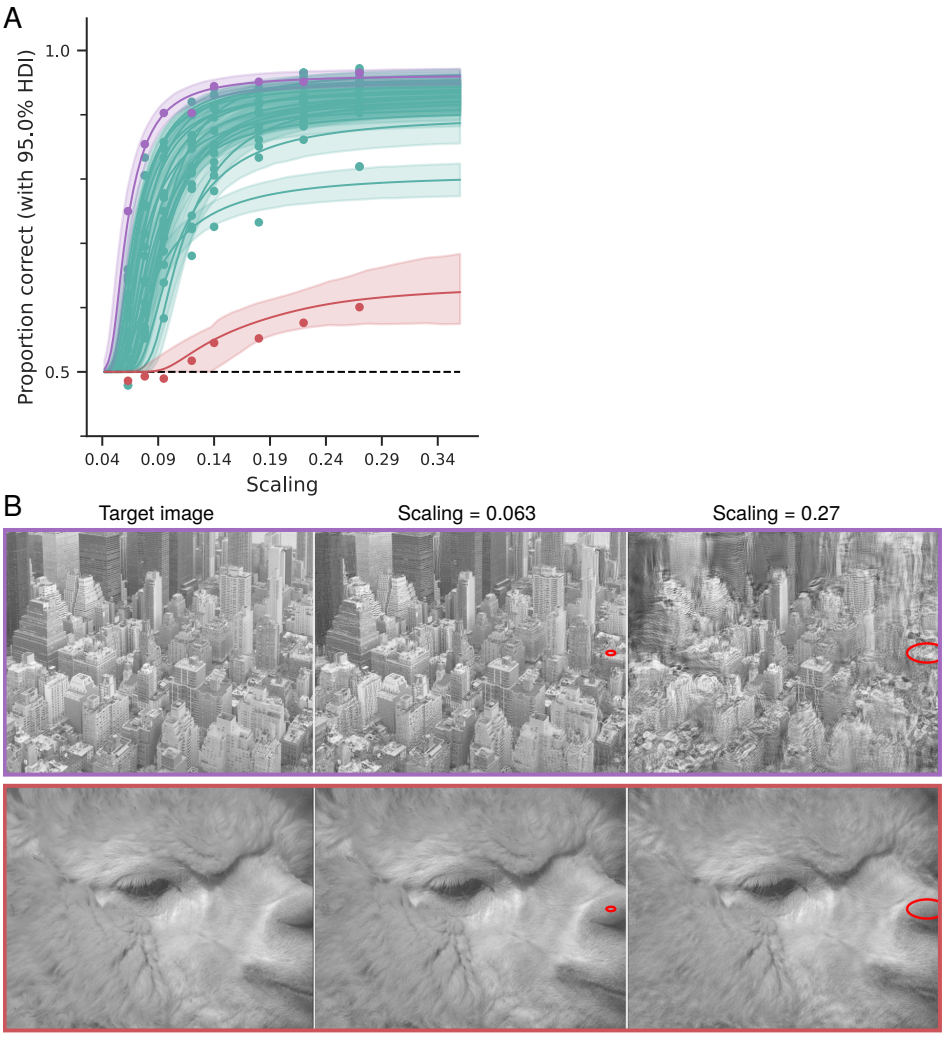


Figure 3.7: The interaction between image content and model sensitivities greatly affects asymptotic performance, most noticeably on the energy model synth vs. synth comparison, while critical scaling does not vary as much. (A) Performance for each image, averaged across subjects, comparing synthesized images to natural images. Most images show similar performance, with one obvious outlier whose performance never rises above 60%. Data points represent the average across subjects, 288 trials per data point for half the images, 144 per data point for the other half. Lines represent the posterior predictive means across subjects, with the shaded region giving the 95% HDI. (B) Example model metamers for two extreme images. The top row (nyc) is the image with the best performance, corresponding to the purple line in A, while the bottom row (llama) is an outlier with by far the worst performance, corresponding to the red line in A. In each row, the leftmost image is the target image, and the next two show model metamers with the lowest and highest tested scaling values for this comparison. The performance on the llama image is so low because the image itself is very similar to pink noise, without a lot of phase structure. Thus, even with larger scaling values, the model metamers are very difficult to distinguish from the target image. The nyc image, on the other hand, has a lot of phase structure, with hard edges that require precise alignment of phase across scales to adequately represent. As the energy model discards phase information, this phase structure is difficult to capture in the model metamers, and so they are relatively easy to distinguish from the target image at all tested scaling values. However, this pattern does not hold in the luminance model (since all target images have $1/f$ frequency distributions, lacking the high-frequency noise the model is insensitive to) or synth vs. synth comparison (since all model metamers lack the phase structure participants are using to identify the target image), where both images have similar, middle-of-the-pack performance (see figure 3.15). Full resolution version of this figure can be found on the OSF

right panel, which is replotted from 3.7, llama and nyc no longer appear as the two ends of the continuum, but lie instead in the middle. For these other models and comparisons, the hard edges of the original nyc are less informative, either because they are no longer present to compare against (in the synth vs. synth comparisons) or because they are no longer exceptionally difficult for the model to capture (for the luminance model original vs. synth comparison). We can also see from this figure that the between-image differences are smaller for these other models and comparisons. For the luminance model, original vs. synth comparison, this is because there is less interaction between the original image features and the model's invariances: the model is insensitive to high frequencies and thus requires smaller windows to adequately represent that information, but all our target images are natural images, with $1/f$ power distributions, and thus there is less information present in higher frequencies. For the energy model synth vs. synth comparison, the two images to distinguish are both synthesized and thus neither contain the edge information that allows subjects to readily distinguish the original from synthesized image in the original vs. synth comparison.

3.4.3 Initializing synthesized images with natural images affects synth vs. synth performance

The difference between the two comparisons shown in figure 3.5 is concerning — which comparison should we rely on when determining the model's critical scaling factor for comparing to human perception? It seems natural to choose the comparison with the smallest critical scaling value, but how do we know there is not some other comparison we could do for these models that would reveal an even smaller critical scaling value? The null space of the pooling models is incredibly large, and so there are many possible model metamers we could synthesize, either for comparison against each other or against the target image. Effectively searching this space is not feasible; we used one method at our disposal: the synthesized images' initialization. All

synthesized images discussed so far (and used in other studies, such as Freeman and Simoncelli [70] and Wallis et al. [219]) were initialized with white noise and this may be biasing our search. The rationale behind initializing with patches of white noise is that such images have maximum entropy and close-to-uniform power across all frequencies. However, white noise images are far from common in the natural world and so may be grouped together perceptually, both by the human visual system and our pooling models, and a similar issue may be occuring for our synthesized images initialized with white noise. In fact, synthesized images initialized with white noise all have a smaller synthesis-model distance with each other than they do with their target image, consistent with them lying clumped together in one portion of model representation space. In an attempt to explore a different portion of this space, we generated another set of model metamers, initializing them with other natural images from our data set, and repeated the comparisons between two synthesized images and a synthesized image and its target.

Figure 3.8(A) shows behavior for these additional comparisons in the energy model, for one participant (note that the curves are different than in figure 3.5 because they only represent one subject's behavior). We can see that changing the initialization image does not affect behavior for the original vs. synth comparison, but that it has a large effect on the synth vs. synth comparison: performance lies between that in the original vs. synth and synth vs. synth comparisons when their synthesized images were initialized with white noise. Importantly, as far as the energy model is concerned, it makes no difference what was used to initialize the synthesized images, they are all model metamers and should be equally confusable with each other and with the natural image their representations match. Clearly, however, that is not the case for human behavior.

Figure 3.9 shows comparisons corresponding to three points that intersect the vertical line on figure 3.8(A). All synthesized images here have the same scaling value, 0.27, but human performance on these comparisons varies from ceiling to chance. The top row shows the easiest comparison, between the original image and a synthesized image initialized with a natural image; sub-00 was able to distinguish these two images with near-perfect accuracy. The bottom row

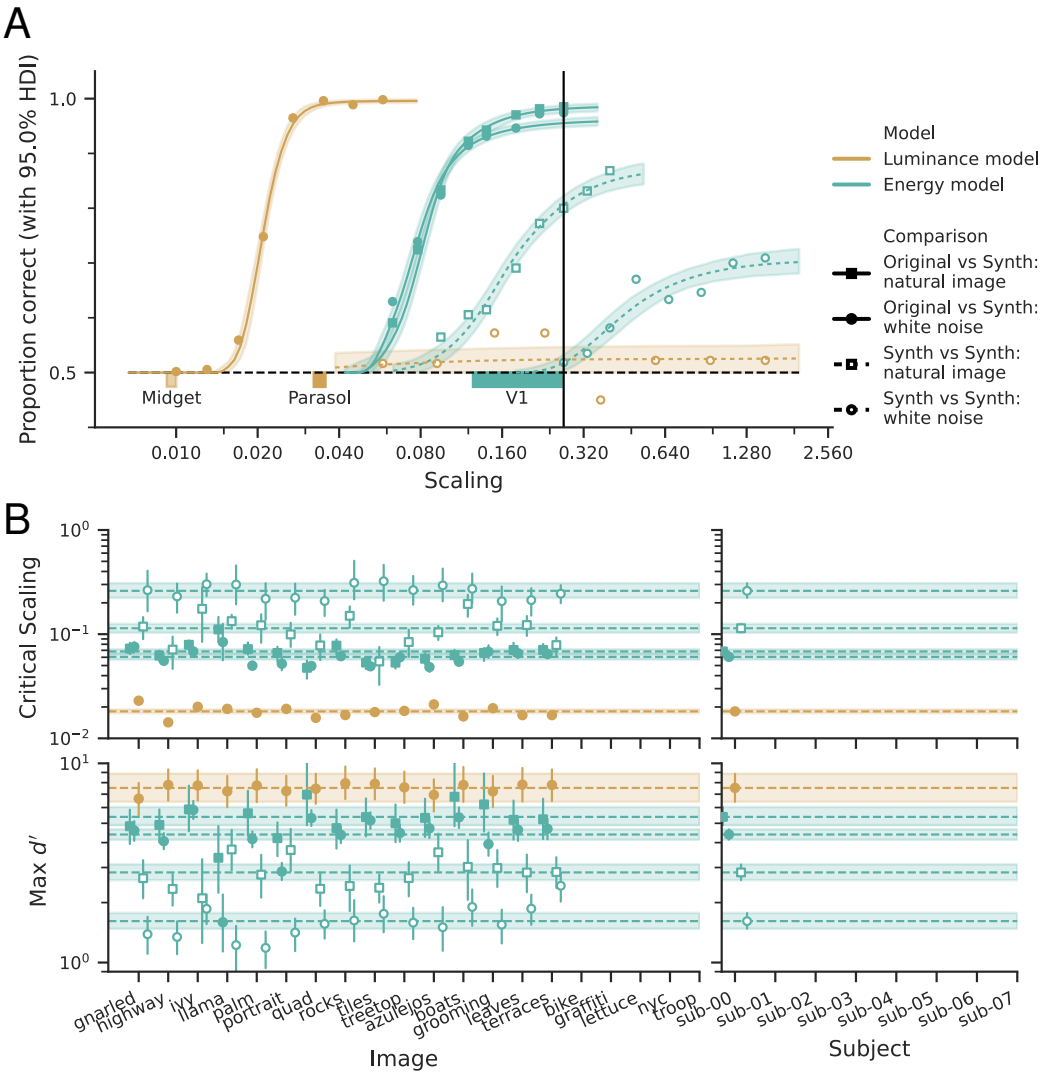


Figure 3.8: Initializing model metamers with natural images does not affect performance in original vs. synth comparison, but reduces critical scaling and increases max d' for the synth vs. synth comparison. (A) Probability correct for one subject, sub-00, as a function of scaling for energy and luminance models, all comparisons (note curves are different from figure 3.5, which averaged all subjects). Data points represent the average images, 540 trials per data point (one subject, fifteen images) except for luminance model synth vs. Synth: white noise comparison, which have 180 per data point (one subject, five images). Lines represent the posterior predictive means across images, with the shaded region giving the 95% HDI. Labeled horizontal bars give the range of physiological scaling values for the associated retinal ganglion cell type or cortical area. Vertical black line represents scaling value where difficulty ran from chance to 100%, based on initialization and comparison, as discussed further in figure 3.9. (B) Parameter values for these comparisons. Top row shows the critical scaling value and the bottom the value of the max d' parameter. Left column presents the values for each image separately for this one subject, while the right presents the values for this subject, averaged across images. In this case, we only present the data for sub-00, as they are the only subject to perform all comparisons. Points represent the posterior means, shaded regions the 95% HDI, and horizontal dashed lines and shaded regions average across all shown images for this subject. Note that the luminance model, synth vs synth: white noise comparison is not shown in this panel, because the data was poorly fit by this curve – as can be seen in panel A, the psychophysical curve is essentially flat at chance and thus the fit had low max d' and low critical scaling), with high uncertainty.

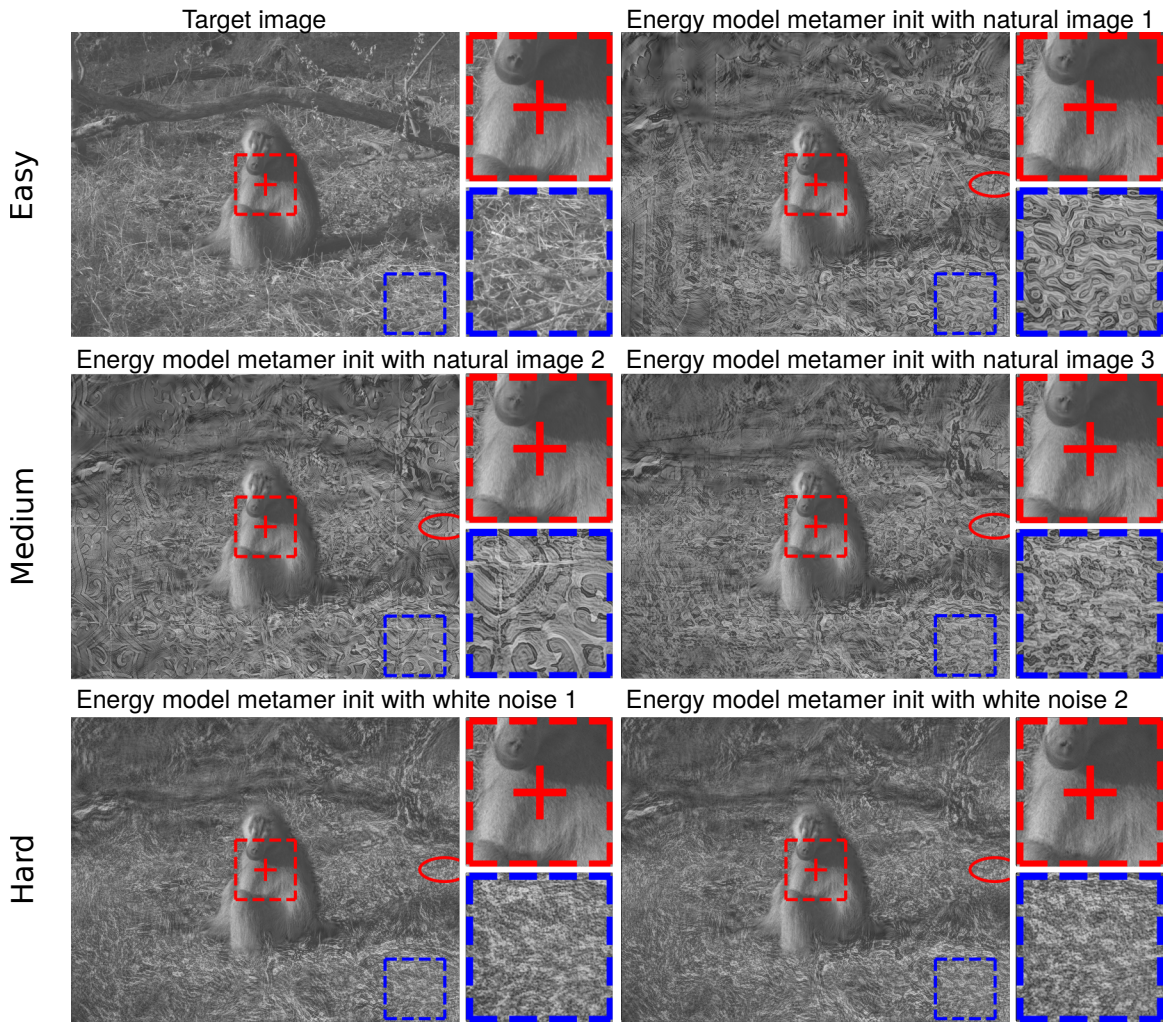


Figure 3.9: With energy model metamers synthesized with scaling value 0.27, task performance varies from ceiling to chance, depending on which comparison is being made, despite the fact that the model predicts all three comparisons should be equivalent. The top row shows the easiest comparison, between the target image and a synthesized image (here initialized with another natural image, bike, but performance is identical when initialized with white noise). With pooling windows of this size, synthesized images are obviously not natural when fixating. The middle rows shows a comparison with about 80% probability correct, between two synthesized images initialized with different natural images. Here, enough features remain from the initial images (tiles on the left, highway on the right) to make these images distinguishable. Finally, the bottom row shows the most difficult comparison, between two synthesized images initialized with different patches of white noise, and humans are insensitive to the differences between the two. Full resolution version of this figure can be found on the OSF.

shows the hardest comparison, between two synthesized images initialized with different samples of white noise. As discussed above, comparing two images of this type is difficult even with large pooling windows; at this scaling level, humans are insensitive to the differences between them,

Model	Comparison	Critical Scaling	Number of Statistics (percentage of image pixels)
Luminance	Original vs. Synth: white noise	0.017	19.6 %
	Synth vs. Synth: white noise	N/A	N/A
Energy	Original vs. Synth: white noise	0.065	34.7 %
	Original vs. Synth: natural image	0.068	31.5 %
	Synth vs. Synth: natural image	0.114	11.6 %
	Synth vs. Synth: white noise	0.252	2.6%

Table 3.1: Models reach critical scaling at different compression rates, and all are undercomplete. Table shows critical scaling (posterior mean over all subjects and images) and number of statistics (as a percentage of number of input image pixels), akin to the compression rate, for each model and comparison. Note that the critical scaling for original vs. synth comparisons does not result in the same number of statistics across models and, in particular, all models are undercomplete at their critical scaling values, i.e., their representations compress the image to some degree. The luminance model synth vs. synth comparison has no critical scaling value, as performance was always at chance. For a given model, the number of statistics decreases quadratically with scaling.

and so performance was at chance. The middle row shows two synthesized images, initialized with different natural images, which the subject was able to distinguish with middling accuracy. When comparing these two images, one can see features in the periphery that remain from the initial image (tiles and highway, respectively). Even when fixating, the subject was able to use these features to distinguish the two images, i.e., the human was sensitive to them while the model was not. This reinforces the notion that our pooling models are incomplete in an important way: at the pooling level that approximately matches V1 receptive fields, humans are sensitive to some statistic present in natural images that our models are discarding.

3.4.4 The models reach critical scaling at different compression rates

Table 3.1 shows all the critical scaling values we were able to compute. We can see that finding a model's critical scaling is not simply a matter of increasing the number of windows until the model's representation is overcomplete or until it reaches some threshold: for the original vs. synth white noise comparison, the luminance model representation is slightly smaller than two-thirds the energy model's. We should also note that, if one were to use the model outputs

as a compressed representation of the image, the number of statistics in each representation is almost certainly an overcount, for several reasons. First, in order to ensure that the Gaussian pooling windows uniformly tile the image, the most peripheral windows in the model have the majority of their mass off the image. This is important for metamer synthesis, to avoid artifacts at the edge of the image, but may not be necessary for a compression application. Second, for the energy model, we did not spend a lot of effort determining how the precise number of scales or orientations affected metamer synthesis, and currently all scales are equally-weighted across the image. As the human visual system is insensitive to high frequencies in the periphery and low frequencies in the fovea, this is probably unnecessary, and so some of these statistics can likely be discarded. Finally, our pooling windows are highly overlapping and thus the pooled statistics are far from independent; this redundancy means that the effective dimensionality of our model representations is far lower than the compression rates above.

3.5 Discussion

We synthesized sets of metamers for two foveated pooling models of human vision with large fields of view. We presented these images to observers in psychophysical experiments and showed that our choice of scaling values spanned an appropriate range of values, and that behavior depends significantly on the nature of the comparison observers are making and to a lesser extent on the natural image whose model representation the synthesized image matched. Critical scaling is smaller for the luminance than for the energy model, and for both models critical scaling is smaller and max d' is larger when discriminating natural and synthesized images than when discriminating two synthesized images. We also found intermediate behavior for synth vs. synth but identical behavior for original vs. synth when the synthesized images were initialized with other natural images.

3.5.1 Luminance pooling is smaller than energy pooling

The largest effect observed in this study is the difference in critical scaling between the energy and luminance models in the original vs. synth condition: the energy model's critical scaling is approximately four times larger than that of the luminance model (figure 3.5 and table 3.1). This is a robust result, observed for all subjects and all target images (figure 3.5(B)). Additionally, the energy model scaling value found here is also smaller than that of the texture model found in Wallis et al. [219] in the original vs. synth condition: the average energy model critical scaling is about three times smaller than the average value for the texture model. Together, these results suggest that the three models are approximating subsequent parts of the visual hierarchy, with larger pooling regions for texture statistics than spectral energy, and larger pooling regions for energy than luminance.

3.5.2 Critical scaling value is smaller for original vs. synth than synth vs. synth comparisons

The difference between the original vs. synth and synth vs. synth comparisons is striking, especially for those synthesized images initialized with white noise. Similar results were observed by Wallis et al. [219] for the mid-ventral model from Freeman and Simoncelli [70] and by Deza, Jonnalagadda, and Eckstein [57] for their deep neural network-derived model. For both the luminance and energy models, original vs. synth was much easier than the synth vs. synth comparison. This is most striking for the luminance model, where discriminating between two synthesized images initialized with white noise is always impossible, regardless of the scaling value. If we were to take the luminance model with its critical scaling value as an observer model, this result is confusing: these synthesized images are not observer model metamers, and so should be discriminable. This speaks to an asymmetry in the metamer testing framework: when we have the correct critical scaling, two model metamers should also be perceptual metamers. However,

when two images are *not* model metamers, we make no predictions about their discriminability; the information that discriminates them at the stage of the visual system that corresponds to our luminance model may still be thrown out further downstream. As the luminance model approximates the retina, there are many brain areas afterwards whose null spaces the differences between these images may still fall into. We can see this in figure 3.6: two such images look like different patches of white noise and while they may be discriminable at the beginning of the visual system, they are not at the level of behavior. In order to close the loop and make predictions about how discriminable pairs of arbitrary images are (not just determining whether those pairs of images will be perceived identically), a complete observer model is needed. The models presented here may serve as the basis for such a model, but more work is needed, tying in with the literature of observer models in vision science (e.g., [8, 26, 191]) and of image quality metrics in computer vision (e.g., [58, 224]).

Combining our results with Wallis et al. [219], we have critical scaling values for both original vs. synth and synth vs. synth comparisons for three pooling models of progressively deeper levels of the visual system. We see that, while the critical scaling value is lower for the original vs. synth comparison for all three models, the gap between the two decreases as the models go up the hierarchy: infinite for the luminance model, to about quadruple for energy, to less than double for texture. One potential explanation for this observation is that there are progressively fewer stages of visual processing to discard information as you go up the hierarchy: the difference between V1 responses to a pair of images may fall into IT's null space, but there are not many steps of processing between IT and the perceptual read out where differences between IT responses can be discarded. This may be why we see no overlap between the critical scaling values for original vs. synth and synth vs. synth comparisons across images in figure 3.5(B), whereas Wallis et al. [219] see substantial overlap. Ultimately, physiological data is needed to investigate these possibilities.

3.5.3 Interpreting the critical scaling value

When we have different potential critical scaling values for a single model, as with the energy model's values from the original vs. synth and synth vs. synth comparisons, which should we trust? We believe that the critical scaling value from the original vs. synth comparison, which will almost certainly be smaller, should be used. Natural images are more likely than synthesized images to include information that the human visual system as a whole has evolved to be sensitive to, rather than information that is discarded at some later stage of processing. Using natural images to initialize synthesis or the development of novel synthesis methods that better explore the space of all possible metamers may reduce the discrepancy between the two conditions, but the original vs. synth comparison provides the strongest test of these models.

We thus have a critical scaling value for our energy model that falls well below the range we would expect from V1 neurons and above that of retinal ganglion cells. How should we interpret this? This likely signifies that the energy model is missing some key component of V1 computation. Pooling models approximate statistics they do not have via smaller windows: for example, a Gabor wavelet can be approximated either by matching its spatial frequency and orientation over its whole extent or by matching the luminance of each of its component lobes. Analogously, our energy model does not represent hard edges well, as seen in figure 3.7B, but begins to approximate them with small enough windows. Thus, we believe that the fact that our energy model's critical scaling fell far below the physiological scaling suggests that there is some important calculation performed in V1 neurons beyond computing the local spectral energy and luminance. We added additional local moments of the pixel values (second through fourth) to the model, and this did not change the resulting model metamers perceptually.

One possible addition is divisive normalization, a "canonical neural computation" [38] first proposed to explain neuronal responses in cat V1 [88]. If the normalization is global, such that all responses are modulated together, this will have no effect on the metamers, but if the

normalization is local, modulating responses based on those of nearby windows, it may have an effect. Local normalization may allow for larger critical scaling values when comparing synthesized against natural images, bringing the value more in alignment with that for the comparison between two synthesized images. For example, one phenomenon captured by normalization is cross-orientation suppression, exhibited by both simple and complex cells in V1 [47], where the presence of orientations orthogonal to the preferred orientation suppress the firing rate of the cell. In order to capture this phenomenon in our normalization-free energy model, we would need windows small enough to separately constrain the preferred orientation in the center and the anti-preferred orientation in the surround, leading to pooling windows smaller than the corresponding neuronal receptive fields. With normalization, however, a larger window could capture this phenomenon with appropriate suppression of the response corresponding to the preferred orientation. Incorporating divisive normalization into the energy model may thus result in a larger critical scaling value. However, implementing local normalization is not trivial, as it requires making poorly-constrained decisions about how far the normalization pool should extend in space, scale, and orientation. Future work is needed to test this and other possibilities.

The luminance model's critical scaling falls between the physiological values for the two most numerous retinal ganglion cell classes, suggesting that matching the mean luminance results in approximately matched retinal outputs. This may seem surprising, as the textbook definition of retinal ganglion cells emphasizes their center-surround nature, which would give rise to a bandpass representation, as opposed to the lowpass representation used here. We used a simpler model for technical reasons: we were unable to construct difference-of-Gaussian windows that uniformly tiled the visual field (there were particular difficulties where windows began and at the image edge), which led to synthesis artifacts. However, it is known that RGCs are not perfectly balanced, that they have a luminance response: Croner and Kaplan [48], for example, show a mean integrated sensitivity ratio between surround and center of 0.55 for both parasol and midget RGCs, with a range of 0.1 to 0.9 (at 40 $\mathrm{cd\,m^{-2}}$, the low end of photopic sensitivity, and this balance

changes with luminance). Furthermore, our goal was not to build a complete retina model but to model spatial averaging of luminance, which we know is computed somewhere in the visual system and is the simplest pooling model we can build, and then investigate potential connections to the physiology. It is intriguing that the critical scaling of our luminance model, which is missing obvious properties of retinal processing, matches the physiological scaling of retinal ganglion cells, while that of our energy model, which better matches the functional properties of V1, is so much lower than the corresponding physiological value, but we have no compelling explanation as to why that might be.

3.5.4 The interaction between image content and model (in)sensitivity affects performance

Similar to Brown et al. [31] and Wallis et al. [219], we find that image content matters. Both of those studies synthesize model metamers based on pooled texture statistics, and Wallis et al. [219] shows that texture-like original images are harder to distinguish from their synthesized images than scene-like ones, while Brown et al. [31] show that, among their textures, original images with higher global and local regularity are the easiest to distinguish from their synthesized images (textures with high regularity include baskets, whereas those with low regularity include fur). This aligns with our result: the most distinguishable pairs include natural images with features not well-captured by the synthesizing model, whereas the least distinguishable include those natural images whose features are all adequately captured.

However, we should note that we found this image-level variability largely in super-threshold performance, and this variability does not constitute a failure of these pooling models. As pointed out by Freeman and Simoncelli [70], asymptotic performance also varies with experimental manipulation, while critical scaling remains relatively unaffected. The metamer paradigm makes strong predictions about what happens when the representation of two stimuli are matched: they

are indistinguishable, and so performance on a discrimination task will be at chance, as captured by the critical scaling value. However, it makes *no* predictions about performance at super-threshold levels, as captured by the max d' parameter. An analogy with color vision might make this point more clear: color matching experiments provide evidence for what spectral distributions of light are perceived as identical colors, but provide no information about whether humans consider blue more similar to green or to red; further investigations are necessary to understand color appearance. Thus, while this image-level variability is worth investigating in order to better understand the sensitivities of our model, it does not much affect the inferences we want to make about the human visual system, and speaks more to the need for a complementary approach.

On the other hand, the differences between the original vs. synth and synth vs. synth comparisons, as discussed in 3.5.3, do represent difficulties for our models, as do the effect of initialization on results in the synth vs. synth comparison. This latter effect implies there are some features present in the natural image seeds that the model is insensitive to, and so they remain in the synthesized images, but that the human is *sensitive* to, and thus are able to use to discriminate between two such synthesized images. The lack of such features is what makes discriminating between two synthesized images initialized with white noise so difficult. This provides further support for the possibility that our energy model is missing some key statistics, as otherwise the image the metamer is initialized with would have no effect. Initializing with natural images is an attempt to sample a broader portion of the manifold of possible metamers, but a more principled way of doing this would involve changing the synthesis procedure to better search image space, resulting in more informative metamers.

To summarize, the metamer contains information the model is insensitive to from the initial image and information the model is sensitive to from the target image. When performing the synth vs. synth task, these are the only sources of information that subjects can use to distinguish the two images, but when performing the original vs. synth task, subjects also have access to model insensitive-information from the target image. The discrepancy between performance on

these two tasks, as well as the outlier performance on certain images, result from this mismatch between human and model sensitivities: humans are able to use this information which the model has discarded.

3.5.5 Window mismodeling may also be an issue

Since pooling windows and statistics trade off against one another, it is also possible that the smaller than expected critical scaling for the energy model speaks to an issue with our pooling windows, rather than with the statistics being pooled. When performing the task, especially the original vs. synth comparisons, subjects reported that the most informative portions of the image were in the mid-periphery, rather than close to fixation or at the edges of the image. This suggests that our windows may be too large in that region (or, equivalently, too small near the fovea and the far periphery), and that pooling window width is not best modeled by a linear function but by something non-linear. Another possibility is that, rather than using the same size windows for all statistics, different statistics should have different size windows, with window size varying inversely with statistic scale, such that high frequency information has the smallest windows and luminance the largest. Either variant will change the interpretation of the scaling parameter, and so careful thought is necessary to determine how to relate such sizes to the linear, constant size captured by the scaling parameter in this and similar studies.

3.5.6 The difficulties of linking physiology and psychophysics

The above issues demonstrate some of the difficulties in making inferences about the link between physiology and psychophysical models. The linking proposition that underlies the metamer paradigm is a "Converse Identity" proposition in the framework proposed by Teller [206]: identical perceptual states imply identical physiological states. In fact, the use of pooling models allows us to go one step further: once we have found the critical scaling value, we can say

that identical model outputs imply identical perceptual states imply identical physiological states. However, our linking proposition is silent on what is implied by *distinct* model outputs and so a complementary approach is required, such as building observer models to predict perceptual distance.

Furthermore, there is a general difficulty with the idea of linking psychophysical pooling to physiological receptive fields in order to find a matching visual area: there is not a single receptive field scaling value for a given area. There are many reasons for this, the most important of which is the heterogeneity of cell types in any given area: V1 cells do not belong to a single population any more than retinal ganglion cells do. Simple and complex cells have different receptive field sizes [105], cells in different layers act differently [204], and the cell sampling issues inherent in electrophysiology mean we may have overlooked other populations [166]. Furthermore, even for a single population of V1 cells, such as complex cells, there is still a fair amount of uncertainty about receptive field size. This uncertainty may result from one of any number of reasons, but an important possibility is the sensitivity of the receptive field size measurement to the stimuli and experimental procedure. Many early receptive field mapping studies swept bars across the visual field (e.g., [74]), defining the receptive field by the bar locations where response was first evoked, resulting in a square receptive field. Later experiments (e.g., [42]) rarely use this method, instead defining center and surround regions separately by enlarging circular stimuli until responses saturate and shrinking annular stimuli until responses begin, respectively. These two methods give fundamentally different answers to the question "what is the size of the receptive field?", and reverse correlation, another common method, gives a third. Furthermore, Ziemba et al. [238] show that, even when using a given method, different types of stimuli (naturalistic textures, spectral noise, or sinusoidal gratings) result in different values for the size of the classical receptive field. All of this suggests that receptive field scaling is, unfortunately, not a simple property of visual areas, but is a function of at least the visual area, cell class, cortical layer, mapping method, and stimulus type. Thus, we should understand that there is a large amount of meta-uncertainty on

the measurements of the physiological scaling value, and this uncertainty may be part of the reason for the mismatch between our psychophysical critical scaling value and the physiological scaling ranges plotted throughout the figures in this paper.

In this study, we measured the spatial extent of pooling for luminance and spectral energy required to generate novel images that function as perceptual metamers for natural images, and showed these critical scaling values were reasonably consistent across subjects and images, as well as robust to initialization choices. While linking these values to physiology is difficult, for the reasons discussed above, we believe this consistency and robustness demonstrates that these models capture important aspects of the foveated processing of visual information. Furthermore, the increase in critical scaling and decrease in ratio between critical scaling for the synth vs. synth and original vs. synth comparisons along with statistic complexity, from luminance to energy to texture [219], suggest they do correspond to different stages of visual processing. Further work is needed to understand the source of the differences between the synth vs. synth and original vs. synth comparisons and to understand the relationship between these perceptual pooling models and neuronal processing.

3.6 Appendix

3.6.1 Dacey data

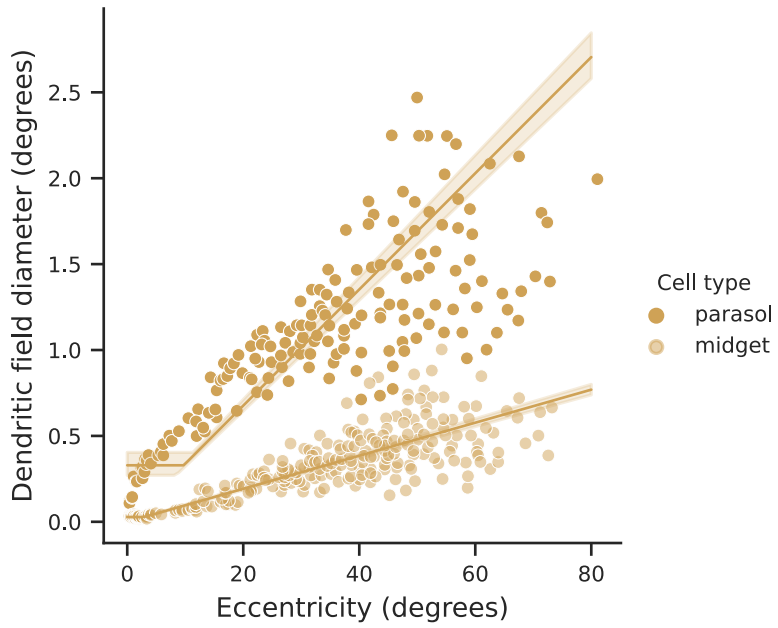


Figure 3.10: Receptive field diameter as a function of eccentricity for retinal ganglion cells. Data includes both on and off subtypes, and is from Dacey and Petersen [50] figure 2B, extracted using WebPlotDigitizer, and then fit with a hinged line through the origin using MCMC. Shaded region denotes the 95% HDI of the fit. The 95% HDI of these slopes are plotted as the range of physiological scaling values in figures 3.5 and 3.8. Eccentricity of retinal ganglion cells within 3 mm (approximately 10 deg) of the fovea have been converted to photoreceptor inner segment eccentricity using a second-order polynomial equation from Curcio, personal communication to the authors of Dacey and Petersen [50], based on direct measurement of human photoreceptor axons. While the hinged line through the origin does not fit the data for the parasol RGCs as a line with a non-zero intercept, it does not qualitatively change our interpretation of the physiological scaling value in this study, and is thus used to be consistent with the analysis of V1 from Freeman and Simoncelli [70] and the definition of scaling for our pooling models.

The range of physiological scaling values for midget and parasol retinal ganglion cells plotted in figures 3.5 and 3.8 come from the data plotted in Dacey and Petersen [50] figure 2B, replotted in figure 3.10 along with the hinged line whose slope we use as the physiological scaling value. This is similar to the procedure used by the authors in Freeman and Simoncelli [70] to get physiological scaling values for V1 and V2. The primary difference is that the Dacey data is anatomical, measuring the diameter of the dendritic field in human postmortem retinal tissue

([70] used electrophysiological measurements from macaques). Eccentricity of retinal ganglion cells within 3 mm (approximately 10 deg) of the fovea have been converted to photoreceptor inner segment eccentricity using a second-order polynomial equation from Curcio, personal communication to the authors of Dacey and Petersen [50], based on direct measurement of human photoreceptor axons. According to Brian Wandell's list of useful vision science numbers, the foveola (the region of the fovea that is completely rod- and capillary-free) is .3 mm (about 1 degree) in diameter. As midget RGCs in the fovea are connected to a single cone, the foveal cone diameter serves as a floor for midget retinal ganglion cell spatial integration at the fovea. Foveal cones have a diameter of 1 to 4 microns [221], which corresponds to .2 to .9 arcmin, two to nine times smaller than the approximately 1.8 arcmin that our hinged line predicts for midget dendritic field diameter at fixation.

As can be seen in figure 3.10, the hinged line with zero intercept is not a good fit for the parasol RGC data (though it does fit the midget data well). Using a non-zero intercept improves the prediction and slightly increases the physiological scaling values much for both classes: the 95% HDI is $[0.0092, 0.01]$ vs. $[.0091, .01]$ for midget RGCs and $[.032, .036]$ vs. $[.025, .028]$ for parasols. However, this does not change how this data relates to our results, and does align our analysis with that of Freeman and Simoncelli [70], as well as matching the hinged-line with zero intercept construction of pooling windows in our models, so we present the slopes from the line with zero intercept.

3.6.2 Differences with Freeman and Simoncelli, 2011

As noted in the text, while the critical scaling we found for the energy model comparing two synthesized images initialized with white noise was comparable to that found in Freeman and Simoncelli [70], subjects' asymptotic performance was lower (average of 60% correct across subjects and images, compared to 85%). This was also noted, for the "mid-ventral", pooled texture-statistic model, in Wallis et al. [219], who ran a control experiment to ensure it doesn't come down

to task differences. As our implementation of the pooling models is completely separate from that of Freeman and Simoncelli [70], we investigated whether differences in window construction or other implementation details could have led to any marked difference. A Jupyter notebook investigating this can be found in the Github repo associated with this project, showing that the windows appear comparable, as do energy model metamers with comparable scaling values. Additionally, if our windows were significantly different from those in Freeman and Simoncelli [70], we would expect them to affect the critical scaling, whose value relates window radial diameter to eccentricity. However, the critical scaling value for the only comparison present in both studies (energy model, synth vs synth white noise comparison) is consistent, so we believe it is unlikely that the some detail of window construction is responsible for the differences between the studies.

One possible explanation for the difference in asymptotic performance is the smaller pixel pitch of our images: 48.5 pixels per degree, as compared to the 19.7 pixels per degree used in Freeman and Simoncelli [70]. To investigate this possibility, we down-sampled our target images by half (to a resolution of 1024 by 1300 pixels) using a Gaussian pyramid (via scikit-image's `transform.pyramid_reduce` function, [33, 220]) then synthesized energy model metamers with identical scaling values and optimization hyperparameters to the energy model synth vs. synth white noise comparison. The psychophysical experiment was run as before, with one subject (sub-00), upsampling the images with nearest neighbor interpolation to present at the same physical size as before, effectively doubling the pixel pitch.

As can be seen in figure 3.11, which shows data for a single subject, the critical scaling value did not change, but asymptotic performance increased slightly. For an intuition, see the example images in figure 3.12, which show that the patterns used to differentiate the two synthesized images are slightly more obvious with a larger pixel pitch.

Beyond the difference in pixel pitch, there are several other potential factors that may contribute to our lower asymptotic performance:

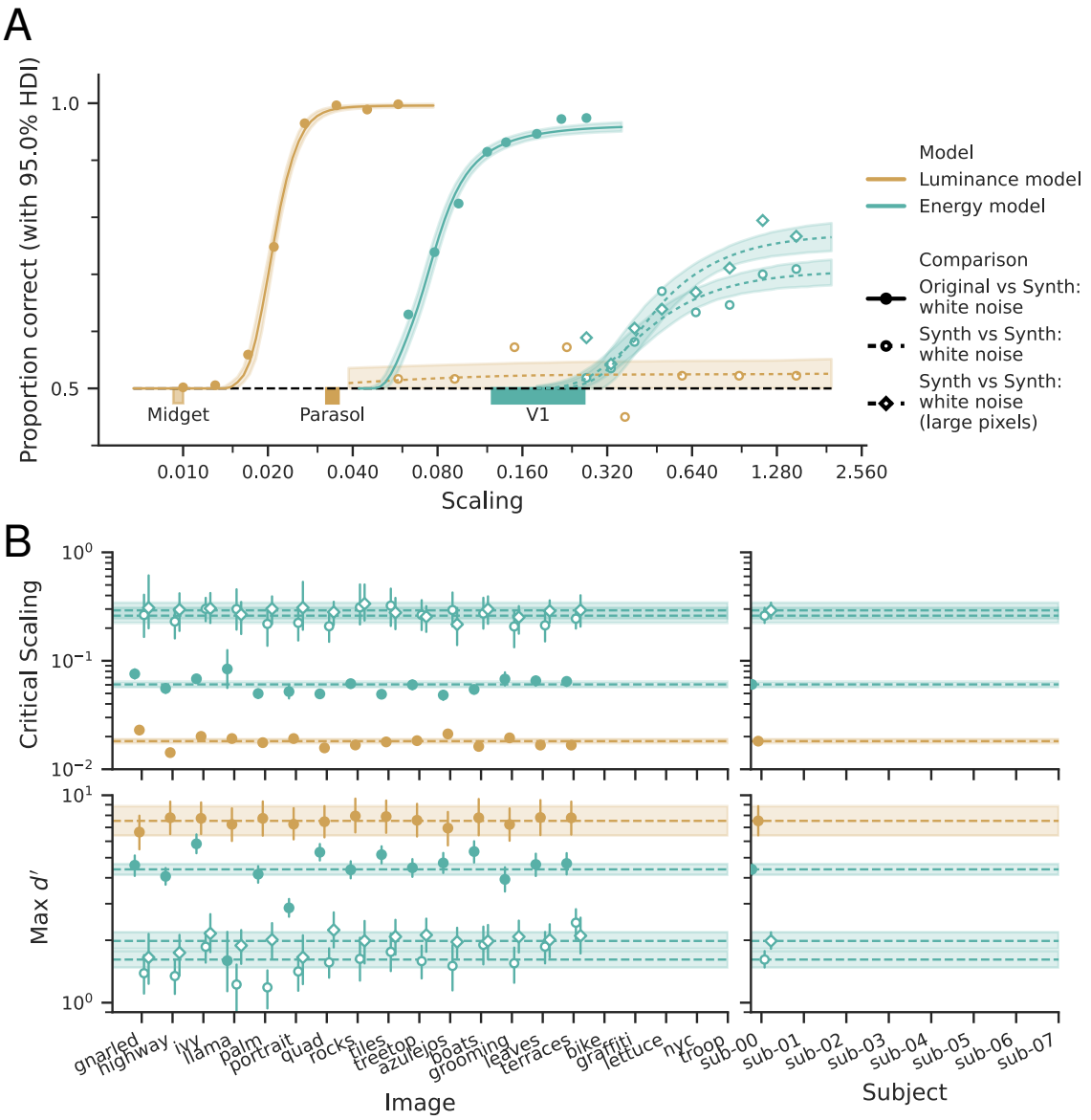


Figure 3.11: Asymptotic performance on the synth vs. synth white noise comparison is slightly higher when images have a larger pixel pitch. (A) Probability correct for one subject, sub-00, as a function of scaling for energy and luminance models. Data points represent the average across images, 540 trials per data point (one subject, fifteen images) except for luminance model synth vs. Synth: white noise comparison, which have 180 per data point (one subject, five images). Lines represent the posterior predictive means across images, with the shaded region giving the 95% HDI. Labeled horizontal bars give the range of physiological scaling values for the associated retinal ganglion cell type or cortical area. (B) Parameter values for these comparisons. Top row shows the critical scaling value and the bottom the value of the max d' parameter. Left column presents the values for each image separately for this one subject, while the right presents the values for this subject, averaged across images. In this case, we only present the data for sub-00, as they are the only subject to perform the task with larger pixel pitch. Points represent the posterior means, shaded regions the 95% HDI, and horizontal dashed lines and shaded regions average across all shown images for this subject. Note that the luminance model, synth vs synth: white noise comparison is not shown in this panel, because the data was poorly fit by this curve – as can be seen in panel A, the psychophysical curve is essentially flat at chance and thus the fit had low max d' and low critical scaling), with high uncertainty.

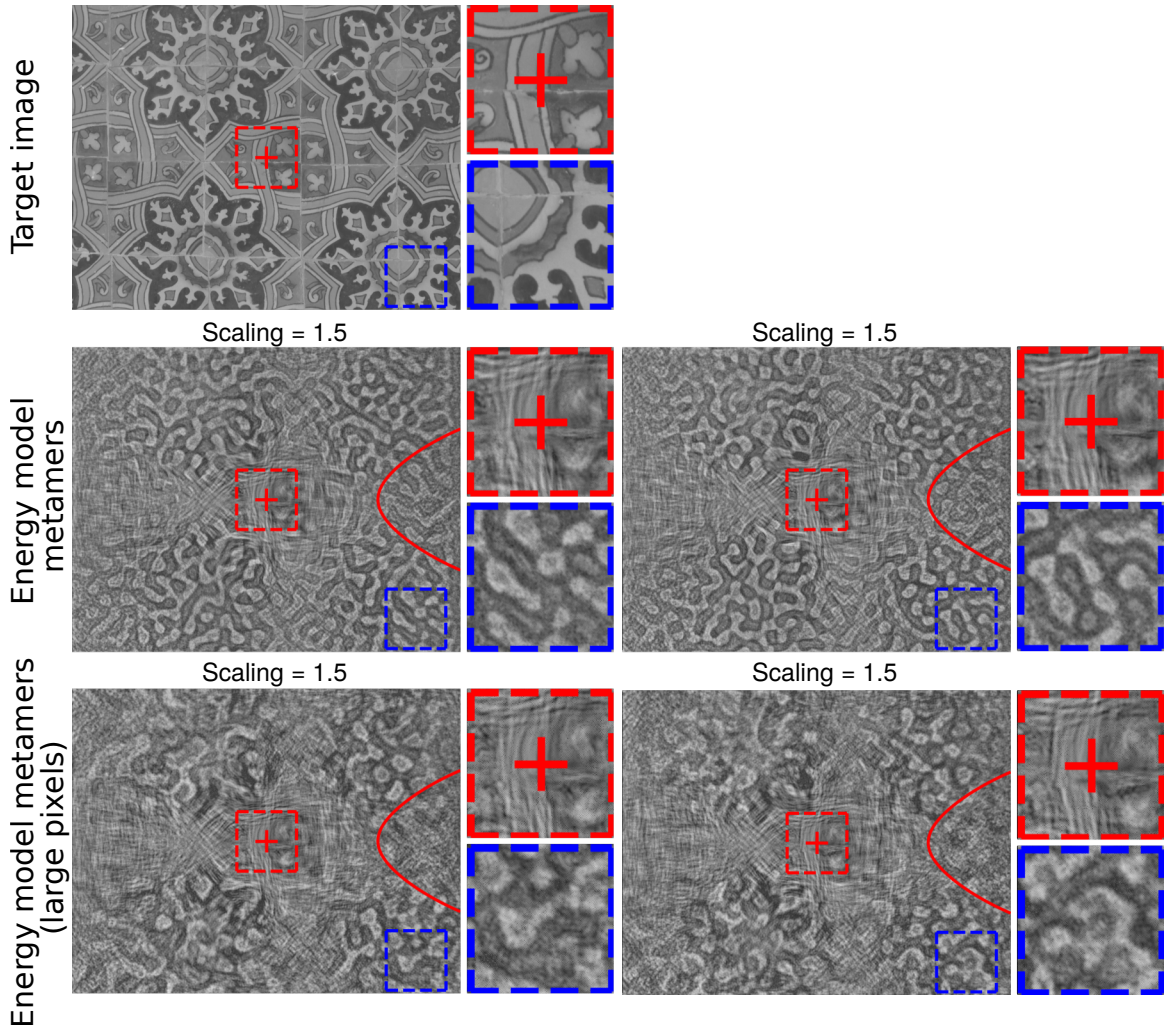


Figure 3.12: Energy model metamers synthesized with larger pixel pitch are slightly easier to distinguish from each other at high scaling values. Top row shows the target image whose representation all presented model metamers were synthesized to match. Middle row shows two energy model metamers initialized with different patches of white noise, which were shown in the synth vs. synth white noise comparison. Bottom row shows two model metamers initialized with same settings but half the pixel resolution, which were shown in the large pixel version of the synth vs. synth white noise comparison. As these images have half the resolution in each direction as the middle row but were presented at the same physical display size, their pixel pitch was doubled, making it comparable to that of Freeman and Simoncelli [70]. With the larger pixel pitch, the snake-like patterns that can be used to distinguish the two synthesized images have a lower spatial frequency (in cycles per degree), which, given their presence in the periphery where spatial frequency sensitivities are lower, may account for the participant's increased ability to distinguish such images. Full resolution version of this figure can be found on the OSF.

- Experimental parameters were determined for the original vs. synth comparison, which is an easier task than synth vs. synth comparison. Longer presentation times or some other configuration may increase asymptotic performance in synth vs. synth comparison (but, as

Freeman and Simoncelli [70] showed, these sorts of experimental manipulations are unlikely to have much effect on the critical scaling, which is the focus of this study).

- Our images are physically larger, with a diameter of 53.6 degrees compared to 26 degrees in 26 in Freeman and Simoncelli [70]. In debriefing, participants reported performing the task by finding particular informative parts of the image (e.g., high contrast edges) and attending there. Larger images may have made these regions harder to find, and the attentional manipulation in Freeman and Simoncelli [70] shows that attending to the most informative region of the image improves asymptotic performance (while leaving critical scaling unchanged). Additionally, Ziemba and Simoncelli [237] showed that the probability of correctly differentiating two samples from the same texture family decreases as those samples get larger (conversely, performance increases with image size when participants are differentiating between two different texture families). Something analogous may be happening here, with two synthesized images initialized with white noise acting similarly to two samples from the same texture family.

- As seen both here and in previous studies [57, 70, 219], asymptotic performance varies more across images and subjects than critical scaling. We have a different subset of images and subjects and so sampling issues may be partly at fault.

All told, there are multiple reasons why asymptotic performance differs between this study and Freeman and Simoncelli [70], which we are unable to comprehensively track down.

3.6.3 Images are physically distinct

There are two properties that perceptual metamers must have: they must be perceptually identical and physically distinct. We have shown our synthesized images are perceptually identical and figure 3.13 shows that the images are physically distinct. The image at the top of that figure is the original natural image whose representation the model metamers matches. The six lower

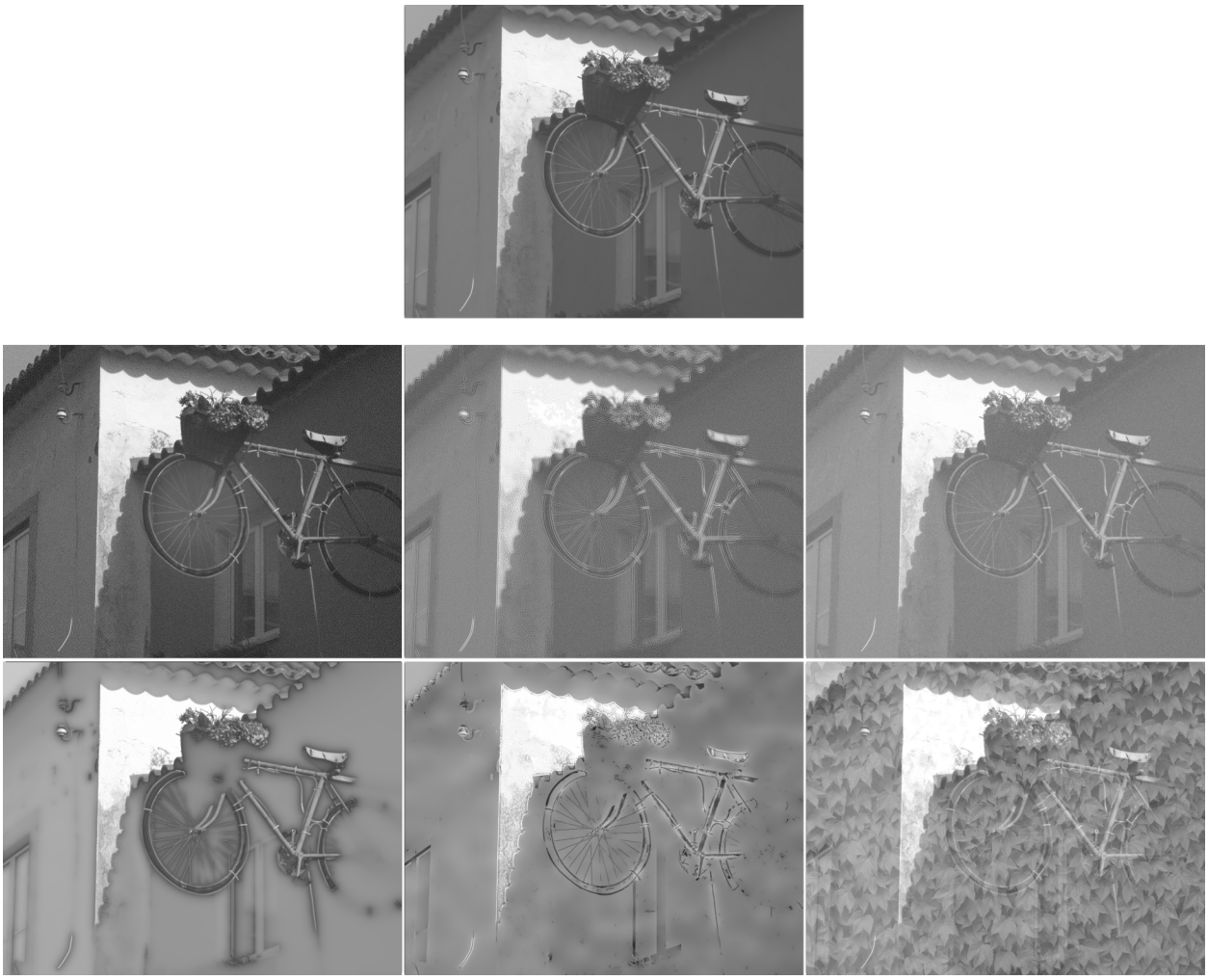


Figure 3.13: Luminance model metamers are substantially physically different from original images. The six bottom images all have approximately the same mean-squared error to the top image, the original natural image. Notice that the luminance model metamer synthesized with scaling value 0.01 (top left) is by far the least discriminable with the natural image when fixating on the center, the others are obviously distinct. This pair of original and synthesized image was chosen to set the MSE because it has the lowest MSE for all luminance metamers when compared with their original image. Full resolution version of this figure can be found on the OSF.

images all have approximately the same mean-squared error to that original image, yet only the luminance model metamer, top left, is confusable with the natural image, the others are easily discriminable from it. The top right image has had high-frequency noise added uniformly across the image and thus shows the importance of foveation: the luminance model metamer also differs from the original image primarily by the addition of high-frequency noise but, by concentrating the noise in the periphery, it is undetectable.

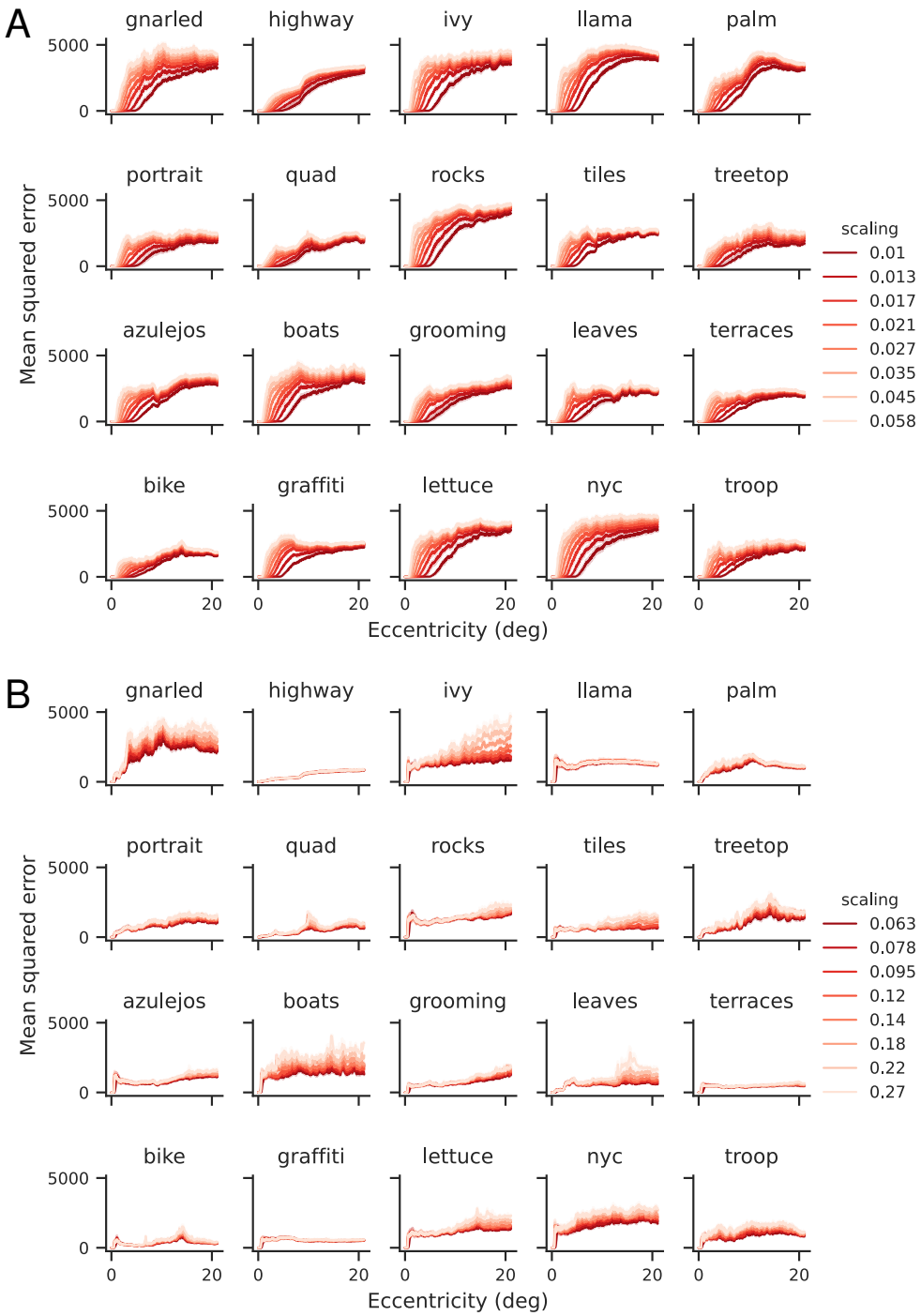


Figure 3.14: Mean-squared error (MSE) betwen model metamers and target images as a function of eccentricity, averaged radially, for the luminance model (A) and energy model (B). Each scaling value is shown as a separate color, and each target image is plotted on a separate axis. For the luminance model, the smallest scaling value has zero MSE out to about 5 degrees, and all scaling values rise with eccentricity and then saturate. For the energy model, all scaling values have non-zero MSE by 1 degree and are indistinguishable beyond that for most images, though for some, such as gnarled and ivy, lower scaling values have higher MSE across eccentricities.

In addition to global mean-squared error, we can also examine the mean-squared error at each eccentricity, as plotted in figure 3.14, which shows the mean-squared error between model metamers and target images as a function of eccentricity for the range of scaling values used in the original vs. synth comparison, for the luminance model (A) and the energy model (B). As discussed earlier, when windows get smaller than a pixel, the only value that will have the same model output is the original pixel value. One might have a similar concern for the energy model: what is the smallest pooling window in which a given spatial frequency can be computed and so will windows containing e.g., 16 pixels, also be uniquely constrained when matching 6 scales? Panel (B) shows this is not the case for our energy model metamers: all scaling values have a non-zero mean-squared error by 1 degree. However, panel (A) shows that the lowest scaling value for the luminance model metamers has a zero mean-squared error until about 5 degrees, and rises beyond that. While lower scaling values do have a smaller mean-squared error across much of the image, that does not guarantee that they are less informative: mean-squared error is a poor perceptual metric [226].

However, when mean-squared error is zero, two images cannot be discriminated between, and so we might worry that the reduced performance for lower scaling values of the luminance model do not reflect the fact that we have found perceptual metamers but that we have removed all information that participants could use to discriminate between the images because of sampling issues with our display. We think this is unlikely for two reasons. First, participants are able to use peripheral information to discriminate between stimuli, including letters [202], gratings, and Vernier lines [60]. Second, our resolution is approximately 48.5 pixels per degree, giving a Nyquist frequency of about 24 cycles per degree. Human grating acuity drops below this frequency by an eccentricity of 2 or 3 degrees [5, 60], suggesting that participants would not be able to use this information even if it were present. Additionally, participants reported that the most informative portions of the image were in the mid-periphery, across all conditions, providing additional evidence that they were not relying on the portion of the image where the lowest

scaling values were matched to the target image in order to perform the task. Further experiments investigating which portion of the image were the most informative in a more systematic way, such as restricting the stimuli to annuli at different eccentricities and comparing the resulting psychophysical curves, would provide clarity on this matter.

3.6.4 Image and subject differences

Figures 3.15 and 3.16 show the performance for each image and subject separately, respectively (collapsing over the other dimension). As can also be seen in figure 3.8(B), there's not much variability in subjects' critical scaling. Additionally, we see that the two extreme images (examined in figure 3.7) are only extremes for the energy model when comparing against natural and synthesized images; there's nothing special about the image contents in and of themselves.

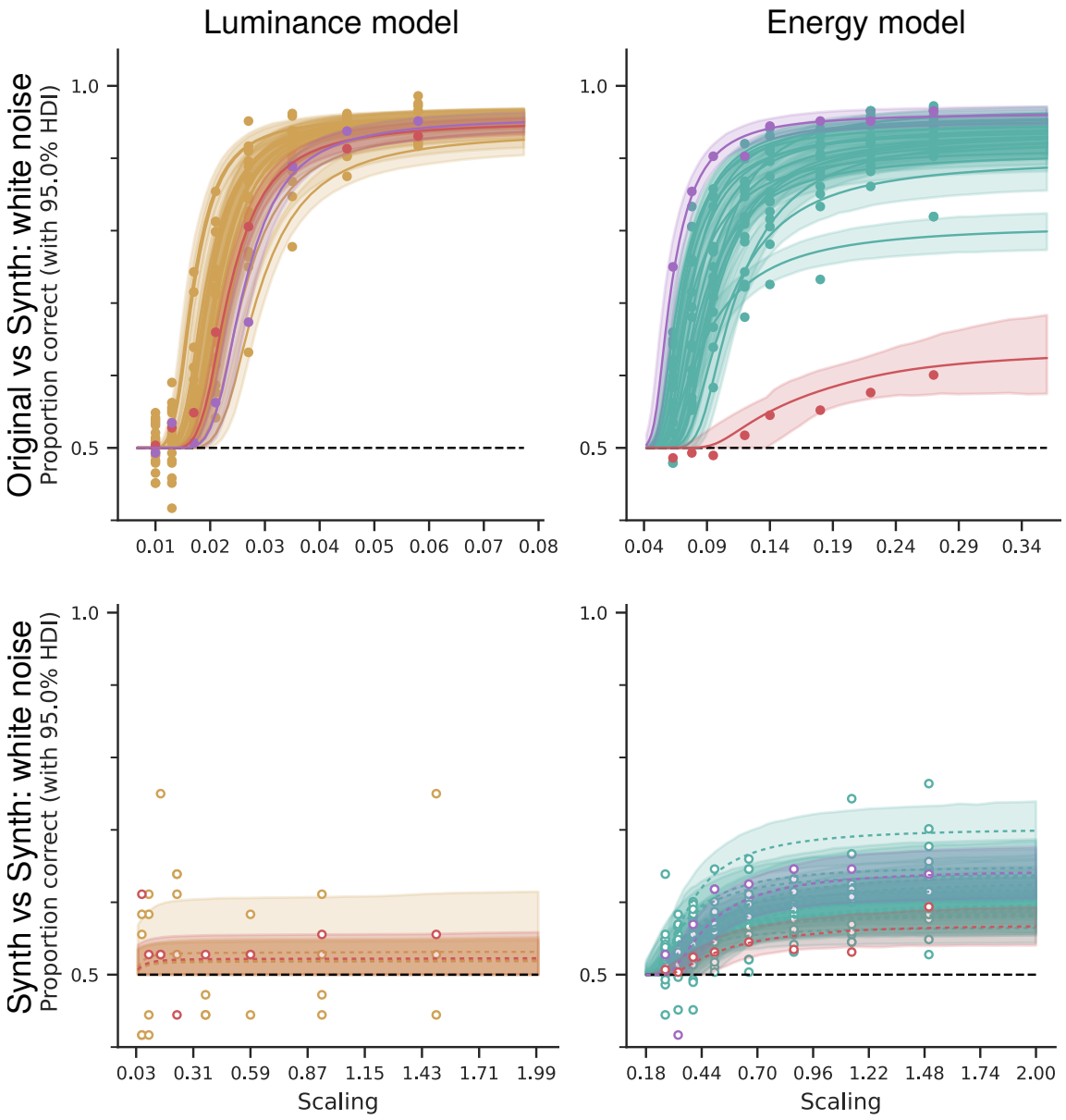


Figure 3.15: Image content only matters for the energy model, original vs. synth comparison. Performance for each image, averaged across subjects, is plotted separately. This is because the image content affects performance based on how it interacts with the models' sensitivities and insensitivities: llama contains little phase structure and is thus easy for the energy model to capture even at larger scaling values, while nyc contains many hard edges, and is thus difficult for the energy model to capture even with the smallest tested windows. However, for the luminance model original vs. synth comparison, the relevant interaction would be with the Fourier spectra: the model is insensitive to high frequencies in its periphery. As all images have $1/f$ spectra everywhere in the image, none are outliers in terms of difficulty. For the synth vs. synth comparisons, on the other hand, by definition, neither displayed image contains those features that the model is insensitive to, which is what enables correct performance in the original vs. synth case. Thus, images are more interchangeable with each other. Colors and symbols are the same as in 3.5, except for the two lines showing the nyc (purple) and llama (red) images, which are colored as in 3.7. Top right subplot is reprinted from 3.7(A). Note that bottom left subplot only contains five images (and in particular does not include the purple line representing nyc), as only a single session of that comparison was run.

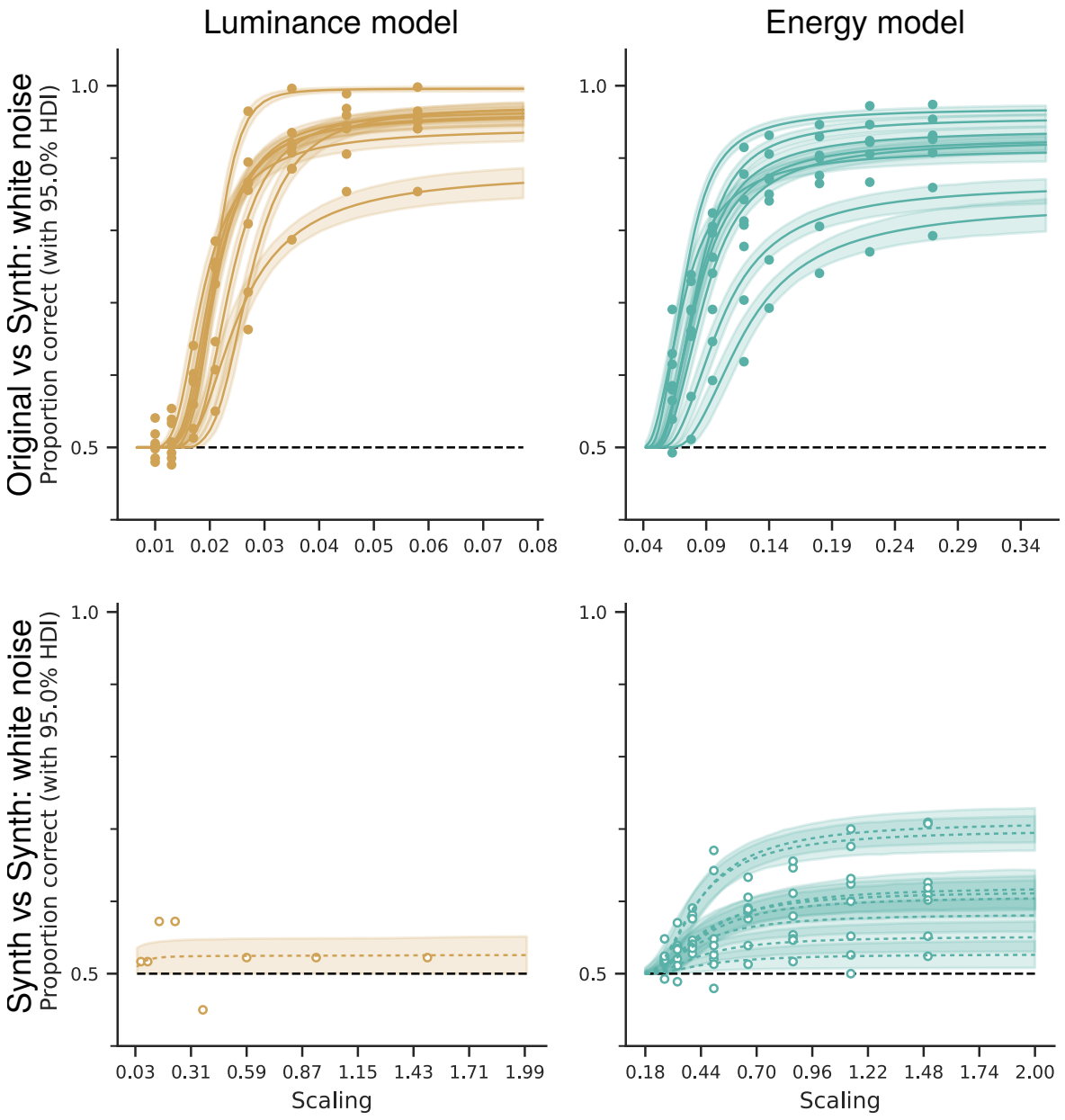


Figure 3.16: Performance does not vary much across subjects, and varies more in terms of max d' than critical scaling. Each line represents the performance for a single subject, averaged across images. Colors and symbols are the same as in 3.5. Note that bottom left subplot only contains a single subject, since only one subject completed this comparison.

4 | PLENOPTIC: AN OPEN-SOURCE PACKAGE FOR IMAGE SYNTHESIS

4.1 Abstract

Computational models are powerful tools for understanding the visual system, enabling researchers to implement their theories and providing specific predictions to test against. However, any experiment or benchmark dataset is necessarily limited, and models often behave unexpectedly on out-of-distribution data, underscoring the difficulty of understanding how a model transforms input to output. Stimulus synthesis provides one method of doing so, by creating novel informative stimuli to test or distinguish models, and several such methods have been developed in the Lab for Computational Vision over the years. Although several of these have been released as source code, they are not easliy generalized for wider usage, limiting their use by the broader scientific community. `plenoptic` is a software package developed by graduate students and postdocs in the lab to provide a unified framework for four such synthesis methods, as well as several models developed in the lab. This package does not provide novel algorithms or models, focusing instead on well-documented, tested, and generalized implementations of methods and models already described in the literature, many of which are demonstrably interesting to the community. This focus on utility over novelty is deliberate, and speaks to a broader under-appreciation of the importance of software and software maintenance within the scientific community more broadly,

which hampers our ability to build cumulative knowledge. This chapter will discuss these issues, as well as the contents of `plenoptic` and the scientific investigations it enables, focusing on the two methods which were my primary focus.

4.2 Introduction

Computational models are generally evaluated on their ability to perform a task, such as classification of images into pre-defined categories or predicting neural activity, but even when performing well on such tasks, models can behave unexpectedly on out-of-distribution data. The burgeoning literature on adversarial examples and robustness in machine learning provides many examples of this, such as the addition of a small amount of noise (invisible to humans) changing the predicted category [203] or the addition of a small elephant to a picture completely changing detected objects' identities and boundaries [186]. Furthermore, vastly different models can perform equivalently well on tasks. For example, Lescroart, Stansbury, and Gallant [138] show that models of Fourier power, subjective distance-to-object, and object categories all account reasonably well for BOLD activity in human mid-level visual areas. While factors such as number of parameters can be used to adjudicate between similarly-performing models, it is unlikely that the models would produce the same predictions for *all* possible images. Most likely, their similar overall performance results from a limited stimulus set, yet finding an effective set of stimuli for model comparison or understanding is difficult, owing to the huge number of possible images. Image synthesis provides one way of generating a stimulus set that allows for effective evaluation of and differentation between models.

The Lab for Computational Vision has developed multiple synthesis methods for better understanding model representations over the years but, while they all share a similar conceptual framework, they were each developed in the context of a single research project, with limited generalizability beyond the project's focus, and across a variety of programming languages, limiting

interoperability.

`plenoptic` was developed to provide a unified framework for image synthesis, and is available on GitHub. This package relies on a python-based machine learning library `pytorch` [169] which has taken off in popularity within the research community. Its implementation of automatic differentiation allows us to implement synthesis methods without requiring the manual differentiation that slowed the development of these synthesis methods and allows the methods to be used by arbitrary models, as long as they meet a short list of requirements. The package also provides some canonical models and computations, such as the steerable pyramid, Portilla-Simoncelli texture statistics [178], and the FrontEnd models describe in Berardino et al. [17].

By providing open source, well-documented, and fairly general implementations of these methods, we hope to enable scientists, especially within the vision science and machine learning communities, to use these methods to better understand and ultimately improve their own models, developing experiments for testing them and performing model comparison. While almost all components of the package have been previously described in the literature, re-implementing them is non-trivial. Thus, creating and maintaining high-quality versions of these methods provides a valuable resource to the research community. Furthermore, simply uploading a zip folder of research code online is not sufficient — to be most useful, the library should be freely available, extendable, easy to install, regularly tested and maintained, thoroughly documented, and should include tutorials to get users started.

This chapter will give an overview of `plenoptic`, using it as a case-study to discuss both the importance of open source software in academic scientific research as well as the conceptual framework of **synthesis** as a way of understanding computational visual models. `plenoptic` includes four methods for synthesis, two of which, `Metamer` and `MADCompetition`, will be the focus of this chapter, as they were my primary contributions to the package. Finally, we close with a brief usage example in a Jupyter notebook.

4.3 Open source software is critical and under-valued

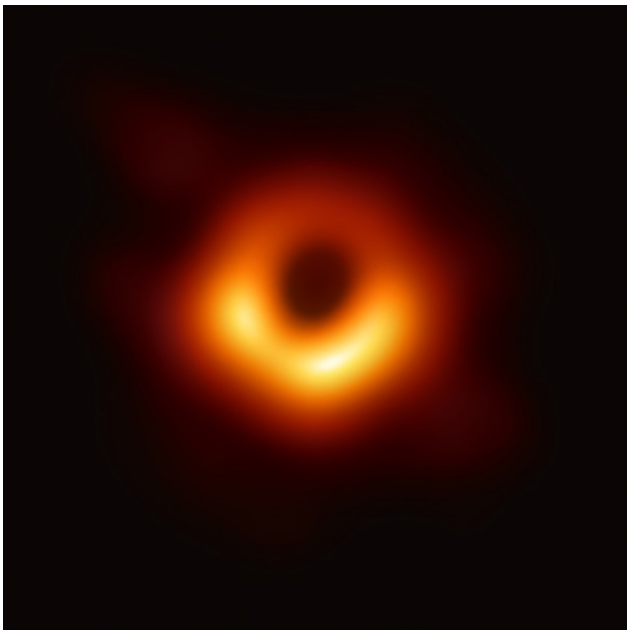


Figure 4.1: Image of supermassive black hole M87, from [229] (image credit: Event Horizon Telescope Collaboration). This image was created using tools from the scientific Python ecosystem, whose maintenance the National Science Foundation subsequently refused to fund for lack of impact.

Open software is critical for modern science, but is under-valued. The anecdote that best exemplifies this situation comes from April 2019. That month, a team of astrophysicists stitched together images from the Event Horizon Telescope to create the first image of a black hole, figure 4.1, a feat that captured headlines around the world. They did this using a pipeline based on packages from the Python research software ecosystem, which includes Matplotlib [106] and NumPy [86], among others. Yet, five days after that announcement, the US National Science Foundation (NSF) denied a grant to support that ecosystem, saying the software did not have "sufficient impact" [161].

These software projects persist largely through the efforts of volunteers. While the NSF awarded 9.6 billion dollars from 1995 to 2016 to grants with "software" in their abstract [40], "grant-based funding is often exhausted shortly after new software is released, and without

support, in-house maintenance of the software and the systems it depends on becomes a struggle" [180]. This funding problem occurs, ironically, just when software is starting to be used by the broader community and when its support becomes more important. The most mature and widely-used projects, such as NumPy and Matplotlib, may seek grants from governmental and non-governmental sources to support their ongoing work. NumFocus is a non-profit that supports much of the Python scientific ecosystem, yet their revenue in 2020 was approximately 5 million dollars [196] to support 44 "sponsored" open source projects (largely in Python, but also in R, Julia, and more). NumFocus received a large grant from NASA that year, but most of their revenue came from private foundations (primarily the Chan Zuckerberg Initiative and the Alfred P. Sloan Foundation) and corporate donations. For comparison, that year New York University alone received 41 million dollars from the NSF [162], and 77 million dollars from the National Institutes of Health (NIH), with an additional 329 million dollars from the NIH for the NYU School of Medicine [158]. In the fiscal year 2020, the NSF disbursed a total of 7.75 billion dollars in grants [164], while the NIH disbursed 30.75 billion in extramural research grants [157] (while these grants support research and education in many fields of science, so does NumFocus's work). While I was writing this chapter, the NSF announced a new funding initiative, Pathways to Enable Open-Source Ecosystems, specifically to support open source ecosystems, rather than specific tools [163]. While this is a positive step, the program is specifically *not* intended to fund existing open-source communities and ecosystems, which is part of a recurring issue (not just limited to software support) where funding sources prefer to support new initiatives rather than existing ones. As a result, many scientific software packages rely on lone or small groups of maintainers who support the package in their spare time, a situation captured by the popular webcomic xkcd in figure 4.2.

Yet this lack of investment stands in contrast with scientists' heavy use and reliance on software. A 2014 survey of British academics [96] found that 92% use research software, a number which is almost certainly higher in the experimental sciences, where it is hard to imagine data analysis

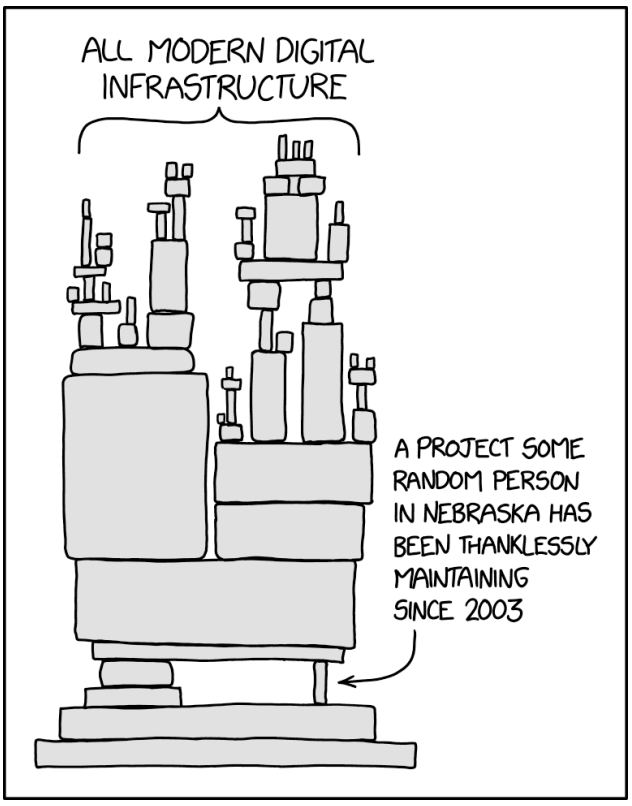


Figure 4.2: xkcd webcomic satirizing the state of support for crucial software, from [154]. The webcomic discusses "modern digital infrastructure", of which scientific software is but a small part, demonstrating that the problems with supporting the maintenance of open source software is not limited to academic science. This was made clear by the discovery of the Heartbleed [61] and Log4Shell [228] vulnerabilities, in 2014 and 2021, respectively.

being performed without the use of software. Furthermore, 69% say their research would not be practical without research software. And, in a 2014 analysis, the "vast majority" of the top 100 all-time cited papers (requiring at least 12,119 citations) describe experimental methods or software [159]. The share of software in that analysis is almost certainly an undercount, given the inconsistencies with which scientists cite software: Howison and Bullard [101] surveyed a random sample of 90 biology articles and found that only 31% to 43% of the software mentions involved formal citations; most were informal mentions that would not be counted by the aforementioned analysis (and this practice of informal mentions holds across journal impact factors and software types).

Software is important because scientists use it. A key goal of a scientific publication is to

enable other researchers to understand and reproduce the steps that led to the result, yet many computational analyses cannot be adequately translated into words or equations [111]. This is even true for pseudocode, popularly used to represent algorithms in the computer science literature, an example of which is Porter's stemming algorithm [208]. Stemming is the process of reducing inflected words to their word stem (e.g., "stemming" has the stem "stem"), an important problem in computational linguistics. In 1980, Martin Porter published the pseudocode for this algorithm, which would go on to be very influential in the field [177]. Many implementations of the algorithm were implemented and distributed, but the author put out an official version on a web site in 2000 because "unfortunately there were numerous variations in functionality among these versions, and this web page was set up primarily to 'put the record straight'and establish a definitive version for distribution" [176]. A pseudocode description of an analysis is often more detailed than those given in neuroscience and psychology journals, which reduces the chance of being able to reproduce an analysis even further. Different teams of scientists will also choose different analytic approaches and pipelines to address the same question, leading to very different outcomes [22, 195], emphasizing the importance of software to live up the ideal of scientific communication enabling reproducibility. As Buckheit and Donoho [32] put it, "an article about computational science in a scientific publication is *not* the scholarship itself, it is merely *advertising* of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures."

Modern scientific analyses are complicated and, while sharing code is an important step in enabling computational reproducibility, it is often insufficient. A zip file full of uncommented scripts with no instructions on how to set up the environment or how to run them is of little use. Fortunately, multiple guides have been put together detailing simple steps to make shared code as useful as possible (e.g., [62]). Scientists can leverage tools created specifically for reproducible workflows (e.g., [77, 80, 149]) as well as make use of or adapt tools from the flourishing community of open-source software development, such as git.

When software becomes more general-purpose than the workflow for a single publication, scientists face additional difficulties. Since the early days of the neuroimaging field, a variety of packages have been developed for the specialized analyses common to magnetic resonance imaging (MRI). These packages are freely available and widely used, with the source code available once one agrees to a simple license, but they do not follow the standard practices of contributorship for open source (probably because they predate the widespread adoption of GitHub and similar web tools which facilitate these practices). Many modern open source packages, including the Python scientific ecosystem described earlier, host their source code on GitHub or similar websites, which allow anyone to view the source code and the entire history of changes. Additionally, these websites allow for easy integration with a variety of continuous integration tools, which run regular tests, including before accepting any changes, to ensure consistent results across time and different systems (such as operating systems and software versions). They also allow for easy versioning, tagging specific moments in the code with human-readable labels such as `v1.2.1`, which allow scientists to specify which version was used in their analysis and, via archiving services such as Zenodo, the creation of a persistent URL to that version (Freesurfer [51], to pick one example, does version their code but does not use persistent URLs and only versions `5.3`, `6.0.0`, `7.1.1`, and `7.2.0` are available on their website). Finally, anyone with a free account on these websites can file a public issue, asking for help or alerting contributors to difficulties with or bugs in the code, as well as propose changes via a pull request, which can be reviewed by developers and, if appropriate, merged into the main version of the code for later release. These practices can be followed without the use of such websites, but the websites facilitate them and make them transparent to users of the software.

Public tests, in particular, are useful. Among other issues, they allow users (and developers!) to be sure that the same results are obtained across operating systems (which cannot be assumed, see [19, 83]). Without public tests, users cannot be sure of what *is* being tested. Several studies have highlighted the need for validation in fMRI analysis packages [63, 136, 137]; in an ideal world,

the authors of those papers would be able to open a pull request adding their validation tests to the testing suite of the packages they investigated. I have highlighted the tools and practices that I believe would allow these packages to manage these issues, but in an environment where software is under-valued and under-appreciated, validation tests are a likely casualty, as they are more complicated to write and maintain than tests that e.g., check whether a line of code runs without failing.

Scientific software lives in a strange gray area; it does not align well with the current incentives of either industry or academia. Industry does a poor job even supporting open software that is obviously and directly crucial to its bottom line, such as web infrastructure (see figure 4.2), let alone scientific software whose connections are less obvious. The exceptions are the large Python deep learning libraries, such as TensorFlow (supported largely by Google) and PyTorch (supported largely by Facebook). This may be because of the relatively porous boundary between industry and academia in machine learning, with researchers moving between the two and often having joint appointments. Industry's support of these libraries may also reduce training costs: as the libraries are open and standard in the field, companies can reasonably expect their interns and full-time hires to already be familiar with them, instead of having to spend time training them. Similarly, using an open source library allows them to accept contributions from a wide range of machine learning practitioners and researchers, including many who do not work at their companies. However, we should note that corporate interests are not those of academic scientists, and so they will make different choices about what to develop and prioritize. This is most obvious for corporations selling closed software: MathWorks, for example, does not provide an official declarative dependency manager for MATLAB, which would facilitate the sharing of open source code and reproducibility of analysis pipelines, and is common in modern open source languages; this has led to the development of three unofficial managers, all built by neuroscientists rather than experts. But even in industry-supported open source software, companies are unlikely to pursue the same objectives as academic scientists; while implementations of standard convolutional

neural networks such as VGG-16 and AlexNet are found in every deep learning package, standard vision science models are nowhere to be found. This is partly the fault of vision scientists, who have not put great effort into sharing and standardizing such models (but see [13, 165]), but any drive to do so is unlikely to come from Google or Facebook. Furthermore, this misalignment of interests also means that for many packages that are essential to scientific practice, such as those that specialize in running psychophysical experiments, industry is not a reliable source of support.

Unfortunately, neither is academia. Publications are the currency of academia, yet they are a poor match for software [110]. Scientific publications serve as the source for a particular scientific claim or evidence, and they almost never change after publication. However, software is constantly evolving: new functionality is added, bugs are fixed, etc. Specific releases should be archived, as described above, but the package as a whole is always in flux. This is a *strength* of software: Gardner et al. [73] found that active maintenance, not the journal impact factor or number of citations for the corresponding article, was the most indicative measure of accuracy for bioinformatic software. Fortunately, a variety of measures are being proposed and systems being built to facilitate the recognition, support, and quality of software in science (e.g., [9, 131]), along with a variety of organizations pushing for these changes, such as US Research Software Engineer Association (and similar organizations in other countries), Software Sustainability Institute, and Research Software Alliance. Scientific software is simply too important to be reliant on a handful of over-worked postdocs and volunteers or the vagaries of industry: academic science must value software as a scientific output in its own right [110, 126].

4.4 Scientific motivation for plenoptic

4.4.1 plenoptic enables cumulative science

As described above, `plenoptic` is an open-source python library developed by a team of graduate students and postdocs in the Lab for Computational Vision (LCV) over the past several years. While the package contains no novel methods or models not already described in the literature, it serves to standardize and generalize a variety of tools developed in the lab over the past several decades, making them available and useful to the broader vision science community. As is known, reimplementing a novel method or model from only the description in the paper is exceedingly difficult: Thimbleby [208] describes the proliferation and continued use of incorrect implementations of Porter's stemming algorithm, which was originally described in pseudo-code. This example arose from pseudo-code, which is generally more explicit than the algorithmic descriptions given in the vision science literature, and in the computer science community, which is more skilled in developing software than the vision science community, so we can only imagine what similar issues in our field must look like. Thus, providing a tested and maintained implementation, when possible, is the ideal situation.

Furthermore, the shared framework provided by `plenoptic` allows researchers to make use of the four provided synthesis methods with the same models, allowing for more expansive experiments and model exploration. Our methods can be used with the models included as part of `plenoptic` or user-created models. Models can be arbitrary: we only require that models be implemented in `pytorch`, accept real-valued four-dimensional input (this `(batch, channel, height, width)` shape is fairly standard in the deep learning community) and return real-valued three- or four-dimensional output (either vector- or image-like). For synthesis results to make sense, arbitrary input (and probably output) values should be interpretable. As the intended use case of this library is for visual models, which often take images as input, this is not too big an ask.

The package also provides tested, differentiable implementations of existing models. These can be used as is or as building blocks in other models (e.g., the Steerable Pyramid is the first stage of the foveated spectral energy model discussed in chapter 3).

Our package builds on `pytorch` [169], a deep learning library popular among researchers. Previously, using a synthesis method required computing the gradients manually, a time-intensive and error-prone process that would need to be repeated whenever the model or algorithm changed. We are able to provide general implementations by leveraging `pytorch`'s automatic differentiation, which automatically computes the gradient of any computational graph implemented in `pytorch`. This powerful tool is supported by a large community of contributors, both in industry and academia, providing a significantly more robust pool of knowledgeable support than LCV would be able to manage by itself. Furthermore, by building directly on top of `pytorch`, researchers can use our synthesis methods with the wide array of standard deep learning models it provides. These models can also be tweaked and their parameters learned, allowing researchers to train on a large database, for example, and then, using the same code object, synthesize images using those learned parameters.

`plenoptic` also follows open-source best practices: the code is available on GitHub under an MIT license, allowing others to view, use, and modify it as they see fit, and invites users to contribute their changes back to the package. Using GitHub and open source licenses also allows us to make use of existing tools to run tests and validations on the CPU before any proposed change, ensuring tutorials stay up-to-date and the code does not break unexpectedly. Through partnership with the Flatiron Institute, we are able to use their computing resources to run all tests on GPUs as well, ensuring consistency across both device types; as GPUs speed up the required computations greatly (approximately 10 times faster for the foveated energy model metamers in chapter 3), this is especially important. By using git (which functions as a public change log) and semantic versioning (along with long-term archiving via Zenodo), we enable users to specify which version of the code was used to generate a given result, facilitating reproducibility. While

following the above requires more work than simply uploading a zip directory to a website, this set of practices, along with regularly-built public documentation, is considered best practice for open-source and research software development [114, 232], and increases the usefulness of the package to the community.

Finally, `plenoptic` serves as a repository of institutional knowledge for LCV. Because of the nature of academic research, with graduate students and postdocs arriving regularly, working on individual projects, and then moving on, building institutional knowledge in a lab and facilitating its transfer so that everything is not forgotten is incredibly difficult. The PI can serve as a hub for this knowledge, but as the lab grows and the PI gains more responsibilities, this tends to fall by the wayside. This leads to students and postdocs wasting time when extending or using someone else's work. `plenoptic` serves as a repository for this knowledge about the included models and synthesis methods: we have learned a lot about the steerable pyramid [198] and Portilla-Simoncelli texture statistics [178], especially, as well as reproducing previously-published synthesis examples, which increased our understanding of how best to use these methods. The comments within the code, documentation, and tutorial notebooks serve as a record of our hard-won understanding, and their central location ensures that all former, current, and future members can access them. As LCV is an active member of the vision science community whose methods are used by labs around the world, such knowledge will also benefit the rest of the field.

4.4.2 Stimulus synthesis as a framework for model understanding

Synthesis is a framework for exploring models that takes advantage of the fact that we can create stimuli, not just rely on existing ones. Computational models take a stimulus as input, perform some computations based on parameters, and return an output. In visual models, the focus of `plenoptic`, the inputs are typically images and the outputs are some abstractions of representation, which are used to predict neural activity or behavior of some kind. Most commonly, researchers use these models alongside experiments, simulating model responses (with fixed

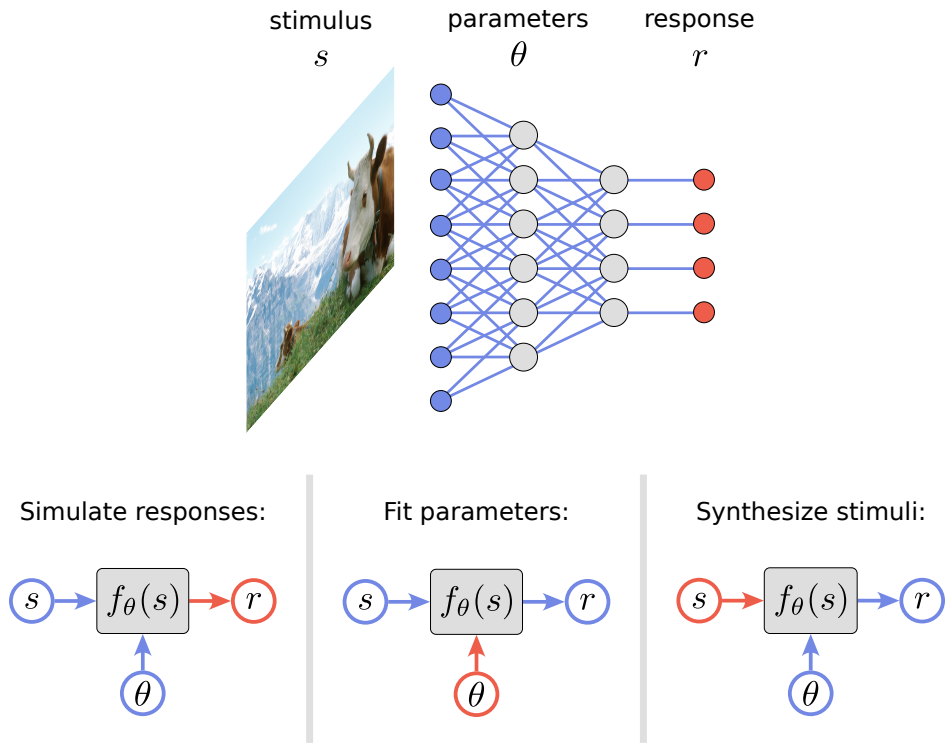


Figure 4.3: Schematic describing relationship between simulate, fit, and synthesis. Computational models take a stimulus as an input and, given some parameters, simulate a response. In vision science, the stimulus is often an image and the response is some behavioral measure, such as discriminability, or a neural response. The most common ways of using these models is to simulate the response, where the stimulus and parameters are held fixed to generate the response, or to fit the parameters, where the stimulus and responses are held fixed and optimization is used to find the best-fitting parameters. However, there's nothing special about the stimulus: we can also hold the response and the parameters constant and use optimization to generate a novel stimulus. `plenoptic` provides a set of tools for synthesizing images. Simulation is used during synthesis, while model fitting is typically done separately, as part of the overall experiment in which synthesized images are used (see figure 4.6 for a psychophysical example which fits parameters after synthesis, but a pre-trained model, such as a neural network trained on ImageNet, could also be used). Original figure created by Eero Simoncelli.

parameters) to a variety of stimuli that are compared against other models, neural responses, or animal behavior. Researchers also often fit the parameters of their model, using optimization to find the parameter values that best align model responses with the output of interest for the tested set of inputs. However, stimuli are not special, and researchers can similarly hold parameters and responses fixed, while using optimization to generate new stimuli (see figure 4.3 for a schematic comparing these procedures). We refer to this process as **synthesis**.

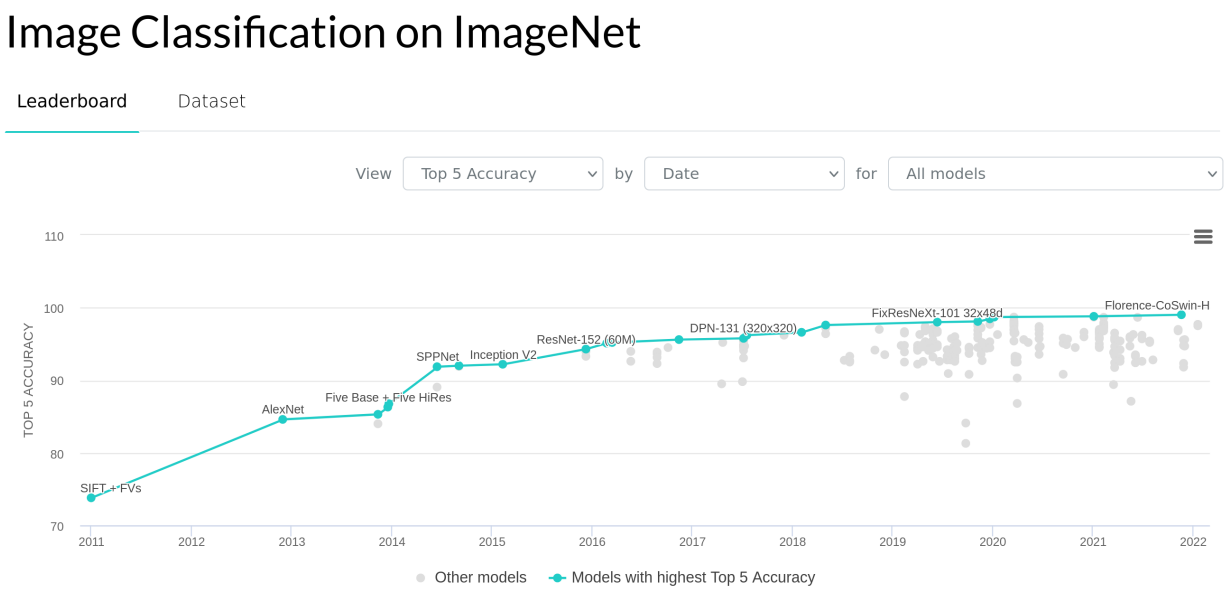


Figure 4.4: Screenshot of top 5 accuracy image classification performance on ImageNet from [167]. After rapid advancement between 2011 and 2015, performance has approached the ceiling, with little improvement from one year to the next and many models performing similarly, despite a large amount of work on this data set. While ImageNet has been instrumental in stimulating work on this area, additional methods for investigating and comparing these models are required.

The goal of synthesis is to explore the input space to improve our understanding of a model's representational space, exploring model insensitivities in various ways. `plenoptic` provides four such methods, which provide different ways to do so. We can generate sets of physically distinct images with the same model representations, **metamers**, to better understand a model's insensitivities: while all synthesized images will share some features with each other and the target image, features that the model is insensitive to will differ. **Eigendistortions** show the most and least noticeable changes to an image for a given model, highlighting the features that a model

is the most and least sensitive to. **Geodesics** give you the shortest path between two images in a model's representational space. Finally, **Maximally Differentiating (MAD) Competition** compares two models, synthesizing an image that one model's finds as different or as similar to the target as possible, while the other model's representation is unchanging. Together, these methods give researchers a set of tools to investigate what features of the input are considered important and unimportant. This focus on model null spaces is often over-looked, but has a long history in vision science, dating back to work on trichromacy in the 19th century [91], the understanding of which allowed for the development of three-channel color displays.

What is the scientific value of generating stimuli? Computational models are generally evaluated on their ability to accurately perform a task, such as predicting neural activity or correctly classifying images, but this focus hides at least two difficulties. First, many competing models can perform approximately equivalently on a task: for example, as of February, 2022, more than 100 models have above 95% top 5 accuracy on ImageNet, with 9 models within a percent of the top performer at 99.02% (see figure 4.4, [167]). Furthermore, the state of the art top 5 accuracy has been at or above 95% since 2016, with an improvement of only 4% in the past six years. With so many models performing the task well, and state of the art performance already so close to ceiling, we need some other way of discriminating between competing models. We could develop another benchmark dataset or use meta-scientific concerns to choose among them (such as number of parameters, training cost, etc.), but synthesis provides another way to do so. Second, models can perform unexpectedly on out-of-distribution data: the burgeoning literature on adversarial examples and robustness in machine learning provides many examples of this, such as the addition of a small amount of noise (invisible to humans) changing the predicted category [203] or the addition of a small elephant to a picture completely changing detected objects' identities and boundaries [186]. Exploring model behavior on *all* possible inputs is impossible, but synthesis provides one mechanism for exploration in a targeted manner.

Note that, while many of the above examples come from the machine learning literature,

similar problems, though less well-documented, occur for neuroscience researchers as well. For example, neuronal responses in primate primary visual cortex are frequently modeled with a linear receptive field followed by some non-linearities. This receptive field is generally modeled as an oriented, bandpass filter, two of the most popular being the steerable pyramid ([70, 198], chapter 3 of this thesis) and a Gabor filter bank [52, 115, 142]. Another approach that has gained popularity in recent years is to predict neural responses using some linear combination of units in a convolutional neural network (e.g., [235]), the filters of which have no simple parametric form. All of these models have been used in the literature, though generally not in direct competition (but see [142] for a step in that direction), and perform adequately. Synthesis would provide an additional tool to distinguish among competing models.

In addition to the synthesis methods present in `plenoptic`, a variety of related concepts have developed over the years in both the vision science and machine learning literatures. These include mongrels [7], eidolons [127], feature inversion [140], adversarial images [203], DeepDream [150], deep visualization [236], style transfer [75], and controversial stimuli [78], among others. All of these methods except for controversial stimuli (which is a variation on MAD Competition) can be understood as either identical to or slight variants on metamers: they hold the representation of the model constant and use optimization to generate novel inputs, starting with another image or a patch of noise. When working with neural networks, many of these methods match representations at various layers, not just the final output, either to understand that layer better [66, 236] or, in the case of style transfer [75], to mix "content" and "style" features that are represented at different layers. Some methods add additional constraints or regularizations: Szegedy et al. [203], the first paper showing adversarial examples, minimized the pixel-wise difference between their initial and synthesized images, while many, such as Mahendran and Vedaldi [140], Mordvintsev, Olah, and Tyka [150], and Yosinski et al. [236], use some form of natural image prior (though the exact details vary) to produce more human-recognizable images (as Feather et al. [66] point out, the importance of using such priors tells us there are important differences between the

representations of these networks and those of humans). All of these examples fall under the umbrella of metamers, synthesizing novel images that match a model's representation.

4.5 Package contents and contributors

`plenoptic` contains three main components: `synthesize`, `simulate`, and `metric`. One of the advantages of providing the implementations in `simulate` and `metric` is that these are differentiable and GPU-compatible, enabling them to be used in our synthesis framework, as well as other applications, and greatly speeding up their performance.

1. `synthesize`: implementations of `Metamer`, `Eigendistortion`, `Geodesic`, and `MADCompetition`.
    - `Metamer` operates on a single model and a single reference image, synthesizing a novel image (initialized by default with a patch of white noise) whose model representation matches that of the reference image [70, 178].
    - `Eigendistortion` operates on a single model and a single reference image, synthesizing the most and least noticeable distortions for that model on that image. An extension of the original, included in our package, allows the user to synthesize the intermediate distortions as well [17].
    - `Geodesic` operates on a single model and a pair of endpoint images, synthesizing the frames that would lie between those endpoints on a straight line through the model's representational space [92].
    - `MADCompetition` operates on a single reference image and two metrics, synthesizing sets of images that are the most or least different from the reference image for one metric, while holding the other metric's value constant [225].
2. `simulate`: implementations of several simple models and model components.

- `models`:
  - `PortillaSimoncelli` is a parametric texture model based on the output of the steerable pyramid, as well as some pixel marginal statistics, which, when used with `Metamer`, enables the synthesis of novel texture samples from within a given family [178].
  - `FrontEnd` contains a family of model architectures that provide simple models of the early visual system, such as `LinearNonlinear`, `OnOff`, and `LuminanceGainControl` [17].
  - `naive` contains simple models for comparison against more complex visual models, such as `Linear`, `Gaussian`, and `CenterSurround`.
- `canonical_computations` provides several components for use in model construction:
  - `SteerablePyramidFreq`: the frequency domain implementation of the steerable pyramid, with variable height and number of orientations, with an option to get either real- and complex-valued coefficients [198].
  - `LaplacianPyramid`, one- and two-dimensional Gaussian filters, a form of local gain control, and functions for transforming between the polar and rectangular representations of complex numbers.

3. `metric`: several perceptual distance metrics developed in the lab over the years, including `ssim`, `ms_ssim`, and `nlpd` (the normalized Laplacian pyramid distance) [133, 223, 224].

4. `tools`: variety of helper functions, including the blurring and downsampling trick used in the steerable pyramid and functions for displaying and animating images of the type synthesized by this package with anti-aliasing safeguards (e.g., by default, display images at their actual resolution, and allow for user to specify "zoom" which up- and down-samples neatly, avoiding e.g., the issue of 1.5 display pixels for each data pixel, etc).

This project was team-built by grad students and postdocs in the Lab for Computational Vision. The following is a rough breakdown of what everyone did, including code and the related documentation, tutorials, and tests:

- Kathryn Bonnen: synthesis class design, Portilla-Simoncelli texture model.
- Billy Broderick: metamer and MAD competition, display and animate functions, tooling for automated documentation builds and CPU tests.
- Lyndon Duong: eigendistortions, tooling for automated GPU tests, FrontEnd models.
- Pierre-Étienne Fiquet: geodesics, metrics.
- Nikhil Parthasarathy: steerable pyramid.
- Teddy Yerxa: geodesics, steerable pyramid tutorials.
- Xinyuan Zhao: metrics.

Additionally, everyone has reviewed each others' pull requests, providing feedback on code, documentation, and tutorials. The first five members have been involved from the beginning of the project and have thus contributed the most to the planning and high-level structure and goals of the package.

4.6 Metamer

`Metamer` is conceptually the simplest of the synthesis methods, and the oldest. In perception, the concept dates back to the color-matching experiments of the 19th century [91] that first provided support for the existence of three cone types (though it would be another hundred years before anatomical evidence was found). Perceptual metamers refer to two images that are physically different but are perceived as identical. In `plenoptic`, "metamers" refer to model

metamers: images that are physically different but have identical representations for a given model. In the Lab for Computational Vision, the first work with model metamers was Portilla and Simoncelli [178], where the authors proposed a set of texture statistics and synthesized texture metamers to show the extent to which the model succeeded and failed to capture the "texturiness" of different images. That paper, however, did not use the word "metamers"; Freeman and Simoncelli [70], where the authors develop a putative model of mid-ventral processing that averages those texture statistics in log-polar windows, is the first paper from the lab to do so.

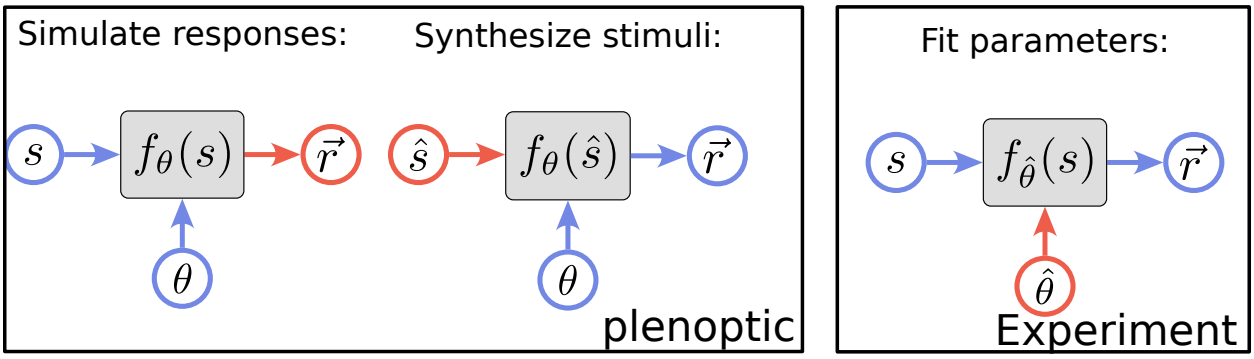


Figure 4.5: `plenoptic` generates metamers, which can then be used in an experiment to find the parameter value(s) for which model metamers are perceptual metamers. A given model, f_θ accepts an input image s and returns a response $\vec{r}$. `Metamer` provides tools for the user to synthesize a novel image, $\hat{s}$, which has the same model response $\vec{r}$. These images (along with the target images s) can be used in an experiment to fit the parameter values $\hat{\theta}$. See section 4.6.1 for more details on that.

Model metamers, as described earlier, are physically distinct images that have the same model representation. As figure 4.5 shows, `Metamer` is one way to synthesize stimuli: given an image s and a model with fixed parameters f_θ, it uses iterative optimization to minimize the difference between the model's representation of s and a new image $\hat{s}$. We include an optional quadratic penalty of values outside some range, since the user probably wants to display the resulting metamer on a screen. The size of that penalty, along with the loss function, optimizer (and learning rate), and initialization for the metamer, can all be set by the user, with reasonable defaults provided by the package. See 4.6.3 for more details.

4.6.1 How to use in experiments

As alluded to above, model metamers are not necessarily perceptual metamers. `plenoptic` provides tools for generating model metamers, which a vision scientist would most likely wish to use in experiments to determine whether and for which parameter values they are also metamers of the visual system. These can be neural metamers, as in Freeman et al. [71], or perceptual metamers, as in Freeman and Simoncelli [70] and the third chapter of this dissertation. I will walk through the logic of that chapter, as it provides an extended example of one way in which `Metamer`, and model metamers more broadly, may be used in vision science experiments.

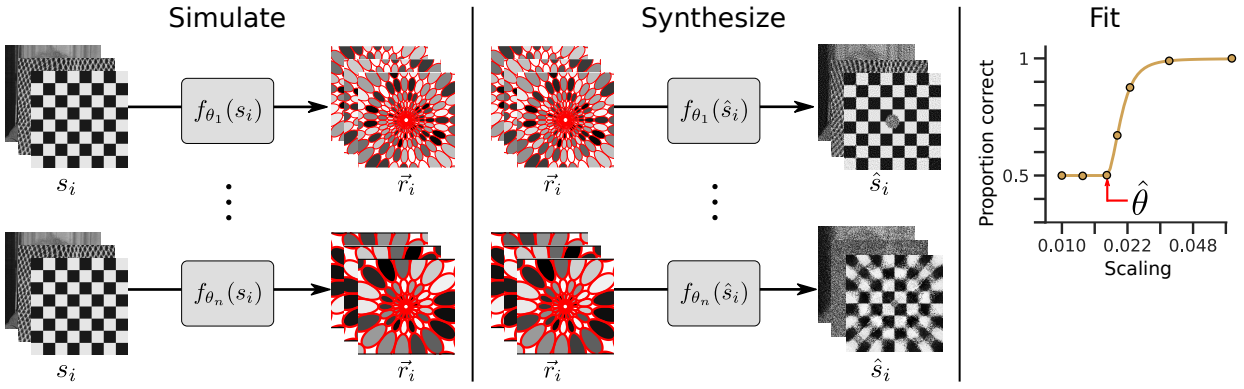


Figure 4.6: Metamers can be used to fit parameters of perceptual models. This figure presents a schematized overview of the project discussed at length in chapter three, for a single model. We developed a pooling model of the early visual system, which averages image statistics in log-Gaussian windows whose diameter grows linearly with eccentricity, then chose a set of images of natural scenes. We simulated the model's responses to these images across an appropriate range of parameter values and synthesized new images with matched responses. We used these images to perform a psychophysical experiment to find the maximum parameter value where model metamers are perceptual metamers.

In my third chapter, I developed foveated models of early visual system processing, synthesized metamers for these models for a range of parameter values, and ran a psychophysics experiment to determine the largest parameter value for which model metamers were also perceptual metamers. The models average image statistics in log-Gaussian windows whose diameter grows linearly with eccentricity: the rate at which these windows grows, the scaling, is the model's only parameter, and the identity of the pooled statistic differentiates the models from each other. We built two models, one which pooled luminance and one which pooled oriented spectral energy, as these are

statistics thought to be computed in the early visual system, after the photoreceptors and before secondary visual cortex. We selected a range of scaling values where we believed performance would go from chance to ceiling, based on visual inspection of the images (the length of time required to synthesize a single metamer, up to 14 days for the smallest scaling value, meant we could not use extensive piloting to determine an appropriate range).

Figure 4.6 shows a schematic of how the experiment was run for a single model. With the model f_θ built and the scaling values $\theta_1, \ldots, \theta_n$ chosen, we selected a set of images of natural scenes s_i. We then ran our model, with its range of scaling values, on each image to generate the responses $\hat{r}_i$. These responses were matched using optimization in order to generate a new set of images $\hat{s}_i$ (in this description and the schematic, simulate and synthesize are shown separately, but `Metamer` handles them together). This step requires a large amount of time and compute resources, though each image is generated independently, allowing for large-scale parallelization (and the use of GPUs, when possible, also speeds up the process considerably). The duration of synthesis scales approximately linearly with the duration of a single forward pass of the model; as multiple forward passes are called on each iteration of synthesis, reducing its duration is the most effective way to speed up synthesis.

With a large set of suitable metamers on hand, the psychophysical experiment can be performed in order to fit model parameters. In our experiment, $\hat{\theta}$ was the largest scaling value where model metamers were perceptual metamers, that is, the largest scaling value where performance was at chance. We showed pairs of images to observers in a two-alternative forced choiced setup, testing participants' ability to discriminate between them. These images were either pairs of synthesized model metamers or a model metamer and its target image; in either case, both images were metameric according to the model with a particular parameter value. After having participants discriminate between many such pairs, we were able to plot proportion correct as a function of scaling and fit psychophysical curves in order to find $\hat{\theta}$. With it, we know the parameter value for which our model's invariances align with those of the human visual system (see chapter three

for greater discussion of the nuances here). Such a result allows for us to make inferences about the extent of spatial pooling of the corresponding image statistic, as well as providing a first step towards the development of a large field of view foveated observer model, which takes arbitrary pairs of images and predicts how discriminable they are.

Note that this particular goal of finding the *largest* parameter value is not the only possible goal when using metamers in an experiment, but follows from our use of these models. As scaling controls the size of the windows in which image statistics are averaged, models with smaller scaling values have smaller null spaces, reducing the set of possible model metamers until the trivial case: a model whose pooling windows are no larger than a pixel anywhere in the image will have no model metamers, as each image's representation will be unique. Such a model tells us nothing about the human visual system. For other models, the fitting experiment may try to find other parameter values or ranges of parameter values, depending on the parameters' interpretation and the researchers' intended inference. Furthermore, the fitting step can be carried out before synthesis. For example, a researcher could fit several candidate models to a large scale image database such as ImageNet and synthesize metamers for each model. They could then run a psychophysics experiment similar to the one described above to compare each model against human perception, determining for which model, rather than for which parameter values, model metamers are perceptual metamers.

4.6.2 Examples

One of the major advantages we get from building `plenoptic` on top of `pytorch` is that synthesis works with arbitrary models with no modifications necessary to the model or `Metamer` (however, work is still required to figure out what hyperparameters are necessary to successfully synthesize something, and those choices will almost certainly vary from model to model). Figure 4.7 shows example model metamers for three different target images from four different models. By examining each of them, we can gain a better understanding of each model's sensitivities and

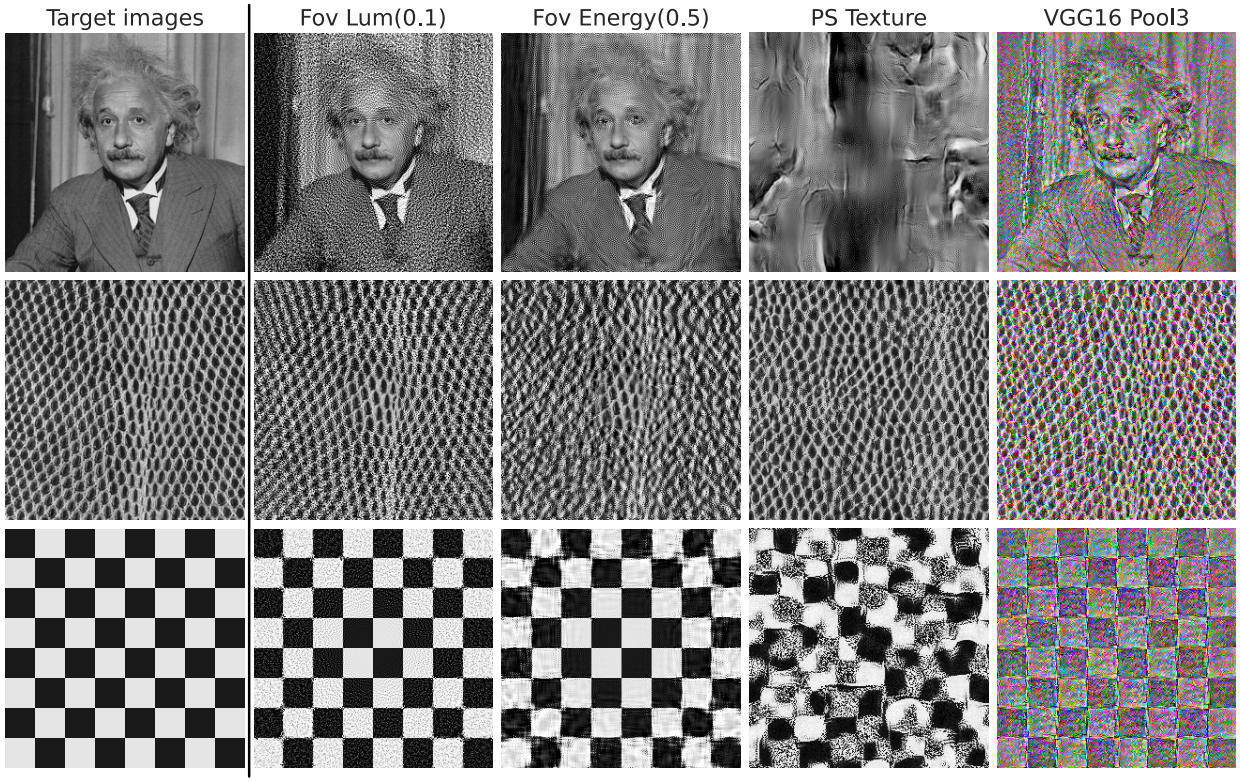


Figure 4.7: Example model metamers generated with `Metamer`. Each row shows metamers generated for a given target image for four different models (from left to right): the foveated luminance and energy models from the third chapter of this dissertation (at a scaling value above the critical scaling, so these are not perceptual metamers), Portilla-Simoncelli texture statistics [178], and the third max pooling layer of VGG16 (configuration D, [199]). Examining these images allows us to gain a better understanding of each model's sensitivities and invariances.

invariances.

The leftmost column shows the three target images, one per row, whose representation each metamer is trying to match. The next two columns show metamers for the foveated models from the third chapter of this dissertation: the foveated luminance model with scaling 0.1 and the foveated spectral energy model with scaling 0.5 (these scaling values are well above the critical scaling value, and is large enough that these model metamers are definitively *not* perceptual metamers; fixation is at the center of the image and the edge corresponds to 3 degrees eccentricity). These model metamers look like what we saw in chapter three: the luminance model metamers has gradually increasing white noise (leftover from the initial image) as you move away from the fovea, while the energy model has a more complex pattern resulting from matching energy but

not phase in larger and larger windows.

The fourth column shows metamers for the Portilla-Simoncelli texture statistics [178]. We reimplemented this model in `pytorch` and included it in our package, with validation tests to ensure that we produce the same outputs on a given image as the original matlab implementation. As these statistics were developed to capture the natural textures, the metamer of reptile skin in the second row looks very convincing, like it may be a perceptual metamer, while the other two appear bizarre. The failure on the image of Einstein is expected, as faces and bodies contain all sorts of structure not captured by the texture statistics (see figure 17 in [178]), and the regular repeating structure of the checkerboard is also difficult to capture (see figure 16 in [178]; though the speckled patterns are most likely because of boundary artifacts).

The final column shows metamers for the third max pooling layer (about halfway through the network) of VGG16, a deep neural network trained on ImageNet (configuration D, [199]). These metamers look like the original image plus RGB noise, which immediately highlights that it's the only model that operates on RGB images, that its sensitivities to color are very different than human sensitivities, and that the model is relatively insensitive to high-frequency noise. Unlike the foveated luminance model, its sensitivity to such noise does not change with spatial position. I also created metamers for all five of VGG16's max pooling layers (not shown): for the earlier pooling layers, metamers appeared to be the original image plus noise, with the intensity of the noise increasing with layer depth until the later layers appear to be simply RGB noise (Feather et al. [66] figure 3 shows similar results when synthesizing model metamers for VGG-19 and several other ImageNet-trained deep networks).

I would like to emphasize that these are all very different models: the first two were developed by me for my third chapter, the third is a reimplementation from the literature and is included in `plenoptic`, and the last is a deep neural network that comes from `pytorch`'s model zoo. Because they all follow the basic model requirements for synthesis (implemented in `pytorch`, accepts four-dimensional tensor inputs, and returns three- or four-dimensional tensor outputs), they

were all compatible with `Metamer`. These models also have vastly different number of statistics, approximately 16,500 for the first two models, 1000 for the texture statistics, and 8.3 million for VGG16 (compared to 65,536 pixels in the input image). Additionally, no manual calculation of gradients was required: when Portilla and Simoncelli developed the model in Portilla and Simoncelli [178], they had to recalculate the gradients by hand every time they changed the model, which made the model's development exceedingly difficult. The automatic gradient calculation baked into `plenoptic` means that researchers are able to focus on the scientific questions that matter to them.

4.6.3 Usage details

A schematic of `Metamer` is shown in figure 4.8. `Metamer` is used to generate model metamers, images that are physically distinct but have the same model representation. We do this by minimizing the objective function found at the top of the figure: $\arg\min_{\hat{s}} \mathcal{L}(f_\theta(s), f_\theta(\hat{s})) + \lambda \mathcal{B}_r(\hat{s})$, where:

- s is the user-supplied image. It must be a four-dimensional `torch.tensor` object, where the dimensions are batch, channel, height, and width. This is the standard way of representing images in `pytorch` (and deep learning libraries more generally). Most commonly, different images are indexed along the batch dimension, while channel contains either the separate RGB channels or the outputs of different convolutional filters. Singleton batch and channel dimensions are allowed (for a single grayscale image) and are the most common use-case so far.

- f_θ is the user-supplied model. It must accept a four-dimensional `torch.tensor` as input and return either a three- or four-dimensional `torch.tensor` as output, corresponding to either a vector- or image-like output (e.g., texture statistics or output of a convolution). In addition to these requirements, the model must also be written in and differentiable by `pytorch`,

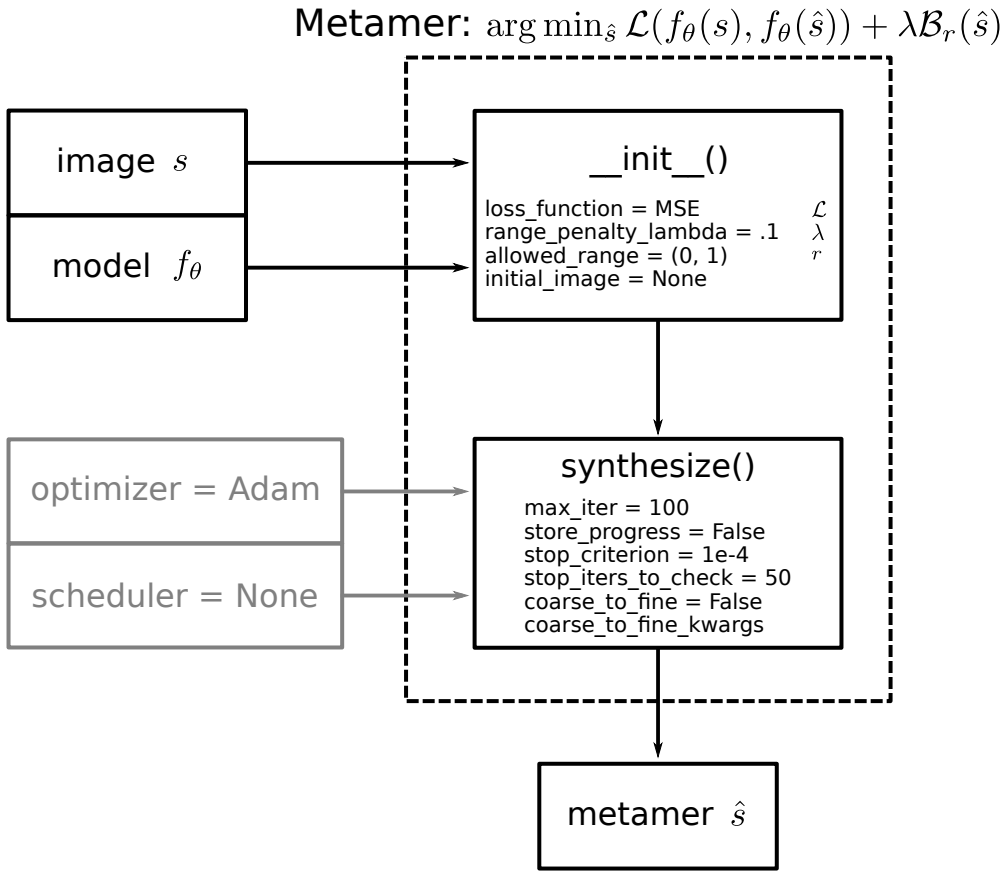


Figure 4.8: Schematic showing how `Metamer` is used. All instances of `Metamer` minimize the objective function found at the top of the figure, with the goal of returning a synthesized image $\hat{s}$, known as a model metamer, that minimizes it. All components of that objective function except for the bounds penalty function, $\mathcal{B}$ (which is a quadratic penalty on pixel values outside the allowed range), are settable by the user, with reasonable defaults provided for the loss function, range penalty lambda, and allowed range. See text for more details.

which requires using `pytorch` functions rather than those of other python libraries, e.g., `torch.fft.fft2` rather than `numpy.fft.fft2`. Finally, the model must accept real-valued inputs and return real-valued outputs, and arbitrary inputs and outputs should have meaning. This last is not a strict technical requirement, but as synthesis returns something that is out-of-distribution, this will lead to the output being more interpretable.

- $\mathcal{L}$ is a loss function that takes two three- or four-dimensional tensors and returns a scalar defining how different they are. A default, the mean-squared error, is supplied, and this will generally work in most cases, though for models that have components of differing scales,

another loss function that weights them in some way may be more helpful.

- λ and r are the weight and allowed range for a quadratic penalty on pixel values in the synthesized metamer image, $\mathcal{B}$. As users generally want to display their synthesized images and may in some cases wish to run experiments that require physically presenting them, some way of constraining their pixel values is desirable. A previous version of `Metamer` "clamped" the image, setting all pixel values outside the range to the minimum or maximum on each iteration, but in practice, a quadratic penalty with varying weight is more effective. Users can set λ and r, but the use of a quadratic penalty for $\mathcal{B}$ is fixed; the user can set $\lambda = 0$ to remove this penalty.

- $\hat{s}$ is the synthesized metamer image, which is produced via iterative optimization.

All of the above are specified by the user upon initialization of the `Metamer` object; python is an object-oriented language, and `Metamer` must thus be instantiated by the user before using it to create model metamers. The user may also supply an `initial_image` from which synthesis will start. If none is supplied, a patch of white noise matching the dimension of the input image will be generated. After initializing `Metamer`, the user may optionally create standard `pytorch` `Optimizer` and `lr_scheduler` objects, which specify how to update the synthesized image given the gradient and how to change the optimizer's learning rate over time, respectively, allowing the user to make use of the existing library of `pytorch` objects or create their own. The user may then pass these to the `synthesize()` method and, if none are provided, `Metamer` defaults to using `Adam` with a learning rate of .01 and `amsgrad=True` with no learning rate scheduler, a relatively conservative choice. Calling `synthesize()`, as the name suggests, starts the synthesis process, which displays a progress bar summarizing how long the synthesis will take and its current state, including the current loss and gradient norm. When calling `synthesize()`, the user may also specify the maximum number of iterations to run synthesis, whether (and how frequently) to store the metamer-in-progress during synthesis for later examination, a stopping criterion, and whether

to use coarse-to-fine optimization (and its criterion for moving onto the next scale). Synthesis can be resumed by calling `synthesize()` again with the same set of options.

Coarse-to-fine optimization is a technique useful for multiscale models and is analogous to optimizing a blurred version of the objective function, then progressively adding finer details in. It requires the `model` to meet an extra specification: it must have an attribute, `scales`, which is a list of the possible scales of the model, and it method must accept a `scales` argument in addition to the image. This argument must be a list containing a subset of the values in the `scales` attribute, and the model should return the subset of its representation corresponding to those values. For example, in my foveated metamers project, `model.scales=[0, 1, 2, 3, 4, 5]`, corresponding to the six scales of the steerable pyramid, and `model(image, scales=[0, 1])` would return only the pooled energy computed from the coarsest two scales. If the model meets these requirements, then `Metamer` can do coarse-to-fine optimization, which tends to reduce the ultimate loss of the synthesized metamer.

Because `plenoptic` is built on top of `pytorch`, we get GPU compatibility basically for free in all synthesis methods. In order to run metamer synthesis on the GPU, your image and model should be sent to the GPU before initializing the `Metamer` object. Then, everything will be run on the GPU. Sending the image to the GPU is as simple as running `image = image.to('cuda')`; since models can be custom-built, the user must generally implement their own `to()` method to do this, which generally requires calling the `to()` methods of any tensor attributes of the model (if using any of the models included in `plenotic` or `pytorch`, `to()` will already be implemented). (The user may also call `metamer.to()` after instantiating the `Metamer` object, but it's recommended to pass everything to the GPU beforehand.)

`Metamer` also provides four other public methods: `to()`, `objective_function`, `save()`, and `load()`. The `to()` method works as described above, calling the `to()` method of all tensor attributes and the model (this also allows the user to change the datatype of synthesis, for example to `torch.float16` or `torch.float64`, half and double precision, respectively). `objective_function()`

accepts a model representation and compares it against the stored `target_representation`, returning the value of the function described above. `save()` and `load()` do as the names suggest, allowing the user to move the `Metamer` object between machines.

Finally, the `plenoptic` package includes several helper functions for visualizing the status of metamer synthesis. These helper functions allow the user to easily display the synthesized metamer, the current loss, and the error in the model's representation (optionally, models may have custom methods to plot their representation, making this more informative). An `animate()` function is also provided to animate all of these plots over time, allowing the user to inspect synthesis progress (this is only available if the user called `synthesize()` with `store_progress!=False`).

In the above description, I referred to metamers throughout as "images", but model metamers can be generated in other signal domains as well, such as video or audio. As long as the specifications of `image` and `model` are followed (most importantly, that the input is four-dimensional and the output is three- or four-dimensional), the process is the same. Such extensions have not been tested so far, however, and the corresponding optimization problem will almost certainly be more difficult.

4.7 MAD Competition

Whereas metamer synthesis allows researchers to evaluate how a single model's invariances align with those of the human visual system, MAD Competition is a method for efficiently testing two competing visual metrics, by generating a set of images with maximally different predictions. The use-case outlined in the original paper, Wang and Simoncelli [225], was to compare two potential models of human perceptual distance, mean-squared error (MSE) and the structural similarity index (SSIM). Both of these functions make predictions about how different two images are, and the authors describe MAD Competition as a means of understanding how the two differ and for generating images for a psychophysical experiment.

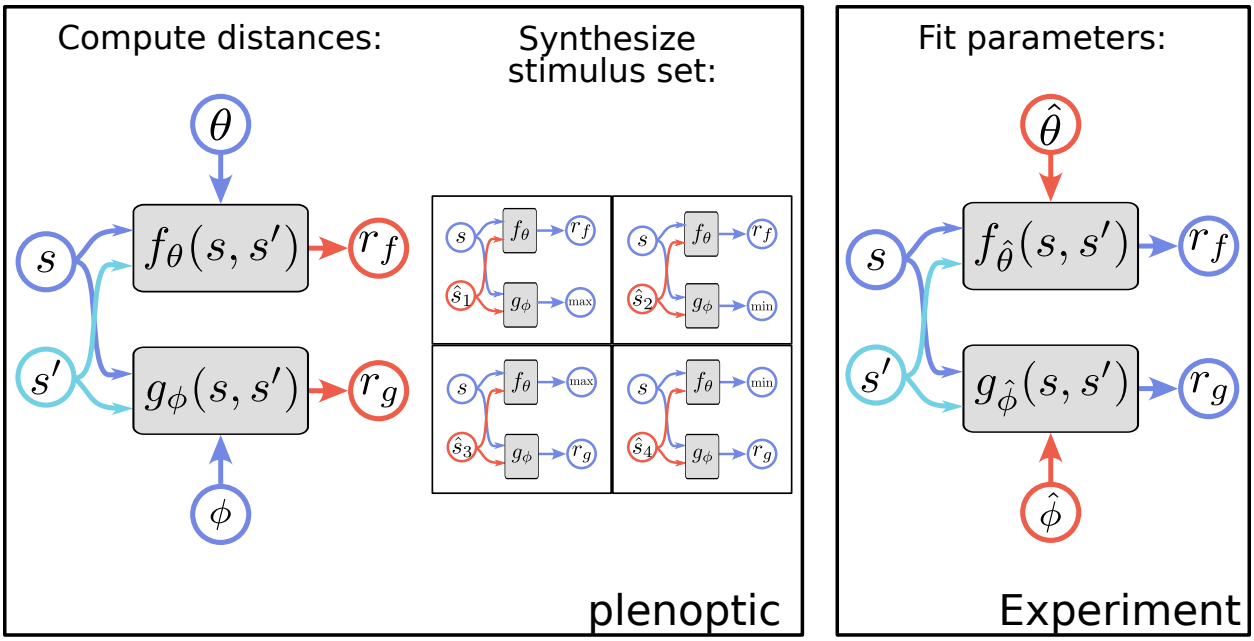


Figure 4.9: `MADCompetition` generates a set of images for efficiently testing two visual metrics which each accept two inputs and return a scalar output giving the distance between them. For a given reference image and noise level, `MADCompetition` generates two pairs of images, holding the distance constant for one metric while minimizing or maximizing the other. Parameter fitting is done in a separate experiment, and can use these images or some other procedure.

Since `MADCompetition` works with visual metrics, rather than models, it has slightly different requirements than the other synthesis methods included in `plenoptic`. Note that we mean "metric" loosely: as far as `MADCompetition` are concerned, metrics must accept two inputs, return a scalar giving the distance between them, and satisfy the identity of indiscernibles, i.e. $f_\theta(x, y) = 0 \Leftrightarrow x = y$ (note that this means we must use $1 - SSIM(x, y)$ instead of $SSIM(x, y)$, as SSIM is a similarity index and so returns a value of 1 on identical images, rather than 0). We do not require that they satisfy the other two properties of mathematical metrics, symmetry or the triangle equality. While this is a different requirement than the other methods, it is straightforward to build a naive metric that satisfies these requirements from any model: take the mean-squared error between the model outputs of two images.

`MADCompetition` takes two such metrics, f_θ and g_ϕ, and a reference image, s, and adds noise to s to get an initial image s'. It then generates a new image whose distance to the reference image

is the same as that of the initial image according to f_θ, while its distance according to g_ϕ is as large as possible, i.e., we synthesize a new image $\arg\max_{\hat{s}_1} g_\phi(s, \hat{s}_1)$ subject to the constraint that $f_\theta(s, \hat{s}_1) = f_\theta(s, s')$. A full set of MAD Competition images consists of four such images: two which hold f_θ constant while maximizing and minimizing g_ϕ and two which hold g_ϕ constant while maximizing and minimizing f_θ. We use iterative optimization to do this, in a slightly different procedure than in Wang and Simoncelli [225]: whereas the original paper projected out the gradient of one model from the other's on each step, we add a term (with user-adjustable weight) that penalizes any divergence from the constant loss, which seems to work better in practice (see section 4.7.4 for more details). Similar to `Metamer`, `MADCompetition` also includes an quadratic range penalty, whose weight, along with the optimizer and learning rate, can be adjusted.

In Wang and Simoncelli [225] and in the description above, s' is created by adding normally-distributed noise to the reference image in order to achieve a target mean-squared error level. This is the default behavior in `plenoptic`, but the user can also specify a separate image (of the same size as the original) to use for s' instead. The usefulness of this will be demonstrated in an example in the next section.

4.7.1 How to use in experiment

In the original paper, the authors compare the suitability of MSE and SSIM as perceptual metrics by generating sets of MAD Competition images with increasing distortion levels and showing them in pairs to participants in an experiment where subjects were asked to choose the image from each pair that had higher peceptual quality (participants were allowed to free view and no time limit was imposed on the decision). The authors found that at low distortion levels, images with fixed MSE and fixed SSIM were both matched in quality, but as the distortion level increased, images with fixed MSE were considered of poorer quality while those with fixed SSIM remained matched in quality up until the highest distortion levels (and were always better than images with fixed MSE). The authors thus conclude that SSIM provides a better model of

perceptual quality than MSE.

While the original paper shows an example of their use with potential image quality metrics, `MADCompetition` can be used for other purposes as well. In the appendix of the third chapter of this dissertation, there is a figure showing that luminance model metamers are substantially physically different from their target images. For luminance model metamers with the smallest scaling values, the differences between the target image and the model metamers become less visible (as expected), and we wanted a way to demonstrate that these metamers were still physically distinct from the target, i.e., that the mean-squared error had not gotten so small that any image with that MSE would be perceptually identical with the target image (this was especially important because the model metamers need to be shrunk from their full size in order to be displayed in a paper or presentation, which makes the high frequency noise patterns they contain less visible). In order to generate more images with the same MSE, I first linearly interpolated between the target image and another natural image (or white noise) until the MSE was matched to that of the luminance model metamer with the lowest MSE between it and its target image. I then used these images as the initial image s' in `MADCompetition`, with the metamer target image serving as the reference image s, holding the MSE constant while minimizing or maximizing $1 - SSIM$ to generate several images with the same MSE as our model metamer. While all images had the same MSE, only one was a luminance model metamer with the target image. The fact that this image was also the only one that was a perceptual metamer with the same image demonstrated that our luminance model metamers were still physically distinct from the target image.

Finally, while the procedure in Wang and Simoncelli [225] compares potential models of image quality against human perception, `MADCompetition` can be used to compare models of other systems as well (see controversial stimuli, [78] for a related concept). For example, it could be used to generate stimuli for comparing models of neural activity. As discussed earlier, Lescroart, Stansbury, and Gallant [138] demonstrate that three models of scene-selective visual areas all predict BOLD responses to a set of images of natural scenes fairly well. As the set of all images

is so vast, finding a set of stimuli that can effectively discriminate between competing models is difficult, especially when those models are of higher-level visual areas whose response properties are not well-understood. `MADCompetition` provides one way of doing so: sets of stimuli that discriminate pairs of models can be generated and shown to participants while recording BOLD activity. As these stimuli have been synthesized to separate model predictions to as large an extent as possible, it is less likely that all models will perform comparably well on predicting the corresponding brain activity.

4.7.2 Simple walkthrough

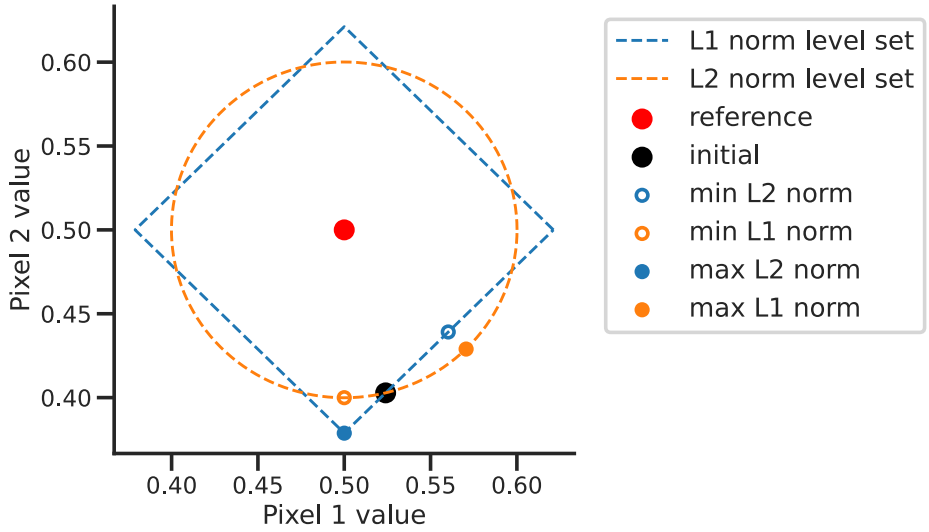


Figure 4.10: Simple example of MAD Competition between L1 and L2 norms on a two-dimensional image. See text for details.

To better understand what MAD Competition does, we will walk through a simple example: we will differentiate the L1 and L2 norms on a two-pixel image $[.5, .5]$. The L1 norm distance between two 2d points (x_0, y_0) and (x_1, y_1) is defined as $|x_0 - x_1| + |y_0 - y_1|$, while the L2 norm distance (or Euclidean distance) is $\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$. Since our input is two dimensional, we can plot this example as a scatter plot, as shown in figure 4.10, with the first pixel value on the x axis and the second on the y. Our reference image is shown as the red point at the center.

To begin, we add a small amount of noise to get our initial image, plotted in black. The distance between this and the reference point is what we will be holding constant for one metric while we minimize or maximize the other. Since we know what the level sets of L1 and L2 norms look like (a diamond and a circle, respectively), we can plot those as well. All the points generated by `MADCompetition` must lie along one of these two level sets, and our initial point lies at their intersection.

To generate a MAD image, we hold one metric constant while changing the other as much as possible. Let's first examine the "max L2 norm" point, shown in solid blue. For this, we have maximized the L2 norm while holding the L1 norm constant. We can see the synthesized output satisfies these constraints by the fact that this point is as far away as possible (in Euclidean distance) from the reference point while still remaining on the L1 level set. This point must thus lie along the axis, so that one pixel has the same value as our reference image, while the other is as different as possible. The other corners of the diamond would also satisfy this; we ended up here because it's closer to our initial image and our optimization found this global optima first (if we had started in a different portion of input space or used a larger learning rate, we may have ended up in one of the other corners).

We can similarly maximize the L1 norm while holding L2 constant, as shown by the solid orange point. Analogous to the point we just discussed, this point lies along the L2 level set and is as far away from the reference as possible. This puts it on the diagonal, so that neither pixel has the same value as in the reference, but they individually have the same difference (the other diagonals would also have worked, but this is the one we were closest to). Minimizing the two norms, shown by the empty points, puts the point along the other norm's level set, as close to the reference as possible. Minimizing the L1 norm puts you along the axes, like maximizing the L2 norm, while minimizing L2 norm puts you along the diagonal.

This simple example develops our intuition for what `MADCompetition` achieves: changing the initial image to move along one metric or the other's level set, ending up as close to or as far away

from the reference as possible. This set of images therefore produces a set of predictions from the two metrics that are *as different as possible*, which allows for more efficient model comparison. In general, we cannot easily visualize pixel space and metrics' the level sets, but the same principle applies in more complicated settings, as we will see in the next section.

4.7.3 More complex examples

We can do the same procedure on a 256 by 256 image, as shown in figure 4.11. We are comparing the same two metrics, the L1 and L2 norms, but we can no longer plot the images nor the level sets in pixel space, as we have moved into a much higher-dimensional space. We can, however, follow the same procedure to use `MADCompetition` to synthesize images. When we do so, we start by adding normally-distributed noise to get our initial image s', shown in the center of the figure. If we look at the image generated by maximizing L1 norm, on the bottom, we see that most of the pixels are unchanged, but that there are many pixels that have been changed to the extremal values: black on white squares, white on black squares. This is the high-dimensional analogue what we saw in the two-dimensional example, where maximizing L2 put us on the axis, with one pixel unchanged and the other changed as much as possible. Similarly, minimizing L1 has the same pattern, but there are fewer changed pixels and their differences with the reference image are not as extreme. This again matches with the two-dimensional example, where minimizing L1 resulted in another image along the axis, but that was closer to the reference image.

Similarly, when we maximize L1 or minimize L2, our image lies along the diagonal: no pixels have the same value as the reference image, all are distributed above or below the reference image values by the same amount, so we end up with different grayish values. Again, we end up with a larger difference when maximizing L1 than when minimizing L2.

From our understanding of L1 and L2 norm, the behavior we have been examining in figures 4.10 and 4.11 does not depend on image content (we would get the same level sets anywhere in pixel space), and we can see that on the image of Einstein shown in figure 4.12. We can see that

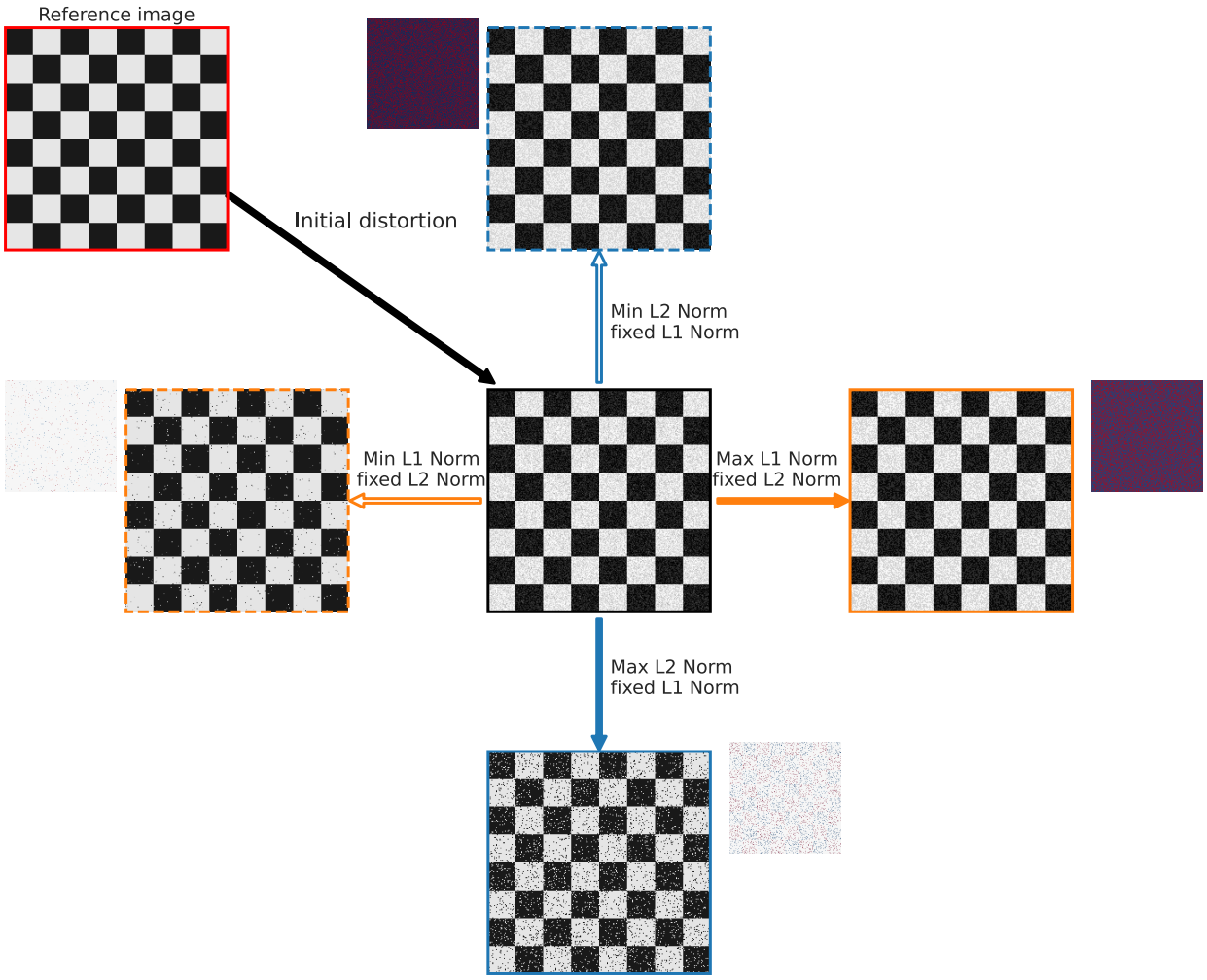


Figure 4.11: MAD Competition between L1 and L2 norms on a checkerboard image reveals similar patterns to the two-dimensional example in figure 4.10. Since images are higher dimensional than before, cannot plot them in pixel space but must visualize them as images. Layout is similar to that in 4.10, with orange borders showing images with fixed L2 norm, blue borders showing images with fixed L1 norm, solid borders showing images that maximized the other metric, and dashed borders showing images that minimized it. The inset next to each synthesized image shows the difference between it and the reference image. See text for explanation.

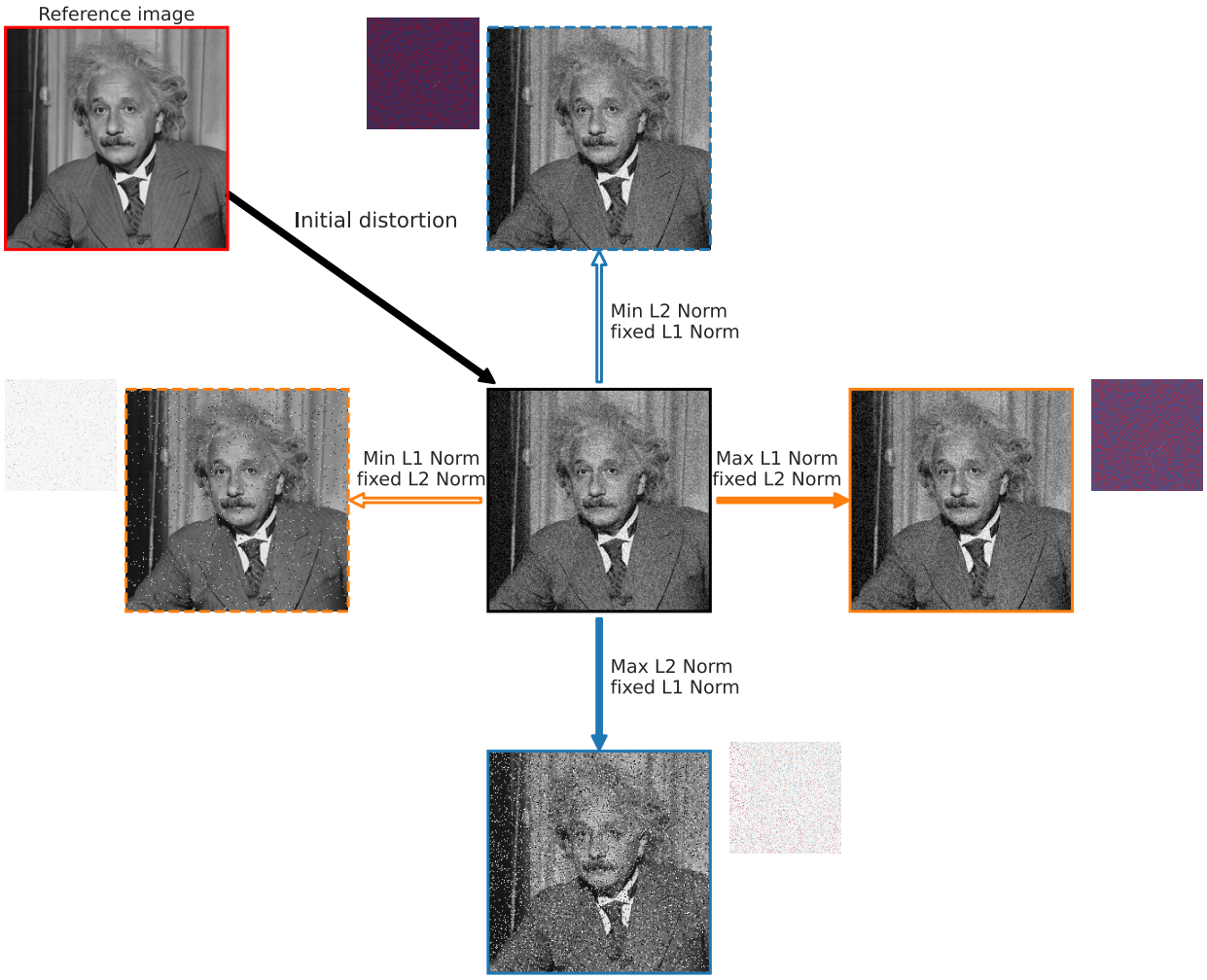


Figure 4.12: For MAD Competition between L1 and L2 norm, image content does not matter: the patterns are the same when using an image of Einstein as a target image as when using a checkerboard. Layout the same as in figure 4.11.

the same patterns of changes to pixel values are present here as in the checkerboard of figure 4.11. This would not be the case for other metrics that were more image content-dependent, such as SSIM or something texture-aware.

Our simple two-dimensional L2 vs. L1 example has given us the intuition to understand these two, slightly more complicated examples. This also provides some evidence that L2 is a better perceptual metric than L1: L2 predicts that the blue-outlined images in figures 4.11 and 4.12 are as different as possible, while L1 predicts that the orange-outlined ones are. L1 also predicts

that the midgray noise seen in the right and top images should result in a worse image than the salt-and-pepper noise present in the bottom and left ones, while L2 predicts the opposite. From examining these images, I agree with the predictions made by L2 over those made by L1, though a psychophysical experiment would be required to say so for sure.

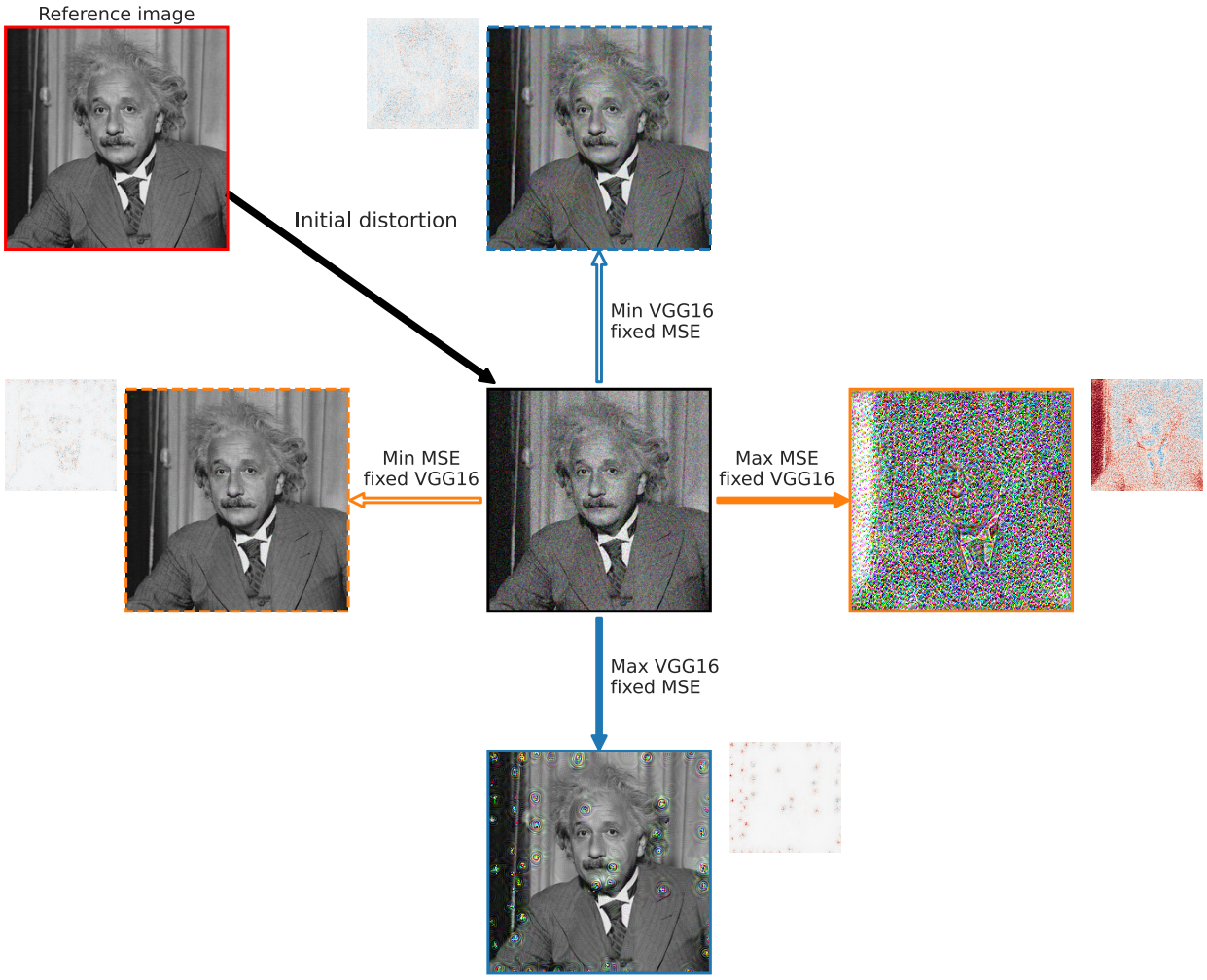


Figure 4.13: MAD Competition between MSE and the third max pooling layer of VGG16 shows that this layer is sensitive to whorl-like patterns and insensitive to high frequencies. Layout is the same as in figure 4.11. See text for more details.

We can, of course, run MAD Competition between other models as well. Figure 4.13 shows the results of comparing MSE and the third max pooling layer of VGG16 (to make this compatible with `MADCompetition`, we take the MSE between the outputs of that layer for the two images). Note

that, unlike the L1 and L2 norms, VGG16 only operates on RGB images; MSE in this example is the average across all three channels. When we maximize VGG16, we end up adding multicolored whorls across the image and otherwise setting all pixels to their value in the reference image. The whorls seem especially common near the edge of the image, which implies there might be some boundary-handling issue: all convolutional steps of VGG16 use zero-padding, so the image edges are especially high contrast. When we maximize MSE while holding VGG16 constant, we get a large amount of high frequency noise across all three RGB channels, but the eyes are pretty much left alone (the noise level also looks slightly lower around the tie). The two images that minimized VGG16 and MSE both appear to have just reduced the noise found in the initial image. When minimizing MSE and holding VGG16 constant, the noise looks concentrated along the edges and the brighter patches of the image, avoiding the dark patches, while it appears more uniform when minimizing VGG16.

This example does not allow us to say that one or the other is a better visual metric without a psychophysics experiment, as both pairs of images appear very distinct, with the maximizing metric image appearing of worse quality than the minimizing metric one. However, it does allow us to get a better sense of VGG16's third max pooling layer's sensitivities and invariances, especially when we examine them together with the model metamers in figure 4.7. This model appears to be especially sensitive to whorl- or eye-like patterns, relatively insensitive to high frequency noise, does not distinguish between the three color channels, does not distinguish between grayscale and not grayscale, and appears convolutional, not concentrating on or avoiding any particular part of the image. If we truly wished to understand VGG16 as a potential model of human vision, we should extend these examples to more images and investigate the other layers to get a sense for how these properties change with processing depth.

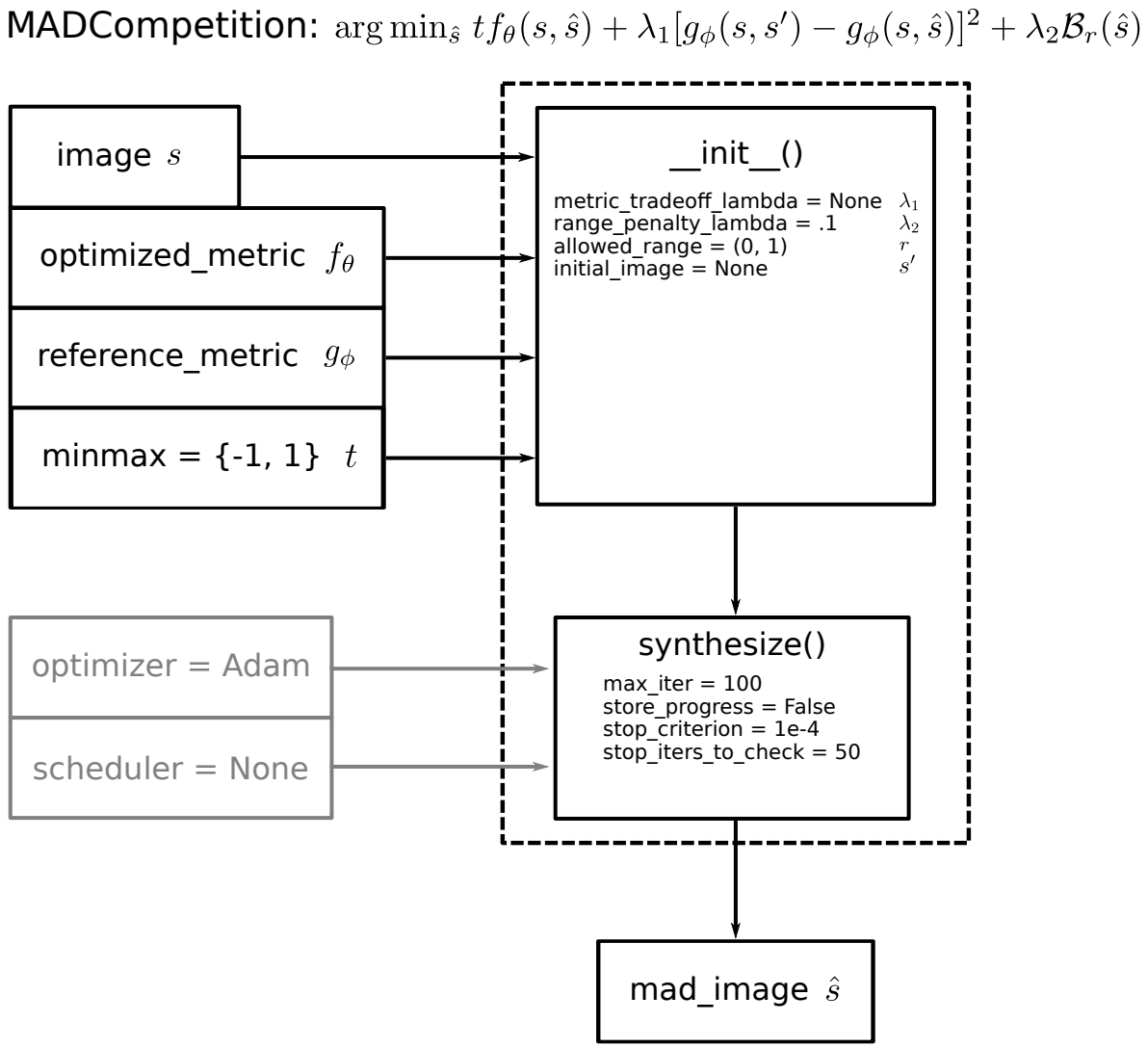


Figure 4.14: Schematic showing how `MADCompetition` is used. All instances of `MADCompetition` minimize the objective function found at the top of the figure, with the goal of returning a synthesized image $\hat{s}$, known as a MAD image, that minimizes it. Note that a single `MADCompetition` instance creates a single MAD image, while a complete set of MAD Competition contains four images: two for each possible value of `minmax` t, switching which metric is f_θ and which is g_ϕ. All components of that objective function except for the bounds penalty function, $\mathcal{B}$ (which is a quadratic penalty on pixel values outside the allowed range), are settable by the user, with reasonable defaults provided for range penalty lambda, initial noise, and allowed range. The metric tradeoff lambda, λ_1 is initialized to the ratio $f_\theta(s, s')/g_\phi(s, s')$, rounded to the nearest power of 10, if left as its default value of None, but, in practice, users will need to spend some time finding an appropriate value. See text for more details.

4.7.4 Usage details

A schematic of `MADCompetition` is shown in figure 4.14. `MADCompetition` is used to generate images which produce the maximally different predictions from two metrics. We do this by minimizing the objective function found at the top of the figure: $\arg\min_{\hat{s}} t f_\theta(s, \hat{s}) + \lambda_1[g_\phi(s, s') - g_\phi(s, \hat{s})]^2 + \lambda_2 \mathcal{B}_r(\hat{s})$, where:

- s is the user-supplied image. It must be a four-dimensional `torch.tensor` object, where the dimensions are batch, channel, height, and width. This is the standard way of representing images in `pytorch` (and deep learning libraries more generally). Most commonly, different images are indexed along the batch dimension, while channel contains either the separate RGB channels or the outputs of different convolutional filters. Singleton batch and channel dimensions are allowed (for a single grayscale image) and are the most common use-case so far.

- f_θ is the user-supplied metric whose representation we will be changing as much as possible. It must accept two four-dimensional `torch.tensor` as input and return a scalar value as output, corresponding to the distance between the two inputs, and must satisfy $f_\theta(s, s) = 0$. In addition to these requirements, the model must also be written in and differentiable by `pytorch`, which requires using `pytorch` functions rather than those of other python libraries, e.g., `torch.fft.fft2` rather than `numpy.fft.fft2`. Finally, the model must accept real-valued inputs and return real-valued outputs, and arbitrary inputs and outputs should have meaning. This last is not a strict technical requirement, but as synthesis returns something that is out-of-distribution, this will lead to the output being more interpretable.

- g_ϕ is the user-supplied metric whose representation we will be holding constant during this instance of MAD synthesis. It must meet the same requirements as f_θ.

- t controls whether we are minimizing or maximizing f_θ by multiplying its loss by 1 or -1.

- λ_1 is the metric tradeoff lambda, which controls how heavily to weight the divergence of g_ϕ loss from 0. If the user does not specify a value, it defaults to the ratio $f_\theta(s, s')/g_\phi(s, s')$ rounded to the neareset power of 10, but in practice, users will need to spend some time finding an appropriate value. If the value is too small, the loss will not be held constant (i.e., $g_\phi(s, s') \neq g_\phi(s, \hat{s})$) and thus the image will not be a MAD image. If its value is too large, $f_\theta(s, \hat{s})$ will change slowly (or not at all).

- s' is the initial image, which fixes the error level. The user can set either a float or a `torch.tensor` object. If a float, we add normally-distributed noise to s to produce an s' such that the MSE error between s and s' is the specified value. If a tensor, it must be the same shape as s and is the initial image.

- λ_2 and r are the weight and allowed range for a quadratic penalty on pixel values in the synthesized image, $\mathcal{B}$. As users generally want to display their synthesized images and may in some cases wish to run experiments that require physically presenting them, some way of constraining their pixel values is desirable. Users can set λ and r, but the use of a quadratic penalty for $\mathcal{B}$ is fixed; the user can set $\lambda = 0$ to remove this penalty.

- $\hat{s}$ is the synthesized image, which is produced via iterative optimization.

A single instance of `MADCompetition` will synthesize a single maximally differentiating image. A complete set contains four such images, two for each possible value of t, switching which metric is f_θ and which is g_ϕ. When generating a set of images, the same reference images s and initial images s' should be used for all four images.

Similarly to `Metamer`, all of the above are specified by the user upon initialization of the `MADCompetition` object. After initialization, the user may optionally create standard `pytorch` `Optimizer` and `lr_scheduler` objects, which specify how to update the synthesized image given the gradient and how to change the optimizer's learning rate over time, respectively, allowing the user to make use of the existing library of `pytorch` objects or create their own. The user may

then pass these to the `synthesize()` method and, if none are provided, `MADCompetition` defaults to using `Adam` with a learning rate of .01 and `amsgrad=True` with no learning rate scheduler, a relatively conservative choice. Calling `synthesize()`, as the name suggests, starts the synthesis process, which displays a progress bar summarizing how long the synthesis will take and its current state, including the current loss and gradient norm. When calling `synthesize()`, the user may also specify the maximum number of iterations to run synthesis, whether (and how frequently) to store the image-in-progress during synthesis for later examination, and a stopping criterion. Synthesis can be resumed by calling `synthesize()` again with the same set of options.

`MADCompetition` also provides four other public methods: `to()`, `objective_function`, `save()`, and `load()`. The `to()` method works as described above, calling the `to()` method of all tensor attributes and the model (this also allows the user to change the datatype of synthesis, for example to `torch.float16` or `torch.float64`, half and double precision, respectively). `save()` and `load()` do as the names suggest, allowing the user to move the `MADCompetition` object between machines. `objective_function()` accepts $\hat{s}$ and computes the value of the loss function described above, using cached versions of s and $g_\phi(s, s')$. Setting up the objective function as a publicly available method allows users to create a subclass of `MADCompetition`, inheriting all its methods with the ability to override `objective_function`, enabling users to extend and build on our work to test out alternative means of synthesizing MAD images.

Finally, the `plenoptic` package includes several helper functions for visualizing the status of synthesis. These helper functions allow the user to easily display the synthesized MAD image and each metric's loss. An `animate()` function is also provided to animate all of these plots over time, allowing the user to inspect synthesis progress (this is only available if the user called `synthesize()` with `store_progress!=False`). If the user has created a full set of four MAD images, two additional functions are provided to quickly display the images or the loss of the two metrics for all four synthesis objects.

4.8 Conclusion

This chapter describes `plenoptic`, an open-source python package developed over the past several years by members of the Lab for Computational Vision. This package presents well-documented, tested, and generalized implementations of synthesis methods which have been developed in the lab over the years, so that they can be used by the broader research community. `plenoptic` provides tools for researchers to better understand their computational models, and facilitates experiments like those presented in the third chapter. This piece of software represents a type of scientific work that is poorly supported at the moment: software which focuses on utility over novelty and which will require maintenance, but which will facilitate the broader adoption of these useful methods. Such software is critical for science but under-appreciated, and academic science should move towards valuing software as a scientific output in its own right. Finally, I discussed two of the methods included in `plenoptic` in more detail, describing what `Metamer` and `MADCompetition` do, how they can be used in experiments, and how to interact with the code, as well as providing some example outputs.

4.9 Example notebook

The following is a pdf export of a jupyter notebook that was put together to give a simple introduction to `plenoptic`. It can also be viewed in its GitHub repository or interacted with on mybinder. (Note that the progress bars do not render in the pdf export or static version, but are visible in the interactive version).

The following provides a gentle introduction to `plenoptic`, showing how to create a simple model that works with our synthesis methods, as well

[1]:
```
import plenoptic as po
import torch
import matplotlib.pyplot as plt
import pytest
import pyrtools as pt

%matplotlib inline
%load_ext autoreload
%autoreload 2
```

To get started, let's create our model (a simple one-channel large Gaussian) and our target image.

[2]:
```
im = po.tools.load_images('data/einstein.pgm')
```

```
/home/billbrod/miniconda3/envs/synth/lib/python3.8/site-
packages/plenoptic/tools/data.py:115: UserWarning: Creating a tensor from a list
of numpy.ndarrays is extremely slow. Please consider converting the list to a
single numpy.ndarray with numpy.array() before converting to a tensor.
(Triggered internally at  /home/conda/feedstock_root/build_artifacts/pytorch-
recipe_1635217151385/work/torch/csrc/utils/tensor_new.cpp:201.)
  images = torch.tensor(images, dtype=torch.float32)
```

To show how easy it is to construct a plenoptic-compliant model, we're going to go ahead and create the model here. We must define an `__init__` method, which initializes the model, and a `forward` method, which defines how the model transforms its input into its output.

[3]:
```
from plenoptic.simulate.canonical_computations import filters
from plenoptic.tools import conv
from torch.nn import functional as F

class Gaussian(torch.nn.Module):
    """Isotropic Gaussian convolutional filter.

    Kernel elements are normalized and sum to one.

    Parameters
    ----------
    kernel_size:
        Size of convolutional kernel.
    """

    def __init__(self, kernel_size):
        super().__init__()
        if isinstance(kernel_size, int):
            kernel_size = (kernel_size, kernel_size)
```

```
        self.kernel_size = kernel_size
        # this is a convenience function we provide for creating 2d gaussian
        # filters
        self.filt = filters.circular_gaussian2d(self.kernel_size, 3.)

    def forward(self, x, **conv2d_kwargs):
        # use circular padding so our output is the same size as our input
        x = conv.same_padding(x, self.kernel_size, pad_mode='circular')
        y = F.conv2d(x, self.filt, **conv2d_kwargs)

        return y
```

[4]:
```
model = Gaussian(8)
```

```
/home/billbrod/miniconda3/envs/synth/lib/python3.8/site-
packages/torch/functional.py:445: UserWarning: torch.meshgrid: in an upcoming
release, it will be required to pass the indexing argument. (Triggered
internally at  /home/conda/feedstock_root/build_artifacts/pytorch-
recipe_1635217151385/work/aten/src/ATen/native/TensorShape.cpp:2157.)
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
```

To work with out synthesis methods, a model must accept a 4d tensor as input and return a 3d or 4d tensor as output. 4d inputs are commonly used for pytorch models, and the dimensions are batch (often, multiple images), channel (often, RGB or outputs of different convolutional filters), height, and width. The output should then either return a 1d vector or a 2d image per batch and channel. If your model operates across channels or batches, that's no problem; for example if the model transforms RGB to grayscale, your input would have 3 channels and your output would have 1.

We can see below that our `Gaussian` model satisfies this constraint:

[5]:
```
print(im.shape)
print(model(im).shape)
```

```
torch.Size([1, 1, 256, 256])
torch.Size([1, 1, 256, 256])
```

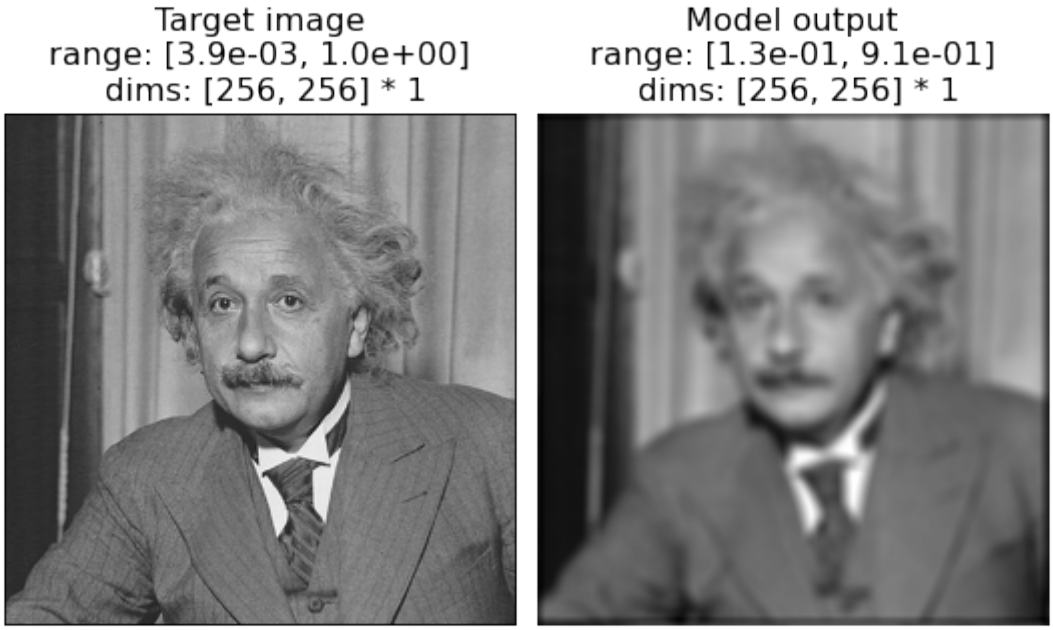
There's also several slightly more abstract constraints:

- Models must be written in PyTorch, because we make use of the its automatic differentiation features.
- Models must accept real-valued inputs and return real-valued outputs. Anything else makes optimization very tricky.
- Arbitrary model inputs and outputs should have meaning. This is easiest with models that, e.g., operate on images and predict something numeric, such as firing rate. Synthesis will return something that is out-of-set and the only constraint we place on its values is that they fall within some range. If your output is categorical or your input is more abstract, synthesis might not be useful for you

Okay, with those caveats, let's continue.

The following shows the image and the model output. We can see that output is, as we would expect, a blurred version of the input.

[6]:
```
h = model.forward(im)
fig = po.imshow(torch.cat([im, h]), title=['Target image', 'Model output'])
```



Let's start with metamer synthesis. To initialize, we only need the model and the image (there are some additional options, but the defaults are fine in this case)

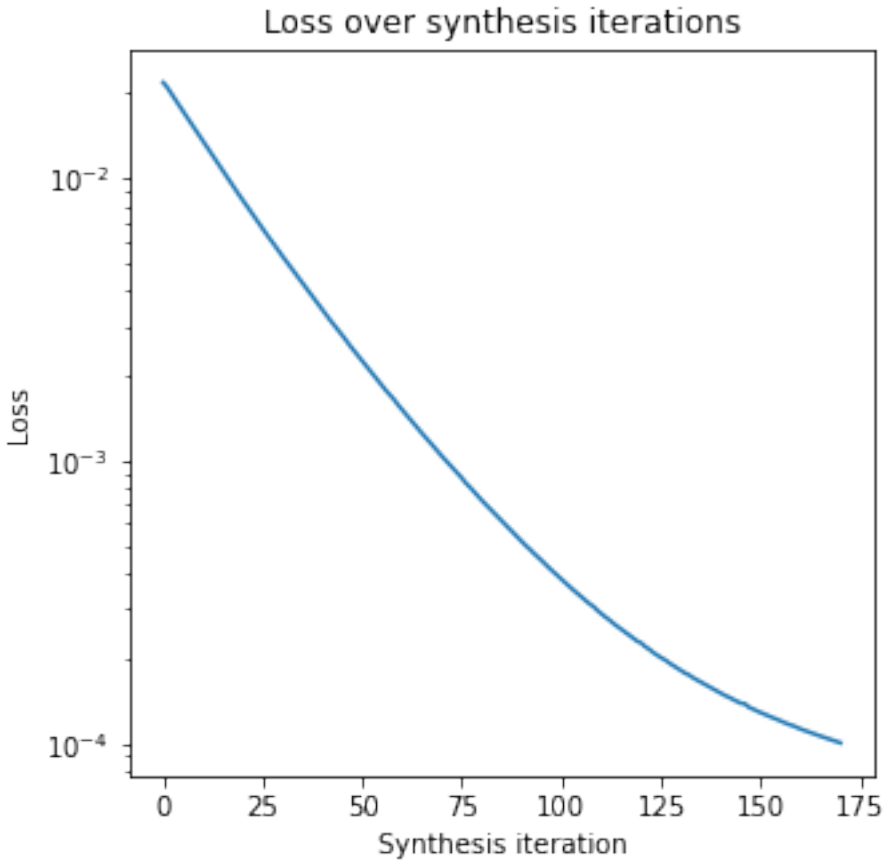
[7]:
```
met = po.synthesize.Metamer(im, model)

# we do have a default optimizer, with a specific lr and other parameters --
# if you want other than the default, create one and pass it to `synthesize`
optim = torch.optim.Adam([met.synthesized_signal], lr=.005)
synth_image = met.synthesize(max_iter=20, optimizer=optim)
# if we call synthesize again, we resume where we left off.
synth_image = met.synthesize(max_iter=150)
```

```
  0%|          | 0/20 [00:00<?, ?it/s]
  0%|          | 0/150 [00:00<?, ?it/s]
```

Let's look at the loss over time!

[8]:
```
fig, ax = plt.subplots(1, 1, figsize=(5, 5))
ax.semilogy(met.losses)
```

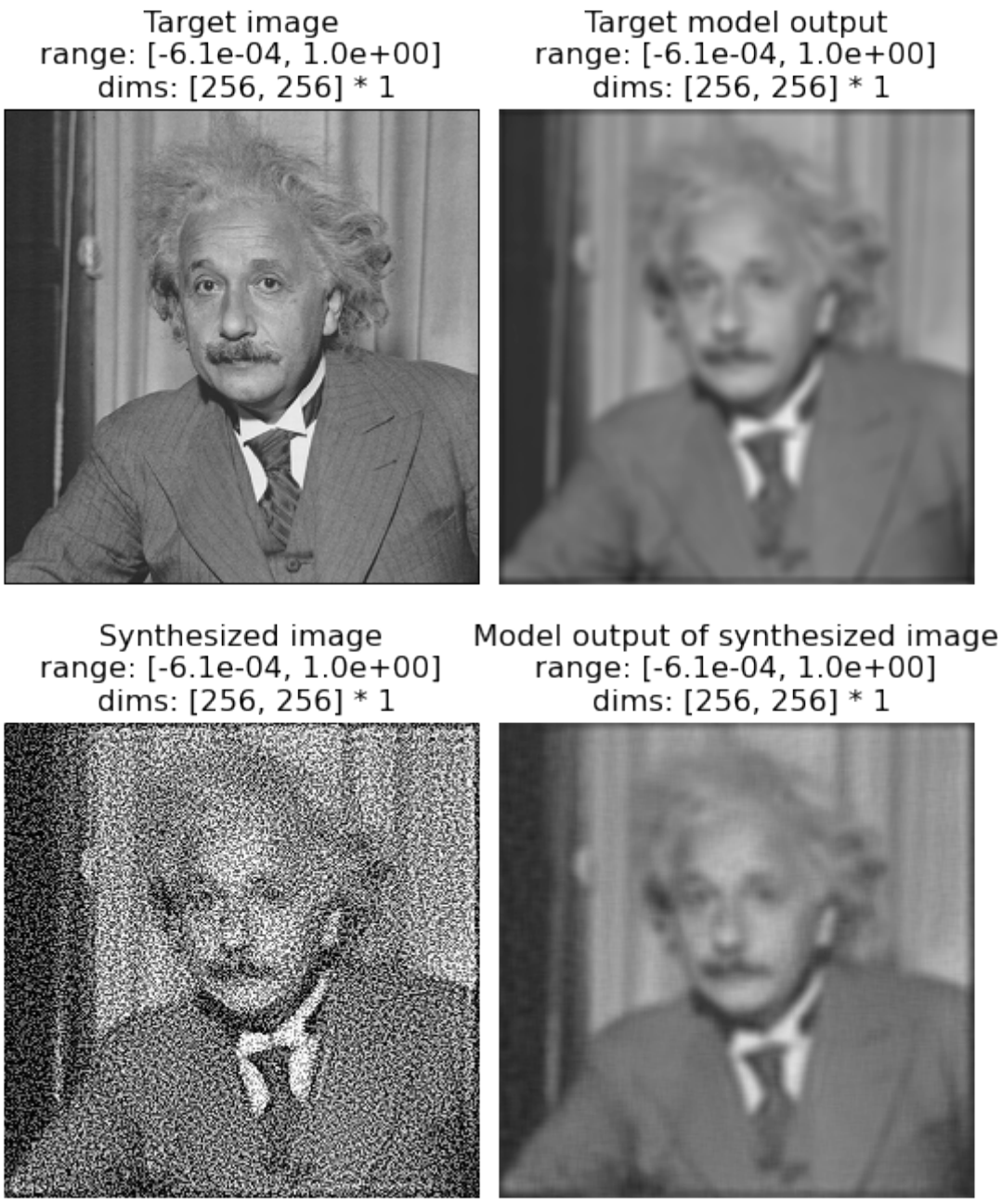
```
ax.set(title="Loss over synthesis iterations", ylabel="Loss",
       xlabel="Synthesis iteration");
```



We can see that the loss is decreasing steadily and has reached a very low value (though it hasn't converged yet).

The following figure compares the target and synthesized images, as well as showing what the model's outputs on these images looks like:

[9]:
```
po.imshow(torch.cat([im, h, synth_image, met.model(synth_image)]), col_wrap=2,
          vrange='auto1',
          title=["Target image", "Target model output", "Synthesized image",
                 "Model output of synthesized image"]);
```



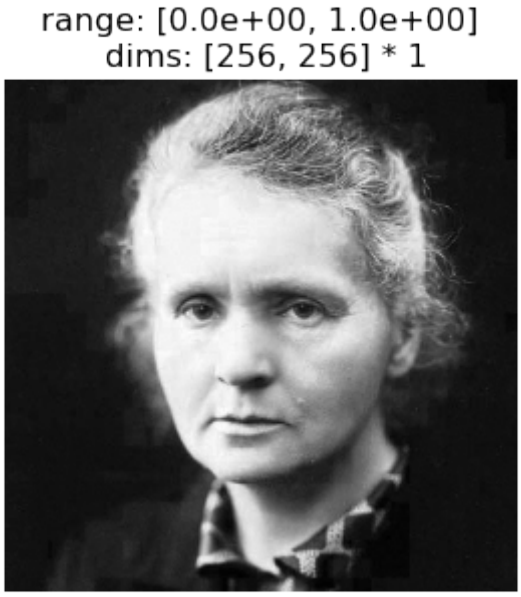
We can see that, even though the target and synthesized images look very different, the two model outputs look basically identical (which matches the exceedingly low loss value we see above). (The left column shows the images and the right column the model outputs; top row shows the target and bottom the synthesized.)

It may seem strange that the synthesized image looks like it has high-frequency noise in it – a Gaussian is a low-pass filter, so why isn't the model metamer just a blurred version of the original image? Indeed, such a blurred image *would* be a model metamer, but it's only one of many.

Gaussians are insensitive to high-frequency information, which not only means that their response doesn't change when you remove that information, but that you can *put any amount* of high frequency information into an image without affecting the model's output. Put another way, you can randomize the contents of the model's null space without affecting its response, and the goal of metamer synthesis is to generate different images that do just that.

We can see this more dramatically by initializing our metamer synthesis with a different image. By default, we initialize with a patch of white noise, but we can initialize with any image of the same size. Let's try with a different natural image, a picture of Marie Curie.

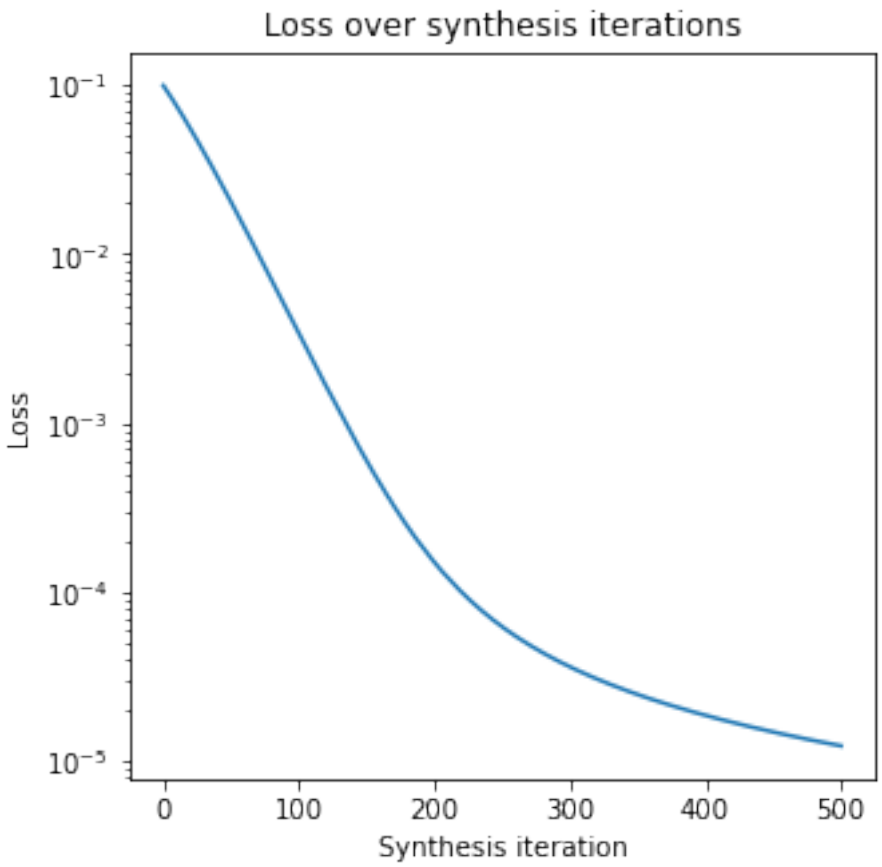
[10]:
```
curie = po.load_images('data/curie.pgm')
po.imshow([curie]);
```



[11]:
```
met = po.synthesize.Metamer(im, model, initial_image=curie, )

# we increase the length of time we run synthesis and decrease the
# stop_criterion, which determines when we think loss has converged
# for stopping synthesis early.
optim = torch.optim.Adam([met.synthesized_signal], lr=.005)
synth_image = met.synthesize(max_iter=500, optimizer=optim, stop_criterion=1e-6)
```

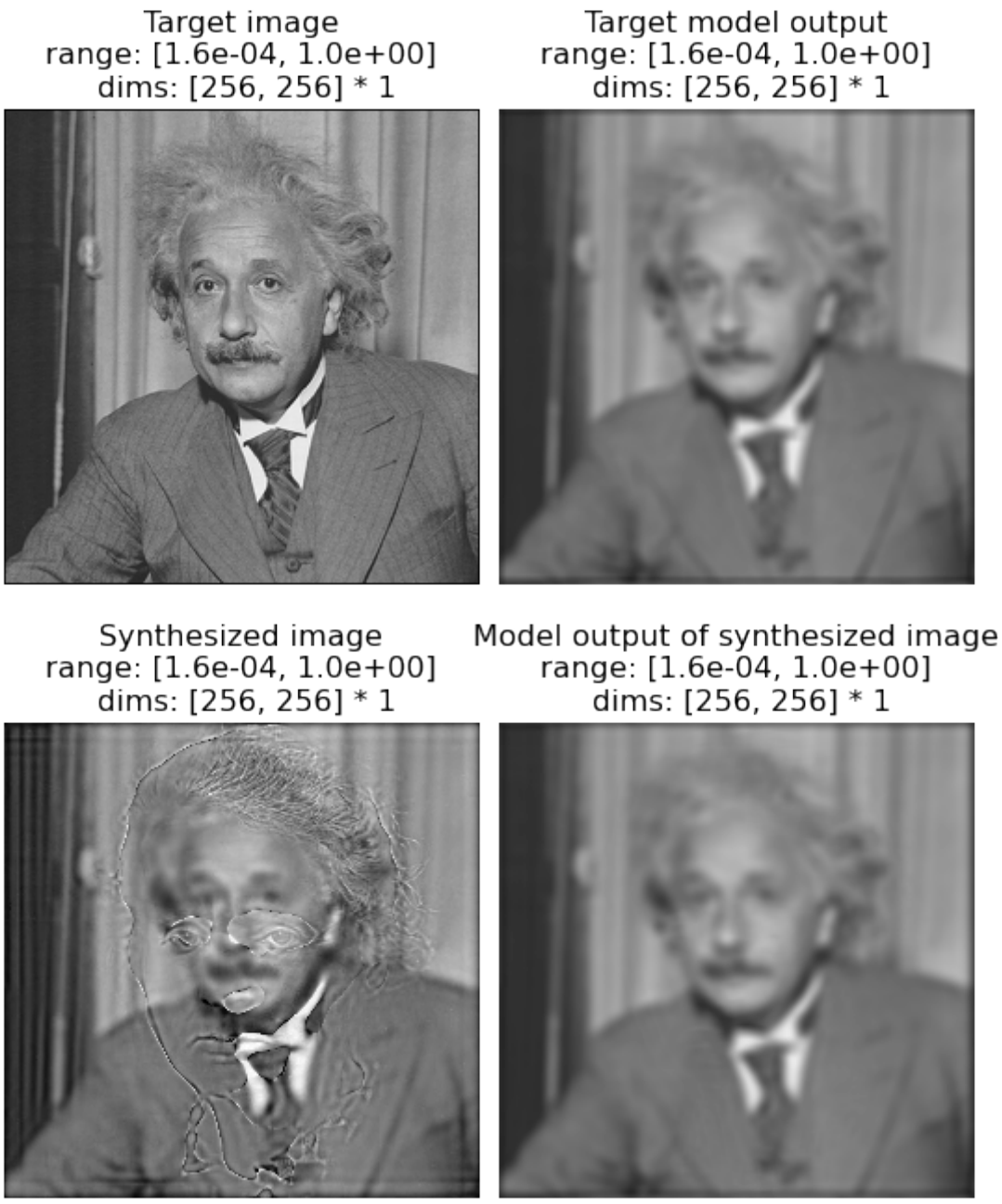
```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[12]:
```
fig, ax = plt.subplots(1, 1, figsize=(5, 5))
ax.semilogy(met.losses)
ax.set(title="Loss over synthesis iterations", ylabel="Loss",
       xlabel="Synthesis iteration");
```



We see that the synthesized image looks quite different from the target and from before, while the model outputs look very similar. Here, our synthesized model metamer looks like a blurry picture of Einstein with a high-frequency "shadow" of Curie added on top. Again, this is because the Gaussian model is insensitive to high frequencies, and thus a model metamer can include *any* high frequency information.

[13]:
```
po.imshow(torch.cat([im, h, synth_image, met.model(synth_image)]), col_wrap=2,
          vrange='auto1',
          title=["Target image", "Target model output", "Synthesized image",
                 "Model output of synthesized image"]);
```



By generating model metamers, we've gained a better understanding of the information our model is invariant to, but what if we want a better understanding of what our model is *sensitive* to? We can use `Eigendistortion` for that.

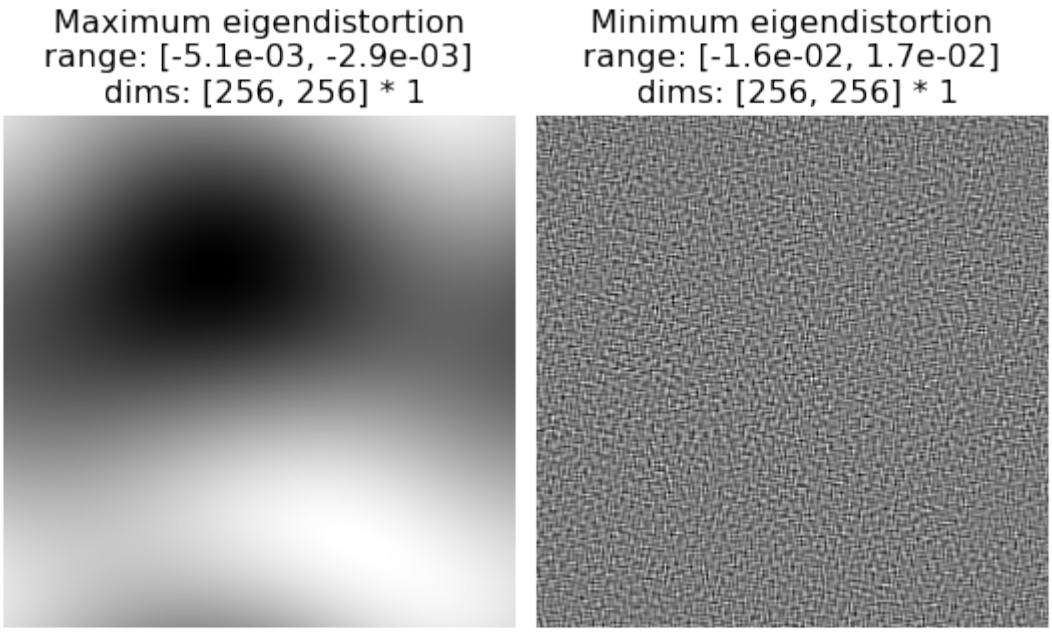
Like `Metamer`, `Eigendistortion` accepts an image and a model as its inputs. By default, it synthesizes the top and bottom eigendistortion, that is, the changes to the input image that the model finds most and least noticeable.

[14]:
```
eig = po.synthesize.Eigendistortion(im, model)
eig.synthesize();
```

```
Initializing Eigendistortion -- Input dim: 65536 | Output dim: 65536

Top k=1 eigendists:   0%|          | 0/1000 [00:00<?, ?it/s]

/home/billbrod/miniconda3/envs/synth/lib/python3.8/site-
packages/plenoptic/synthesize/eigendistortion.py:356: UserWarning: torch.qr is
deprecated in favor of torch.linalg.qr and will be removed in a future PyTorch
release.
The boolean parameter 'some' has been replaced with a string parameter 'mode'.
Q, R = torch.qr(A, some)
should be replaced with
Q, R = torch.linalg.qr(A, 'reduced' if some else 'complete') (Triggered
internally at  /home/conda/feedstock_root/build_artifacts/pytorch-
recipe_1635217151385/work/aten/src/ATen/native/BatchLinearAlgebra.cpp:1937.)
  v_new = torch.qr(Fv)[0]  # (ortho)normalize vector(s)

Top k=1 eigendists computed | Tolerance 1.00E-07 reached.

Bottom k=1 eigendists:   0%|          | 0/1000 [00:00<?, ?it/s]
```

Let's examine those distortions:

[15]:
```
po.imshow(eig.synthesized_signal, title=['Maximum eigendistortion',
                                         'Minimum eigendistortion']);
```



We can see they make sense: the most noticeable distortion is a very low-frequency modification to the image, with a period of about half the image. The least noticeable, on the other hand, is very high-frequency, which matches our understanding from the metamer example above.

This brief introduction hopefully demonstrates how you can use `plenoptic` to better understand your model representations! There's much more that can be done with both these methods, as well as two additional methods, `MADCompetition` and `Geodesic`, to explore.

Bibliography

[1] Edward H. Adelson and James R. Bergen. "Spatiotemporal Energy Models for the Perception of Motion". In: *Journal of the Optical Society of America A* 2.2 (Feb. 1985), p. 284. DOI: 10.1364/josaa.2.000284.

[2] Edward H. Adelson and James R. Bergen. "The Plenoptic Function and the Elements of Early Vision". In: *Computational Models of Visual Processing*. MIT Press, 1991, pp. 3–20.

[3] Sara Aghajari, Louis N. Vinke, and Sam Ling. "Population Spatial Frequency Tuning in Human Early Visual Cortex". In: *Journal of Neurophysiology* 123.2 (Feb. 2020), pp. 773–785. DOI: 10.1152/jn.00291.2019.

[4] Duane G Albrecht and David B Hamilton. "Striate Cortex of Monkey and Cat: Contrast Response Function." In: *Journal of neurophysiology* 48.1 (1982), pp. 217–237.

[5] Stephen J Anderson, Kathy T Mullen, and Robert F Hess. "Human Peripheral Spatial Resolution for Achromatic and Chromatic Stimuli: Limits Imposed By Optical and Retinal Factors." In: *The Journal of Physiology* 442.1 (1991), pp. 47–64.

[6] Jesper L.R. Andersson, Stefan Skare, and John Ashburner. "How To Correct Susceptibility Distortions in Spin-Echo Echo-Planar Images: Application To Diffusion Tensor Imaging". In: *NeuroImage* 20.2 (Oct. 2003), pp. 870–888. DOI: 10.1016/s1053-8119(03)00336-7.

[7] B. Balas, L. Nakano, and R. Rosenholtz. "A Summary-Statistic Representation in Peripheral Vision Explains Visual Crowding". In: *Journal of Vision* 9.12 (Nov. 2009), pp. 13–13. DOI: 10.1167/9.12.13.

[8] Martin S. Banks, Wilson S. Geisler, and Patrick J. Bennett. "The Physical Limits of Grating Visibility". In: *Vision Research* 27.11 (Jan. 1987), pp. 1915–1924. DOI: 10.1016/0042-6989(87)90057-5.

[9] Lorena A Barba et al. *Giving software its due through community-driven review and publication*. Apr. 2019. DOI: 10.31219/osf.io/f4vx6.

[10] Antoine Barbot, Shutian Xue, and Marisa Carrasco. "Asymmetries in visual acuity around the visual field". In: *Journal of Vision* 21.1 (2021), pp. 2–2.

[11] Patrick J. Bennett and Filomeno Cortese. "Masking of Spatial Frequency in Visual Memory Depends on Distal, Not Retinal, Frequency". In: *Vision Research* 36.2 (Jan. 1996), pp. 233–238. DOI: 10.1016/0042-6989(95)00085-e.

[12] Noah C. Benson et al. "Correction of Distortion in Flattened Representations of the Cortical Surface Allows Prediction of V1-v3 Functional Organization From Anatomy". In: *PLoS Comput Biol* 10.3 (Mar. 2014), pp. 1–9. DOI: 10.1371/journal.pcbi.1003538.

[13] Noah C. Benson et al. *From Retina to Extra-striate cortex: Forward Models of Visual Input; Toward a Standard Cortical Observer*. Invited talk. Optical Society of America, Oct. 2017.

[14] Noah C Benson and Jonathan Winawer. "Bayesian analysis of retinotopic maps". In: *eLife* 7 (Dec. 2018). Ed. by Mark Schira and Joshua I Gold, e40224. ISSN: 2050-084X. DOI: 10.7554/eLife.40224.

[15] Noah C Benson et al. "Cortical magnification in human visual cortex parallels task performance around the visual field". In: *eLife* 10 (Aug. 2021). Ed. by Ming Meng et al., e67685. ISSN: 2050-084X. DOI: 10.7554/eLife.67685.

[16] Noah Benson et al. “The HCP 7T Retinotopy Dataset: A new resource for investigating the organization of human visual cortex”. In: *Journal of Vision* 18.10 (2018), pp. 215–215.

[17] A Berardino et al. “Eigen-distortions of hierarchical representations”. In: *Adv. Neural Information Processing Systems (NIPS*17)*. Ed. by I Guyon et al. Vol. 30. Presented at: Neural Information Processing Systems 30, Dec 2017, Long Beach, CA. Curran Associates, Inc., Dec. 2017, pp. 1–10.

[18] James R. Bergen and Edward H. Adelson. “Early Vision and Texture Perception”. In: *Nature* 333.6171 (May 1988), pp. 363–364. DOI: 10.1038/333363a0.

[19] Jayanti Bhandari Neupane et al. “Characterization of Leptazolines A-D, Polar Oxazolines From the Cyanobacterium Leptolyngbya Sp., Reveals a Glitch With the "Willoughby-Hoye" Scripts for Calculating Nmr Chemical Shifts”. In: *Organic Letters* 21.20 (2019), pp. 8449–8453. DOI: 10.1021/acs.orglett.9b03216.

[20] Eli Bingham et al. “Pyro: Deep Universal Probabilistic Programming”. In: *arXiv preprint arXiv:1810.09538* (2018).

[21] AB Bonds. “Role of Inhibition in the Specification of Orientation Selectivity of Cells in the Cat Striate Cortex”. In: *Visual neuroscience* 2.1 (1989), pp. 41–55.

[22] Rotem Botvinik-Nezer et al. “Variability in the Analysis of a Single Neuroimaging Dataset By Many Teams”. In: *Nature* 582.7810 (2020), pp. 84–88.

[23] George E. P. Box. “Science and Statistics”. In: *Journal of the American Statistical Association* 71.356 (Dec. 1976), p. 791. DOI: 10.2307/2286841.

[24] Geoffrey M. Boynton et al. “Linear Systems Analysis of Functional Magnetic Resonance Imaging in Human V1”. In: *The Journal of Neuroscience* 16.13 (July 1996), pp. 4207–4221. DOI: 10.1523/jneurosci.16-13-04207.1996.

[25] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.2.21. 2018.

[26] C. Bradley, J. Abrams, and W. S. Geisler. "Retina-V1 Model of Detectability Across the Visual Field". In: *Journal of Vision* 14.12 (Oct. 2014), pp. 22–22. DOI: 10.1167/14.12.22.

[27] Björn Brembs. "Reliable Novelty: New Should Not Trump True". In: *PLOS Biology* 17.2 (Feb. 2019), e3000117. DOI: 10.1371/journal.pbio.3000117.

[28] Matthew Brett et al. "nipy/nibabel: 3.2.1". In: (Nov. 2020). DOI: 10.5281/zenodo.4295521.

[29] William F Broderick, Eero P Simoncelli, and Jonathan Winawer. "Mapping Spatial Frequency Preferences Across Human Primary Visual Cortex". In: *Journal of Vision* 22.4 (2022), pp. 3–3.

[30] Stephen P. Brooks and Andrew Gelman. "General Methods for Monitoring Convergence of Iterative Simulations". In: *Journal of Computational and Graphical Statistics* 7.4 (Dec. 1998), pp. 434–455. DOI: 10.1080/10618600.1998.10474787.

[31] Rachel Brown et al. *Efficient Dataflow Modeling of Peripheral Encoding in the Human Visual System*. 2021.

[32] Jonathan B. Buckheit and David L. Donoho. "WaveLab and Reproducible Research". In: *Wavelets and Statistics*. Springer New York, 1995, pp. 55–81. DOI: 10.1007/978-1-4612-2544-7_5.

[33] Peter J Burt and Edward H Adelson. "The Laplacian pyramid as a compact image code". In: *Readings in computer vision*. Elsevier, 1987, pp. 671–679.

[34] György Buzsáki. "The Brain–cognitive Behavior Problem: a Retrospective". In: *Eneuro* 7.4 (2020).

[35] Fergus W Campbell and John G Robson. "Application of Fourier Analysis To the Visibility of Gratings". In: *The Journal of physiology* 197.3 (1968), p. 551.

[36] John Canny. “A Computational Approach To Edge Detection”. In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), pp. 679–698.

[37] M. Carandini. “Do We Know What the Early Visual System Does?” In: *Journal of Neuroscience* 25.46 (Nov. 2005), pp. 10577–10597. ISSN: 1529-2401. DOI: 10.1523/jneurosci.3726-05.2005.

[38] Matteo Carandini and David J. Heeger. “Normalization As a Canonical Neural Computation”. In: *Nature Reviews Neuroscience* (Nov. 2011). ISSN: 1471-0048. DOI: 10.1038/nrn3136.

[39] T. A. Carlson. “Orientation Decoding in Human Visual Cortex: New Insights From an Unbiased Perspective”. In: *Journal of Neuroscience* 34.24 (June 2014), pp. 8373–8383. ISSN: 1529-2401. DOI: 10.1523/jneurosci.0548-14.2014.

[40] Jeffrey C. Carver et al. “Conceptualization of a Us Research Software Sustainability Institute (URSSI)”. In: *Computing in Science Engineering* 20.3 (2018), pp. 4–9. DOI: 10.1109/MCSE.2018.03221924.

[41] Patrick Cavanagh. “Functional Size Invariance Is Not Provided By the Cortical Magnification Factor”. In: *Vision Research* 22.11 (Jan. 1982), pp. 1409–1412. DOI: 10.1016/0042-6989(82)90231-0.

[42] James R. Cavanaugh, Wyeth Bair, and J. Anthony Movshon. “Nature and Interaction of Signals From the Receptive Field Center and Surround in Macaque V1 Neurons”. In: *Journal of Neurophysiology* 88.5 (Nov. 2002), pp. 2530–2546. DOI: 10.1152/jn.00692.2001.

[43] James R. Cavanaugh, Wyeth Bair, and J. Anthony Movshon. “Selectivity and Spatial Distribution of Signals From the Receptive Field Surround in Macaque V1 Neurons”. In: *Journal of Neurophysiology* 88.5 (Nov. 2002), pp. 2547–2556. DOI: 10.1152/jn.00693.2001.

[44] C. Cherici et al. “Precision of Sustained Fixation in Trained and Untrained Observers”. In: *Journal of Vision* 12.6 (June 2012), pp. 31–31. DOI: 10.1167/12.6.31.

[45] EJ Chichilnisky. "A Simple White Noise Analysis of Neuronal Light Responses". In: *Network: computation in neural systems* 12.2 (2001), p. 199.

[46] Open Science Collaboration et al. "Estimating the reproducibility of psychological science". In: *Science* 349.6251 (2015), aac4716.

[47] M. Concetta Morrone, D.C. Burr, and Lamberto Maffei. "Functional Implications of Cross-Orientation Inhibition of Cortical Visual Cells. I. Neurophysiological Evidence". In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 216.1204 (Oct. 1982), pp. 335–354. DOI: 10.1098/rspb.1982.0078.

[48] Lisa J. Croner and Ehud Kaplan. "Receptive Fields of P and M Ganglion Cells Across the Primate Retina". In: *Vision Research* 35.1 (Jan. 1995), pp. 7–24. DOI: 10.1016/0042-6989(94)e0066-t.

[49] Dany V D'Souza et al. "Dependence of Chromatic Responses in V1 on Visual Field Eccentricity and Spatial Frequency: an Fmri Study". In: *JOSA A* 33.3 (2016), A53–A64.

[50] D. M. Dacey and M. R. Petersen. "Dendritic Field Size and Morphology of Midget and Parasol Ganglion Cells of the Human Retina." In: *Proceedings of the National Academy of Sciences* 89.20 (Oct. 1992), pp. 9666–9670. DOI: 10.1073/pnas.89.20.9666.

[51] Anders M Dale, Bruce Fischl, and Martin I Sereno. "Cortical Surface-Based Analysis: I. Segmentation and Surface Reconstruction". In: *Neuroimage* 9.2 (1999), pp. 179–194.

[52] Joel Dapello et al. "Simulating a Primary Visual Cortex At the Front of CNNs Improves Robustness To Image Perturbations". In: *bioRxiv* (June 2020). DOI: 10.1101/2020.06.16.154542.

[53] John G Daugman. "Entropy Reduction and Decorrelation in Visual Coding By Oriented Neural Receptive Fields". In: *IEEE Transactions on Biomedical Engineering* 36.1 (1989), pp. 107–114.

[54] Russell L. De Valois, Duane G. Albrecht, and Lisa G. Thorell. “Spatial Frequency Selectivity of Cells in Macaque Visual Cortex”. In: *Vision Research* 22.5 (Jan. 1982), pp. 545–559. DOI: 10.1016/0042-6989(82)90113-4.

[55] Russell L De Valois, Duane G Albrecht, and Lisa G Thorell. “Cortical cells: bar and edge detectors, or spatial frequency filters?” In: *Frontiers in visual science*. Springer, 1978, pp. 544–556.

[56] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pp. 248–255.

[57] Arturo Deza, Aditya Jonnalagadda, and Miguel P. Eckstein. “Towards Metamerism via Foveated Style Transfer”. In: *International Conference on Learning Representations*. 2019.

[58] Keyan Ding et al. “Image Quality Assessment: Unifying Structure and Texture Similarity”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1. DOI: 10.1109/tpami.2020.3045810.

[59] Serge O. Dumoulin and Brian A. Wandell. “Population Receptive Field Estimates in Human Visual Cortex”. In: *NeuroImage* 39.2 (Jan. 2008), pp. 647–660. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2007.09.034.

[60] Robert O. Duncan and Geoffrey M. Boynton. “Cortical Magnification Within Human Primary Visual Cortex Correlates With Acuity Thresholds”. In: *Neuron* 38.4 (May 2003), pp. 659–671. DOI: 10.1016/s0896-6273(03)00265-4.

[61] Zakir Durumeric et al. “The Matter of Heartbleed”. In: *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, Nov. 2014. DOI: 10.1145/2663716.2663755.

[62] Stephen J Eglen et al. “Toward Standard Practices for Sharing Computer Code and Programs in Neuroscience”. In: *Nature Neuroscience* 20.6 (2017), pp. 770–773.

[63] Anders Eklund et al. “Empirically Investigating the Statistical Validity of SPM, FSL and AFNI for Single Subject fMRI Analysis”. In: *International Symposium on Biomedical Imaging*. 2015.

[64] Christina Enroth-Cugell and John G Robson. “The Contrast Sensitivity of Retinal Ganglion Cells of the Cat”. In: *The Journal of physiology* 187.3 (1966), pp. 517–552.

[65] Reza Farivar et al. “Non-Uniform Phase Sensitivity in Spatial Frequency Maps of the Human Visual Cortex”. In: *The Journal of Physiology* 595.4 (Feb. 2017), pp. 1351–1363. ISSN: 0022-3751. DOI: 10.1113/jp273206.

[66] Jenelle Feather et al. “Metamers of neural networks reveal divergence from human perceptual systems.” In: *NeurIPS*. 2019, pp. 10078–10089.

[67] David A. Feinberg et al. “Multiplexed Echo Planar Imaging for Sub-Second Whole Brain FMRI and Fast Diffusion Imaging”. In: *PLoS ONE* 5.12 (Dec. 2010). Ed. by Pedro Antonio Valdes-Sosa, e15710. DOI: 10.1371/journal.pone.0015710.

[68] Michael C. Frank. “N-Best Evaluation for Academic Hiring and Promotion”. In: *Trends in Cognitive Sciences* 23.12 (Dec. 2019), pp. 983–985. DOI: 10.1016/j.tics.2019.09.010.

[69] J. Freeman, D. J. Heeger, and E. P. Merriam. “Coarse-Scale Biases for Spirals and Orientation in Human Visual Cortex”. In: *Journal of Neuroscience* 33.50 (Dec. 2013), pp. 19695–19703. ISSN: 1529-2401. DOI: 10.1523/jneurosci.0889-13.2013.

[70] Jeremy Freeman and Eero P Simoncelli. “Metamers of the ventral stream”. In: *Nature Neuroscience* 14.9 (Aug. 2011), pp. 1195–1201. DOI: 10.1038/nn.2889.

[71] Jeremy Freeman et al. “A functional and perceptual signature of the second visual area in primates”. In: *Nature Neuroscience* 16.7 (May 2013), pp. 974–981. DOI: 10.1038/nn.3402.

[72] L. Gagnon et al. “Quantifying the Microvascular Origin of BOLD-fMRI from First Principles with Two-Photon Microscopy and an Oxygen-Sensitive Nanoprobe”. In: *Journal of Neuroscience* 35.8 (Feb. 2015), pp. 3663–3675. DOI: 10.1523/jneurosci.3555-14.2015.

[73] Paul P. Gardner et al. “Sustained Software Development, Not Number of Citations Or Journal Choice, Is Indicative of Accurate Bioinformatic Software”. In: *Genome Biology* 23.1 (Feb. 2022). DOI: 10.1186/s13059-022-02625-x.

[74] R. Gattass, C. G. Gross, and J. H. Sandell. “Visual Topography of V2 in the Macaque”. In: *The Journal of Comparative Neurology* 201.4 (1981), pp. 519–539. ISSN: 1096-9861. DOI: 10.1002/cne.902010405.

[75] L. A. Gatys, A. S. Ecker, and M. Bethge. “A Neural Algorithm of Artistic Style”. In: *arXiv* (Aug. 2015).

[76] Samuel J. Gershman. *Just looking: the innocent eye in neuroscience*. 2021.

[77] SS Ghosh et al. “A Very Simple, Re-Executable Neuroimaging Publication [version 1; Peer Review: 2 Approved With Reservations]”. In: *F1000Research* 6.124 (2017). DOI: 10.12688/f1000research.10783.1.

[78] Tal Golan, Prashant C Raju, and Nikolaus Kriegeskorte. “Controversial Stimuli: Pitting Neural Networks Against Each Other As Models of Human Cognition”. In: *Proceedings of the National Academy of Sciences* 117.47 (2020), pp. 29330–29337.

[79] Krzysztof J. Gorgolewski et al. “Nipype”. In: (2018). DOI: 10.5281/zenodo.596855.

[80] Krzysztof Gorgolewski et al. “Nipype: a Flexible, Lightweight and Extensible Neuroimaging Data Processing Framework in Python.” In: *Front Neuroinform* 5 (Aug. 2011), p. 13. ISSN: 1662-5196. DOI: 10.3389/fninf.2011.00013.

[81] Robbe L.T. Goris, Eero P. Simoncelli, and J. Anthony Movshon. “Origin and Function of Tuning Diversity in Macaque Visual Cortex”. In: *Neuron* 88.4 (Nov. 2015), pp. 819–831. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2015.10.009.

[82] Douglas N Greve and Bruce Fischl. “Accurate and Robust Brain Image Alignment Using Boundary-Based Registration”. In: *NeuroImage* 48.1 (2009), pp. 63–72. ISSN: 1095-9572. DOI: 10.1016/j.neuroimage.2009.06.060.

[83] Ed H. B. M. Gronenschild et al. “The Effects of Freesurfer Version, Workstation Type, and Macintosh Operating System Version on Anatomical Volume and Cortical Thickness Measurements”. In: *PLoS ONE* 7.6 (June 2012). Ed. by Satoru Hayasaka, e38234. DOI: 10.1371/journal.pone.0038234.

[84] Olivia Guest and Andrea E. Martin. “How Computational Modeling Can Force Theory Building in Psychological Science”. In: *PsyArXiv* (Feb. 2020). DOI: 10.31234/osf.io/rybh9.

[85] Koen V. Haak, Frans W. Cornelissen, and Antony B. Morland. “Population Receptive Field Dynamics in Human Visual Cortex”. In: *PLoS ONE* 7.5 (May 2012). Ed. by Mark W. Greenlee, e37686. DOI: 10.1371/journal.pone.0037686.

[86] Charles R Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[87] Haldan Keffer Hartline. “The Response of Single Optic Nerve Fibers of the Vertebrate Eye To Illumination of the Retina”. In: *American Journal of Physiology-Legacy Content* 121.2 (1938), pp. 400–415.

[88] David J Heeger. “Normalization of Cell Responses in Cat Striate Cortex”. In: *Visual neuroscience* 9.2 (1992), pp. 181–197.

[89] David J Heeger, Eero P Simoncelli, and J Anthony Movshon. “Computational Models of Cortical Visual Processing”. In: *Proceedings of the National Academy of Sciences* 93.2 (1996), pp. 623–627.

[90] D.W. Heeley and B. Timney. "Meridional Anisotropies of Orientation Discrimination for Sine Wave Gratings". In: *Vision Research* 28.2 (Jan. 1988), pp. 337–344. DOI: 10.1016/0042-6989(88)90162-9.

[91] H. Helmholtz. "LXXXI. On the Theory of Compound Colours". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 4.28 (1852), pp. 519–534. DOI: 10.1080/14786445208647175.

[92] O. J. Hénaff and E. P. Simoncelli. "Geodesics of Learned representations". In: *ArXiv e-prints* (Nov. 2015).

[93] Linda Henriksson et al. "Spatial Frequency Tuning in Human Retinotopic Visual Areas". In: *Journal of Vision* 8.10 (2008), p. 5. DOI: 10.1167/8.10.5.

[94] Dora Hermes et al. "An image-computable model for the stimulus selectivity of gamma oscillations". In: *eLife* 8 (Nov. 2019), e47035. DOI: 10.7554/elife.47035.

[95] Robert F. Hess et al. "Selectivity As Well As Sensitivity Loss Characterizes the Cortical Spatial Frequency Deficit in Amblyopia". In: *Human Brain Mapping* 30.12 (June 2009), pp. 4054–4069. ISSN: 1065-9471. DOI: 10.1002/hbm.20829.

[96] Simon Hettrick. *softwaresaved/software_in_research_survey_2014: Software in research survey*. Version 1.0. Feb. 2018. DOI: 10.5281/zenodo.1183562.

[97] Marc M. Himmelberg, Jonathan Winawer, and Marisa Carrasco. "Stimulus-dependent contrast sensitivity asymmetries around the visual field". In: *Journal of Vision* 20.9 (Sept. 2020), p. 18. DOI: 10.1167/jov.20.9.18.

[98] Marc M Himmelberg et al. "Cross-dataset reproducibility of human retinotopic maps". In: *Neuroimage* 244 (2021), p. 118609.

[99] Matthew D Hoffman and Andrew Gelman. "The No-U-turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo." In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1593–1623.

[100] JC Horton and WF Hoyt. "The Representation of the Visual Field in Human Striate Cortex: A Revision of the Classic Holmes Map". In: *Archives of Ophthalmology* 109.6 (1991), pp. 816–824. DOI: 10.1001/archopht.1991.01080060080030.

[101] James Howison and Julia Bullard. "Software in the Scientific Literature: Problems With Seeing, Finding, and Using Software Mentioned in the Biology Literature". In: *Journal of the Association for Information Science and Technology* 67.9 (May 2015), pp. 2137–2155. DOI: 10.1002/asi.23538.

[102] S. Hoyer and J. Hamman. "Xarray: N-D Labeled Arrays and Datasets in Python". In: *Journal of Open Research Software* 5.1 (2017). DOI: 10.5334/jors.148.

[103] D. H. Hubel and T. N. Wiesel. "Ferrier Lecture - Functional Architecture of Macaque Monkey Visual Cortex". In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 198.1130 (July 1977), pp. 1–59. DOI: 10.1098/rspb.1977.0085.

[104] David H Hubel and Torsten N Wiesel. "Receptive Fields of Single Neurones in the Cat's Striate Cortex". In: *The Journal of physiology* 148.3 (1959), p. 574.

[105] David H Hubel and Torsten N Wiesel. "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex". In: *The Journal of physiology* 160.1 (1962), pp. 106–154.

[106] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[107] Naoum P. Issa, Christopher Trepel, and Michael P. Stryker. "Spatial Frequency Maps in Cat Visual Cortex". In: *The Journal of Neuroscience* 20.22 (Nov. 2000), pp. 8504–8514. DOI: 10.1523/jneurosci.20-22-08504.2000.

[108] Pavel Izmailov et al. “Averaging Weights Leads To Wider Optima and Better Generalization”. In: *arXiv preprint arXiv:1803.05407* (2018).

[109] Akshay V. Jagadeesh and Justin L. Gardner. “Texture-Like Representation of Objects in Human Visual Cortex”. In: *bioRxiv* (2022). DOI: 10.1101/2022.01.04.474849.

[110] Caroline Jay, Robert Haines, and Daniel S. Katz. “Software Must Be Recognised As an Important Output of Scholarly Research”. In: *International Journal of Digital Curation* 16.1 (Dec. 2021), p. 6. DOI: 10.2218/ijdc.v16i1.745.

[111] C Jay et al. “The Challenges of Theory-Software Translation [version 1; Peer Review: 2 Approved, 1 Approved With Reservations]”. In: *F1000Research* 9.1192 (2020). DOI: 10.12688/f1000research.25561.1.

[112] Mark Jenkinson and Stephen Smith. “A Global Optimisation Method for Robust Affine Registration of Brain Images”. In: *Medical Image Analysis* 5.2 (2001), pp. 143–156. ISSN: 1361-8415. DOI: 10.1016/S1361-8415(01)00036-6.

[113] Mark Jenkinson et al. “Improved Optimization for the Robust and Accurate Linear Registration and Motion Correction of Brain Images”. In: *NeuroImage* 17.2 (2002), pp. 825–841. ISSN: 1053-8119. DOI: 10.1006/nimg.2002.1132.

[114] RC Jiménez et al. “Four Simple Recommendations To Encourage Best Practices in Research Software [version 1; Peer Review: 3 Approved]”. In: *F1000Research* 6.876 (2017). DOI: 10.12688/f1000research.11407.1.

[115] J. P. Jones and L. A. Palmer. “The Two-Dimensional Spatial Structure of Simple Receptive Fields in Cat Striate Cortex”. In: *Journal of Neurophysiology* 58.6 (Dec. 1987), pp. 1187–1211. DOI: 10.1152/jn.1987.58.6.1187.

[116] Yukiyasu Kamitani and Frank Tong. “Decoding the visual and subjective contents of the human brain.” In: *Nature neuroscience* 8.5 (2005), pp. 679–685. ISSN: 1097-6256. DOI: 10.1038/nn1444.

[117] Kendrick N. Kay et al. “A Two-Stage Cascade Model of BOLD Responses in Human Visual Cortex”. In: *PLoS Comput Biol* 9.5 (May 2013). Ed. by Jörn Diedrichsen, e1003079. DOI: 10.1371/journal.pcbi.1003079.

[118] Kendrick N. Kay et al. “Glmdenoise: a Fast, Automated Technique for Denoising Task-Based Fmri Data”. In: *Frontiers in Neuroscience* 7 (2013). ISSN: 1662-453X. DOI: 10.3389/fnins.2013.00247.

[119] Kendrick N. Kay et al. “Identifying Natural Images From Human Brain Activity”. In: *Nature* 452.7185 (Mar. 2008), pp. 352–355. ISSN: 1476-4687. DOI: 10.1038/nature06713.

[120] Kendrick N Kay and Jason D Yeatman. “Bottom-up and top-down computations in word- and face-selective cortex”. In: *eLife* 6 (Feb. 2017), e22341. DOI: 10.7554/elife.22341.

[121] Kendrick Kay et al. “A critical assessment of data quality and venous effects in sub-millimeter fMRI”. In: *NeuroImage* 189 (2019), pp. 847–869.

[122] Georgios A. Keliris et al. “Estimating average single-neuron visual receptive field sizes by fMRI”. In: *Proceedings of the National Academy of Sciences* 116.13 (Mar. 2019), pp. 6425–6434. DOI: 10.1073/pnas.1809612116.

[123] Shaiyan Keshvari and Ruth Rosenholtz. “Pooling of Continuous Features Provides a Unifying Account of Crowding”. In: *Journal of Vision* 16.3 (Feb. 2016), p. 39. DOI: 10.1167/16.3.39.

[124] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *ArXiv e-prints* (Dec. 2014).

[125] Thomas Kluyver et al. “Jupyter Notebooks - a publishing format for reproducible computational workflows”. In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by Fernando Loizides and Birgit Scmidt. Netherlands: IOS Press, 2016, pp. 87–90.

[126] Rebecca Knowles, Bilal A. Mateen, and Yo Yehudi. "We Need To Talk About the Lack of Investment in Digital Research Infrastructure". In: *Nature Computational Science* 1.3 (Mar. 2021), pp. 169–171. DOI: 10.1038/s43588-021-00048-5.

[127] Jan Koenderink et al. "Eidolons: Novel Stimuli for Vision Research". In: *Journal of Vision* 17.2 (Mar. 2017), p. 7. DOI: 10.1167/17.2.7.

[128] John K. Kruschke. *Doing Bayesian Data Analysis*. Second. Elsevier, 2015. DOI: 10.1016/c2012-0-00477-2.

[129] Stephen W Kuffler. "Discharge Patterns and Functional Organization of Mammalian Retina". In: *Journal of neurophysiology* 16.1 (1953), pp. 37–68.

[130] Ravin Kumar et al. "Arviz a Unified Library for Exploratory Analysis of Bayesian Models in Python". In: *Journal of Open Source Software* 4.33 (2019), p. 1143. DOI: 10.21105/joss.01143.

[131] Anna-Lena Lamprecht et al. "Towards Fair Principles for Research Software". In: *Data Science* 3.1 (2020), pp. 37–59.

[132] Michael F Land and Dan-Eric Nilsson. *Animal eyes*. 2nd. Oxford University Press, 2012.

[133] Valero Laparra et al. "Perceptually Optimized Image Rendering". In: *Journal of the Optical Society of America A* 34.9 (Aug. 2017), p. 1511. DOI: 10.1364/josaa.34.001511.

[134] Y. LeCun et al. "Backpropagation Applied To Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (Dec. 1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

[135] Adrian T Lee, Gary H Glover, and Craig H Meyer. "Discrimination of large venous vessels in time-course spiral blood-oxygen-level-dependent magnetic-resonance functional neuroimaging". In: *Magnetic resonance in medicine* 33.6 (1995), pp. 745–754.

[136] Garikoitz Lerma-Usabiaga, Jonathan Winawer, and Brian A. Wandell. "Population Receptive Field Shapes in Early Visual Cortex Are Nearly Circular". In: *The Journal of Neuroscience* 41.11 (Feb. 2021), pp. 2420–2427. DOI: 10.1523/jneurosci.3052-20.2021.

[137] Garikoitz Lerma-Usabiaga et al. "A Validation Framework for Neuroimaging Software: the Case of Population Receptive Fields". In: *PLoS computational biology* 16.6 (2020), e1007924.

[138] Mark D. Lescroart, Dustin E. Stansbury, and Jack L. Gallant. "Fourier Power, Subjective Distance, and Object Categories All Provide Plausible Models of Bold Responses in Scene-Selective Visual Areas". In: *Frontiers in Computational Neuroscience* 9 (Nov. 2015). ISSN: 1662-5188. DOI: 10.3389/fncom.2015.00135.

[139] Mark Liberman. "Obituary: Fred Jelinek". In: *Computational Linguistics* 36.4 (2010), pp. 595–599.

[140] Aravindh Mahendran and Andrea Vedaldi. "Understanding deep image representations by inverting them". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2015. DOI: 10.1109/cvpr.2015.7299155.

[141] Christopher J Markiewicz et al. "The Openneuro Resource for Sharing of Neuroscience Data". In: *eLife* 10 (Oct. 2021). Ed. by Thorsten Kahnt et al., e71774. ISSN: 2050-084X. DOI: 10.7554/eLife.71774.

[142] Tiago Marques, Martin Schrimpf, and James J. DiCarlo. "Multi-Scale Hierarchical Neural Network Models That Bridge From Single Neurons in the Primate Primary Visual Cortex To Object Recognition Behavior". In: *bioRxiv* (2021). DOI: 10.1101/2021.03.01.433495.

[143] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York, NY, USA: Henry Holt and Co., Inc., 1982. ISBN: 0716715678.

[144] J.P. Maury. *Newton: Understanding the Cosmos*. New horizons in history series. Thames and Hudson, 1992. ISBN: 9780500300237.

[145] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[146] Gil Menda et al. “Visual Perception in the Brain of a Jumping Spider”. In: *Current Biology* 24.21 (Nov. 2014), pp. 2580–2585. ISSN: 0960-9822. DOI: 10.1016/j.cub.2014.09.029.

[147] Genevieve Milliken, Sarah Nguyen, and Vicky Steeves. “A Behavioral Approach to Understanding the Git Experience”. In: *Proceedings of the 54th Hawaii International Conference on System Sciences*. Hawaii International Conference on System Sciences, 2021. DOI: 10.24251/hicss.2021.872.

[148] Steen Moeller et al. “Multiband Multislice GE-EPI At 7 Tesla, With 16-fold Acceleration Using Partial Parallel Imaging With Application To High Spatial and Temporal Whole-Brain fMRI”. In: *Magnetic Resonance in Medicine* 63.5 (Apr. 2010), pp. 1144–1153. DOI: 10.1002/mrm.22361.

[149] Felix Mölder et al. “Sustainable data analysis with Snakemake”. In: *F1000Research* 10 (Apr. 2021), p. 33. DOI: 10.12688/f1000research.29032.2.

[150] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. *DeepDream - a code example for visualizing Neural Networks*. 2015. URL: https://ai.googleblog.com/2015/07/deepdream-code-example-for-visualizing.html (visited on 03/01/2022).

[151] Christina Moutsiana et al. “Cortical idiosyncrasies predict the perception of object size”. In: *Nature Communications* 7.1 (June 2016). DOI: 10.1038/ncomms12110.

[152] J Anthony Movshon, Ian D Thompson, and David J Tolhurst. “Receptive Field Organization of Complex Cells in the Cat’s Striate Cortex.” In: *The Journal of physiology* 283.1 (1978), pp. 79–99.

[153] J Anthony Movshon, Ian D Thompson, and David J Tolhurst. "Spatial Summation in the Receptive Fields of Simple Cells in the Cat's Striate Cortex." In: *The Journal of physiology* 283.1 (1978), pp. 53–77.

[154] Randall Munroe. *xkcd: Dependency*. 2020. URL: https://xkcd.com/2347/ (visited on 03/04/2022).

[155] Danielle J. Navarro. "Between the Devil and the Deep Blue Sea: Tensions Between Scientific Judgement and Statistical Model Selection". In: *Computational Brain & Behavior* (Nov. 2018). DOI: 10.1007/s42113-018-0019-z.

[156] L. R. Newsome. "Visual Angle and Apparent Size of Objects in Peripheral Vision". In: *Perception & Psychophysics* 12.3 (May 1972), pp. 300–304. DOI: 10.3758/bf03207209.

[157] NIH. *FY 2020 By the Numbers: Extramural Investments in Research*. 2021. URL: https://nexus.od.nih.gov/all/2021/04/21/fy-2020-by-the-numbers-extramural-investments-in-research/ (visited on 03/03/2022).

[158] NIH. *NIH Awards by Location and Organization*. 2021. URL: https://report.nih.gov/award/index.cfm?ot=&fy=2020&state=NY&ic=&fm=&orgid=&distr=NY12&rfa=&om=n&pid=&view=statedetail (visited on 03/03/2022).

[159] Richard Van Noorden, Brendan Maher, and Regina Nuzzo. "The Top 100 Papers". In: *Nature* 514.7524 (Oct. 2014), pp. 550–553. DOI: 10.1038/514550a.

[160] Brian A Nosek, Jeffrey R Spies, and Matt Motyl. "Scientific Utopia: Ii. Restructuring Incentives and Practices To Promote Truth Over Publishability". In: *Perspectives on Psychological Science* 7.6 (2012), pp. 615–631.

[161] Anna Nowogrodzki. "How To Support Open-Source Software and Stay Sane". In: *Nature* 571.7763 (2019), pp. 133–135. DOI: 10.1038/d41586-019-02046-0.

[162] NSF. *Award Summary Information*. 2021. URL: https://dellweb.bfa.nsf.gov/Top50Inst2/default.asp (visited on 03/03/2022).

[163] NSF. *Pathways to Enable Open-Source Ecosystems (POSE)*. 2022. URL: https://beta.nsf.gov/funding/opportunities/pathways-enable-open-source-ecosystems-pose (visited on 03/10/2022).

[164] NSF. *Workbook: NSF by Numbers*. 2022. URL: https://tableau.external.nsf.gov/views/NSFbyNumbers/AwardObligationwithAxis?%5C%3AisGuestRedirectFromVizportal=y&%5C%3Aembed=y&%5C%3Alinktarget=_blank&%5C%3Atoolbar=top (visited on 03/03/2022).

[165] Cheryl Olman et al. "Building a Better Model of V1". In: *Journal of Vision* 17.10 (Aug. 2017), p. 780. DOI: 10.1167/17.10.780.

[166] Bruno A Olshausen and David J Field. "How Close Are We To Understanding V1?" In: *Neural computation* 17.8 (2005), pp. 1665–1699.

[167] Papers with Code. *ImageNet Benchmark (Image Classification)*. 2022. URL: https://paperswithcode.com/sota/image-classification-on-imagenet?metric=Top%5C%205%5C%20Accuracy (visited on 02/28/2022).

[168] Adam Paszke et al. "Automatic differentiation in PyTorch". In: *NIPS-W*. 2017.

[169] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019, pp. 8024–8035.

[170] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[171] Jonathan Peirce et al. "PsychoPy2: Experiments in behavior made easy". In: *Behavior Research Methods* 51.1 (Feb. 2019), pp. 195–203. DOI: 10.3758/s13428-018-01193-y.

[172] Du Phan, Neeraj Pradhan, and Martin Jankowiak. "Composable Effects for Flexible and Accelerated Probabilistic Programming in Numpyro". In: *arXiv preprint arXiv:1912.11554* (2019).

[173] Jonathan W. Pillow et al. "Spatio-Temporal Correlations and Visual Signalling in a Complete Neuronal Population". In: *Nature* 454.7207 (July 2008), pp. 995–999. DOI: 10.1038/nature07140.

[174] Russell A. Poldrack. "The Costs of Reproducibility". In: *Neuron* 101.1 (Jan. 2019), pp. 11–14. DOI: 10.1016/j.neuron.2018.11.030.

[175] Daniel A Pollen and Steven F Ronner. "Visual Cortical Neurons As Localized Spatial Frequency Filters". In: *IEEE Transactions on Systems, Man, and Cybernetics* 5 (1983), pp. 907–916.

[176] Martin Porter. *Porter Stemming Algorithm*. 2006. URL: https://tartarus.org/martin/PorterStemmer/ (visited on 03/04/2022).

[177] Martin F Porter. "An Algorithm for Suffix Stripping". In: *Program* (1980).

[178] Javier Portilla and Eero P Simoncelli. "A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients". In: *International journal of computer vision* 40.1 (2000), pp. 49–70.

[179] Angela Potochnik. "Idealization and Many Aims". In: *Philosophy of Science* 87.5 (Dec. 2020), pp. 933–943. DOI: 10.1086/710622.

[180] Andreas Prlić and James B. Procter. "Ten Simple Rules for the Open Development of Scientific Software". In: *PLOS Computational Biology* 8.12 (Dec. 2012), pp. 1–3. DOI: 10.1371/journal.pcbi.1002802.

[181] Santiago Ramon y Cajal. *Advice for a Young Investigator*. Trans. by Larry W. Swanson and Neely Swanson. Bradford book. MIT Press, 1999. ISBN: 9780262681506.

[182] Jeff Reback et al. *pandas-dev/pandas: Pandas 1.2.3*. Sept. 2021. DOI: 10.5281/zenodo.4572994.

[183] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. "On the Convergence of Adam and Beyond". In: *ArXiv e-prints* (2019).

[184] Dario L. Ringach. "Spatial Structure and Symmetry of Simple-Cell Receptive Fields in Macaque Primary Visual Cortex". In: *Journal of Neurophysiology* 88.1 (July 2002), pp. 455–463. DOI: 10.1152/jn.2002.88.1.455.

[185] Iris van Rooij. "Psychological Models and Their Distractors". In: *Nature Reviews Psychology* (Feb. 2022). DOI: 10.1038/s44159-022-00031-5.

[186] A. Rosenfeld, R. Zemel, and J. K. Tsotsos. "The Elephant in the Room". In: *ArXiv e-prints* (Aug. 2018).

[187] Zvi N Roth, David J Heeger, and Elisha P Merriam. "Stimulus vignetting and orientation selectivity in human visual cortex". In: *eLife* 7 (Aug. 2018). Ed. by Floris P de Lange and Sabine Kastner, e37241. ISSN: 2050-084X. DOI: 10.7554/eLife.37241.

[188] Nicole C. Rust et al. "Spatiotemporal Elements of Macaque V1 Receptive Fields". In: *Neuron* 46.6 (June 2005), pp. 945–956. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2005.05.021.

[189] Yuka Sasaki et al. "Local and Global Attention Are Mapped Retinotopically in Human Occipital Cortex". In: *Proceedings of the National Academy of Sciences* 98.4 (2001), pp. 2077–2082.

[190] Martin Schrimpf et al. "Integrative Benchmarking To Advance Neurally Mechanistic Models of Human Intelligence". In: *Neuron* (2020).

[191] Heiko H. Schütt and Felix A. Wichmann. "An Image-Computable Psychophysical Spatial Vision Model". In: *Journal of Vision* 17.12 (Oct. 2017), p. 12. DOI: 10.1167/17.12.12.

[192] Eric L. Schwartz. “Computational Anatomy and Functional Architecture of Striate Cortex: a Spatial Mapping Approach To Perceptual Coding”. In: *Vision Research* 20.8 (Jan. 1980), pp. 645–669. DOI: 10.1016/0042-6989(80)90090-5.

[193] Robert M Shapley and Jonathan D Victor. “The Effect of Contrast on the Transfer Properties of Cat Retinal Ganglion Cells.” In: *The Journal of physiology* 285.1 (1978), pp. 275–298.

[194] Charles S. Sherrington. *The integrative action of the nervous system*. Yale University Press, 1906.

[195] Raphael Silberzahn et al. “Many Analysts, One Data Set: Making Transparent How Variations in Analytic Choices Affect Results”. In: *Advances in Methods and Practices in Psychological Science* 1.3 (2018), pp. 337–356.

[196] Leah Silen. *NumFocus Annual Report 2020*. 2021. URL: https://numfocus.org/wp-content/uploads/2021/03/NumFOCUS-AnnualReport-2020.pdf (visited on 03/03/2022).

[197] Maria Fatima Silva et al. “Radial Asymmetries in Population Receptive Field Size and Cortical Magnification Factor in Early Visual Cortex”. In: *NeuroImage* 167 (Feb. 2018), pp. 41–52. DOI: 10.1016/j.neuroimage.2017.11.021.

[198] E P Simoncelli and W T Freeman. “The Steerable Pyramid: A flexible architecture for multi-scale derivative computation”. In: *Proc 2nd IEEE Int’l Conf on Image Proc (ICIP)*. Vol. III. Washington, DC: IEEE Sig Proc Society, Oct. 1995, pp. 444–447. DOI: 10.1109/ICIP.1995.537667.

[199] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *arXiv* (2014).

[200] Paul Smaldino. “Models Are Stupid, and We Need More of Them”. In: *Computational Models in Social Psychology*. Ed. by R. R. Vallacher, A. Nowak, and S. J. Read. Psychology Press, 2017.

[201] Stephen M. Smith et al. “Advances in Functional and Structural MR Image Analysis and Implementation As FSL”. In: *NeuroImage* 23 (Jan. 2004), S208–S219. DOI: 10.1016/j.neuroimage.2004.07.051.

[202] S. Song, D. M. Levi, and D. G. Pelli. “A Double Dissociation of the Acuity and Crowding Limits To Letter Identification, and the Promise of Improved Visual Screening”. In: *Journal of Vision* 14.5 (May 2014), pp. 3–3. DOI: 10.1167/14.5.3.

[203] Christian Szegedy et al. “Intriguing properties of neural networks”. In: (Dec. 2013), pp. 1–10.

[204] Shiming Tang et al. “Complex Pattern Selectivity in Macaque Primary Visual Cortex Revealed By Large-Scale Two-Photon Imaging”. In: *Current Biology* 28.1 (Jan. 2018), 38–48.e3. DOI: 10.1016/j.cub.2017.11.039.

[205] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. 2020. DOI: 10.5281/zenodo.3509134.

[206] Davida Y Teller. “Linking Propositions”. In: *Vision research* 24.10 (1984), pp. 1233–1246.

[207] L. Thaler et al. “What Is the Best Fixation Target? the Effect of Target Shape on Stability of Fixational Eye Movements”. In: *Vision Research* 76 (Jan. 2013), pp. 31–42. DOI: 10.1016/j.visres.2012.10.012.

[208] Harold Thimbleby. “Explaining Code for Publication”. In: *Software: Practice and Experience* 33.10 (2003), pp. 975–1001. DOI: 10.1002/spe.537.

[209] Gašper Tkačik et al. “Natural Images From the Birthplace of the Human Eye”. In: *PLoS ONE* 6.6 (June 2011). Ed. by David C. Burr, e20409. DOI: 10.1371/journal.pone.0020409.

[210] Meropi Topalidou et al. “A Long Journey into Reproducible Computational Neuroscience”. In: *Frontiers in computational neuroscience* 9.28 (2015). DOI: 10.3389/fncom.2015.00028.

[211] John S Tregoning and Jason E McDermott. “Ten Simple Rules To Becoming a Principal Investigator”. In: *PLOS Computational Biology* 16.2 (2020), e1007448.

[212] “Understanding Visual Representation by Developing Receptive-Field Models”. In: *Visual Population Codes*. Cambridge, MA: The MIT Press, 2011. DOI: 10.7551/mitpress/8404.003.0009.

[213] David Van Dijk, Ohad Manor, and Lucas B Carey. “Publication Metrics and Success on the Academic Job Market”. In: *Current Biology* 24.11 (2014), R516–R517.

[214] David C Van Essen and Charles H Anderson. “Information processing strategies and pathways in the primate visual system”. In: *An introduction to neural and electronic networks*. Ed. by Zornetzer et al. 2nd ed. Academic Press New York, 1995, pp. 45–76.

[215] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[216] Jonathan D Victor. “The Dynamics of the Cat Retinal X Cell Centre.” In: *The Journal of physiology* 386.1 (1987), pp. 219–246.

[217] B Vintch, J A Movshon, and E P Simoncelli. “A convolutional subunit model for neuronal responses in macaque V1”. In: *J Neurosci* 35 (44 Nov. 2015), pp. 14829–14841. DOI: 10.1523/JNEUROSCI.2815-13.2015.

[218] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: https://doi.org/10.1038/s41592-019-0686-2.

[219] Thomas SA Wallis et al. “Image Content Is More Important Than Bouma’s Law for Scene Metamers”. In: *eLife* 8 (Apr. 2019). DOI: 10.7554/elife.42512.

[220] Stéfan van der Walt et al. “Scikit-Image: Image Processing in Python”. In: *PeerJ* 2 (June 2014), e453. ISSN: 2167-8359. DOI: 10.7717/peerj.453.

[221] Brian A Wandell. *Foundations of vision.* Sunderland, MA: Sinauer Associates, 1995.

[222] Brian A Wandell and Jonathan Winawer. "Computational Neuroimaging and Population Receptive Fields". In: *Trends in cognitive sciences* 19.6 (2015), pp. 349–357.

[223] Z. Wang, E.P. Simoncelli, and A.C. Bovik. "Multiscale structural similarity for image quality assessment". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003.* IEEE, 2003. DOI: 10.1109/acssc.2003.1292216.

[224] Z. Wang et al. "Image Quality Assessment: From Error Visibility To Structural Similarity". In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pp. 600–612. DOI: 10.1109/tip.2003.819861.

[225] Z Wang and E P Simoncelli. "Maximum Differentiation (MAD) Competition: A Methodology for Comparing Computational Models of Perceptual Discriminability". In: *Journal of Vision* 8.12 (Sept. 2008), pp. 1–13. DOI: 10.1167/8.12.8.

[226] Zhou Wang and A.C. Bovik. "Mean Squared Error: Love It Or Leave It? A New Look At Signal Fidelity Measures". In: *IEEE Signal Processing Magazine* 26.1 (Jan. 2009), pp. 98–117. DOI: 10.1109/msp.2008.930649.

[227] Michael L. Waskom. "seaborn: statistical data visualization". In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021.

[228] Wikipedia contributors. *Log4Shell - Wikipedia, The Free Encyclopedia.* 2022. URL: https://en.wikipedia.org/w/index.php?title=Log4Shell&oldid=1073831176 (visited on 03/04/2022).

[229] Wikipedia contributors. *Messier 87 — Wikipedia, The Free Encyclopedia.* 2022. URL: https://en.wikipedia.org/w/index.php?title=Messier_87&oldid=1075712488 (visited on 03/07/2022).

[230] Michael O. Wilkinson et al. “Neural Bandwidth of Veridical Perception Across the Visual Field”. In: *Journal of Vision* 16.2 (Jan. 2016), p. 1. DOI: 10.1167/16.2.1.

[231] Rick A. Williams et al. “Oblique Effects in Normally Reared Monkeys (Macaca nemestrina): Meridional Variations in Contrast Sensitivity Measured With Operant Techniques”. In: *Vision Research* 21.8 (Jan. 1981), pp. 1253–1266. DOI: 10.1016/0042-6989(81)90230-3.

[232] Greg Wilson et al. “Best Practices for Scientific Computing”. In: *PLoS Biology* 12.1 (2014). ISSN: 15449173. DOI: 10.1371/journal.pbio.1001745.

[233] Jonathan Winawer et al. “Mapping Hv4 and Ventral Occipital Cortex: The Venous Eclipse”. In: *Journal of Vision* 10.5 (2010), p. 1. DOI: 10.1167/10.5.1.

[234] Junqian Xu et al. “Evaluation of Slice Accelerations Using Multiband Echo Planar Imaging At 3t”. In: *NeuroImage* 83 (Dec. 2013), pp. 991–1001. DOI: 10.1016/j.neuroimage.2013.07.055.

[235] D. L. K. Yamins et al. “Performance-Optimized Hierarchical Models Predict Neural Responses in Higher Visual Cortex”. In: *Proceedings of the National Academy of Sciences* 111.23 (May 2014), pp. 8619–8624. ISSN: 1091-6490. DOI: 10.1073/pnas.1403112111.

[236] Jason Yosinski et al. “Understanding Neural Networks Through Deep Visualization”. In: *arXiv preprint arXiv:1506.06579* (2015).

[237] Corey M. Ziemba and Eero P. Simoncelli. “Opposing Effects of Selectivity and Invariance in Peripheral Vision”. In: *Nature Communications* 12.1 (July 2021). DOI: 10.1038/s41467-021-24880-5.

[238] Corey M. Ziemba et al. “Contextual Modulation of Sensitivity To Naturalistic Image Structure in Macaque V2”. In: *Journal of Neurophysiology* 120.2 (Aug. 2018), pp. 409–420. DOI: 10.1152/jn.00900.2017.