

Supplementary Materials

In this supplementary section, we provide additional details on our computational framework for modeling the emergence of human color vision. Section A covers the simulation of optic nerve signals from hyperspectral scene images and our modeling of retina circuitry. Section B details the design of our simulated cortical model. Section C provides a step-by-step description of how we simulate formal color matching function tests of color dimensionality, *CMF-SIM*. In Section D, we probe whether the emergence of color vision depends on cortical learning, by introducing comparative, baseline cortical models. Section E simulates variant cell expression phenotypes after gene therapy in squirrel monkeys (Mancuso et al., 2009), showing robustness of the simulation result that addition of a new cell opsin gene enhances color dimensionality. Finally, Section F discusses our tetrachromacy simulation, highlighting the differences between the human Q cone (between M and L cones) and the pigeon Q cone (between S and M cones).

Upon acceptance, we will make our entire codebase available along with trained model weights and comprehensive documentation.

Table 1: Glossary of terms used in our framework

Scalars	
t, x, y	Time, world view coordinates
u, v	Optic nerve image coordinates
N	N -dimensional vector represents cortical color
Eye Simulation Engine	
$S_t(x, y)$	Scene image, spectrum at each pixel
Υ	Retina encoding of scene into time-averaged optic nerve image: $\Upsilon(S_t) = O_t$
Cortical Image Functions	
$O_t(u, v)$	Time-averaged optic nerve image
$A_t(u, v)$	Cone activations image
$\tilde{V}_t(u, v)$	Visual percept image
$V_t(x, y)$	Unwarped visual percept image
$\hat{O}_{t+dt}(u, v)$	Predicted optic nerve image, time $t + dt$
Neural Buckets (Learnable Parameters)	
C	Color type of each cone
P	Position of each cone in visual field
W	Lateral inhibition weights to neighboring cones
D	Demosaicing operator parameters
M	Motion estimation operator parameters
Learned Cortical Function Operators	
Φ, Ψ	Φ decodes O_t into V_t , and Ψ re-encodes V_t into O_t . $\Phi = \Phi_P \circ \Phi_C \circ \Phi_W$, and $\Psi = \Psi_W \circ \Psi_C \circ \Psi_P$.
Φ_W, Ψ_W	Transforms O_t into A_t , and vice-versa.
Φ_C, Ψ_C	Transforms A_t into \tilde{V}_t , and vice versa.
Φ_P, Ψ_P	Transforms \tilde{V}_t into V_t , and vice versa.
$\mu_M(O_1, O_2)$	Estimates world translation (x, y) between two optic nerve images
$\Omega_{(x, y)}$	Translates an image laterally by (x, y) . $\Omega_{\mu_M(O_t, O_{t+dt})}(V_t) = V_{t+dt}$

A SIMULATION ENGINE OF BIOLOGICAL EYES

We simulate the optic nerve signals from hyperspectral scene images, simulate eye gaze movements, and model the retina circuitry. We describe the details of these simulations in the following.

A.1 DATABASE OF HYPERSPECTRAL SCENE IMAGES

We model the scene viewed by the eye using a dataset comprising 900 hyperspectral images from everyday scenes, captured with a hyperspectral camera (Arad et al., 2022). Each image in the dataset is of dimension $512 \text{ pixels} \times 482 \text{ pixels} \times 31 \text{ spectral channels}$, where each pixel is represented by its spectral power distribution (“spectra” below). To generate our training data set, we cropped thousands of 482×482 patches from these hyperspectral images, ensuring sufficient resolution to avoid aliasing effects, induced by varying cone cell densities (Section A.3.2).

While only hyperspectral images are used during learning simulation, RGB images can be presented during test-time by simulating a projector with specific spectral power distributions (SPDs) for the R, G, and B channels (Cottaris et al., 2019), effectively converting each RGB image into a spectral representation before passing into the retina model. This test-time use of RGB images does not affect learning and is purely for illustrative purposes in the figures (e.g., Figure 3).

A.2 EYE GAZE MOVEMENTS – FIXATIONAL DRIFT

We simulate small eye movement as fixational eye drift with a random walk (Young & Smithson 2021). This produces a stream of the current scene image translating across the retina. Over a time interval of duration dt , we sample the change in eye motion from a uniform probability distribution $(dX, dY) \sim \mathcal{U}\{(-15, 15) \times (-15, 15)\}$ where 15 corresponds to 15 pixels in the scene image S .

The Mona Lisa video in Supplementary Video also incorporates manually-authored saccades (e.g. from hand to mouth) for illustrative purposes, but the learning simulation uses only fixational drift.

A.3 RETINA CIRCUITRY MODEL

Our model of light detection and signal encoding in the retina is the textbook model (Rodieck, 1998) of the photopic, midget, “private-line” pathway (see Figure 2). Details of our implementation follow.

A.3.1 CONE MOSAIC AND SPECTRAL SAMPLING

In our simulation, we model 256×256 cone cells, with the random distribution of L, M and S cone cells on the retina with a relative probability of 0.63, 0.32 and 0.05 from (Sabesan et al., 2015). We model L, M and S spectral response using the template response function from (Carroll et al., 2000), with spectral peaks of 560, 530, 419 nm, respectively. We model photon-shot noise in the photoreceptor activation values, simulating a signal-to-noise ratio of approximately 100. The simulation omits foveal tritanopia (Williams et al., 1981).

A.3.2 FOVEATION AND CONE POSITIONS

We model cone cell spatial density on the retina in accordance with (Curcio et al., 1990), while enforcing retinotopy. Starting from a regular grid of cells, we computationally perturb positions until the target spatial distribution is stochastically achieved, while constraining each cell to be always confined by its neighbors. We model biological variation in cell locations, which represents a challenge for cortical learning, by adding multi-resolution positional randomness (Perlin, 1985).

A.3.3 CENTER-SURROUND LATERAL INHIBITION

We mathematically model textbook lateral inhibition in retinal signals as a Difference-of-Gaussians (DoG) convolution kernel (Enroth-Cugell et al., 1983), with parameters fitted from electrophysiology (Wool et al., 2018). Specifically our DoG kernel has standard deviations of 0.15 and 0.9 cone diameters, respectively, for the positive center and negative surround Gaussians. Significantly, the relative amplitudes are such that the kernel has a non-zero mean of 0.09 (Lennie et al., 1991; Wool et al., 2018). We convolve the array of cone activation values by this DoG kernel to compute outputs to bipolar cells.

A.3.4 ON/OFF PATHWAYS AND OPTIC NERVE SPIKING

We model textbook on- and off- connections from cone activations to bipolar cells, forwarding to on- and off- retinal ganglion cells (RGC)s. The on- activation is modeled by a rectified linear unit activation function, $\text{ReLU}(x) = \max(0, x)$, where x is the laterally-inhibited cone output. The off- activation is modeled as $\text{ReLU}(-x)$. Optic nerve spikes are the outputs at RGC axons, with action potentials are generated with Leaky Integrate and Fire model (Lapicque, 1907; Abbott, 1999; Arbib, 2003).

B CORTICAL MODEL

B.1 TIME-AVERAGED OPTIC NERVE SIGNALS

We model the first step of cortical processing as re-combination of the on- and off- RGC action potentials followed by time-averaging. This results in an image stream O_t with real-valued pixel values that are proportional to the laterally-inhibited output from cone cells. O_t is the sole input to the remainder of the cortical model.

B.2 PREDICTION PIPELINE AND LEARNING OBJECTIVE

Figure 7 shows an elaboration of the cortical prediction pipeline introduced in Figure 3.1 and summarized in Section 4. This pipeline represents an existence-proof simulation that the hypothesized self-supervised learning on O_t can successfully produce: (1) emergence of vision with the correct color dimensionality, and (2) inference of invariant retinal properties, including learned neural buckets for cone spectral identities C , cell positions P and lateral inhibition weights W . Our results show that both can be simultaneously achieved by learning to minimize the error in predicting the fluctuations in optic nerve signal values.

To re-summarize Section 4, the prediction is made by applying the learned functions for decoding Φ , translation Ω and re-encoding Ψ to the optic nerve signal O_t , to obtain a predicted optic nerve signal $\hat{O}_{t+dt} = \Psi(\Omega(\Phi(O_t)))$. The learning objective function is to minimize prediction error, the difference between predicted and real optic nerve images at time $t + dt$: $E_{\text{prediction}} = \|O_{t+dt} - \hat{O}_{t+dt}\|_2^2 = \|O_{t+dt} - \Psi(\Omega(\Phi(O_t)))\|_2^2$. The decoder Φ and re-encoder Ψ functions are each factorized into a pipeline, such that: $\Phi = \Phi_P \circ \Phi_C \circ \Phi_W$ and $\Psi = \Psi_W \circ \Psi_C \circ \Psi_P$, where each sub-function is an operator conditioned on its corresponding neural bucket, C , P and W . The following sections describe the implementation of each learnable neural bucket and its associated cortical sub-function.

B.2.1 LEARNING LATERAL INHIBITION NEIGHBOR WEIGHTS W

In the factorized function pipeline, sub-decoder Φ_W conceptually inverts lateral inhibition to transform optic nerve image O_t into cone activation image A_t . Conversely, sub-encoder Ψ_W reproduces lateral inhibition to transform A_t into O_t . These operators are modeled, respectively, as inverse and forward convolution operators, implemented by pixel-wise division and multiplication in the Fourier domain. The learnable parameters for this operator are neural bucket W , representing the Fourier transform image pixels of the lateral inhibition DoG kernel’s Fourier transform. The resolution of W is set equal to the image resolution of O_t .

B.2.2 LEARNING COLOR TYPES C AND INTERPOLATION D

In the factorized function pipeline, sub-decoder Φ_C conceptually transforms cone activation image A_t (scalar-valued pixels) into a full-color visual percept image \tilde{V}_t (N -dimensional vector pixels). Conversely, Ψ_C re-encodes \tilde{V}_t into A_t . These functions depend on two neural buckets of learnable parameters, C and D . C is an image of N -dimensional vectors, in which the cortical model learns to represent the spectral type of the source cone associated with each pixel in A_t . D are the parameter weights of a learnable function ϕ_D that spatially interpolates color across the image, which we implement as a convolutional neural network based on the U-Net (Ronneberger et al., 2015) architecture.

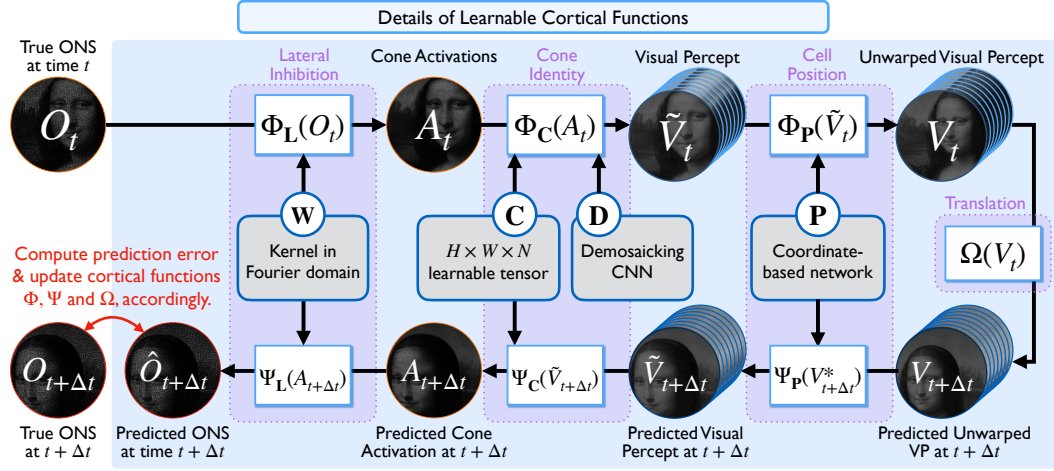


Figure 7: Pipeline of decoding, translation and re-encoding cortical functions. The optic nerve signal image O_t is decoded into visual percept image V_t . The percept is translated in accordance with eye motion over time dt , then re-encoded back into a prediction of the optic nerve signal \hat{O}_{t+dt} at a short time in the future.

Mathematically, Φ_C and Ψ_C are defined as $\Phi_C(A_t) = \phi_D(A_t \otimes C)$, where \otimes denotes element-wise multiplication; $\Psi_C(\tilde{V}_t) = \tilde{V}_t \odot C$, where \odot denotes element-wise dot product. To maintain Φ_C and Ψ_C as pseudo-inverses, we constrain $C \odot C$ to be an image with all pixels equal to 1.

B.2.3 LEARNING CELL POSITIONS P

As shown in Fig. 2.3.F, optic nerve images are spatially distorted due to foveation, whereas visual perception is apparently undistorted. In the factorized function pipeline, sub-decoder Φ_P conceptually transforms $\tilde{V}_t(u, v)$, which is foveated and spatially distorted, into the final visual percept image $V_t(x, y)$, which is Euclidean and undistorted. Conversely, Ψ_P re-warps $V_t(x, y)$ into $\tilde{V}_t(u, v)$.

We define a learnable spatial warping function $\phi_P(u, v) = (x, y)$, implemented using normalizing flow (Rezende & Mohamed, 2015; Dinh et al., 2016) that is an invertible function. Neural bucket P comprises the learnable parameters of this normalizing flow network. Then $\Phi_P(\tilde{V}_t)(x, y) = \tilde{V}_t(\phi_P^{-1}(x, y))$, and $\Psi_P(V_t)(u, v) = \tilde{V}_t(\phi_P(u, v))$.

B.3 LEARNING EYE MOTION ESTIMATION M

In the learning loop, cortical sub-function Ω translates the visual percept image according to the spatial motion (dx, dy) that occurs during the brief period between t and $t + dt$. Where does (dx, dy) come from? Here, we describe how the cortex can learn a helper cortical function $\mu_M(O_t, O_{t+dt})$ that estimates (dx, dy) directly from the optic nerve stream values at times t and $t + dt$, maintaining the strict operation of the cortical learning model purely from optic nerve signals.

We model learning of $\mu_M(O_t, O_{t+dt})$ with neural bucket M by optimizing:

$$M = \underset{M, P}{\operatorname{argmin}} \|B(O_{t+dt}) - \Psi_P(\Omega_{\mu_M(O_t, O_{t+dt})}(\Phi_P(B(O_t))))\|_2^2,$$

where B is a convolution operator that low-pass filters the optic nerve images, effectively halving the highest frequencies; Φ_P aims to dewarp the optic nerve image into Euclidean coordinates, and Ψ_P is the inverse operator that rewarps the image. Details are discussed also in Video 16:04.

B.4 EFFICIENT LEARNING IMPLEMENTATION & ALTERNATIVE REPRESENTATION OF INTERNAL PERCEPTS

We simulate the self-supervised learning by parallel numerical optimization of all neural buckets by applying the stochastic gradient descent algorithm to minimize the prediction error metric. In

practice, this is implemented by repeating the following step thousands of times until convergence: pick a batch of different scene images from the database; generate the optic nerve signals at time t with the retina encoding model; send this into the cortical model and execute the decode / translate / re-encode functions with current neural bucket parameters to estimate the signal at $t + dt$; compute the actual optic nerve signal at $t + dt$ with the retina encoding model; compute the prediction error by taking the difference between prediction and actual signal; backpropagate the prediction error to update neural bucket parameters.

For efficient processing of cortical functions, we avoid computing full-resolution undistorted visual percept images $V_t(x, y)$ during the learning simulation. Instead, we directly compute $\tilde{V}_{t+dt}(u, v)$ by optical flow of the pixels in $\tilde{V}_t(u, v)$, where the optical flow map is defined by $(\Psi_P \circ \Omega_{\mu_M(O_t, O_{t+dt})} \circ \Phi_P)$. This is mathematically equivalent to the learning implementation described in Section B.2 and the cortical model continues to learn cell positions P , and therefore decoder Φ can still be applied to compute undistorted visual percepts when desired.

C COLOR MATCHING FUNCTION TESTS & *CMF-SIM*

In the classical color matching theory of [Maxwell (1856)] and [Grassmann (1853)], an observer compares a test color spectrum, t , and a set of K primary color spectra, p_i for $i = 1, \dots, K$ and tunes real weights α_i for each primary until a color match is achieved. If a particular weight has a negative value, primary light of that magnitude is added to the test color t rather than to the other primary colors. Mathematically, a color match is achieved when the matching spectrum $\sum_{i=1}^K \alpha_i^+ p_i$ and test spectrum $t + \sum_{i=1}^K (-\alpha_i)^+ p_i$ appear identical. Here, α_i^+ is defined as $\max(\alpha_i, 0)$.

We simulate such classical color matching by formulating the retina model plus learned cortical model as a black-box color observer in the colorimetric sense. That is, we take the test and matching spectra, simulate the stimulation on different parts of the retina for test and match color patches, apply the retina encoding model Υ and the learned cortical decoder Φ to each separately, and iteratively update the weights α_i until there is no further improvement in the computed perceptual difference $E = \|\Phi(\Upsilon(\sum_{i=1}^K \alpha_i^+ p_i)) - \Phi(\Upsilon(t + \sum_{i=1}^K (-\alpha_i)^+ p_i))\|_2^2$. Note that the optimal E will not be close to zero if the primaries cannot match test color t .

The color dimensionality of an observer is formally equal to the minimum number of primary colors required to match any test color. *CMF-SIM* simulates thousands of color matches to rigorously compute this dimensionality for any given retina model plus cortical model. In *CMF-SIM*, we simulate classical color matching function (CMF) tests of exhaustively attempting to match test colors equal to each monochromatic wavelength (100 samples from 400 to 700 nm), with a linear combination of K primary colors. For the primary colors, we use distinct, monochromatic spectra. We start CMF tests with a single primary, and add primaries one-by-one until all test wavelengths in the CMF can be matched successfully. The pseudocode for *CMF-SIM* is described in Algorithm 1.

For example, for our cortical model to be formally measured as trichromat, we must show that there do not exist any 1 or 2 primaries that can pass the CMF tests, and show 3 primaries that succeed. A brute-force approach would be to exhaustively test all possible sets of K primaries; for efficiency, we instead perform stratified sampling to randomly generate a large number (e.g. 500) distinct sets of primaries for each choice of K . We require that all of these possible primary sets fail, with the aggregate perceptual error across all wavelengths being greater than a threshold ϵ , before incrementing the CMF tests to $N+1$ primaries.

D BASELINE RESULTS – TESTING IF CORTICAL INFERENCE IS REQUIRED FOR COLOR VISION

A fundamental question about color perception is whether cortical inference is necessary, or if human color perception arises directly from hardwired neural circuits. The main paper presents analysis an existence proof that cortical inference can indeed result in color vision of the correct dimensionality. Here, we add evidence that cortical inference of some kind is necessary, by modeling and testing three baseline cortical models with limited or no cortical processing.

Algorithm 1: Pseudocode for *CMF-SIM***Data:** WAVELENGTHS, MAX_TRIALS, Retinal process Υ , Cortical decoder Φ **Function** CMF-SIM():

```

    base_errors  $\leftarrow$  ComputeBaseErrors()
    color_dimensionality  $\leftarrow$  FindMinimumPrimaries(base_errors)
    return color_dimensionality

```

Function ComputeBaseErrors():

```

    Initialize an array errors
    foreach wavelength  $\lambda$  in WAVELENGTHS do
         $(x, y)_A = \text{RandomLocation}()$ 
         $(x, y)_B = \text{RandomLocation}()$ 
        colorPatchA  $\leftarrow$  MonochromaticColorPatch( $\lambda, (x, y)_A$ )
        colorPatchB  $\leftarrow$  MonochromaticColorPatch( $\lambda, (x, y)_B$ )
        perceptA  $\leftarrow \Phi(\Upsilon(\text{colorPatchA}))$ 
        perceptB  $\leftarrow \Phi(\Upsilon(\text{colorPatchB}))$ 
        errors[ $\lambda$ ]  $\leftarrow$  PerceptualError(perceptA,  $(x, y)_A$ , perceptB,  $(x, y)_B$ )
    return errors

```

Function FindMinimumPrimaries(base_errors):

```

    num_primaries  $\leftarrow$  0
    repeat
        num_primaries  $\leftarrow$  num_primaries + 1
        all_tests_passed  $\leftarrow$  true
        trial  $\leftarrow$  0
        repeat
            trial  $\leftarrow$  trial + 1
            Randomly initialize primaries  $p = \{p_1, \dots, p_{\text{num\_primaries}}\}$ 
            foreach wavelength  $\lambda$  in WAVELENGTHS do
                Zero initialize coefficients  $\alpha = \{\alpha_1, \dots, \alpha_{\text{num\_primaries}}\}$  for primary  $p$ 
                repeat
                     $(x, y)_A, (x, y)_B = \text{RandomLocations}()$ 
                    positive_alpha, negative_alpha  $\leftarrow \alpha_i^+$  for all  $i, (-\alpha_i)^+$  for all  $i$ 
                    testColorPatch  $\leftarrow$  MonochromaticColorPatch( $\lambda, (x, y)_A +$ 
                        WeightedColorPatch(negative_alpha,  $p, (x, y)_A$ )
                    matchColorPatch  $\leftarrow$  WeightedColorPatch(positive_alpha,  $p, (x, y)_B$ )
                    testPercept  $\leftarrow \Phi(\Upsilon(\text{testColorPatch}))$ 
                    matchPercept  $\leftarrow \Phi(\Upsilon(\text{matchColorPatch}))$ 
                    errors[ $\lambda$ ]  $\leftarrow$  PerceptualError(testPercept,  $(x, y)_A$ , matchPercept,  $(x, y)_B$ )
                    coefficients  $\alpha \leftarrow \text{GradientDescent}(p, \alpha, \text{target}, \text{match}, \text{errors})$ 
                until convergence
            if error  $\geq$  base_errors[ $\lambda$ ] then
                all_tests_passed  $\leftarrow$  false
                break
        until trial = MAX_TRIALS
    until all_tests_passed
    return num_primaries

```

For our first baseline, we assume an extreme case where no cortical learning occurs, treating raw optic nerve signals directly as internal percepts (i.e., the cortical decoder function Φ is an identity function). We apply *CMF-SIM* directly to the optic nerve signals generated by a trichromat retina. Figure 8.1 presents analysis that this model fails to generate vision that can be recognized as color consistent. The analysis is to perform baseline color matching experiments where the test and match

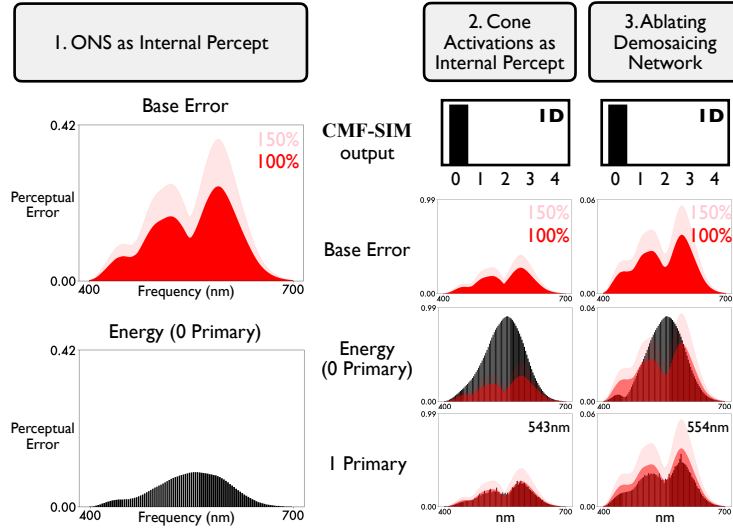


Figure 8: Comparison *CMF-SIM* results of our baseline cortical models with limited or no inferential processing. In all of these models, the input optic nerve stream comes from a retina with 3 cone cells as in regular human trichromacy. 1. The first baseline model represents no cortical learning involvement in human visual perception, by setting the internal percept directly equal to the input optic nerve signal. The resulting base error, the perceptual error between the same color patch at two different retinal locations, results in higher errors, compared to the energy of the internal percept. This means that this cortical model fails to recognize the same color consistently across different patches of the retina, so it is procedurally impossible to even try color matching function tests against reference match colors. Formally then, this model fails resolve color vision. 2. In the second baseline model, we treat cone cell activations as internal percepts, omitting lateral inhibition from the eye simulation; *CMF-SIM* measures the resulting internal percept as 1D color vision, failing to resolve trichromacy. 3. The third baseline model is an ablation model in which we omit the demosaicking network D from our proposed model of Section B; *CMF-SIM* also measures the resulting internal percept as 1D color. Red base errors are superimposed on energy / 1 primary error plots for visual comparison, and 150% of these base errors are applied as threshold determination.

spectral functions are identical. This “base perceptual error” is larger than the energy of each color percept itself, meaning that the emergent vision fails to recognize a patch as the same color across different parts of the retina.

The second baseline model is a variant of the first, where we change the optic nerve stream to directly transfer cone activation values, omitting the encoding complications of foveated warping and lateral inhibition. This is intended as a far simpler encoding, to pressure test whether color vision can be detected in the spectrally encoded cone values without further processing. However, *CMF-SIM* formally measures that this model results in 1D color (Figure 8.2), instead of the expected 3D color from such a retina. This result shows that cortical processing is required even on cone activations to produce color vision of the correct color dimensionality.

The third baseline model adds another perspective by taking the full-featured model detailed in Section B but omitting the demosaicking network ϕ_D (as defined in Section B.2.2). As shown in Figure 8.3, *CMF-SIM* measures the resulting percepts of this model also as 1D, providing evidence of the importance of demosaicking process in the emergence of color vision.

E VARIATIONS IN CELL EXPRESSION AFTER GENE THERAPY

As described in Section 6 we simulate experiments aimed at boosting color dimensionality Mancuso et al. (2009). In our simulation, approximately 60% of M cones are affected by gene therapy, with three possible scenarios for their modification. First, affected M cones are completely transformed into pure L cones (Figure 9.1). Second, affected M cones equally express M and L opsins (Figure 9.2).

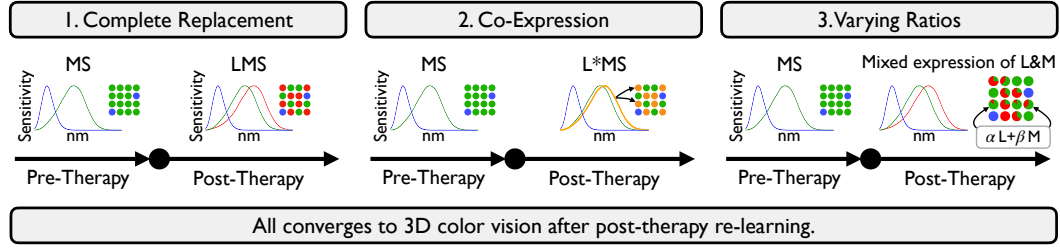


Figure 9: Three scenarios of cell expression after gene therapy adding a third cone type: 1. affected M cones transform into L cones, 2. affected M cones express 50:50 M and L opsins, and 3. affected M cones express M and L opsins at spatially-varying, random ratios (i.e. $\alpha, \beta \in [0, 1], \alpha + \beta = 1$). *CMF-SIM* shows that all cases converge to 3D color vision after post-therapy re-learning.

Third, affected M cones express M and L opsins in random ratios, that is $\alpha L + \beta M$, where α and β spatially vary across the retina (Figure 9.3). In all cases, we confirm that our cortical model acquires 3D color vision after adaptation (i.e. re-learning), formally measured by *CMF-SIM*. This makes intuitive sense if we consider the emergent vision from linear systems theory. Each different opsin can be interpreted as a basis vector for the resulting linear color space. In this view, cells containing a mixture of two cones (M and L in this case) can be interpreted as linear combinations of two basis vectors, which cannot increase the dimensionality of the linear space.

F PROBING EFFECT OF SPECTRAL RESPONSE AND ENVIRONMENTAL COLORS IN ACQUIRING TETRACHROMATIC VISION

In Section 6, tests of tetrachromatic eyes assumed the fourth cone type was a pigeon Q cone, with peak sensitivity at 506 nm, between the S and M cone peak sensitivities (419 nm and 530 nm, respectively). In this section, we report additional experiments using instead a human Q cone, modeled with peak sensitivity at 545 nm between M and L peak sensitivities (530 nm and 560 nm, respectively). As shown in the spectral graphs of Figure 10, the pigeon Q cone is more decorrelated from L, M, S, which in principle may more easily support emergence of 4D color vision.

Further, we probe the effect of visual environment on the emergence of tetrachromacy, simulating the learning of color vision with three different input scene image datasets. The first environment is the set of real hyperspectral photographs of everyday scenes (Arad et al., 2022) as described in Section A.1. However, natural hyperspectral images are thought to be lacking in human tetrachromatic colors (Lee et al., 2024), so in the second and third datasets we augment the 900 hyperspectral images with 100 tetrachromatic color patches. In the second dataset, the colors are sampled from the tetrachromatic hue sphere as computed using recently developed N -dimensional color theory for tetrachromacy (Lee et al., 2024), customized to each of the observers here (i.e. with human Q or pigeon Q, respectively). In the third dataset, the colors are sampled from an idealized tetrachromatic color space in which each of the four cone channels is allowed to take any value, allowing theoretically maximum levels of chromatic contrast. Since natural hyperspectral images have been found lacking in human tetrachromatic colors (Lee et al., 2024), in principle we may expect the likelihood that 4D color vision emerges to increase across these three simulated environments.

Indeed, in line with theoretical intuition, Figure 10 shows that 4D color vision emerges, as measured formally by *CMF-SIM*, only for the third visual environment in the case of the human Q cone, but 4D color vision emerges with any of the visual environments for the pigeon Q cone. Our simulations suggest strong genetic and environmental effects on emergence of tetrachromatic color vision.

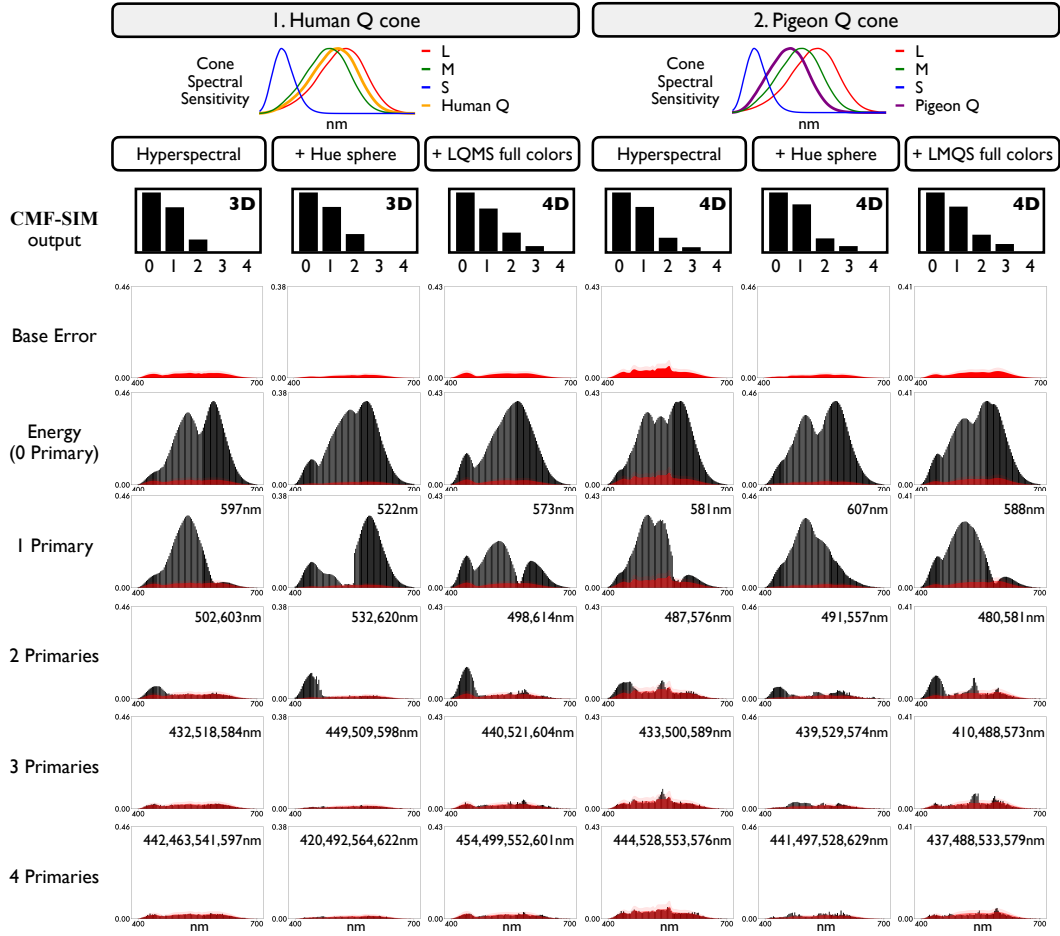


Figure 10: Results of expanded experiments of boosting color dimensionality from 3D to 4D, by addition of a Q cone to trichromatic L, M, S cones. Panel 1 models addition of a Q cone from natural human tetrachromacy, and panel 2 is for a Q cone from pigeon vision; spectral responses are shown. Each panel compares emergence of color vision with three different visual environments: real hyperspectral images (Arad et al., 2022); synthetic images containing tetrachromatic colors sampled from the hue sphere customized to that color observer (Lee et al., 2024); synthetic images containing idealized tetrachromatic colors with chromatic contrast between cone channels at the theoretical maximum. The simulation results show that 4D color vision only emerges for the human Q cone with the third environment, while it emerges for the pigeon Q cone under all three environments.

G FURTHER EVALUATION OF COLOR EMERGENCE IN CORTICAL MODEL

G.1 VALIDATION OF SIMULATED COLOR MATCHING FUNCTIONS AGAINST HUMAN PSYCHOPHYSICAL DATA

Our simulation produces results that are highly consistent with the psychophysical measurements of actual human subjects. Specifically, we compared the output of *CMF-SIM* with the empirical color matching function data reported in Stiles & Burch (1955). To match the experimental setup in Stiles & Burch (1955), we employed the same monochromatic color primaries at 444nm, 526nm, and 645nm. As shown in Figure 11, the CMFs resulting from color matching simulation in *CMF-SIM* closely align with the empirical color matching function curves, providing further validation of our simulated cortical model as an accurate representation of human trichromatic color vision.

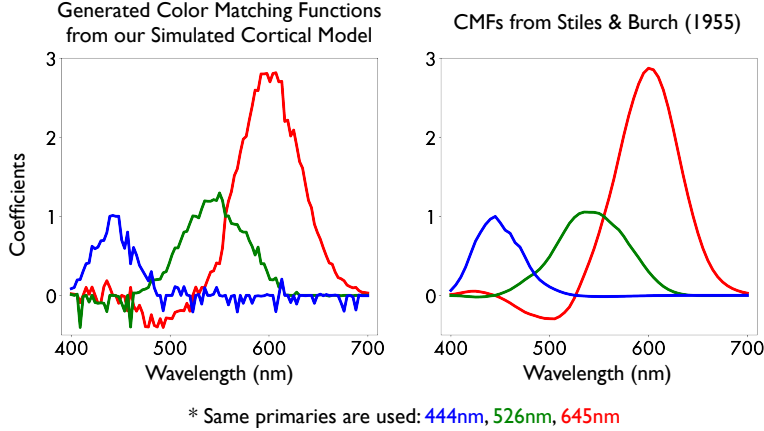


Figure 11: Side-by-side comparisons of two color-matching functions (CMFs) are presented. Left: The CMFs generated from our simulated cortical model, trained using a trichromatic retina. Right: The CMFs obtained from [Stiles & Burch \(1955\)](#), based on real human subjects. To ensure a direct comparison, our simulation used the same set of spectral primaries (i.e., 444nm, 526nm, and 645nm). This comparison highlights the validity of our trained cortical model as an accurate representation of a color observer.

G.2 ROBUST EMERGENCE OF CORRECT COLOR DIMENSIONALITY IN CORTICAL MODEL

The emergence of correct color dimensionality in our simulations remained consistent across variations in initialization, designed to mimic the natural biological variability between humans ([Carroll et al., 2002](#); [Hofer et al., 2005](#)). We tested this rigorously by varying both cone type ratios and initial noise parameters. First, we altered the L:M cone ratios in the retinal patch, testing 2:1 (original), 1:1, and 1:2, as well as an equal L:M:S ratio of 1:1:1. In all cases, the framework consistently converged to 3D color vision, demonstrating adaptability to different cone distributions. Second, we sampled the random seed for noise parameters, including: photon-shot noise during photoreceptor activation, lateral inhibition noise, Perlin noise for cell position randomization, and cortical model parameter initialization. Despite these changes, the model always converged to 3D color vision when trained with a trichromat retina, underscoring the robustness of the simulated cortical learning to biological and physical variability.

To further illustrate the robustness of our cortical model, we present its performance across a diverse range of input stimuli during testing, as shown in Figure [12](#). The figure demonstrates the Neural-Scoped internal percept of our cortical model trained with a trichromatic retina. Specifically:

1. For hyperspectral images derived from RGB inputs, converted using the method described in Section [A.1](#), the model successfully reconstructs accurate color percepts (3rd column) from optic nerve signals (2nd column), which have been time-averaged for improved visualization. Additionally, the model demonstrates complete robustness to various hue variations, accurately reconstructing color percepts even under significant changes in the chromatic properties of the input stimuli.
2. For hyperspectral images from a standard dataset ([Arad et al., 2022](#)), the model accurately reconstructs color percepts (6th column) from optic nerve signals (5th column). To aid clarity, we include RGB-projected versions of the hyperspectral images in the 4th column.

This highlights the model’s ability to generalize across varying input types and maintain robust performance under diverse conditions.

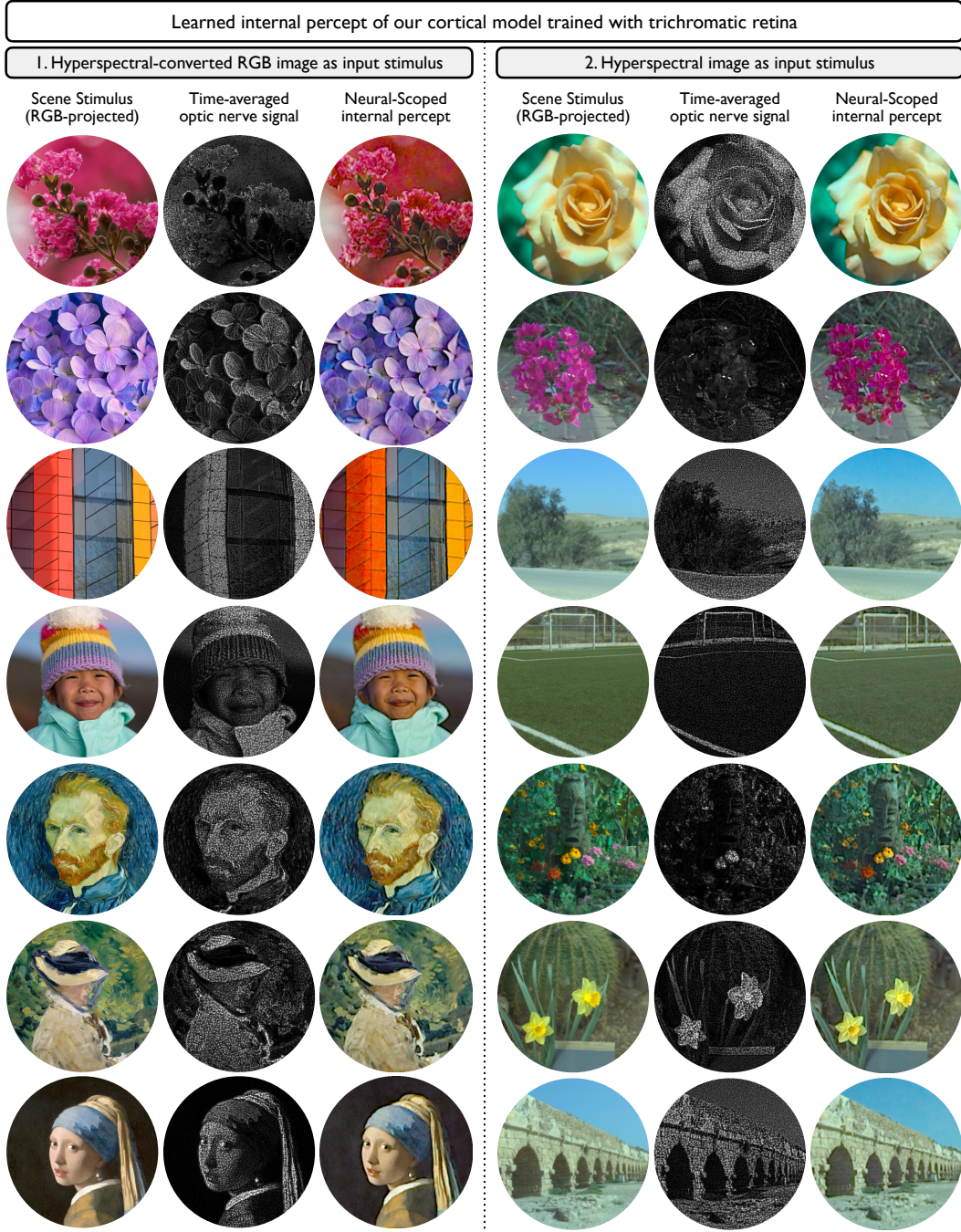


Figure 12: Our learned cortical model demonstrates robust performance across a wide variety of inputs during testing. Here, we present the neural-scoped internal percept of our cortical model trained with a trichromatic retina: 1. For hyperspectral images derived from RGB inputs (Section A.1), the model accurately reconstructs the color percept (3rd column) from optic nerve signals (2nd column, time-averaged for clearer visualization). 2. For hyperspectral images (Arad et al., 2022), the model similarly produces accurate color percepts (6th column) from optic nerve signals (5th column). For visual clarity, we show RGB-projected hyperspectral images in the 4th column.

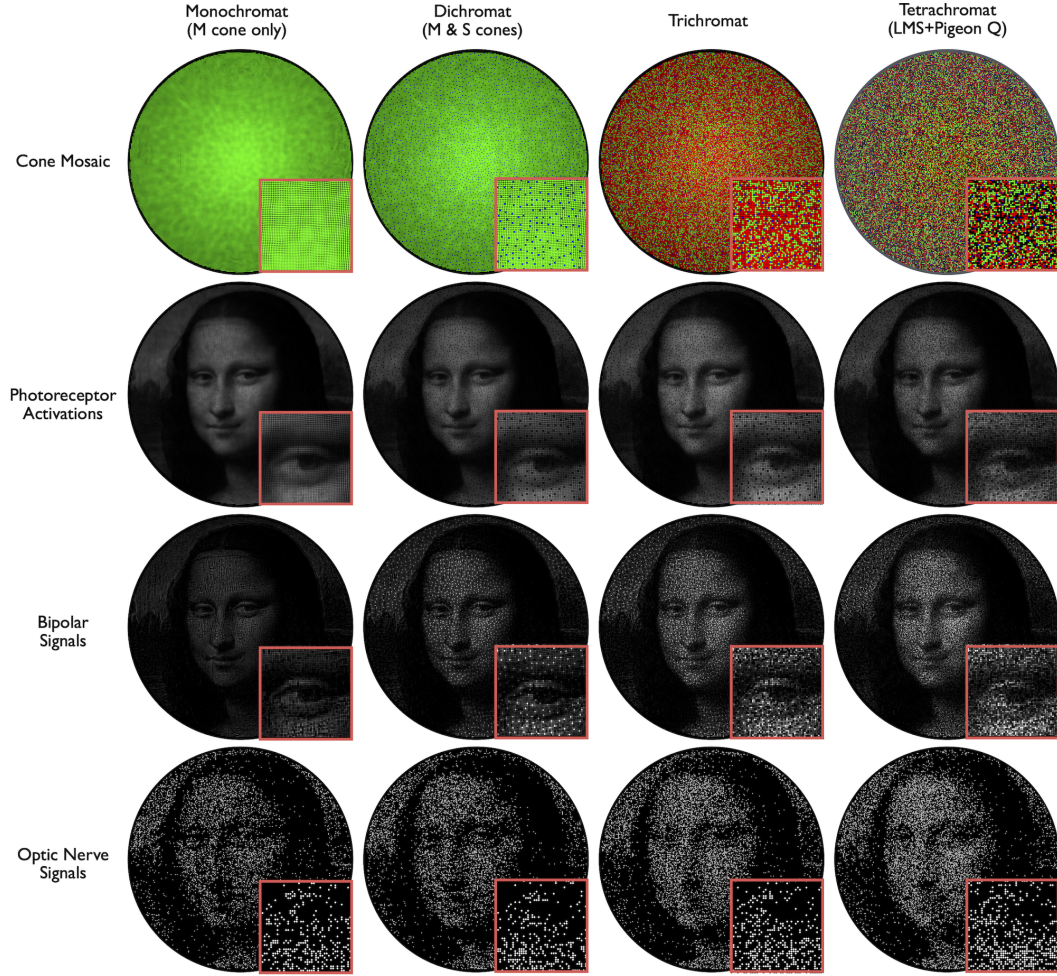


Figure 13: Expanded version of Figure 2.4, showing the full field of view with simulated retinal ganglion cell outputs (i.e., spiking optic nerve signals in the last row). Red boxes highlight close-up details of the full-sized data near the left eye of the Mona Lisa. L, M, S, and Q cones are visualized as red, green, blue, and black, respectively, in the first row.

H TABLE OF IMAGES WITH VARYING NUMBER OF CONES FOR PHOTORECEPTOR ACTIVATIONS AND RGC SPIKES

To enhance intuition and visual clarity, we present a table in Figure 13, where the number of cone types in the simulation varies across columns. The rows display images of photoreceptor activations, bipolar signals, and optic nerve signals.

The second row in Figure 13 reveals that photoreceptor activations become increasingly noisy as additional cone types are added to the retinal mosaic. The third row shows bipolar signals, computed by applying a center-surround lateral inhibition kernel to the photoreceptor activations, introducing greater complexity compared to the photoreceptor layer. By the time optic nerve signals (spikes) are generated, differences between cone mosaics become almost indistinguishable, underscoring the cortical challenge of extracting color vision with the correct dimensionality.

We also present a variant of Figure 13 that features the balloon image in Figure 14.

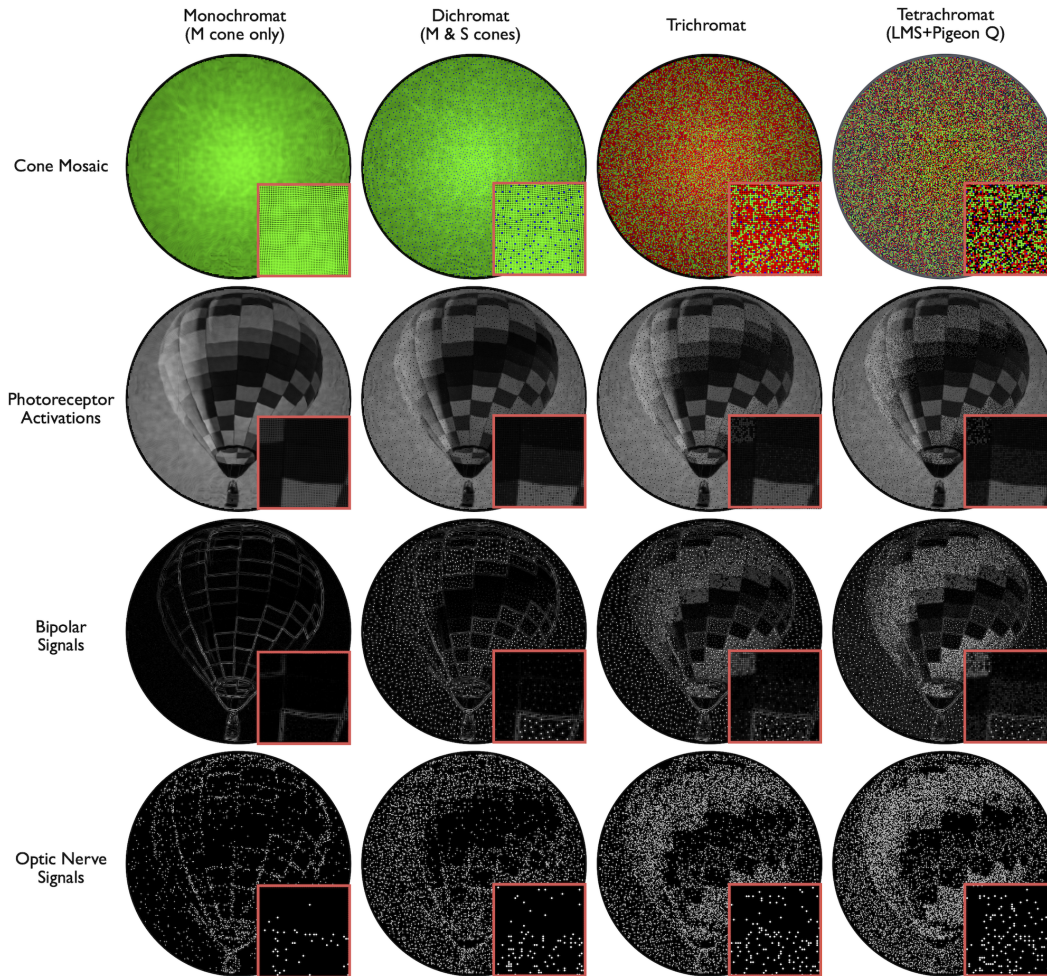


Figure 14: Expanded version of Figure 2.4, a variant of Figure 13, featuring the balloon image.