

---

# Explicitly Modeling Subcortical Vision with a Neuro-Inspired Front-End Improves CNN Robustness

---

Lucas Piper<sup>1,2</sup>, Arlindo L. Oliveira<sup>1,2</sup>, Tiago Marques<sup>3,4</sup>

<sup>1</sup>INESC-ID, Lisboa, Portugal

<sup>2</sup>Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

<sup>3</sup>Breast Cancer Research Program, Champalimaud Foundation, Lisboa, Portugal

<sup>4</sup>Faculdade de Medicina de Lisboa, Universidade de Lisboa, Portugal

lucaspiper99@tecnico.ulisboa.pt

## Abstract

Convolutional neural networks (CNNs) trained on object recognition achieve high task performance but continue to exhibit vulnerability under a range of visual perturbations and out-of-domain images, when compared with biological vision. Prior work has demonstrated that coupling a standard CNN with a front-end (VOneBlock) that mimics the primate primary visual cortex (V1) can improve overall model robustness. Expanding on this, we introduce Early Vision Networks (EVNets), a new class of hybrid CNNs that combine the VOneBlock with a novel Subcortical-Block, whose architecture draws from computational models in neuroscience and is parameterized to maximize alignment with subcortical responses reported across multiple experimental studies. Without being optimized to do so, the assembly of the SubcorticalBlock with the VOneBlock improved V1 alignment across most standard V1 benchmarks, and better modeled extra-classical receptive field phenomena. In addition, EVNets exhibit stronger emergent shape bias and outperform the base CNN architecture by 9.3% on an aggregate benchmark of robustness evaluations, including adversarial perturbations, common corruptions, and domain shifts. Finally, we show that EVNets can be further improved when paired with a state-of-the-art data augmentation technique, surpassing the performance of the isolated data augmentation approach by 6.2% on our robustness benchmark. This result reveals complementary benefits between changes in architecture to better mimic biology and training-based machine learning approaches.<sup>1</sup>

## 1 Introduction

Convolutional neural networks (CNNs) have achieved remarkable performance across a range of object recognition benchmarks [1, 2, 3, 4, 5], yet they remain vulnerable when faced with common corruptions [6], domain shifts [7, 8, 9], and adversarial perturbations [10, 11]. These vulnerabilities not only limit deployment in real-world settings but also underscore fundamental disparities between computer vision models and primate vision [7, 8]. In response, recent work has introduced biologically inspired models that integrate neuroscientific computations into CNN pipelines [12, 13, 14, 15, 16]. A prominent example is the VOneNet family [12], which improves adversarial and corruption robustness by combining a biologically constrained front-end — the VOneBlock — to conventional CNN architectures. This block simulates processing in primate primary visual cortex (V1) via a fixed-weight, empirically constrained Gabor filter bank (GFB), nonlinearities reflecting responses of V1 simple and complex cells, and a neural noise generator. While VOneNets represent a key

---

<sup>1</sup>Code and model weights available at <https://github.com/lucaspiper99/evnet/>.

step towards neurally-aligned vision models, they abstract away the hierarchical processing in the early visual system, notably omitting subcortical circuits such as the retina and lateral geniculate nucleus (LGN). This raises the question of whether refining upstream processing to V1, by explicitly modeling subcortical processing, yields further gains in robustness and alignment with biology. To address this question, we present the following key contributions:

- We introduce a novel fixed-weight CNN front-end called the SubcorticalBlock designed to capture key computations in the retina and the LGN. This module is instantiated from neuroscientific models and is explicitly parameterized to produce responses aligned with a broad set of experimentally observed subcortical response properties.
- We introduce Early Vision Networks (EVNets), a new class of hybrid CNNs that combines two biologically-grounded modules, the VOneBlock with the new SubcorticalBlock, as a multi-stage front-end for a standard CNN architecture.
- We show that without any explicit optimization for V1 predictivity, EVNets improve neural and behavioral alignment with primate vision, outperforming both standard CNNs and VOneNets across multiple benchmarks. In particular, EVNets capture extra-classical RF phenomena more accurately, increase V1 tuning property alignment, and better mimic human inductive biases through a stronger emergence of shape bias.
- We demonstrate that EVNets deliver enhanced robustness across a diverse battery of evaluations, including adversarial attacks, common image corruptions, and domain shifts, and that these gains generalize to other architecture back-ends.
- We show that EVNets trained with a state-of-the-art (SOTA) data augmentation technique yield additive improvements in robustness, highlighting the complementary effects of architectural priors and training-based strategies.

## 1.1 Related Work

**Modeling subcortical vision.** The Difference-of-Gaussian (DoG) model emerged as the foundational linear framework for characterizing spatial summation over the receptive field (RF) of subcortical cells [17, 18]. Subsequent extensions modeled extra-classical RF properties, including contrast gain control and surround suppression [19, 20]. Building on this, divisive normalization [21, 22] and cascading linear-nonlinear (LN) models [23] improved subcortical predictivity by incorporating the interaction between different visual processing stages. More recently, CNNs outperformed prior models in predicting subcortical responses to visual stimuli [24].

**Applications of subcortical vision.** Beyond modeling subcortical vision, DoG-based filtering emerged as a solution to edge detection [25], while Retinex theory [26] and lightness models [27] used spatial normalization to achieve color constancy. Subsequent frameworks such as scale-space representations [28] and multiscale Laplacian pyramids [29] generalized these computations into hierarchical contrast and boundary encoding. Recent work has reintroduced these ideas into deep architectures, by embedding explicit center-surround pathways for illumination-robust classification [30] and by unrolling Retinex-inspired optimization within CNNs for low-light image enhancement [31].

**Improving perturbation robustness.** CNN robustness improvements have largely been driven by data augmentation techniques [32, 33, 34]. To this end, standard benchmarks evaluate models under image corruptions [6] and alternative renditions [7, 8, 33, 35, 36]. Recent work highlights that composing augmentations [32, 33, 37], especially when integrated with architectural changes [14], forwards the SOTA under this regime. Notably, PRIME augmentation [34] samples semantically-aligned transformations from maximum entropy distributions. In parallel, the vulnerability of CNNs to white-box adversarial perturbations has catalyzed extensive research [10, 38, 39] with adversarial training emerging as the dominant paradigm for improving robustness [10].

**Measuring alignment with primate vision.** A growing suite of metrics has emerged to quantify the alignment between models and the primate vision [40, 41, 42, 43]. Metrics such as shape bias [8] have been instrumental in measuring model-human behavioral alignment, while a parallel line of research emphasizes representational alignment through the comparison of model tuning properties to those observed in neural data [40, 41, 44]. Complementing these efforts, the BrainScore platform [42,

43] provides a unified benchmark that integrates neural recordings and behavioral data across multiple visual cortical areas, including V1, V2, V4, and IT, alongside behavioral and task-driven metrics.

**Building neuro-inspired models.** Introducing biological computations into CNNs has consistently enhanced their robustness to input perturbations [12, 14, 15, 16, 45]. Gains have been observed with convolutional layers aligned to early visual RFs [15, 16], push-pull inhibition motifs inspired by V1 [14, 45], and by introducing the VOneBlock [12, 13]. Besides improving adversarial and corruption robustness, VOneNets achieved improved accuracy in V1 predictivity, and, when combined with V1 divisive normalization, further sharpened alignment with V1 response properties and corruption robustness [13, 44]. Finally, the systematic composition of hallmark V1 computations into CNNs recently achieved SOTA performance on explaining V1 predictivity and tuning properties [46].

## 2 Methods

Inspired by the hierarchical organization of early visual processing culminating in V1, we introduce EVNets, a new family of neuro-inspired CNNs that build upon the VOneNet framework. EVNets incorporate modular fixed-weight front-ends that reflects the functional stages of the early primate visual pathway (Fig. 1). This architecture comprises three key components: the SubcorticalBlock, modeling response characteristics of foveal neurons in the retina and the LGN; a variant of the VOneBlock, which models classical RF properties of V1 neurons; and a standard CNN back-end architecture. Together, the front-end blocks instantiate a composite cascading LN model of early cortical vision. EVNets adopt a spike-count formulation, abstracting over temporal dynamics, focusing solely on spatial encoding. The EVNet operates over a 7deg field-of-view (FoV), reduced from the 8deg used in the original VOneBlock [12]. We also extend the GFB to higher spatial frequencies (SFs), improving alignment with the SF tuning properties of primate V1 [47] (cf. Supplementary Material A for an overview of VOneNets and our modifications). We trained three random seeds for both the VOneNet and the EVNet model families using a ResNet50 [4] back-end architecture. All models were trained on the ImageNet-1k dataset [1], with clean accuracy evaluated on the standard validation split. Additional training details are provided in Supplementary Material E.4.

### 2.1 Architecture

The SubcorticalBlock simulates spatial summation over the RF of parvocellular (P) and magnocellular (M) cells, processing them as separate parallel pathways. To account for both classical and extra-classical properties, each pathway comprises a light adaptation stage, a DoG convolutional layer, a contrast normalization stage, and a noise generator that simulates subcortical noise statistics.

**DoG convolution.** Spatial summation over the RF and center-surround antagonism is modeled by incorporating a set of DoG filters described by

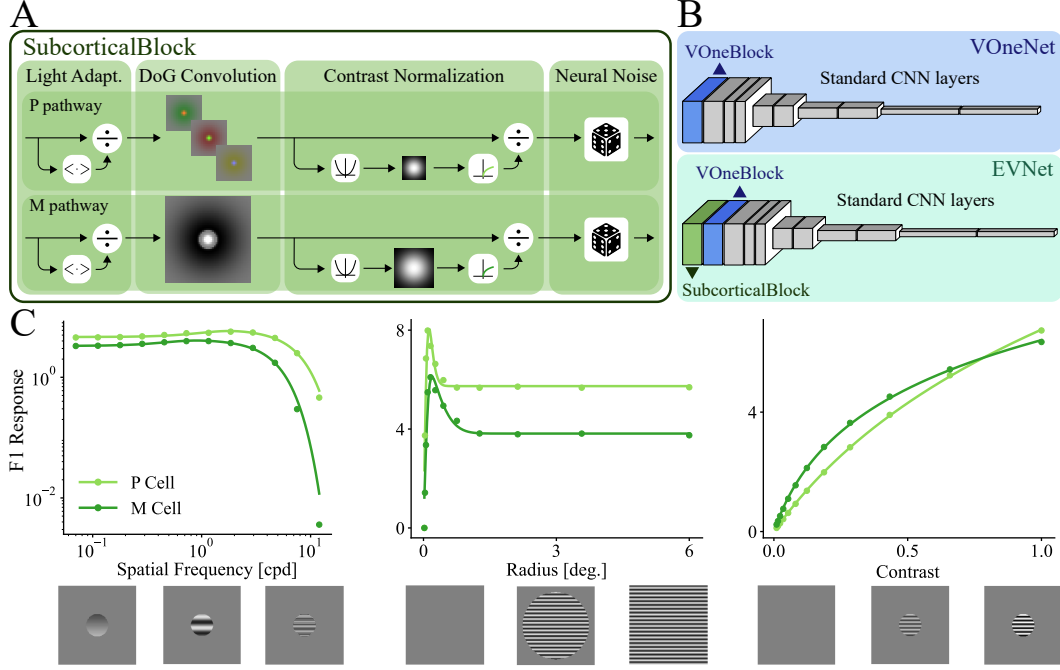
$$w_{\text{DoG}}(x, y) = \exp\left(-\frac{x^2 + y^2}{r_c^2}\right) - (k_s/k_c) \exp\left(-\frac{x^2 + y^2}{r_s^2}\right), \quad (1)$$

where  $r_c$  and  $r_s$  are the center and surround radii, and  $k_s/k_c$  is the peak contrast sensitivity ratio. We simulate biological color-opponent pathways characteristic of the different types of cells, with the P-cell stream incorporating red-green, green-red, and blue-yellow opponency, whereas M cells reflect achromaticity by incorporating a DoG filter with no color tuning [48, 49].

**Light adaptation.** To mimic the subcortical mechanism of light adaptation [21, 23, 50], we introduce a biologically inspired module that performs global luminance normalization. This transformation modulates the input  $\mathbf{x}$  as

$$\mathbf{x}_{\text{LA}} = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\bar{\mathbf{x}}}, \quad (2)$$

where  $\mathbf{x}_{\text{LA}}$  is the light-adapted input and the  $\bar{\mathbf{x}}$  denotes the average pixel intensity across channels and spatial dimensions of the input, ensuring a null output for pixel values matching the global mean.



**Figure 1: Simulating primate early visual processing as CNN front-end blocks.** **A** The SubcorticalBlock integrates two parallel processing pathways for P and M cells with a light-adaptation layer, a DoG convolutional layer, a contrast-normalization layer and a neural noise generator. **B** Both VOneNets and EVNets comprise an initial block designed to simulate a specific stage of the visual system, followed by a standard CNN architecture. VOneNets include a VOneBlock and EVNets include both a SubcorticalBlock and a VOneBlock. **C** SF, size, and contrast tuning curves (left to right) for two example subcortical neurons with example frames from the drifting gratings stimulus set shown below. Markers indicate the F1 component of the cell response and the solid line depicts the fitted response functions used for parameterizing response properties (cf. Supplementary Material E.2). Notably, the SubcorticalBlock exhibits hallmark LGN phenomena, including contrast saturation and surround suppression, with stronger modulation observed in M cells.

**Contrast normalization.** To model the adaptive gain control mechanisms characteristic of early visual processing [21, 22, 23, 51, 52], we introduce a contrast normalization stage that normalizes activations by a local estimate of stimulus contrast. The normalized response is computed by

$$\mathbf{x}_{\text{CN}} = \frac{\mathbf{x}_{\text{DoG}}}{(c_{50} + \sqrt{\mathbf{x}_{\text{DoG}}^2 * \mathbf{w}_{\text{CN}}})^n}, \quad (3)$$

where  $\mathbf{x}_{\text{DoG}}$  denotes the pre-normalized activation,  $\mathbf{w}_{\text{CN}}$  is a Gaussian kernel defining the contrast integration pooling window,  $c_{50}$  is a semi-saturation constant controlling sensitivity, and  $n$  governs the strength of the nonlinearity.

**Push-pull pattern.** The canonical push-pull pattern, emerging from the antagonistic interaction between ON- and OFF-center cells in early visual circuits, can be functionally approximated by subtracting rectified responses of opposite polarity pathways [53, 54]. Assuming ON and OFF cells of the same type share identical spatial profiles and gain, their RFs differ only in polarity. Under this assumption, since Equation 3 is antisymmetric with respect to  $\mathbf{x}_{\text{DoG}}$ , the contrast-normalized responses of the ON and OFF pathways,  $\mathbf{x}_{\text{CN}}^{\pm}$ , would satisfy  $\mathbf{x}_{\text{CN}}^+ = -\mathbf{x}_{\text{CN}}^-$ . Thus, applying rectification to both signals and computing their differences gives

$$\max(\mathbf{x}_{\text{CN}}^+, 0) - \max(-\mathbf{x}_{\text{CN}}^+, 0) = \mathbf{x}_{\text{CN}}^+. \quad (4)$$



Accordingly, our implementation bypasses explicit rectification and subtraction steps, instead operating directly on the signed contrast-normalized signal to improve computational efficiency without sacrificing functional fidelity.

**Noise Generator.** Neuronal responses in the primate visual system exhibit trial-to-trial variability with distinct stochastic signatures across processing stages. In V1, spike count variability closely follows a Poisson distribution, where the variance is equal to the mean, corresponding to a Fano Factor equal to one [55]. Conversely, subcortical neurons exhibit sub-Poisson variability, characterized by a spike count variance lower than the mean [56, 57]. To faithfully capture this hierarchical structure of neural noise, we implement a dual-source noise injection mechanism in which we add independent Gaussian noise to each unit of both front-end blocks scaled accordingly. This noise is calibrated at the unit level to maintain an overall variability with a unit Fano factor, while trial-to-trial activations at the VOneBlock output exhibit heteroskedasticity consistent with V1 measurements [55]. Prior to noise injection, unit activations are linearly rescaled such that their mean response to a stimulus aligns with the empirically observed spike count of the corresponding primate neuronal population over a 50-ms integration window [12, 56].

## 2.2 Subcortical-Aligning Parameterization

Despite the wealth of empirical data on primate LGN and the existence of various fitted models [21, 22, 23, 51, 52], the heterogeneity of modeling approaches across studies limits the direct reuse of parameters in the SubcorticalBlock, while complicating synthesis from the broader literature. To address this, we introduce a novel neurophysiologically-constrained hyperparameter tuning strategy designed to produce responses that best match the mean neuronal response properties of an LGN neuronal population taken from prior studies and modeling strategies. Specifically, we selected a total of  $N = 6$  different response property distributions measured at foveal LGN to ensure alignment in SF tuning, size tuning, and contrast sensitivity. The individual properties are: center, surround, excitation and inhibition radii [18, 58, 59], suppression index [58] and saturation index [51].

We conducted a series of *in silico* experiments, presenting each cell with drifting gratings, quantifying each response property through the first harmonic (F1) of the cell’s response. We then performed hyperparameter search via Bayesian optimization [60], minimizing the loss

$$\mathcal{L} = \sum_{i=1}^N \left[ \log_2 \left( \frac{R_i(\mathbf{f}_{1,i})}{\bar{r}_i} \right) \right]^2. \quad (5)$$

For each response property  $i$ ,  $\bar{r}_i$  is the mean of the empirical response property distribution,  $\mathbf{f}_{1,i}$  is a vector of F1 responses produced by the SubcorticalBlock cell when subjected to the experiment-specific stimulus set, and  $R_i$  maps the F1 responses to the response property. This mapping often involved an intermediate model-fitting step, with the response property computed from the fitted parameters. Figure 1C shows the F1 responses of both P and M cells, example frames of the stimulus set used for each experiment and response model curves obtained at convergence (cf. Supplementary Material E.2 for in-depth description of experiment methodology).

## 2.3 Model Evaluation

**Alignment with primate vision.** Shape bias was measured using a cue-conflict dataset [8] that combines shapes and textures of ImageNet samples using style transfer [61]. To quantify the correspondence between model internal representations and V1 responses, we assessed the activations from the first block of each model using two complementary BrainScore [42, 43] benchmarks: V1 neural predictivity [41] and V1 response property [40]. V1 neural predictivity evaluates the degree to which model features can account for the variance in primate V1 responses via partial least squares (PLS) regression. In contrast, V1 response property quantifies RF tuning similarity by comparing the distributions of 22 response properties, extracted from the same first-block activations, to empirical V1 distributions. These properties span 7 functional categories: orientation and SF tuning, response selectivity, RF size, surround modulation, texture modulation, and response magnitude.

**Robustness evaluation.** To quantify robustness, we report a mean Robustness Score, defined as the mean top-1 accuracy across a diverse set of common corruptions, adversarial attacks, and domain

shifts. To assess whether EVNets offer complementary gains to SOTA robustness training methods, we trained a standard ResNet50, a VOneResNet50 and an EVResNet50 using PRIME [34]. We included adversarial training with an  $L_\infty$  constraint of  $\|\delta\|_\infty = 4/255$  [62] ( $AT_{L_\infty}$ ) as a baseline.

**Image corruptions.** We evaluated model corruption robustness by measuring top-1 accuracy on the ImageNet-C dataset [6], which comprises 75 distinct corrupted variants of the ImageNet validation set. These corruptions are organized into 15 types applied at five severity levels, reflecting a specific real-world image degradation. The corruption types are further grouped into four broad categories: noise, blur, weather, and digital perturbations.

**Domain shifts.** To assess each model’s generalization under domain shifts, we averaged top-1 accuracies across five ImageNet-derived datasets that focus on renditions partially addressable by early visual processing mechanisms. Specifically, we used ImageNet-R [33], ImageNet-Cartoon [35], ImageNet-Drawing [35], ImageNet-Sketch [36], and the 16-class Stylized-ImageNet [7, 8] (Stylized<sub>16</sub>-ImageNet). Each dataset introduces representational changes such as abstraction, stylization, or domain-specific distortions while preserving the underlying semantic structure.

**Adversarial attacks.** Following Dapello et al. [12], we evaluated robustness to white-box adversarial attacks by applying untargeted Projected Gradient Descent (PGD) [10] on 5000 images of the ImageNet validation set. Attacks were carried out under  $L_\infty$ ,  $L_2$  and  $L_1$  norm constraints and the perturbation budgets used were  $\|\delta\|_\infty \in [1/1020, 1/255, 4/255, 16/255]$ ,  $\|\delta\|_2 \in [0.15, 0.6, 1.2, 2.4]$  and  $\|\delta\|_1 \in [40, 160, 640, 2560]$ . We used 64 PGD iterations with step size of  $\|\delta\|_p/32$ . Additional implementation details are provided in Supplementary Material C.1.

**EVNet variants.** We trained seven EVResNet50 variants derived from the full EVNet by performing six targeted ablations and one architectural addition (two seeds each). The ablations individually removed the P- and M-cell pathways, the contrast normalization and light adaptation layers, the VOneBlock, and the subcortical noise generator. When removing subcortical noise, cortical noise in the VOneBlock output was amplified to remain Poisson-distributed, consistent with the original VOneNet [12]. The final variant introduced an LGN–V2 skip connection [63] by concatenating the SubcorticalBlock output with the VOneNet bottleneck. While all EVResNet50 variants were tested for primate vision alignment and robustness, adversarial evaluations were restricted to a reduced attack set including only the two weakest perturbation strengths of each norm constraint.

**Additional experiments.** To test whether improvements in adversarial robustness were not due to gradient masking [38], we performed a battery of controls according to the best practices [38, 39, 64] (cf. Supplementary Material C.1). We further evaluated the generalization of EVNet front-ends across back-end architectures by integrating them with EfficientNet-B0 [5] and CORnet-Z [65] (cf. Supplementary Material D.2). Finally, we assessed performance gains from multi-pass ensemble inference in Supplementary Material D.3.

## 3 Results

### 3.1 EVNets Improve Neuronal and Behavioral Alignment with Primate Vision

We evaluated whether coupling the SubcorticalBlock with the VOneBlock improved alignment with primate V1. As illustrated in Figure 2, the inclusion of the SubcorticalBlock upstream of the VOneBlock introduces hallmark extra-classical RF response properties absent from the VOneBlock alone. In particular, we observe an increased surround suppression in the size tuning curve and a non-linear contrast-sensitivity curve.

Motivated by these observations, we used the BrainScore platform [42, 43] to quantitatively evaluate V1 alignment. As shown in Table 1, while VOneNets outperform in V1 predictivity, EVResNet50 achieves a higher mean response property score than both the ResNet50 and VOneResNet50 models. Notably, the highest gains are observed for the surround modulation and RF size tuning, both associated with the increased surround suppression. A tradeoff in SF tuning is observed, which can be attributed to the fact that no changes were done to the GFB to account for the upstream processing.

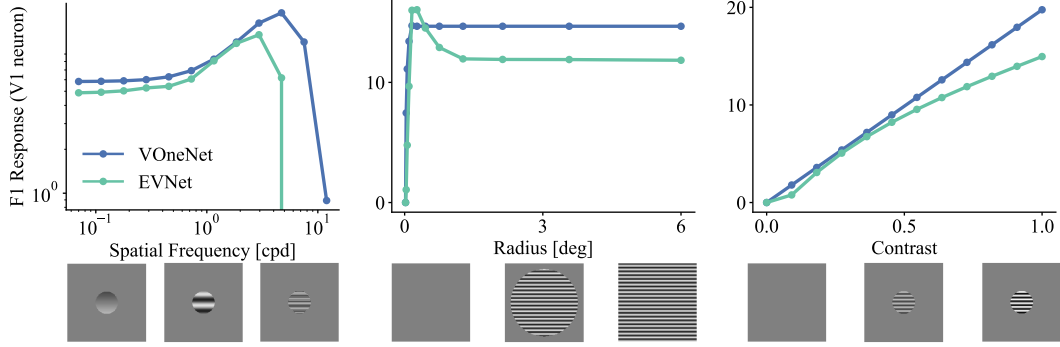


Figure 2: **Subcortical preprocessing improves explainability of extra-classical RF properties in V1.** SF, size, and contrast tuning curves (left to right) for an example neuron in the VOneBlock with and without subcortical preprocessing (M cell). Example frames from the drifting gratings stimuli are shown below. VOneBlock neurons in isolation exhibit predominantly classical RF effects but when coupled with subcortical processing exhibit behaviors consistent with those empirically observed, such as enhanced surround modulation and non-linear contrast responses[66, 67]. See Supplementary Material B.1 for empirical V1 tuning curves.

Table 1: **EVResNet50 outperforms baselines on mean V1 response property alignment, and shape bias.** BrainScore [42, 43] V1 alignment scores and shape bias [8] for ResNet50, VOneResNet50, and EVResNet50. Values indicate mean  $\pm$  SD ( $n = 3$  seeds).

Model	V1 Predict.	V1 Resp. Prop.	V1 Response Properties							
			Orient. Tuning	SF Tuning	RF Size	Surround Mod.	Texture Mod.	Resp. Select.	Resp. Magn.	Shape Bias [%]
ResNet50	.271 $\pm .002$	.637 $\pm .008$	.822 $\pm .027$	.754 $\pm .026$	.214 $\pm .002$	.389 $\pm .023$	.792 $\pm .028$	.621 $\pm .010$	.865 $\pm .012$	18.8 $\pm 1.2$
VOneResNet50	<b>.375</b> $\pm .002$	.754 $\pm .006$	<b>.859</b> $\pm .005$	<b>.969</b> $\pm .001$	.482 $\pm .041$	.373 $\pm .003$	<b>.919</b> $\pm .004$	<b>.792</b> $\pm .003$	.884 $\pm .002$	31.6 $\pm 1.2$
<b>EVResNet50</b>	.364 $\pm .000$	<b>.826</b> $\pm .000$	.854 $\pm .009$	.950 $\pm .000$	<b>.726</b> $\pm .001$	<b>.614</b> $\pm .004$	.916 $\pm .001$	.781 $\pm .000$	<b>.933</b> $\pm .000$	<b>48.9</b> $\pm 2.4$

Among the suit of primate visual behavior alignment metrics, shape bias has emerged as a particularly informative proxy of human-aligned inductive biases and out-of-domain (OOD) generalization [7, 8]. Motivated by the hypothesis that shape bias may originate in early visual computations, we evaluated shape bias in EVNets (Tab. 1) and observed a substantial increase of 30.1% relative to the standard ResNet50 and of 17.3% relative to the VOneNet model, suggesting that EVNets may confer not only improved neuronal alignment but also behavioral traits more consistent with primate perception.

### 3.2 EVNets Improve Robustness Across an Aggregate Benchmark

When tested on clean images (Tab. 2), our VOneNets variant achieves a 1.2% improvement over the original VOneNet [12], while EVNets displayed a performance drop of 1.3%, when compared to the same reported accuracy. We further examined whether the increased alignment of EVNets with primate vision, also leads to improved robustness.

**Image corruptions.** Across most corruption categories and in terms of mean corruption accuracy, EVNets consistently outperformed both VOneNets and the base ResNet50, which only retained an advantage in weather corruptions. Notably, the most pronounced gains were observed on noise corruptions, where EVNets outperformed VOneNets by 3.7% while effectively preserving the same cumulative Fano factor as VOneNets (cf. Fig. C4 for accuracy across individual corruptions).

**Domain shifts.** Table 3 summarizes our OOD generalization results. EVNets consistently outperforms both the baseline ResNet50 and VOneNets across the majority of benchmarks, also surpassing these baselines on the mean domain shift accuracy. The only dataset where EVNets underperform is ImageNet-Sketch, where the base ResNet50 exhibits a marginal advantage.

Table 2: **EVResNet50 outperforms baselines on most image corruption types and on mean corruption accuracy.** Clean and corrupted top-1 accuracies averaged across severities and corruptions for ResNet50, VOneResNet50 and EVResNet50. Values indicate mean  $\pm$  SD ( $n = 3$  seeds).

Model	Mean [%]	Corruption Types				Clean [%]
		Noise [%]	Blur [%]	Weather [%]	Digital [%]	
ResNet50	38.8 $\pm$ 0.5	29.2 $\pm$ 0.6	34.6 $\pm$ 0.4	<b>36.1<math>\pm</math>0.5</b>	49.5 $\pm$ 0.6	<b>75.4<math>\pm</math>0.1</b>
VOneResNet50	40.4 $\pm$ 0.1	35.9 $\pm$ 0.5	34.8 $\pm$ 0.1	32.6 $\pm$ 0.1	52.2 $\pm$ 0.1	72.9 $\pm$ 0.1
<b>EVResNet50</b>	<b>41.9<math>\pm</math>0.2</b>	<b>39.6<math>\pm</math>0.3</b>	<b>37.5<math>\pm</math>0.1</b>	30.6 $\pm$ 0.2	<b>53.5<math>\pm</math>0.1</b>	70.4 $\pm$ 0.1

Table 3: **EVResNet50 outperforms baselines on most domain shift datasets and on overall mean domain shift accuracy.** Top-1 accuracies on ImageNet-{Cartoon, Drawing, R, Sketch, Stylized<sub>16</sub>} for ResNet50, VOneResNet50 and EVResNet50. Values indicate mean  $\pm$  SD ( $n = 3$  seeds).

Model	Mean [%]	Cartoon [%]	Drawing [%]	R [%]	Sketch [%]	Stylized <sub>16</sub> [%]
ResNet50	33.4 $\pm$ 0.2	51.2 $\pm$ 0.7	20.9 $\pm$ 0.6	35.4 $\pm$ 0.1	<b>23.3<math>\pm</math>0.1</b>	36.3 $\pm$ 1.2
VOneResNet50	37.1 $\pm$ 0.4	55.5 $\pm$ 0.2	30.5 $\pm$ 0.4	37.5 $\pm$ 0.1	23.1 $\pm$ 0.3	<b>38.8<math>\pm</math>1.1</b>
<b>EVResNet50</b>	<b>38.1<math>\pm</math>0.3</b>	<b>57.1<math>\pm</math>0.3</b>	<b>33.9<math>\pm</math>0.2</b>	<b>38.1<math>\pm</math>0.2</b>	22.7 $\pm$ 0.2	38.6 $\pm$ 1.1

**Adversarial attacks.** EVNets improves adversarial robustness across most perturbation norms and attack strengths when compared to VOneNets (Tab. 4). While VOneNets obtained a marginal advantage under the weakest attack strengths for  $L_\infty$  and  $L_2$  norm constraints, EVNets improved robustness on the remaining attack settings, exhibiting also a smaller gap between clean and adversarial accuracy.

Table 4: **EVResNet50 outperforms baselines on most adversarial perturbations and on mean adversarial robustness.** Top-1 accuracies for the ResNet50, VOneResNet50 and EVResNet50 models. Values indicate mean  $\pm$  SD ( $n = 3$  seeds).

Model	Mean [%]	$\ \delta\ _\infty$				$\ \delta\ _2$				$\ \delta\ _1$			
		$\frac{1}{1020}$ [%]	$\frac{1}{255}$ [%]	$\frac{4}{255}$ [%]	$\frac{16}{255}$ [%]	0.15 [%]	0.6 [%]	2.4 [%]	9.6 [%]	40 [%]	160 [%]	640 [%]	2560 [%]
ResNet50	8.3 $\pm$ 0.2	23.4 $\pm$ 0.8	0.4 $\pm$ 0.0	0.2 $\pm$ 0.0	<b>0.2</b> $\pm$ 0.0	37.2 $\pm$ 1.0	1.8 $\pm$ 0.2	0.2 $\pm$ 0.0	<b>0.2</b> $\pm$ 0.0	33.6 $\pm$ 0.7	1.7 $\pm$ 0.2	0.2 $\pm$ 0.0	<b>0.2</b> $\pm$ 0.0
VOneResNet50	26.1 $\pm$ 0.1	62.6 $\pm$ 0.4	30.4 $\pm$ 0.3	1.2 $\pm$ 0.1	0.0 $\pm$ 0.0	<b>66.2</b> $\pm$ 0.3	42.3 $\pm$ 0.2	4.2 $\pm$ 0.1	0.0 $\pm$ 0.0	<b>64.5</b> $\pm$ 0.3	37.3 $\pm$ 0.6	3.9 $\pm$ 0.2	0.0 $\pm$ 0.0
<b>EVResNet50</b>	<b>28.3</b> $\pm$ 0.2	<b>62.7</b> $\pm$ 0.2	<b>38.8</b> $\pm$ 0.6	<b>3.0</b> $\pm$ 0.1	0.0 $\pm$ 0.0	65.1 $\pm$ 0.0	<b>48.0</b> $\pm$ 0.3	<b>7.4</b> $\pm$ 1.8	0.0 $\pm$ 0.0	64.0 $\pm$ 0.2	<b>44.5</b> $\pm$ 0.4	<b>6.0</b> $\pm$ 1.8	0.0 $\pm$ 0.0

Table 5 summarizes all the robustness results described above and presents the Robustness Score for the evaluated models. The EVResNet50 model improves the Robustness Score by 9.3% over the base ResNet50 and by 1.6% over our VOneNet variant.

### 3.3 Combining EVNets with Data Augmentation Provides Cumulative Gains

In our evaluation of the Robustness Score for the EVResNet50 with PRIME data augmentation (Tab. 5), we find that this combined strategy yields cumulative performance gains beyond those of either component alone. Furthermore, when comparing these results with those obtained by augmenting a ResNet50 with adversarial training and augmenting a VOneNet with PRIME, no configuration surpasses the additive gains achieved by the EVResNet50 trained with PRIME.

### 3.4 EVNet Variants Reveal Competing Drivers of Primate Vision Alignment and Robustness

Selective ablation of specific SubcorticalBlock components affected primate vision alignment in different aspects (see Supplementary Material D.1). The components that caused a greater drop in V1

Table 5: **EVResNet50 achieves a higher Robust Score than ResNet50 and VOneResNet50 and, when coupled with PRIME, surpasses SOTA data augmentation approaches.** Robustness Score, clean and perturbed top-1 accuracies for the ResNet50, VOneResNet50 and EVResNet50 models; for ResNet50 with two data augmentation approaches:  $AT_{L_\infty}$  and PRIME; and for VOneResNet50 and EVResNet50 with PRIME. Values indicate mean  $\pm$  SD ( $n = 3$  seeds).

Model	Robust. Score [%]	Perturbations			Clean [%]
		Adversarial [%]	Corrupt. [%]	Domain Shift [%]	
ResNet50	26.8 $\pm$ 0.2	8.3 $\pm$ 0.2	39.1 $\pm$ 0.3	33.3 $\pm$ 0.1	<b>75.4<math>\pm</math>0.1</b>
VOneResNet	34.5 $\pm$ 0.1	26.1 $\pm$ 0.1	40.4 $\pm$ 0.2	37.1 $\pm$ 0.4	72.9 $\pm$ 0.0
<b>EVResNet50</b>	<b>36.1<math>\pm</math>0.2</b>	<b>28.3<math>\pm</math>0.2</b>	<b>41.9<math>\pm</math>0.2</b>	<b>38.1<math>\pm</math>0.3</b>	70.4 $\pm$ 0.1
ResNet50 + $AT_{L_\infty}$ [62]	34.4	<b>31.3</b>	32.5	39.5	62.4
ResNet50 + PRIME	36.5 $\pm$ 0.1	14.3 $\pm$ 0.2	52.6 $\pm$ 0.2	42.8 $\pm$ 0.2	<b>76.0<math>\pm</math>0.1</b>
VOneResNet50 + PRIME	42.1 $\pm$ 0.1	28.7 $\pm$ 0.3	<b>53.4<math>\pm</math>0.0</b>	<b>44.3<math>\pm</math>0.3</b>	74.0 $\pm$ 0.1
<b>EVResNet50 + PRIME</b>	<b>42.7<math>\pm</math>0.2</b>	30.7 $\pm$ 0.8	53.2 $\pm$ 0.1	44.2 $\pm$ 0.1	72.0 $\pm$ 0.1

response property benchmarks were the M-cell pathway and the contrast normalization, which greatly affected the RF size and surround modulation. Surprisingly, removing either the P-cell pathway or light adaptation greatly increased shape bias.

Table 6: **Model components contribution to robustness vary greatly.** Robustness Score, clean and perturbed top-1 accuracies for all EVResNet50 variants. ResNet50 and VOneResNet50 included for reference but not in the comparison. Values indicate mean  $\pm$  SD ( $n = 2$  seeds).

Model	Robust. Score* [%]	Perturbations			Clean [%]
		Adversarial* [%]	Corrupt. [%]	Domain Shift [%]	
ResNet50	29.5 $\pm$ 0.3	16.4 $\pm$ 0.5	39.1 $\pm$ 0.3	33.3 $\pm$ 0.1	75.4 $\pm$ 0.1
VOneResNet50	42.7 $\pm$ 0.1	50.5 $\pm$ 0.1	40.4 $\pm$ 0.2	37.1 $\pm$ 0.4	72.9 $\pm$ 0.0
EVResNet50	44.6 $\pm$ 0.1	53.8 $\pm$ 0.2	41.9 $\pm$ 0.2	38.1 $\pm$ 0.3	70.5 $\pm$ 0.0
– P Cells	38.4 $\pm$ 0.0	46.7 $\pm$ 0.2	34.9 $\pm$ 0.1	33.6 $\pm$ 0.0	60.7 $\pm$ 0.1
– M Cells	<b>44.8<math>\pm</math>0.1</b>	54.0 $\pm$ 0.2	42.2 $\pm$ 0.1	38.2 $\pm$ 0.3	70.3 $\pm$ 0.1
– Light Adapt.	44.3 $\pm$ 0.3	<b>55.1<math>\pm</math>0.5</b>	40.4 $\pm$ 0.2	37.4 $\pm$ 0.2	69.8 $\pm$ 0.2
– Contrast Norm.	44.4 $\pm$ 0.2	53.5 $\pm$ 0.1	41.7 $\pm$ 0.4	38.0 $\pm$ 0.1	70.7 $\pm$ 0.0
– Subcort. Noise	42.6 $\pm$ 0.0	48.5 $\pm$ 0.1	41.9 $\pm$ 0.2	37.4 $\pm$ 0.1	<b>72.6<math>\pm</math>0.2</b>
– VOneBlock	44.3 $\pm$ 0.3	51.8 $\pm$ 0.5	<b>42.7<math>\pm</math>0.2</b>	38.2 $\pm$ 0.1	71.6 $\pm$ 0.0
+ LGN-V2 Connect.	<b>44.8<math>\pm</math>0.0</b>	53.9 $\pm$ 0.2	42.1 $\pm$ 0.1	<b>38.3<math>\pm</math>0.2</b>	70.7 $\pm$ 0.1

\* Computed with a reduced attack set (two perturbations per norm constraint); not comparable to Table 5.

In terms of performance, the single component that had the largest impact on both clean accuracy and robustness was the P-cell pathway (Tab. 6). On the other hand, removing the M-cell pathway or contrast normalization had no impact on clean accuracy and robustness. Light adaptation had only minor effects on specific robustness benchmarks and a small drop in clean accuracy. Removing subcortical stochasticity improved clean accuracy while decreasing adversarial robustness. Interestingly, the omission of the VOneBlock barely affected overall robustness, revealing even to be the best variant under image corruptions and an improvement in clean accuracy, accompanied by a small drop in adversarial robustness. The inclusion of LGN-V2 skip connections had little to no impact in performance across all perturbations types and clean images.

## 4 Discussion

In this work, we present a new family of neuro-inspired CNNs that not only display enhanced robustness across a broad spectrum of perturbations but also achieve stronger alignment with primate vision. Previous works have incorporated biologically inspired mechanisms such as DoG filtering [15, 16,

30], divisive normalization [13, 68], and noise injection [37] within CNN architectures. In contrast, our SubcorticalBlock introduces a principled segregation of computations into parallel P- and M-cell pathways constrained by prior empirical neurophysiological findings. The EVNet architecture introduces a modular, cascading model of V1 that incorporates stage-specific architectural priors, offering a compelling alternative to V1-inspired CNN paradigms. The V1-alignment improvements observed in EVNets are most pronounced for surround modulation and RF size. These effects likely arise from the normalization mechanisms within the SubcorticalBlock, which induce local competition and lead to extraclassical RF responses not present in VOneNets. Despite these improvements, benchmark scores for surround modulation and RF size are still relatively low, suggesting that normalization mechanisms at the V1 level, either caused by recurrent or feedback circuits, are also needed for a better alignment [13, 68]. While EVNets do not entirely solve the longstanding problem of robust generalization, our results also underscore the critical importance of subcortical processing in shaping early visual representations: while the DoG filtering improves features selectivity, performs low-pass filtering, mitigating high-frequency noise, the normalization layers promote local competition and dynamic range compression, reducing sensitivity to input perturbations. Notably, we demonstrate that meaningful improvements to V1 modeling can be achieved by exclusively refining upstream stages. The cumulative gain shown by combining the SubcorticalBlock with the VOneBlock reveals a biologically-plausible potential for compositionality in inductive biases, where each module targets distinct axes of visual invariance, together producing synergistic improvements in perturbation robustness. Specifically, while the SubcorticalBlock primarily encodes invariance to luminance and contrast, the VOneBlock focuses more on spatial and polarity invariance. Finally, we show that integrating biologically-inspired architectures with standard data augmentation techniques leads to synergistic improvements in robustness, surpassing data augmentation alone and adversarial training, the most effective methods for improving adversarial and corruption robustness. Together, these results provide further evidence that neuroscience-driven inductive biases and machine learning heuristics are not mutually exclusive, but can in fact be complementary.

While the observed robustness gains are compelling, there are also some trade-offs. Specifically, our model abstracts subcortical processing by instantiating only four channels that reflect the average spatial response profiles observed in the LGN, rather than capturing the full heterogeneity of subcortical cells. An interesting observation is that the M-cell pathway contributes marginally to downstream performance, which is in line with the classical view of M cells small contribution to the ventral stream. However, this result presents some inconsistencies with more recent literature [69]. Additionally, enhancements in perturbation robustness are consistently accompanied by modest reductions in clean image accuracy, a common tension in robustness research [12, 14]. Beyond these considerations, subsequent work could examine the possibility of initializing from neuro-inspired weights while allowing task-driven fine-tuning of the front-ends used. For V1 alignment, we kept the VOneBlock unchanged as upstream processing minimally affected its responses aside from the improved extra-classical RF effects. However, the slight drop in V1 alignment on some benchmarks likely reflects unforeseen interactions between modules. Future work could adapt the GFB to account for subcortical preprocessing and add normalization mechanisms at the V1 level to evaluate if these changes further enhance alignment and robustness. Finally, while our approach may seem to contrast the “Bitter Lesson” [70], the trajectory of neurally aligned vision tells a more nuanced story. Prior work indicates that scaling alone does not yield more brain-like representations [71], and large-scale benchmarks [72] show that neural alignment with primate visual cortex saturates with model size — architectures with stronger biological priors often align better than transformers. Our work leverages this insight by embedding early vision-inspired inductive biases to achieve the efficiency and alignment characteristic of biological vision.

**Broader impact.** This work advances computer vision by proposing a biologically grounded alternative to vision transformers [73]. By simulating early vision, we bridge the gap between human and machine vision, improving robustness, transparency, and interpretability, with potential benefits for bias reduction and accessibility. Fully replicating human vision remains challenging and calls for further research. We acknowledge that some foundational insights stem from animal studies, emphasizing the ethical responsibility to minimize harm and favor alternative approaches. Leveraging existing biological data, as done here, may reduce reliance on new animal experimentation.

## Acknowledgments

This work was supported by the project Center for Responsible AI reference no. C628696807-00454142, and by national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT) under projects UID/50021/2025 and UID/PRR/50021/2025. We thank Ajani Stewart, Joel Dapello and James J. DiCarlo for initial explorations on the architectural space of EVNets, and Kartik Pradeepan for assistance in using the BrainScore benchmarking platform.

## References

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012.
- [2] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: Computational and Biological Learning Society, 2015, pp. 1–14.
- [3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. “Going deeper with convolutions”. In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 2015, pp. 1–9.
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.
- [5] Mingxing Tan and Quoc Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 6105–6114.
- [6] Dan Hendrycks and Thomas Dietterich. “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations”. In: *International Conference on Learning Representations*. 2019.
- [7] Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. “Partial success in closing the gap between human and machine vision”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. 2021.
- [8] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.” In: *International Conference on Learning Representations*. 2019.
- [9] Robert Geirhos, Kristof Meding, and Felix A. Wichmann. “Beyond Accuracy: Quantifying Trial-by-Trial Behaviour of CNNs and Humans by Measuring Error Consistency”. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS’20*. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.
- [10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. “Towards Deep Learning Models Resistant to Adversarial Attacks”. In: *International Conference on Learning Representations*. 2018.
- [11] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *International Conference on Learning Representations*. 2015.
- [12] Joel Dapello, Tiago Marques, Martin Schrimpf, Franziska Geiger, David Cox, and James J DiCarlo. “Simulating a Primary Visual Cortex at the Front of CNNs Improves Robustness to Image Perturbations”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 13073–13087.
- [13] Andrew Cirincione, Reginald Verrier, Artiom Bic, Stephanie Olaiya, James J. DiCarlo, Lawrence Udeigwe, and Tiago Marques. “Implementing Divisive Normalization in CNNs Improves Robustness to Common Image Corruptions”. In: *SVRHM 2022 Workshop @ NeurIPS*. 2022.

- [14] Guru Swaroop Bennabhaktula, Enrique Alegre, Nicola Strisciuglio, and George Azzopardi. “PushPull-Net: Inhibition-Driven ResNet Robust to Image Corruptions”. In: *Pattern Recognition: 27th International Conference, ICPR 2024, Kolkata, India, December 1–5, 2024, Proceedings, Part VIII*. Kolkata, India: Springer-Verlag, 2024, pp. 391–408. ISBN: 978-3-031-78185-8. DOI: 10.1007/978-3-031-78186-5\_26.
- [15] Benjamin D. Evans, Gaurav Malhotra, and Jeffrey S. Bowers. “Biological convolutions improve DNN robustness to noise and generalisation”. In: *Neural Networks* 148 (2022), pp. 96–110. ISSN: 0893-6080.
- [16] Akhilesh Adithya, Basabdatta Sen Bhattacharya, and Michael Hopkins. “Robustness of Biologically-Inspired Filter-Based ConvNet to Signal Perturbation”. In: *Artificial Neural Networks and Machine Learning – ICANN 2023*. Ed. by Lazaros Iliadis, Antonios Papaionidas, Plamen Angelov, and Chrisina Jayne. Cham: Springer Nature Switzerland, 2023, pp. 394–406. ISBN: 978-3-031-44204-9.
- [17] Stephen W. Kuffler. “Discharge Patterns And Functional Organization Of Mammalian Retina”. In: *Journal of Neurophysiology* 16.1 (1953). PMID: 13035466, pp. 37–68.
- [18] R.W. Rodieck. “Quantitative analysis of cat retinal ganglion cell response to visual stimuli”. In: *Vision Research* 5.12 (1965), pp. 583–601. ISSN: 0042-6989.
- [19] R M Shapley and J 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.
- [20] Samuel G. Solomon, Barry B. Lee, and Hao Sun. “Suppressive Surrounds and Contrast Gain in Magnocellular-Pathway Retinal Ganglion Cells of Macaque”. In: *Journal of Neuroscience* 26.34 (2006), pp. 8715–8726. ISSN: 0270-6474.
- [21] Matteo Carandini and David J. Heeger. “Normalization as a canonical neural computation”. In: *Nature Reviews Neuroscience* 13.1 (Jan. 2012), pp. 51–62. ISSN: 1471-0048.
- [22] Vincent Bonin, Valerio Mante, and Matteo Carandini. “The Suppressive Field of Neurons in Lateral Geniculate Nucleus”. In: *Journal of Neuroscience* 25.47 (2005), pp. 10844–10856. ISSN: 0270-6474.
- [23] Valerio Mante, Vincent Bonin, and Matteo Carandini. “Functional Mechanisms Shaping Lateral Geniculate Responses to Artificial and Natural Stimuli”. In: *Neuron* 58.4 (2008), pp. 625–638. ISSN: 0896-6273.
- [24] Lane T. McIntosh, Niru Maheswaranathan, Aran Nayebi, Surya Ganguli, and Stephen A. Baccus. “Deep learning models of the retinal response to natural scenes”. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS’16*. Barcelona, Spain: Curran Associates Inc., 2016, pp. 1369–1377. ISBN: 9781510838819.
- [25] David Marr and Ellen Hildreth. “Theory of Edge Detection”. In: *Proceedings of the Royal Society of London. Series B, Biological Sciences* 207.1167 (1980), pp. 187–217. DOI: 10.1098/rspb.1980.0020.
- [26] Edwin H. Land and John J. McCann. “Lightness and Retinex Theory”. In: *J. Opt. Soc. Am.* 61.1 (Jan. 1971), pp. 1–11. DOI: 10.1364/JOSA.61.000001.
- [27] Berthold K.P. Horn. “Determining lightness from an image”. In: *Computer Graphics and Image Processing* 3.4 (1974), pp. 277–299. ISSN: 0146-664X. DOI: [https://doi.org/10.1016/0146-664X\(74\)90022-7](https://doi.org/10.1016/0146-664X(74)90022-7).
- [28] J. J. Koenderink and A. J. van Doorn. “Representation of local geometry in the visual system”. In: *Biological Cybernetics* 55.6 (Mar. 1987), pp. 367–375. ISSN: 1432-0770. DOI: 10.1007/BF00318371.
- [29] P. Burt and E. Adelson. “The Laplacian Pyramid as a Compact Image Code”. In: *IEEE Transactions on Communications* 31.4 (1983), pp. 532–540. DOI: 10.1109/TCOM.1983.1095851.
- [30] Zahra Babaiee, Ramin Hasani, Mathias Lechner, Daniela Rus, and Radu Grosu. “On-Off Center-Surround Receptive Fields for Accurate and Robust Image Classification”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 478–489.
- [31] Liu Risheng, Ma Long, Zhang Jiaao, Fan Xin, and Luo Zhongxuan. “Retinex-inspired Unrolling with Cooperative Prior Architecture Search for Low-light Image Enhancement”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2021.



- [32] Dan Hendrycks\*, Norman Mu\*, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. “AugMix: A Simple Method to Improve Robustness and Uncertainty under Data Shift”. In: *International Conference on Learning Representations*. 2020.
- [33] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. “The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 8320–8329.
- [34] Apostolos Modas, Rahul Rade, Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. “PRIME: A Few Primitives Can Boost Robustness to Common Corruptions”. In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner. Cham: Springer Nature Switzerland, 2022, pp. 623–640. ISBN: 978-3-031-19806-9.
- [35] Tiago Salvador and Adam M Oberman. “ImageNet-Cartoon and ImageNet-Drawing: two domain shift datasets for ImageNet”. In: *ICML 2022 Shift Happens Workshop*. 2022.
- [36] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. “Learning Robust Global Representations by Penalizing Local Predictive Power”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 10506–10518.
- [37] Evgenia Rusak, Lukas Schott, Roland S. Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. “A Simple Way to Make Neural Networks Robust Against Diverse Image Corruptions”. In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Cham: Springer International Publishing, 2020, pp. 53–69. ISBN: 978-3-030-58580-8.
- [38] Anish Athalye, Nicholas Carlini, and David Wagner. “Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples”. In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. July 2018.
- [39] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. “On Evaluating Adversarial Robustness”. In: *arXiv preprint arXiv:1902.06705* (2019).
- [40] 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.
- [41] Jeremy Freeman, Corey M. Ziemba, David J. Heeger, Eero P. Simoncelli, and J. Anthony Movshon. “A functional and perceptual signature of the second visual area in primates”. In: *Nature Neuroscience* 16.7 (July 2013), pp. 974–981. ISSN: 1546-1726. DOI: 10.1038/nn.3402.
- [42] Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, Kailyn Schmidt, Daniel L. K. Yamins, and James J. DiCarlo. “Brain-Score: Which Artificial Neural Network for Object Recognition is most Brain-Like?” In: *bioRxiv preprint* (2018).
- [43] Martin Schrimpf, Jonas Kubilius, Michael J Lee, N Apurva Ratan Murty, Robert Ajemian, and James J DiCarlo. “Integrative Benchmarking to Advance Neurally Mechanistic Models of Human Intelligence”. In: *Neuron* (2020).
- [44] Stephanie Olaiya, Tiago Marques, and James J. DiCarlo. “Measuring the Alignment of ANNs and Primate V1 on Luminance and Contrast Response Characteristics”. In: *SVRHM 2022 Workshop @ NeurIPS*. 2022.
- [45] Nicola Strisciuglio, Manuel Lopez-Antequera, and Nicolai Petkov. “Enhanced robustness of convolutional networks with a push–pull inhibition layer”. In: *Neural Comput. Appl.* 32.24 (Dec. 2020), pp. 17957–17971. ISSN: 0941-0643.
- [46] Galen Pogoncheff, Jacob Granley, and Michael Beyeler. “Explaining V1 Properties with a Biologically Constrained Deep Learning Architecture”. In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.
- [47] 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 (1982), pp. 545–559. ISSN: 0042-6989.

- [48] Ungsoo Samuel Kim, Omar A. Mahroo, John D. Mollon, and Patrick Yu-Wai-Man. “Retinal Ganglion Cells—Diversity of Cell Types and Clinical Relevance”. In: *Frontiers in Neurology* 12 (2021). ISSN: 1664-2295.
- [49] T N Wiesel and D H Hubel. “Spatial and chromatic interactions in the lateral geniculate body of the rhesus monkey.” In: *Journal of Neurophysiology* 29.6 (1966). PMID: 4961644, pp. 1115–1156.
- [50] Alexander Berardino, Johannes Ballé, Valero Laparra, and Eero Simoncelli. “Eigen-Distortions of Hierarchical Representations”. In: *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 3533–3542. ISBN: 9781510860964.
- [51] R. T. Raghavan, Jenna G. Kelly, J. Michael Hasse, Paul G. Levy, Michael J. Hawken, and J. Anthony Movshon. “Contrast and Luminance Gain Control in the Macaque’s Lateral Geniculate Nucleus”. In: *eNeuro* 10.3 (2023).
- [52] Valerio Mante, Robert A. Frazor, Vincent Bonin, Wilson S. Geisler, and Matteo Carandini. “Independence of luminance and contrast in natural scenes and in the early visual system”. In: *Nature Neuroscience* 8.12 (Dec. 2005), pp. 1690–1697. ISSN: 1546-1726.
- [53] Matteo Carandini and David J. Heeger. “Summation and Division by Neurons in Primate Visual Cortex”. In: *Science* 264.5163 (1994), pp. 1333–1336.
- [54] J A Hirsch, J M Alonso, R C Reid, and L M Martinez. “Synaptic integration in striate cortical simple cells”. en. In: *The Journal of Neuroscience* 18.22 (Nov. 1998), pp. 9517–9528.
- [55] W R Softky and C Koch. “The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs”. en. In: *The Journal of Neuroscience* 13.1 (Jan. 1993), pp. 334–350.
- [56] Loïc Daumail, Brock M. Carlson, Blake A. Mitchell, Michele A. Cox, Jacob A. Westerberg, Cortez Johnson, Paul R. Martin, Frank Tong, Alexander Maier, and Kacie Dougherty. “Rapid adaptation of primate LGN neurons to drifting grating stimulation”. In: *Journal of Neurophysiology* 129.6 (2023). PMID: 37162181, pp. 1447–1467. DOI: 10.1152/jn.00058.2022.
- [57] V. J. Uzzell and E. J. Chichilnisky. “Precision of Spike Trains in Primate Retinal Ganglion Cells”. In: *Journal of Neurophysiology* 92.2 (2004). PMID: 15277596, pp. 780–789. DOI: 10.1152/jn.01171.2003. eprint: <https://doi.org/10.1152/jn.01171.2003>.
- [58] Samuel G. Solomon, Andrew J. R. White, and Paul R. Martin. “Extraclassical Receptive Field Properties of Parvocellular, Magnocellular, and Koniocellular Cells in the Primate Lateral Geniculate Nucleus”. In: *Journal of Neuroscience* 22.1 (2002), pp. 338–349. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.22-01-00338.2002.
- [59] Michael P. Sceniak, Dario L. Ringach, Michael J. Hawken, and Robert Shapley. “Contrast’s effect on spatial summation by macaque V1 neurons”. In: *Nature Neuroscience* 2.8 (Aug. 1999), pp. 733–739. ISSN: 1546-1726. DOI: 10.1038/11197.
- [60] J. Močkus. “On bayesian methods for seeking the extremum”. In: *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*. Ed. by G. I. Marchuk. Berlin, Heidelberg: Springer Berlin Heidelberg, 1975, pp. 400–404. ISBN: 978-3-540-37497-8.
- [61] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. “Image Style Transfer Using Convolutional Neural Networks”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2414–2423. DOI: 10.1109/CVPR.2016.265.
- [62] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. *Robustness (Python Library)*. 2019. URL: <https://github.com/MadryLab/robustness>.
- [63] J. Bullier and H. Kennedy. “Projection of the lateral geniculate nucleus onto cortical area V2 in the macaque monkey”. In: *Experimental Brain Research* 53.1 (Dec. 1983), pp. 168–172. ISSN: 1432-1106. DOI: 10.1007/BF00239409.
- [64] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings*. 2014.
- [65] Jonas Kubilius, Martin Schrimpf, Aran Nayeibi, Daniel Bear, Daniel L. K. Yamins, and James J. DiCarlo. “CORnet: Modeling the Neural Mechanisms of Core Object Recognition”. In: (2018).
- [66] 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 (2002). PMID: 12424292, pp. 2530–2546. DOI: 10.1152/jn.00692.2001.

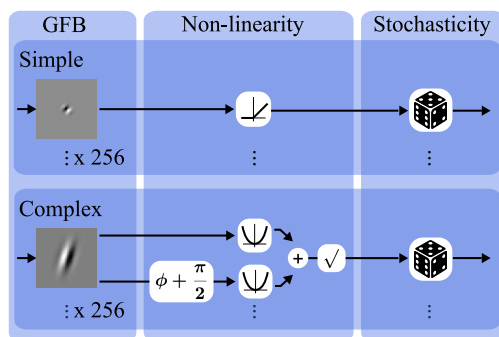
- [67] Gary Sclar, John H.R. Maunsell, and Peter Lennie. “Coding of image contrast in central visual pathways of the macaque monkey”. In: *Vision Research* 30.1 (1990), pp. 1–10. ISSN: 0042-6989. DOI: [https://doi.org/10.1016/0042-6989\(90\)90123-3](https://doi.org/10.1016/0042-6989(90)90123-3).
- [68] Michelle Miller, SueYeon Chung, and Kenneth D. Miller. “Divisive Feature Normalization Improves Image Recognition Performance in AlexNet”. In: *International Conference on Learning Representations*. 2022.
- [69] Nádia Canário, Lília Jorge, M.F. Loureiro Silva, Mário Alberto Soares, and Miguel Castelo-Branco. “Distinct preference for spatial frequency content in ventral stream regions underlying the recognition of scenes, faces, bodies and other objects”. In: *Neuropsychologia* 87 (2016), pp. 110–119. ISSN: 0028-3932. DOI: <https://doi.org/10.1016/j.neuropsychologia.2016.05.010>.
- [70] Richard S. Sutton. *The Bitter Lesson*. accessed: 2025-10-23. Mar. 2019. URL: <http://www.incompleteideas.net/IncIdeas/BitterLesson.html>.
- [71] Jonas Kubilius, Martin Schrimpf, Kohitij Kar, Rishi Rajalingham, Ha Hong, Najib Majaj, Elias Issa, Pouya Bashivan, Jonathan Prescott-Roy, Kailyn Schmidt, Aran Nayebi, Daniel Bear, Daniel L Yamins, and James J DiCarlo. “Brain-Like Object Recognition with High-Performing Shallow Recurrent ANNs”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.
- [72] Abdulkadir Gokce and Martin Schrimpf. “Scaling Laws for Task-Optimized Models of the Primate Visual Ventral Stream”. In: *Forty-second International Conference on Machine Learning*. 2025.
- [73] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2021.
- [74] Nicole C. Rust, Odelia Schwartz, J. Anthony Movshon, and Eero P. Simoncelli. “Spatiotemporal Elements of Macaque V1 Receptive Fields”. In: *Neuron* 46.6 (2005), pp. 945–956. ISSN: 0896-6273.
- [75] J P Jones and L A Palmer. “The two-dimensional spatial structure of simple receptive fields in cat striate cortex”. en. In: *Journal of Neurophysiology* 58.6 (Dec. 1987), pp. 1187–1211.
- [76] Edward H. Adelson and James R. Bergen. “Spatiotemporal energy models for the perception of motion”. In: *J. Opt. Soc. Am. A* 2.2 (Feb. 1985), pp. 284–299.
- [77] Russell L. De Valois, E. William Yund, and Norva Hepler. “The orientation and direction selectivity of cells in macaque visual cortex”. In: *Vision Research* 22.5 (1982), pp. 531–544. ISSN: 0042-6989.
- [78] Dario L. Ringach. “Spatial Structure and Symmetry of Simple-Cell Receptive Fields in Macaque Primary Visual Cortex”. In: *Journal of Neurophysiology* 88.1 (2002). PMID: 12091567, pp. 455–463.
- [79] Avinash Baidya, Joel Dapello, James J. DiCarlo, and Tiago Marques. “Combining Different V1 Brain Model Variants to Improve Robustness to Image Corruptions in CNNs”. In: *SVRHM 2021 Workshop @ NeurIPS*. 2021.
- [80] P. H. Schiller, B. L. Finlay, and S. F. Volman. “Quantitative studies of single-cell properties in monkey striate cortex. III. Spatial frequency”. In: *Journal of Neurophysiology* 39.6 (1976). PMID: 825623, pp. 1334–1351.
- [81] Ankit Rohatgi. *WebPlotDigitizer*. Version 5.2. URL: <https://automeris.io>.
- [82] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. “Adversarial Robustness Toolbox v1.2.0”. In: *CoRR* 1807.01069 (2018).
- [83] Xinrui Wang and Jinze Yu. “Learning to Cartoonize Using White-Box Cartoon Representations”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 8087–8096. DOI: 10.1109/CVPR42600.2020.00811.
- [84] Cewu Lu, Li Xu, and Jiaya Jia. “Combining sketch and tone for pencil drawing production”. In: *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*. NPAR ’12. Annecy, France: Eurographics Association, 2012, pp. 65–73. ISBN: 9783905673906.

- [85] Xun Huang and Serge Belongie. “Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 1510–1519. DOI: 10.1109/ICCV.2017.167.
- [86] George A. Miller. “WordNet: a lexical database for English”. In: *Commun. ACM* 38.11 (Nov. 1995), pp. 39–41. ISSN: 0001-0782.
- [87] Lisa J. Croner and Ehud Kaplan. “Receptive fields of P and M ganglion cells across the primate retina”. In: *Vision Research* 35.1 (1995), pp. 7–24. ISSN: 0042-6989.
- [88] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.
- [89] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. “Re-thinking the Inception Architecture for Computer Vision”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.
- [90] Leslie N. Smith and Nicholay Topin. *Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates*. 2018. arXiv: 1708.07120 [cs.LG].

# Supplementary Material

## A VOneNets

VOneNets [12] are convolutional neural networks (CNNs) augmented with a biologically-inspired, fixed-weight front-end simulating primary visual cortex (V1), termed the VOneBlock. This front-end is structured as a linear–nonlinear–Poisson (LNP) cascade [74], incorporating a Gabor filter bank (GFB) [75], nonlinearities for both simple and complex cells [76], and a stochastic spiking mechanism modeling neuronal variability [55] (cf. Fig. A1). The GFB parameters are sampled from empirical distributions of orientation preference, spatial frequency (SF) tuning, and receptive field (RF) size [47, 77, 78]. The channels are split evenly between simple and complex cells, and a Poisson-like noise generator is applied to emulate spiking variability. The full implementation is available at <https://github.com/dicarlo1ab/vonetnet> under a GNU General Public License v3.0.



**Figure A1: VOneNets simulate V1 processing upstream of standard CNNs.** Each VOneNet incorporates a biologically-constrained front-end, the VOneBlock, preceding a conventional CNN. The VOneBlock consists of a fixed-weight Gabor filter bank (GFB) parameterized by empirical distributions, nonlinearities emulating simple and complex cell responses, and a stochastic component that injects Poisson-like noise to mimic V1 neuronal variability. Adapted from Baidya et al. [79].

To construct a VOneNet, we replaced the initial block of each base architecture with the VOneBlock, along with a channel-matching bottleneck layer to maintain architectural compatibility between the front-end and the downstream convolutional stack. Consistent with the original VOneResNet50 model [12], we replaced a single convolutional layer, batch normalization, nonlinearity, and a max-pooling operation and preserved the original configuration of 512 channels within the VOneBlock, allocating 256 channels to each cell type (simple and complex). When using an EfficientNet-B0, we substituted the initial convolution, batch normalization, and activation with the VOneBlock. Since this initial block has a total stride of 2, we decreased the stride of the VOneBlock accordingly. Furthermore, given the reduced channel dimensionality in the EfficientNet-B0 where the second stage expects only 32 channels, in contrast to 64 in ResNet50 we downscaled the VOneBlock, employing 128 channels per cell type. Finally, for the CORnet-Z we removed a single convolutional layer, nonlinearity, and max-pooling operation. Since this first block has the same combined stride and output dimension as the ResNet50, no additional modifications were necessary.

We modified the VOneBlock by adjusting the field of view (FoV) to 7deg (down from the original 8deg) and increased the SF range of the GFB to 0.5 – 8.0 cpd (from 0.5 – 5.6cpd). This modification allows us to better match empirical V1 distributions, while maintaining a similar safety margin with respect to the Nyquist SF. Additionally, to maintain consistency with upstream processing, we configured the GFB to uniformly sample a single channel from the input, regardless of any preceding subcortical transformations. Finally, to ensure methodological consistency in the spike-based activation regime across both the SubcorticalBlock (cf. Section E.2) and the VOneBlock, we imposed a unified temporal integration window of 50ms. In alignment with Table C2 of Dapello et al. [12], we applied a linear scaling factor to the VOneBlock outputs such that the mean evoked response to a batch of natural images from ImageNet matched the target stimulus response of 0.655 spikes. This scaling factor was computed independently for the VOneNet and for each EVnet variant described in this study.

## B Primate Vision Alignment

### B.1 Empirical V1 Tuning Curves

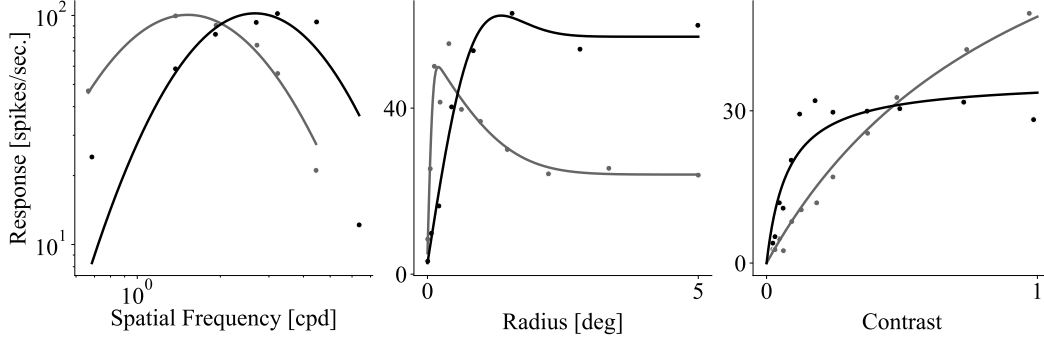


Figure B2: **Examples of empirical V1 tuning curves retrieved from the literature.** SF, size, and contrast tuning curves (left to right) for example V1 neurons. **Left** SF tuning curve of a simple (gray) and complex (black) cell to drifting grating stimuli. Markers represent the total number of F1 responses to gratings of different SF normalized to the best response and the solid line depicts a quadratic fit for purposes of illustrating the tuning profile (adapted from Figure 10 in Schiller et al. [80]). **Middle** Size tuning curve of two complex cells of V1 with distinct degrees of surround suppression under high contrasts. Gray depicts the a cell from V1 layer 4B under 0.15 contrast and black represents a cell from V1 layer 6 under 0.31 contrast. Markers represent each cell’s F1 response to differently-sized gratings and the line depicts the predicted response of a fitted DoG model discussed in the original article (adapted from Figure 1 in Sceniak et al. [59]). **Right** Contrast tuning curve of two simple V1 cell from the least (gray) and most (black) contrast sensitive thirds of their respective population. Marks indicate F1 response and the solid line depicts a fitted response model discussed in the original article (adapted from Figure 2 in Sclar et al. [67]). Data points extracted via WebPlotDigitalizer [81].

### B.2 Shape-bias

In contrast to humans, who predominantly rely on shape cues for object recognition, ImageNet-trained CNNs have been shown to exhibit a strong bias toward texture-based representations [8]. Measuring shape bias thus serves as a proxy for alignment with human inductive biases. We evaluate this using the cue conflict dataset from Geirhos et al. [8], where images contain conflicting shape and texture cues (e.g., a cat-shaped image with elephant texture). While humans tend to classify by shape, ImageNet-trained CNNs often prefer texture. A model’s shape bias is computed as the proportion of shape-consistent predictions out of all shape- or texture-consistent responses.

### B.3 BrainScore

The BrainScore platform [42, 43] is a standardized benchmarking suite for evaluating how brain-like artificial neural networks (ANNs) are. In the context of object recognition, BrainScore compares model activations against neural recordings from primate visual areas and human behavioral data. For early visual processing, V1 predictivity is quantified via the FreemanZiemba2013 [41] neural benchmark, while response properties in V1 are assessed using the Marques2020 [40] benchmark. BrainScore aggregates multiple such benchmarks into a composite score that reflects a model’s alignment with neural and behavioral patterns observed in biological systems. Researchers can submit their models for evaluation at <https://www.brain-score.org/>.

### B.4 V1 Predictivity

To predict model’s ability to predict single-neuron responses in V1, we employed a dataset [41] comprising responses from 102 V1 neurons to 450 unique 4deg image patches, spanning both naturalistic textures and noise stimuli. Predictivity was measured as the explained variance using partial least squares (PLS) regression under a 10-fold cross-validation scheme

Table B1: **Detailed results for V1 response property alignment.** BrainScore [42, 43] V1 alignment scores for ResNet50, VOneResNet50, and EVResNet50. Values indicate mean  $\pm$  SD ( $n = 3$  seeds).

Category	Resp. Property	Models		
		ResNet50 [4]	VOneResNet50	EVResNet50
Orientation	Orientation Selective	0.975 $\pm$ 0.024	<b>0.999<math>\pm</math>0.001</b>	<b>0.999<math>\pm</math>0.001</b>
	Circ. Variance (CV)	<b>0.818<math>\pm</math>0.011</b>	0.742 $\pm$ 0.013	0.754 $\pm$ 0.001
	Orth./Pref. Ratio	<b>0.855<math>\pm</math>0.023</b>	0.717 $\pm$ 0.014	0.710 $\pm$ 0.001
	CV Bandwidth Ratio	0.740 $\pm$ 0.024	<b>0.763<math>\pm</math>0.005</b>	0.762 $\pm$ 0.001
	Pref. Orientation	0.943 $\pm$ 0.046	<b>0.985<math>\pm</math>0.004</b>	0.968 $\pm$ 0.000
	Orth./Pref.-CV Diff.	0.766 $\pm$ 0.016	<b>0.885<math>\pm</math>0.004</b>	0.869 $\pm$ 0.001
	Or. Bandwidth	0.659 $\pm$ 0.086	0.922 $\pm$ 0.010	<b>0.952<math>\pm</math>0.000</b>
Spatial Frequency	Peak SF	0.551 $\pm$ 0.047	<b>0.961<math>\pm</math>0.002</b>	<b>0.961<math>\pm</math>0.001</b>
	SF Bandwidth	0.826 $\pm$ 0.019	<b>0.962<math>\pm</math>0.006</b>	0.937 $\pm$ 0.000
	SF Selective	0.886 $\pm$ 0.053	<b>0.983<math>\pm</math>0.005</b>	0.951 $\pm$ 0.000
Response Selectivity	Texture Selective	0.678 $\pm$ 0.008	<b>0.800<math>\pm</math>0.004</b>	0.774 $\pm$ 0.001
	Modulation Ratio	0.349 $\pm$ 0.009	<b>0.737<math>\pm</math>0.002</b>	0.736 $\pm$ 0.000
	Texture Var. Ratio	<b>0.794<math>\pm</math>0.014</b>	0.703 $\pm$ 0.011	0.694 $\pm$ 0.001
	Texture Sparseness	0.663 $\pm$ 0.032	<b>0.927<math>\pm</math>0.002</b>	0.920 $\pm$ 0.000
RF Size	Grating Sum. Field	0.272 $\pm$ 0.005	0.547 $\pm$ 0.016	<b>0.716<math>\pm</math>0.003</b>
	Surround Diameter	0.156 $\pm$ 0.000	0.361 $\pm$ 0.015	<b>0.736<math>\pm</math>0.000</b>
Surround Mod.	Surround Sup. Index	0.389 $\pm$ 0.023	0.373 $\pm$ 0.003	<b>0.614<math>\pm</math>0.004</b>
Texture Modulation	Abs. Texture Mod. Idx.	<b>0.978<math>\pm</math>0.019</b>	0.942 $\pm$ 0.004	0.934 $\pm$ 0.000
	Texture Mod. Idx.	0.606 $\pm$ 0.040	0.897 $\pm$ 0.011	<b>0.898<math>\pm</math>0.001</b>
Response Magnitude	Max. Texture	0.939 $\pm$ 0.002	0.906 $\pm$ 0.010	<b>0.951<math>\pm</math>0.001</b>
	Max. DC	<b>0.873<math>\pm</math>0.053</b>	0.824 $\pm$ 0.008	0.885 $\pm$ 0.001
	Max. Noise	0.783 $\pm$ 0.018	0.923 $\pm$ 0.006	<b>0.965<math>\pm</math>0.000</b>

## B.5 V1 Response Properties

Marques et al. [40] introduced a novel model-to-brain comparison framework that bypasses conventional fitting procedures, instead relying on *in silico* neurophysiology to establish direct, one-to-one correspondences between artificial and V1 neurons. By probing models with canonical stimulus sets such as drifting gratings and texture pattern, the method quantifies alignment through a normalized similarity metric grounded in the Kolmogorov-Smirnov distance, capturing the distributional match of neural response properties. Critically, this framework enables rigorous benchmarking against prior neurophysiological studies without requiring raw recordings, effectively transforming existing literature into executable V1-aligning tests. In total, this method focuses on 22 distinct response characteristics, organized into seven functional domains: orientation tuning, spatial frequency tuning, receptive field size, surround modulation, texture modulation, response selectivity, and response magnitude. Table B1 presents all individual response properties along with the scores obtained for the ResNet50, the VOneResNet50 and EVResNet50 models.

## C Image Perturbations

### C.1 Adversarial Attacks

To evaluate white-box robustness, we employed Projected Gradient Descent (PGD) [10] on top of a subset of 5000 images from the ImageNet validation split. PGD is a widely adopted first-order attack that has proven effective against biologically inspired models such as VOneNets [12]. We selected this attack due to its scalability to large datasets (e.g., 5k ImageNet samples), and its compatibility with deterministic inference pipelines, which avoids the pitfalls of stochastic defenses that may artificially degrade attack success rates [12]. We run PGD for  $N = 64$  iterations, where each step follows the update rule:

$$\mathbf{x}^{t+1} = \mathcal{P}_{\mathbf{x}+\mathcal{S}} \left( \mathbf{x}^t + \alpha \operatorname{sgn} \left( \nabla_{\mathbf{x}^t} \mathcal{L}(\theta, \mathbf{x}^t, \mathbf{y}) \right) \right),$$

where  $\mathbf{x}^t$  denotes the adversarial input at iteration  $t$ ,  $\mathcal{L}$  is the cross-entropy loss function, and  $\mathcal{P}_{\mathbf{x}+\mathcal{S}}$  projects back onto the perturbation set  $\mathcal{S}$  centered at the clean input  $\mathbf{x}$ . Under an  $L_\infty$  threat model,  $\mathcal{S}$  corresponds to a box constraint, while for  $L_1$  or  $L_2$  norms with a perturbation budget  $\epsilon$ , the gradient direction is rescaled at each iteration to have the respective norm  $\alpha$ , and the projection ensures the final adversarial input  $\mathbf{x}_{\text{adv}}$  satisfies  $\|\mathbf{x}_{\text{adv}} - \mathbf{x}\|_p \leq \epsilon$ . We used a step size of  $\alpha = \epsilon/32$  and performed a total of 12 attacks carried out under  $L_\infty$ ,  $L_2$  and  $L_1$  norm constraints at four perturbation budgets each:  $\|\delta\|_\infty \in [1/1020, 1/255, 4/255, 16/255]$ ,  $\|\delta\|_2 \in [0.15, 0.6, 1.2, 2.4]$  and  $\|\delta\|_1 \in [40, 160, 640, 2560]$ . We used the Adversarial Robustness ToolBox v1.17.1 [82] to conduct all the attacks.

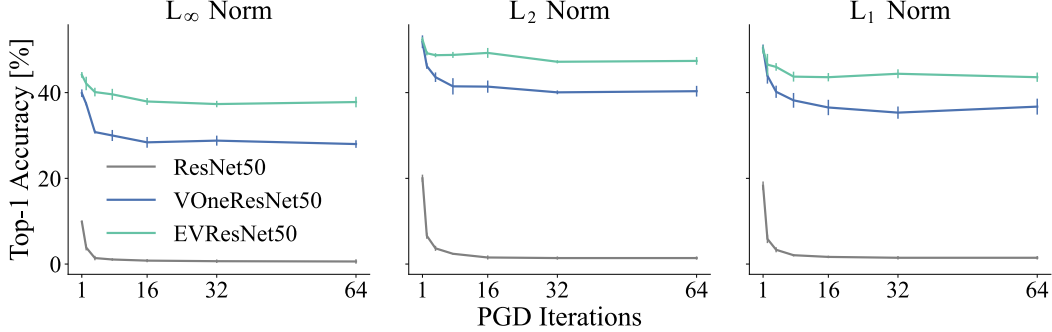


Figure C1: **Adversarial robustness is evaluated at convergence of PGD iterations.** Top-1 white-box accuracy iteration curves for PGD attacks with  $\|\delta\|_\infty = 1/255$ ,  $\|\delta\|_2 = 0.6$ ,  $\|\delta\|_1 = 160$  constraints for ResNet50, VOneResNet50 and EVResNet50 models, evaluated on 500 images. The step size was adjusted to be  $\epsilon$  for 1 iterations, and  $2\epsilon/N$ , in the remaining cases. Increasing the number of PGD iteration steps increases attack effectiveness only up to roughly 32 iterations. Lines indicate the mean accuracy and shaded error bars denote SD ( $n = 3$  seeds).

Due to the inherent stochasticity in our models, special considerations were necessary to enable gradient-based adversarial optimization. We first applied the reparameterization trick [64], which permits gradient flow through stochastic nodes by expressing random variables as deterministic functions of noise. To obtain reliable gradient estimates for PGD, we further adopted the approach of Athalye et al. [38], replacing  $\nabla f$  with an average over multiple stochastic forward passes. Specifically, we estimate gradients as

$$\nabla f \approx \frac{1}{k} \sum_{i=1}^k \nabla_i f,$$

where each  $\nabla_i f$  corresponds to a gradient computed using an independent Monte Carlo sample. We set  $k = 10$ , similarly to prior work on VOneNets, given that the additional noise source in the SubcorticalBlock did not mask the gradients any further.



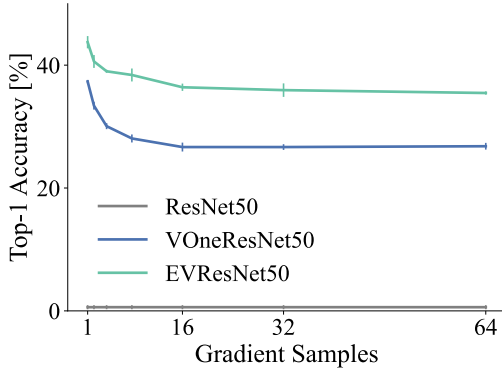


Figure C2: **Increasing the number of Monte Carlo gradient samples has limited impact on white-box attack effectiveness.** White-box PGD accuracy evaluated under  $\|\delta\|_\infty = 1/255$  with 64 PGD iterations on 500 images. Increasing  $k$  from 10 to 64 leads to only marginal decreases in accuracy for both VOneResNet50 and EVResNet50, indicating that additional samples do not substantially strengthen the attack. Lines indicate the mean accuracy and error bars denote SD ( $n = 3$  seeds).

To verify the reliability of our adversarial evaluation pipeline, we conducted a suite of sanity checks on a 500-image subset from the ImageNet validation set. In line with the recommendations of Athalye et al. [38] and Carlini et al. [39], we confirmed that top-1 accuracy decreases monotonically as a function of perturbation strength across all norm constraints (Tab. 4). Additionally, we verified that increasing the number of PGD iterations increased attack effectiveness (Fig. C1) and that increasing the number of gradient samples in the Monte Carlo approximation did not lead to a substantial increase in attack success (Fig. C2), further supporting the completeness of our threat model.

## C.2 Image Corruptions

The ImageNet-C dataset [6] consists of 15 different corruption types, each at 5 levels of severity for a total of 75 different perturbations applied to validation images of the ImageNet validation split. Accuracy improvement on these datasets should be indicative of model robustness gains, given that it comprises, in total, 75 diverse corruptions. The individual corruption types are Gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transform, pixelated, and JPEG compression. The individual corruption types are grouped into 4 categories: noise, blur, weather, and digital effects. Examples of image corruptions are presented in Figure C3. The ImageNet-C dataset is publicly available at <https://github.com/hendrycks/robustness> under Creative Commons Attribution 4.0 International.

## C.3 Domain Shifts

**ImageNet-R** The ImageNet-R dataset [33], consists of a curated set of 200 classes from the ImageNet validation set. This dataset includes 30,000 images featuring renditions in various artistic styles, such as paintings, sketches, and cartoons, designed to test a model’s ability to generalize beyond natural image statistics. ImageNet-R is publicly available at <https://github.com/hendrycks/imagenet-r>.

**ImageNet-Cartoon & ImageNet-Drawing** The ImageNet-Cartoon and ImageNet-Drawing datasets [35], are two domain shift benchmarks derived from the ImageNet validation set by applying label-preserving style transformations. ImageNet-Cartoon contains images transformed into cartoon-like renditions using a GAN-based framework [83], while ImageNet-Drawing comprises colored pencil sketch versions of the same images created via an image processing pipeline [84]. These datasets challenge models to generalize beyond natural image statistics, revealing significant accuracy drops—on average 18 and 45 percentage points, respectively—when standard ImageNet-trained models are evaluated. Both datasets are publicly available at <https://zenodo.org/records/6801109> under Creative Common Attribution 4.0 International.

**ImageNet-Sketch.** The ImageNet-Sketch dataset [36] is a large-scale benchmark designed to evaluate OOD generalization in image classification. It contains 50,000 black-and-white sketch-style images, with 50 images for each of the 1,000 classes in the ImageNet validation set, collected independently using keyword queries like “sketch of [class name]”. Unlike perturbation-based datasets, ImageNet-Sketch represents a significant domain shift in both texture and color, challenging



Figure C3: **Examples of common image corruptions from the ImageNet-C dataset at intermediate severity (level 3)** The first row shows the original image and three noise corruptions; the second row displays blur corruptions; the third row presents weather-related corruptions; and the fourth row illustrates digital corruptions.

models trained on natural images to rely on global structure rather than local textural cues. The dataset is publicly available at <https://www.kaggle.com/datasets/wanghaohan/imagenetsketch>.

**Stylized<sub>16</sub>-ImageNet.** Stylized-ImageNet [8] is created by introducing different painting styles into ImageNet images through Adaptive Instance Normalization style transfer [85]. While texture cues are replaced by those in the paintings, overall shape is preserved. Since the original dataset was introduced primarily for training purposes and models exhibited extremely low performance, we instead used a subset of Stylized-ImageNet as used in Geirhos et al. [7]. This subset focuses on 16 basic categories (e.g., airplane, dog) that are supersets of 227 ImageNet classes within the WordNet hierarchy [86]. We followed the same approach as the original article, where the probability distribution over ImageNet classes is mapped to this 16-class distribution by averaging the probabilities of corresponding fine-grained classes. This 16-class Stylized ImageNet along with the code for probability aggregation is publicly available at <https://github.com/bethgelab/model-vs-human>.

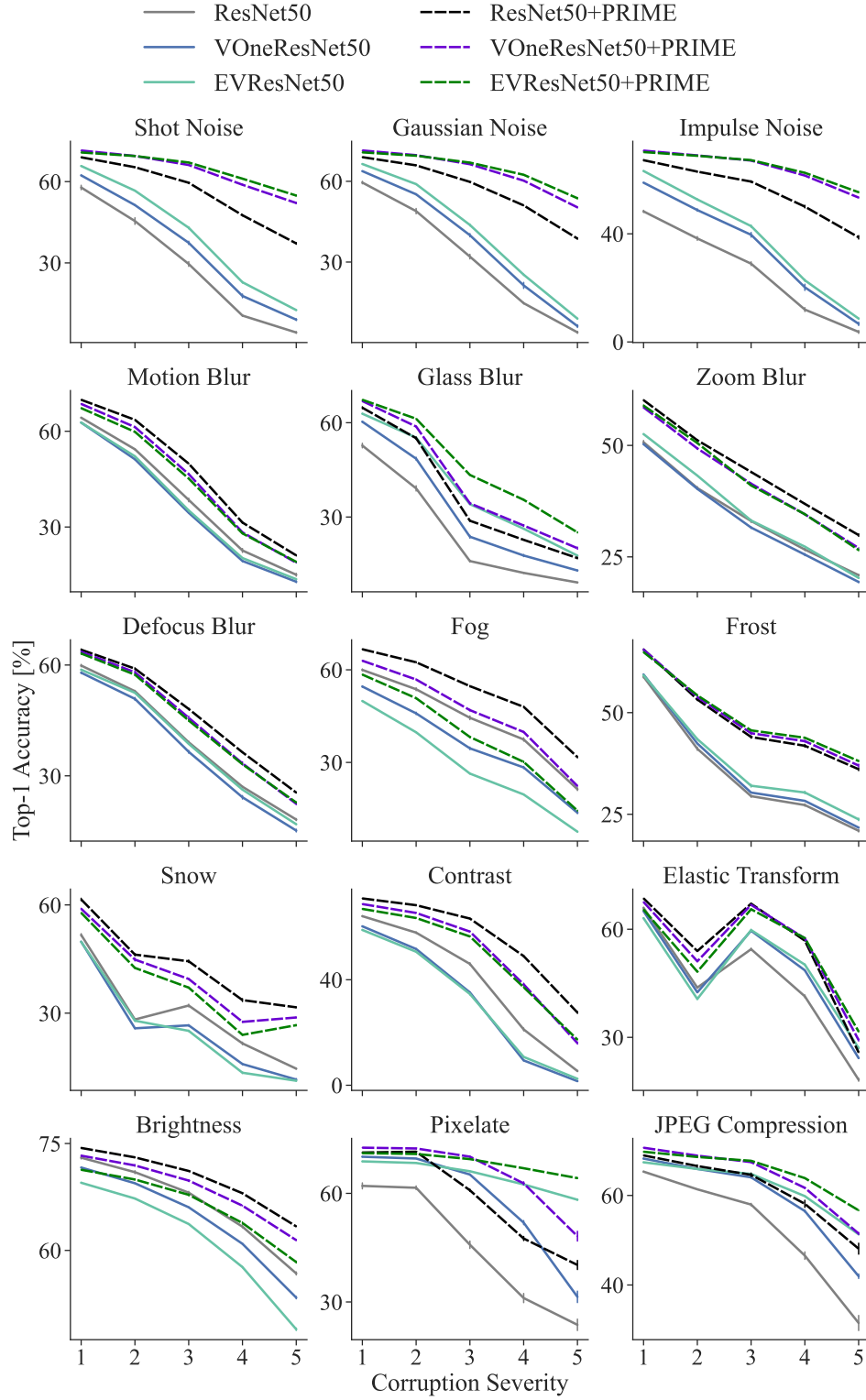


Figure C4: **Detailed results for common corruptions benchmarks.** Top-1 accuracy across 5 severity levels for the 15 individual common corruptions of ImageNet-C. Lines indicate the mean top-1 accuracy and error bars denote SD ( $n = 3$  seeds)

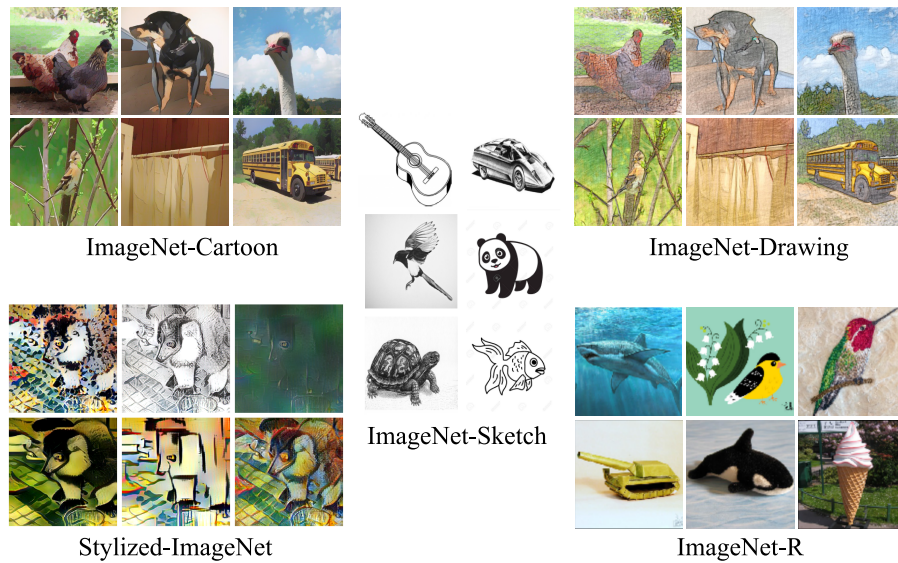


Figure C5: Examples from domain shift benchmark datasets derived from ImageNet. **ImageNet-Cartoon** (adopted from Salvador and Oberman [35]) features object representations in cartoon style. **ImageNet-Drawing** (adopted from Salvador and Oberman [35]) includes images rendered as colored pencil-like hand drawings. **ImageNet-Sketch** (adopted from Wang et al. [36]) consists of black-and-white sketches emphasizing contours. **Stylized-ImageNet** (adopted from Geirhos et al. [8]) applies and replaces the original textures with those of random paintings. **ImageNet-R** (adopted from Hendrycks et al. [33]) contains various renditions of ImageNet classes, including painting, cartoon, origami, toy, embroidery, and sculpture styles.

## D Additional Experiments

### D.1 EVNet Variants

Table D1: **Detailed results for EVResNet50 variants, including ablations, on brain-alignment metrics.** BrainScore [42, 43] V1 alignment scores and shape bias [8] for all EVResNet50 variants. ResNet50 and VOneResNet50 included for reference but not in the comparison. Values indicate mean  $\pm$  SD ( $n = 2$  seeds).

Model	V1 Predict.	V1 Resp. Prop.	V1 Response Properties							Shape Bias [%]
			Orient. Tuning	SF Tuning	RF Size	Surround Mod.	Texture Mod.	Resp. Select.	Resp. Magn.	
ResNet50	.271 $\pm .002$	.637 $\pm .008$	.822 $\pm .027$	.754 $\pm .026$	.214 $\pm .002$	.389 $\pm .023$	.792 $\pm .028$	.621 $\pm .010$	.865 $\pm .012$	18.8 $\pm 1.2$
VOneResNet50	.375 $\pm .002$	.754 $\pm .006$	.859 $\pm .005$	.969 $\pm .001$	.482 $\pm .041$	.373 $\pm .003$	.919 $\pm .004$	.792 $\pm .003$	.884 $\pm .002$	31.6 $\pm 1.2$
EVResNet50	.364 $\pm .000$	.826 $\pm .000$	.854 $\pm .009$	.950 $\pm .000$	.726 $\pm .001$	.614 $\pm .004$	.916 $\pm .001$	.781 $\pm .000$	.933 $\pm .000$	48.9 $\pm 2.4$
– P Cells	.350 $\pm .006$	<b>.845</b> $\pm .001$	.854 $\pm .002$	.945 $\pm .000$	<b>.738</b> $\pm .006$	<b>.737</b> $\pm .001$	.910 $\pm .007$	.764 $\pm .001$	<b>.965</b> $\pm .005$	<b>77.8</b> $\pm 1.7$
– M Cells	.368 $\pm .006$	.763 $\pm .000$	.858 $\pm .004$	.950 $\pm .000$	.527 $\pm .000$	.393 $\pm .007$	.906 $\pm .006$	.781 $\pm .004$	.926 $\pm .000$	49.9 $\pm 1.0$
– Light Adapt.	.364 $\pm .001$	.826 $\pm .001$	.859 $\pm .002$	<b>.951</b> $\pm .000$	.720 $\pm .006$	.630 $\pm .008$	.908 $\pm .009$	.780 $\pm .005$	.936 $\pm .006$	70.0 $\pm 3.3$
– Contrast Norm.	<b>.374</b> $\pm .007$	.768 $\pm .002$	<b>.868</b> $\pm .002$	.936 $\pm .001$	.562 $\pm .008$	.370 $\pm .001$	<b>.921</b> $\pm .008$	<b>.784</b> $\pm .005$	.938 $\pm .006$	48.0 $\pm 2.8$

Table D2: **Detailed results for EVResNet50 variants, including ablations, on common corruption categories.** Clean and corrupted top-1 accuracies averaged across corruptions types for all EVResNet50 variants. ResNet50 and VOneResNet50 included for reference but not in the comparison. Values indicate mean  $\pm$  SD ( $n = 2$  seeds).

Model	Mean [%]	Corruption Types				
		Noise [%]	Blur [%]	Weather [%]	Digital [%]	Clean [%]
ResNet50	38.8±0.5	29.2±0.6	34.6±0.4	36.1±0.5	49.5±0.6	75.4±0.1
VOneResNet50	40.4±0.1	35.9±0.5	34.8±0.1	32.6±0.1	52.2±0.1	72.9±0.1
EVResNet50	41.9±0.2	39.6±0.3	37.5±0.1	30.6±0.2	53.5±0.1	70.4±0.1
– P Cells	34.9±0.1	<b>42.7±0.1</b>	26.8±0.1	19.6±0.1	45.7±0.1	60.7±0.1
– M Cells	42.1±0.1	41.1±0.2	37.7±0.0	30.6±0.1	53.3±0.1	70.3±0.1
– Light Adaptation	40.4±0.2	35.2±0.0	<b>37.9±0.2</b>	29.2±0.2	52.2±0.1	69.8±0.2
– Contrast Norm.	41.7±0.4	39.2±1.1	37.6±0.4	30.4±0.4	53.4±0.1	70.7±0.0
– Subcort. Noise	41.9±0.2	39.9±0.5	35.8±0.1	32.2±0.5	53.8±0.0	<b>72.6±0.2</b>
– VOneBlock	<b>42.7±0.2</b>	41.2±0.9	37.4±0.2	<b>32.6±0.1</b>	<b>54.2±0.1</b>	71.6±0.1
+ LGN-V2 Conn.	42.1±0.1	40.0±0.1	37.6±0.3	30.8±0.1	53.8±0.0	70.7±0.1

Table D3: **Detailed results for EVResNet50 variants, including ablations, on domain-shift accuracy.** Top-1 accuracies on ImageNet- $\{\text{Cartoon, Drawing, R, Sketch, Stylized}_{16}\}$  for all EVResNet50 variants. ResNet50 and VOneResNet50 included for reference but not in the comparison. Values indicate mean  $\pm$  SD ( $n = 2$  seeds).

Model	Mean [%]	Cartoon [%]	Drawing [%]	R [%]	Sketch [%]	Stylized <sub>16</sub> [%]
ResNet50	33.4 $\pm$ 0.2	51.2 $\pm$ 0.7	20.9 $\pm$ 0.6	35.4 $\pm$ 0.1	23.3 $\pm$ 0.1	36.3 $\pm$ 1.2
VOneResNet50	37.1 $\pm$ 0.4	55.5 $\pm$ 0.2	30.5 $\pm$ 0.4	37.5 $\pm$ 0.1	23.1 $\pm$ 0.3	38.8 $\pm$ 1.1
EVResNet50	38.1 $\pm$ 0.3	57.1 $\pm$ 0.3	33.9 $\pm$ 0.2	38.1 $\pm$ 0.2	22.7 $\pm$ 0.2	38.6 $\pm$ 1.1
– P Cells	33.6 $\pm$ 0.0	46.3 $\pm$ 0.2	20.1 $\pm$ 0.2	36.3 $\pm$ 0.3	<b>24.7<math>\pm</math>0.1</b>	<b>40.7<math>\pm</math>0.3</b>
– M Cells	<b>38.2<math>\pm</math>0.2</b>	57.0 $\pm$ 0.2	34.4 $\pm$ 0.5	38.0 $\pm$ 0.0	22.8 $\pm$ 0.2	38.9 $\pm$ 0.9
– Light Adaptation	37.4 $\pm$ 0.2	56.2 $\pm$ 0.2	34.1 $\pm$ 0.5	37.4 $\pm$ 0.1	21.3 $\pm$ 0.4	38.1 $\pm$ 1.1
– Contrast Norm.	38.0 $\pm$ 0.1	56.9 $\pm$ 0.2	33.7 $\pm$ 0.4	38.1 $\pm$ 0.3	22.7 $\pm$ 0.6	38.6 $\pm$ 0.9
– Subcort. Noise	37.4 $\pm$ 0.1	56.5 $\pm$ 0.0	31.0 $\pm$ 0.3	37.6 $\pm$ 0.0	23.0 $\pm$ 0.1	38.7 $\pm$ 0.1
– VOneBlock	<b>38.2<math>\pm</math>0.1</b>	<b>57.2<math>\pm</math>0.0</b>	<b>34.7<math>\pm</math>0.3</b>	<b>38.2<math>\pm</math>0.2</b>	23.0 $\pm$ 0.3	38.1 $\pm$ 0.6
+ LGN-V2 Conn.	38.3 $\pm$ 0.2	57.0 $\pm$ 0.2	34.5 $\pm$ 0.2	38.0 $\pm$ 0.2	22.8 $\pm$ 0.2	39.0 $\pm$ 0.5

Table D4: **Detailed results for EVResNet50 variants, including ablations, on adversarial robustness.** Top-1 accuracies for all EVResNet50 variants on limited adversarial set. ResNet50 and VOneResNet50 included for reference but not in the comparison. Values indicate mean  $\pm$  SD ( $n = 2$  seeds).

Model	Mean [%]	$\ \delta\ _{\infty}$		$\ \delta\ _2$		$\ \delta\ _1$	
		$\frac{1}{1020}$ [%]	$\frac{1}{255}$ [%]	0.15 [%]	0.6 [%]	40 [%]	160 [%]
ResNet50	16.4 $\pm$ 0.5	23.4 $\pm$ 0.8	0.4 $\pm$ 0.0	37.2 $\pm$ 1.0	1.8 $\pm$ 0.2	33.6 $\pm$ 0.7	1.7 $\pm$ 0.2
VOneResNet50	50.5 $\pm$ 0.1	62.6 $\pm$ 0.4	30.4 $\pm$ 0.3	66.2 $\pm$ 0.3	42.3 $\pm$ 0.2	64.5 $\pm$ 0.3	37.3 $\pm$ 0.6
EVResNet50	53.8 $\pm$ 0.2	62.7 $\pm$ 0.2	38.8 $\pm$ 0.6	65.1 $\pm$ 0.0	48.0 $\pm$ 0.3	64.0 $\pm$ 0.2	44.5 $\pm$ 0.4
– P Cells	46.7 $\pm$ 0.2	53.4 $\pm$ 0.3	33.4 $\pm$ 0.1	55.6 $\pm$ 0.5	42.4 $\pm$ 0.2	55.2 $\pm$ 0.4	40.3 $\pm$ 0.1
– M Cells	<b>54.0<math>\pm</math>0.1</b>	62.6 $\pm$ 0.3	40.1 $\pm$ 0.2	64.8 $\pm$ 0.1	<b>48.4<math>\pm</math>0.2</b>	64.1 $\pm$ 0.1	44.0 $\pm$ 0.7
– Light Adapt.	55.1 $\pm$ 0.5	<b>62.8<math>\pm</math>0.5</b>	<b>42.1<math>\pm</math>0.4</b>	65.0 $\pm$ 0.7	50.3 $\pm$ 0.9	64.0 $\pm$ 0.2	<b>46.3<math>\pm</math>0.3</b>
– Contrast Norm.	53.3 $\pm$ 0.1	62.7 $\pm$ 0.2	38.2 $\pm$ 0.4	<b>65.2<math>\pm</math>0.3</b>	47.3 $\pm$ 0.2	64.3 $\pm$ 0.3	43.5 $\pm$ 0.0
– Subcort. Noise	48.5 $\pm$ 0.1	60.5 $\pm$ 0.1	28.2 $\pm$ 0.8	64.2 $\pm$ 0.1	39.6 $\pm$ 0.3	62.7 $\pm$ 0.1	35.8 $\pm$ 0.3
– VOneBlock	51.8 $\pm$ 0.5	62.2 $\pm$ 0.3	34.3 $\pm$ 0.6	<b>65.2<math>\pm</math>0.8</b>	44.8 $\pm$ 0.2	63.9 $\pm$ 0.9	40.4 $\pm$ 0.4
+ LGN-V2 Conn.	53.9 $\pm$ 0.2	<b>62.8<math>\pm</math>0.3</b>	38.7 $\pm$ 0.1	65.0 $\pm$ 0.8	<b>48.4<math>\pm</math>0.2</b>	<b>64.5<math>\pm</math>0.2</b>	44.2 $\pm$ 0.2

## D.2 EVNet Backend Generalization

Similarly to the results obtained with EVResNet50, the EVEfficientNet-B0 and EVCORnet-Z models consistently outperform their corresponding base model across most corruption categories, as well as in mean corruption accuracy, as shown in Table D5. However, this improvement comes with a greater relative drop in clean image accuracy compared to the ResNet50-based models. This steeper drop likely reflects architectural differences in sensitivity to input statistics. EfficientNet-B0 employs compound scaling and aggressive architecture search to optimize performance specifically for standard ImageNet inputs [5], making it more susceptible to deviations introduced by our biologically inspired preprocessing. In contrast, ResNet50, with its more generic design appears to be more adaptable to altered input distribution. Similarly, compared to ResNet50, the compact architecture of CORnet-Z exhibits a lower degree of feature redundancy which, when coupled with a mismatch between the inductive biases imposed by the front-end and those the network was designed to exploit, can limit its flexibility to adapt to the upstream processing. When evaluated on domain shift datasets, both EVEfficientNet-B0 and EVCORnet-Z surpass their base models on most benchmarks, as reported in Table D6, with the only exception being ImageNet-Sketch, mimicking the same pattern as observed with the EVResNet50. Both EVEfficientNet-B0 and EVCORnet-Z exhibit substantial improvements across all norm constraints (Table D7) and, when aggregated into the Robustness Score (Table D8), EVNets consistently surpass their respective base architectures, reinforcing the effectiveness of back-end generalization.

Table D5: **EVNets outperforms base models on most image corruption types and on mean corruption accuracy, across different backend architectures.** Clean and corrupted top-1 accuracies averaged across severities and corruptions for EfficientNet-B0, EVEfficientNet-B0, CORnet-Z and EVCORnet-Z. Values indicate mean  $\pm$  SD ( $n = 2$  seeds).

Model	Mean [%]	Corruption Types				Clean [%]
		Noise [%]	Blur [%]	Weather [%]	Digital [%]	
EfficientNet-B0	30.3±0.4	18.7±0.2	26.6±0.0	<b>29.1±0.3</b>	40.8±1.2	<b>68.1±0.1</b>
EVEfficientNet-B0	<b>34.1±0.0</b>	<b>30.7±0.2</b>	<b>30.5±0.3</b>	24.3±0.1	<b>45.0±0.1</b>	61.4±0.4
CORnet-Z	18.0±0.0	6.4±0.1	17.0±0.0	<b>12.4±0.3</b>	29.1±0.1	<b>53.2±0.1</b>
EVCORnet-Z	<b>21.3±0.0</b>	<b>20.0±0.0</b>	<b>18.2±0.1</b>	11.6±0.0	<b>30.4±0.0</b>	44.7±0.0

Table D6: **EVNets outperforms base models on most OOD datasets and on mean domain shift accuracy, across different backend architectures.** Top-1 accuracies on ImageNet-{Cartoon, Drawing, R, Sketch, Stylized<sub>16</sub>} for EfficientNet-B0, EVEfficientNet-B0, CORnet-Z and EVCORnet-Z. Values indicate mean  $\pm$  SD ( $n = 2$  seeds).

Model	Mean [%]	Cartoon [%]	Drawing [%]	R [%]	Sketch [%]	Stylized <sub>16</sub> [%]
EfficientNet-B0	28.9 $\pm$ 0.2	40.2 $\pm$ 0.4	17.1 $\pm$ 0.5	29.9 $\pm$ 0.2	<b>17.3<math>\pm</math>0.5</b>	40.0 $\pm$ 0.7
EVEfficientNet-B0	<b>33.5<math>\pm</math>0.2</b>	<b>49.1<math>\pm</math>0.0</b>	<b>28.2<math>\pm</math>0.2</b>	<b>31.5<math>\pm</math>0.2</b>	16.8 $\pm$ 0.3	<b>41.8<math>\pm</math>1.0</b>
CORnet-Z	19.5 $\pm$ 0.2	30.9 $\pm$ 0.2	12.5 $\pm$ 0.3	21.0 $\pm$ 0.2	<b>9.5<math>\pm</math>0.1</b>	23.8 $\pm$ 1.8
EVCORnet-Z	<b>21.1<math>\pm</math>0.3</b>	<b>32.9<math>\pm</math>0.1</b>	<b>16.2<math>\pm</math>0.2</b>	<b>21.3<math>\pm</math>0.0</b>	9.0 $\pm$ 0.1	<b>26.0<math>\pm</math>1.1</b>

## D.3 EVNet Inference Ensembling

To evaluate whether combining the stochastic activations of EVNets across multiple forward passes leads to cumulative performance gains, we conducted an ensemble analysis varying both ensemble size and the stage at which activations are aggregated. Specifically, we compared ensembles that averaged activations at three points in the network: (1) the logit layer, (2) the final embedding stage (layer4, before the global average pooling), and (3) immediately after the VOneBlock bottleneck. We found that averaging later representations at the embedding or logit level yielded marginal but consistent improvements across clean, corruption, and domain-shift evaluations (Fig. D1). In contrast, averaging activations after the bottleneck reduced performance, with the exception of a small



Table D7: **EVNets outperforms base models on most adversarial perturbations and on mean adversarial robustness, across different backend architectures.** Top-1 accuracies for EfficientNet-B0, EVEfficientNet-B0, CORnet-Z and EVCORnet-Z. Values indicate mean  $\pm$  SD ( $n = 2$  seeds).

Model	Mean [%]	$\ \delta\ _\infty$		$\ \delta\ _2$		$\ \delta\ _1$	
		$\frac{1}{1020}$ [%]	$\frac{1}{255}$ [%]	0.15 [%]	0.6 [%]	40 [%]	160 [%]
EfficientNet-B0	20.9 $\pm 0.1$	35.0 $\pm 0.6$	2.0 $\pm 0.1$	43.5 $\pm 0.3$	5.1 $\pm 0.1$	36.2 $\pm 0.2$	3.3 $\pm 0.1$
EVEfficientNet-B0	<b>45.6</b> $\pm 0.5$	<b>53.2</b> $\pm 0.5$	<b>31.1</b> $\pm 0.2$	<b>55.8</b> $\pm 0.6$	<b>40.8</b> $\pm 0.5$	<b>55.0</b> $\pm 0.6$	<b>37.7</b> $\pm 0.6$
CORnet-Z	17.6 $\pm 0.3$	24.4 $\pm 0.7$	0.7 $\pm 0.0$	34.8 $\pm 0.6$	5.4 $\pm 0.1$	34.7 $\pm 0.6$	5.5 $\pm 0.0$
EVCORnet-Z	<b>31.1</b> $\pm 0.6$	<b>36.7</b> $\pm 0.9$	<b>19.0</b> $\pm 0.8$	<b>38.8</b> $\pm 0.6$	<b>26.9</b> $\pm 0.4$	<b>39.0</b> $\pm 0.6$	<b>26.2</b> $\pm 0.4$

Table D8: **EVNets outperforms base models on Robustness Score, across different backend architectures.** Robustness Score, clean and perturbed top-1 accuracies for EfficientNet-B0, EVEfficientNet-B0, CORnet-Z and EVCORnet-Z. Values indicate mean  $\pm$  SD ( $n = 2$  seeds).

Model	Robust. Score* [%]	Perturbations			Clean [%]
		Adversarial* [%]	Corrupt. [%]	Domain Shift [%]	
EfficientNet-B0	26.7 $\pm 0.1$	20.9 $\pm 0.1$	30.3 $\pm 0.4$	28.9 $\pm 0.2$	<b>68.1<math>\pm 0.1</math></b>
EVEfficientNet-B0	<b>39.7<math>\pm 0.3</math></b>	<b>45.6<math>\pm 0.4</math></b>	<b>34.1<math>\pm 0.0</math></b>	<b>33.5<math>\pm 0.1</math></b>	61.4 $\pm 0.4$
CORnet-Z	18.4 $\pm 0.0$	17.6 $\pm 0.3$	18.0 $\pm 0.0$	19.5 $\pm 0.2$	<b>53.2<math>\pm 0.1</math></b>
EVCORnet-Z	<b>24.5<math>\pm 0.3</math></b>	<b>31.1<math>\pm 0.6</math></b>	<b>21.3<math>\pm 0.0</math></b>	<b>21.1<math>\pm 0.3</math></b>	44.7 $\pm 0.0$

performance gain by the two-model ensemble when evaluated on ImageNet-C. This degradation likely arises because the bottleneck lies immediately downstream of the noise-injection stage, and averaging at this point effectively diminishes the stochastic variability that the EVNet leverages during training. We did not evaluate adversarial performance in this setting, as generating adversarial samples already requires an ensemble of forward passes, and using an additional ensemble for evaluation would compound computational costs to an impractical level.

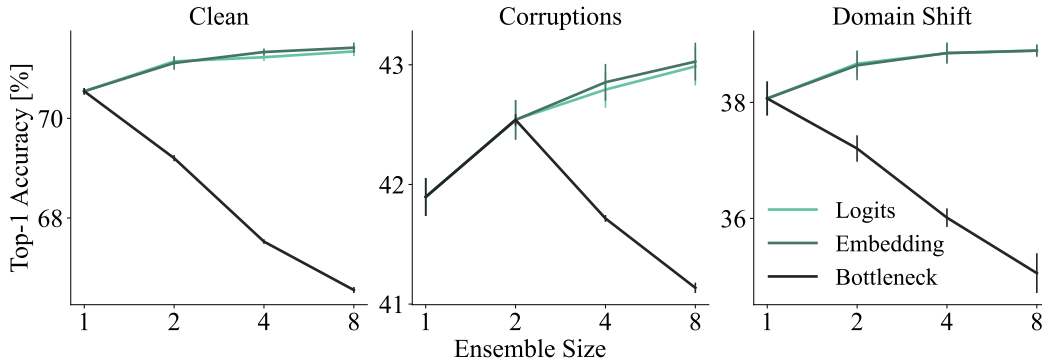


Figure D1: **Averaging EVNet logits or final embeddings yields slight performance improvements, whereas averaging bottleneck activations degrades accuracy.** Accuracy under clean, corruption, and domain-shift images for EVResNet50 ensembles of varying sizes. “Logits” denotes ensembles averaged at the logit layer, “Embeddings” refers to averaging activations in layer4 (prior to global average pooling), and “Bottleneck” indicates averaging immediately after the VOneBlock bottleneck. Lines indicate mean top-1 accuracy and error bars represent SD ( $n = 3$  seeds).



## E Implementation Details

### E.1 Grating Experiments

When tuning the SubcorticalBlock and when measuring its response properties, we presented 12 frames of drifting sine-wave gratings with phase shifts of 30 degrees in the interval  $[0, 360[$  degrees. Grating orientation was set to horizontal and the diameter, SF and contrast was chosen to most accurately replicate the response-property studies used for tuning (see Table B1 for the reference studies from which the stimulus set properties were taken). The background area not covered by the grating was set to 50% gray. To characterize the response properties of the VOneBlock, we adopted the same experimental paradigm as detailed above.

### E.2 SubcorticalBlock Implementation

To parameterize the fixed weights of the SubcorticalBlock, we developed a novel tuning strategy that optimizes alignment with average neuronal response properties of SF tuning, size tuning, and contrast sensitivity, using Bayesian optimization. Table E1 shows the reference values and values obtained for the six subcortical response properties. This procedure was applied independently to the P and M pathways within the SubcorticalBlock. While several hyperparameters were directly optimized, Gaussian kernel sizes were indirectly determined by computing the kernel size necessary to elicit 75% of the their total integrated response. Specifically, this formulation was used with the surround Gaussian in the DoG kernel and with the Gaussian kernel of contrast normalization layer.

**Light adaptation pooling size.** Because the primate visual system exhibits both global and local forms of light adaptation, we initially modeled luminance adaptation as a spatially local process, implemented via Gaussian filtering analogous to our contrast normalization layer. Interestingly, during Bayesian optimization, the learned filter radius consistently expanded to encompass nearly the entire image, suggesting that global rather than local adaptation better supported LGN response property prediction. To reduce computational overhead, we therefore adopted a global luminance normalization strategy.

Table E1: **Reference and tuned response property values for the SubcorticalBlock.** Response property values specific to P and M cells used to tune the SubcorticalBlock, shown alongside reference values from the original studies from which they were sourced. Six response properties were used for tuning: center, surround, excitation and inhibition radii; suppression index; and saturation index.

Resp. Property	Reference		SubcorticalBlock	
	P cells	M cells	P cells	M cells
Center Radius [deg] [58]	0.042	0.063	0.041	0.064
Surround Radius [deg] [58]	0.279	0.602	0.289	0.620
Excitation Radius [deg] [58]	0.236	0.289	0.094	0.125
Inhibition Radius [deg] [58]	0.564	0.869	0.226	0.609
Suppression Index [58]	0.808	0.719	0.710	0.610
Saturation Index [51]	0.095	0.365	0.200	0.410

**Search space.** When defining the search space for each variable in our Bayesian optimization framework, our primary objective was to minimize the introduction of inductive biases by employing search spaces as broad as feasible. In many cases, this was straightforward — for instance, we constrained parameters like the semisaturation constant,  $c_{50}$  to lie within physically meaningful bounds. However, for parameters such as the center and surround radii of the DoG filters, the radius of the Gaussian used in the contrast normalization layer, and the ratio of peak contrast sensitivity, we adopted a more heuristic approach. Specifically, we drew on values reported in the neuroscience literature to inform the bounds of the search space. For the DoG center and surround radii, we defined symmetric search intervals of centered around values reported in the reference study to which we aimed to maximize alignment [58]. Similarly, the bounds for the normalization radius were guided by reported relationships between the suppressive field and the surround Gaussian [22]. The same principle was applied to the contrast sensitivity ratio [87]. Table E2 provides a comprehensive

overview of all parameters tuned, including the corresponding search bounds, the literature references used to guide their selection, where applicable, and the final hyperparameters.

Table E2: **SubcorticalBlock hyperparameters and search space used for tuning.** Minimum ( $x_{\min}$ ) and maximum ( $x_{\max}$ ) bounds used in the Bayesian optimization for each hyperparameter of the SubcorticalBlock and hyperparameter optima obtained ( $x^*$ ). Values describe center radius of the DoG ( $r_c$ ); surround radius of the DoG ( $r_s$ ); peak contrast sensitivity ratio ( $k_s/k_c$ ); contrast normalization pooling radius ( $r_{\text{CN}}$ ); semisaturation constant ( $c_{50}$ ); contrast normalization exponent ( $n$ ). While not obtained through optimization, the kernel sizes used ( $k_{\text{DoG}}$  and  $k_{\text{CN}}$ ) are also presented. For a subset of these hyperparameters, literature references were used to inform the choice of search bounds.

Layer	$x$	P cells			M cells			Ref.
		$x_{\min}$	$x_{\max}$	$x^*$	$x_{\min}$	$x_{\max}$	$x^*$	
DoG	$r_c$ [deg]	0.034	0.050	0.047	0.050	0.76	0.76	[58]
Conv.	$r_s$ [deg]	0.223	0.335	0.224	0.482	0.722	0.534	[58]
	$k_s/k_c$	-0.068	-0.003	-0.12	-0.037	-0.002	-0.004	[87]
	$k_{\text{DoG}}$	—	—	19	—	—	33	—
Contrast Norm.	$r_{\text{CN}}$ [deg]	0.140	0.419	0.419	0.301	0.903	0.902	[22]
	$c_{50}$	0.01	1.0	1.0	0.01	1.0	0.19	—
	$n$	0.01	1.0	1.0	0.01	1.0	0.81	—
	$k_{\text{CN}}$	—	—	43	—	—	69	—

**Bayesian optimization.** We employed Bayesian optimization using the `gp_minimize` function from the Scipy library [88]. The optimization was performed over a defined parameter space for 640 evaluations, with 64 initial points generated using a Sobol sequence. The acquisition function was probabilistically selected among Lower Confidence Bound (LCB), Expected Improvement (EI), and Probability of Improvement (PI) at each iteration. The exploration-exploitation balance was controlled using  $\kappa = 1.96$  for LCB and  $\xi = 0.01$  for EI and PI.

### E.3 EVNet Variants

For all EVNet variants, we re-estimated the scaling factor applied to the VOneBlock whenever it was included, and adjusted the V1 noise Fano factor such that the accumulated Fano factor was 1. Apart from these modifications, most variants were derived by simply performing the modifications described in previous sections, with the two exceptions detailed below.

**Contrast normalization ablation.** Because the light adaptation and contrast normalization layers operate in close synchrony, removing the contrast normalization layer substantially destabilized training. In particular, the absence of contrast normalization caused activations within the SubcorticalBlock to explode, primarily due to excessively high responses from the light adaptation mechanism. This effect was most pronounced when image (or image crops) contained small, bright regions surrounded by dark backgrounds — conditions that produce low mean activations but locally high responses in Equation 2. To mitigate this, we modified the light adaptation layer’s mean computation to ignore pixel values below a threshold of  $\epsilon = 0.05$ , effectively preventing spurious amplification of isolated bright pixels.

**LGN–V2 skip connection.** When incorporating the skip connections between the SubcorticalBlock and the VOneBlock bottleneck, we maintained a total channel dimensionality of 64 at the input to the backend model. Of these, 60 channels originated from the bottleneck output, while the remaining 4 channels were adapted activation maps from the SubcorticalBlock. Given that the VOneBlock operates with a stride of 4, we applied a  $5 \times 5$  max-pooling operation with the same stride to the SubcorticalBlock activations prior to concatenation, ensuring spatial alignment and consistent feature scaling across pathways.

#### E.4 Training Details

All models were trained on an internal cluster, using 48GB NVIDIA A40 GPUs with Python 3.11, PyTorch 2.2 with CUDA 11.7, taking roughly tree days to train.

**Preprocessing.** During training, images were randomly horizontally flipped with a probability of 0.5, then resized and cropped to  $224 \times 224$  pixels. Images were normalized by subtracting and dividing by [0.5, 0.5, 0.5], with the exception of model that included the light adaptation layer of the SubcorticalBlock. During evaluation, images were resized to 256 pixels on the shorter side, followed by a center crop to  $224 \times 224$  pixels, and the same normalization was applied.

**Loss function and optimization.** Models were trained using a cross-entropy loss between ground-truth labels and predicted logits, with label smoothing [89] of 0.1. When using the ResNet50 and CORnet-Z architectures, optimization was performed using stochastic gradient descent with momentum set to 0.9 and weight decay of  $5 \times 10^{-4}$ . For EfficientNet-B0, we used RMSProp with a momentum of 0.9, smoothing constant of 0.9, and a denominator stability term of 1.0. Training was conducted for 50 epochs with a batch size of 256. We employed the 1-Cycle learning rate policy [90], where the learning rate was initialized at 4% of the maximum learning rate, increased up to maximum at 30% of the total training steps, and then annealed to  $4 \times 10^{-4}\%$  of the maximum following a cosine schedule. For the ResNet50, the maximum learning rate was set to 0.1; for the EfficientNet-B0, it was set to 0.256; and, for CORnet-Z, it was set to 0.05. When using PRIME [34], we fine-tuned a standardly trained model for an additional 50 epochs using the same training protocol, except with a maximum learning rate of 0.01 reached at 10% of the training schedule.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The main claims summarized in the abstract are properly supported by the introduction of the our novel architecture in Methods and the results described in Results.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations of our modeling approach and of our results is discussed in the Discussion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The information needed to reproduce the main experimental results is described in Methods and in Implementation Details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

#### 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The model weights will be made publicly available and the code repository will be well-documented with instructions for environment setup, training and testing our models.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All training and testing details are specified in Methods and in Implementation Details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper reports standard deviations for all results obtained with our architectures, but does not include for all models.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Details on computer resources used for training and testing are detailed in Training Details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research does not violate the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The societal impact of this research is discussed on the last paragraph of the Discussion.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

#### 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

#### 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All model authors are appropriately credited in both the main paper and the Supplementary Material. The same applies to the datasets used in this work.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.



- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: The code that will be included in the repository will be well-commented and comprehensive documentation with instructions for environment setup, model training, and evaluation will also be provided.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

#### 16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.