

Supplementary Material

A VOneNets

VOneNets [1] 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 [2], incorporating a Gabor filter bank (GFB) [3], nonlinearities for both simple and complex cells [4], and a stochastic spiking mechanism modeling neuronal variability [5] (cf. Fig. A1). The GFB parameters are sampled from empirical distributions of orientation preference, spatial frequency (SF) tuning, and receptive field (RF) size [6, 7, 8]. 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.

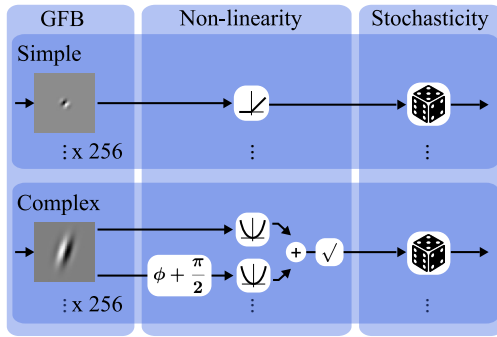


Figure A1: VOneNets simulate V1 processing upstream of standard CNNs. Each VOneNet incorporates a biologically-constrained front-end, the VOneBlock, preceding a conventional CNN. The VOneBlock consists of a fixed-weight Gabor filter bank (GFB) parameterized by empirical distributions, nonlinearities 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. [9].

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 [1], 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. [1], 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

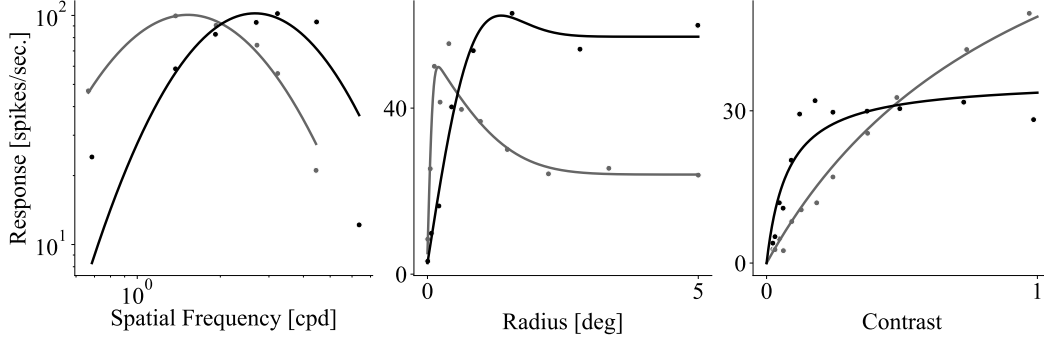


Figure B2: **Examples of empirical V1 tuning curves retrieved from the literature.** SF, size, and contrast tuning curves (left to right) for example V1 neurons. **Left** SF tuning curve of a simple (gray) and complex (black) cell to drifting grating stimuli. Markers represent the total number of F1 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. [10]). **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. [11]). **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. [12]). Data points extracted via WebPlotDigitalizer [13].

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 [14]. 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. [14], 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 [15, 16] 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 [17] neural benchmark, while response properties in V1 are assessed using the Marques2020 [18] 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 [17] 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 [15, 16] V1 alignment scores for ResNet50, VOneResNet50, and EVResNet50. Values indicate mean \pm SD ($n = 3$ seeds).

Category	Resp. Property	Models		
		ResNet50 [19]	VOneResNet50	EVResNet50
Orientation	Orientation Selective	0.975 \pm 0.024	0.999\pm0.001	0.999\pm0.001
	Circ. Variance (CV)	0.818\pm0.011	0.742 \pm 0.013	0.754 \pm 0.001
	Orth./Pref. Ratio	0.855\pm0.023	0.717 \pm 0.014	0.710 \pm 0.001
	CV Bandwidth Ratio	0.740 \pm 0.024	0.763\pm0.005	0.762 \pm 0.001
	Pref. Orientation	0.943 \pm 0.046	0.985\pm0.004	0.968 \pm 0.000
	Orth./Pref.-CV Diff.	0.766 \pm 0.016	0.885\pm0.004	0.869 \pm 0.001
	Or. Bandwidth	0.659 \pm 0.086	0.922 \pm 0.010	0.952\pm0.000
Spatial Frequency	Peak SF	0.551 \pm 0.047	0.961\pm0.002	0.961\pm0.001
	SF Bandwidth	0.826 \pm 0.019	0.962\pm0.006	0.937 \pm 0.000
	SF Selective	0.886 \pm 0.053	0.983\pm0.005	0.951 \pm 0.000
Response Selectivity	Texture Selective	0.678 \pm 0.008	0.800\pm0.004	0.774 \pm 0.001
	Modulation Ratio	0.349 \pm 0.009	0.737\pm0.002	0.736 \pm 0.000
	Texture Var. Ratio	0.794\pm0.014	0.703 \pm 0.011	0.694 \pm 0.001
	Texture Sparseness	0.663 \pm 0.032	0.927\pm0.002	0.920 \pm 0.000
RF Size	Grating Sum. Field	0.272 \pm 0.005	0.547 \pm 0.016	0.716\pm0.003
	Surround Diameter	0.156 \pm 0.000	0.361 \pm 0.015	0.736\pm0.000
Surround Mod.	Surround Sup. Index	0.389 \pm 0.023	0.373 \pm 0.003	0.614\pm0.004
Texture Modulation	Abs. Texture Mod. Idx.	0.978\pm0.019	0.942 \pm 0.004	0.934 \pm 0.000
	Texture Mod. Idx.	0.606 \pm 0.040	0.897 \pm 0.011	0.898\pm0.001
Response Magnitude	Max. Texture	0.939 \pm 0.002	0.906 \pm 0.010	0.951\pm0.001
	Max. DC	0.873\pm0.053	0.824 \pm 0.008	0.885 \pm 0.001
	Max. Noise	0.783 \pm 0.018	0.923 \pm 0.006	0.965\pm0.000

B.5 V1 Response Properties

Marques et al. [18] 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) [20] 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 [1]. 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 [1]. 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_∞ threat model, \mathcal{S} corresponds to a box constraint, while for L_1 or L_2 norms with a perturbation budget ϵ , the gradient direction is rescaled at each iteration to have the respective norm α , and the projection ensures the final adversarial input \mathbf{x}_{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_∞ , 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 [21] to conduct all the attacks.

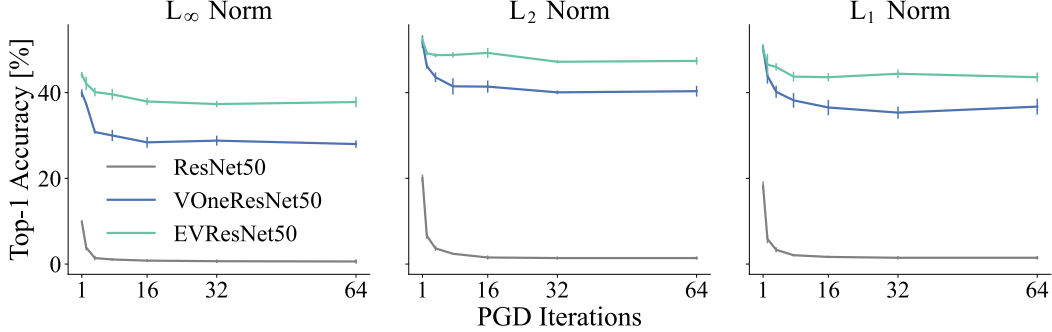


Figure C1: **Adversarial robustness is evaluated at convergence of PGD iterations.** Top-1 white-box accuracy iteration curves for PGD attacks with $\|\delta\|_\infty = 1/255$, $\|\delta\|_2 = 0.6$, $\|\delta\|_1 = 160$ constraints for ResNet50, VOneResNet50 and EVResNet50 models, evaluated on 500 images. The step size was adjusted to be ϵ 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 [22], 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. [23], replacing ∇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.

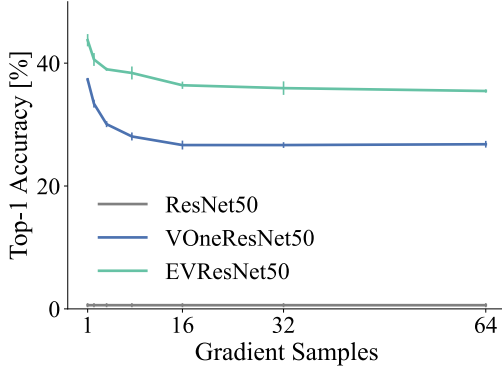


Figure C2: **Increasing the number of Monte Carlo gradient samples has limited impact on white-box attack effectiveness.** White-box PGD accuracy evaluated under $\|\delta\|_\infty = 1/255$ with 64 PGD iterations on 500 images. Increasing k from 10 to 64 leads to only marginal decreases in accuracy for 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. [23] and Carlini et al. [24], 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 [25] 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 [26], 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 [27], 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 [28], while ImageNet-Drawing comprises colored pencil sketch versions of the same images created via an image processing pipeline [29]. 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 [30] 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>.

Stylized₁₆-ImageNet. Stylized-ImageNet [14] is created by introducing different painting styles into ImageNet images through Adaptive Instance Normalization style transfer [31]. 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. [32]. This subset focuses on 16 basic categories (e.g., airplane, dog) that are supersets of 227 ImageNet classes within the WordNet hierarchy [33]. 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>.

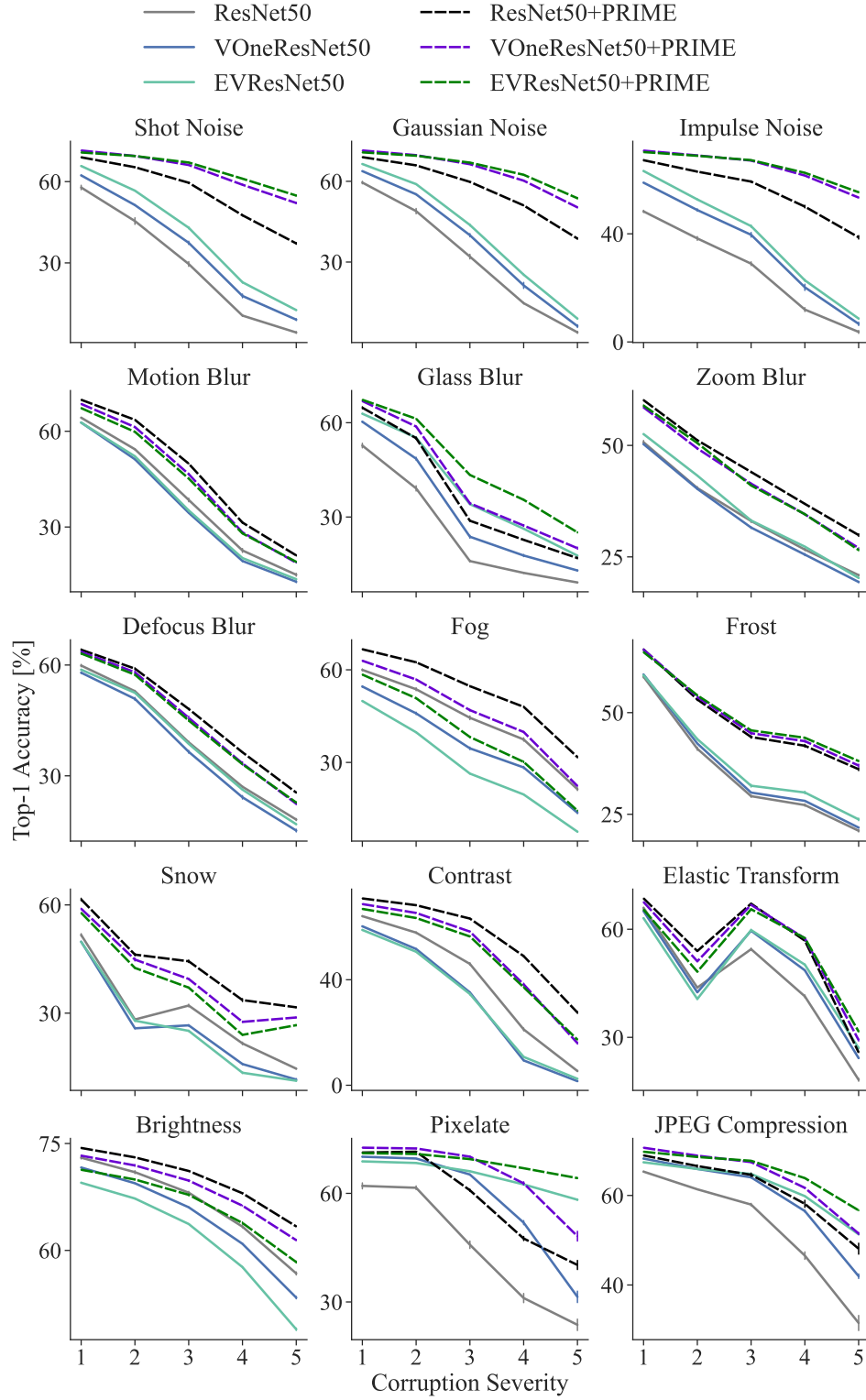


Figure C4: **Detailed results for common corruptions benchmarks.** Top-1 accuracy across 5 severity levels for the 15 individual common corruptions of ImageNet-C. Lines indicate the mean top-1 accuracy and error bars denote SD ($n = 3$ seeds)

D Additional Experiments

D.1 EVNet Variants

Table D1: **Detailed results for EVResNet50 variants, including ablations, on brain-alignment metrics.** BrainScore [15, 16] V1 alignment scores and shape bias [14] 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 ± 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 ± 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 ± 2.4
– P Cells	.350 $\pm .006$.845 $\pm .001$.854 $\pm .002$.945 $\pm .000$.738 $\pm .006$.737 $\pm .001$.910 $\pm .007$.764 $\pm .001$.965 $\pm .005$	77.8 ± 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 ± 1.0
– Light Adapt.	.364 $\pm .001$.826 $\pm .001$.859 $\pm .002$.951 $\pm .000$.720 $\pm .006$.630 $\pm .008$.908 $\pm .009$.780 $\pm .005$.936 $\pm .006$	70.0 ± 3.3
– Contrast Norm.	.374 $\pm .007$.768 $\pm .002$.868 $\pm .002$.936 $\pm .001$.562 $\pm .008$.370 $\pm .001$.921 $\pm .008$.784 $\pm .005$.938 $\pm .006$	48.0 ± 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	42.7±0.1	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	37.9±0.2	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	72.6±0.2
– VOneBlock	42.7±0.2	41.2±0.9	37.4±0.2	32.6±0.1	54.2±0.1	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 ₁₆ [%]
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	24.7\pm0.1	40.7\pm0.3
– M Cells	38.2\pm0.2	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	38.2\pm0.1	57.2\pm0.0	34.7\pm0.3	38.2\pm0.2	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	54.0\pm0.1	62.6 \pm 0.3	40.1 \pm 0.2	64.8 \pm 0.1	48.4\pm0.2	64.1 \pm 0.1	44.0 \pm 0.7
– Light Adapt.	55.1 \pm 0.5	62.8\pm0.5	42.1\pm0.4	65.0 \pm 0.7	50.3 \pm 0.9	64.0 \pm 0.2	46.3\pm0.3
– Contrast Norm.	53.3 \pm 0.1	62.7 \pm 0.2	38.2 \pm 0.4	65.2\pm0.3	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	65.2\pm0.8	44.8 \pm 0.2	63.9 \pm 0.9	40.4 \pm 0.4
+ LGN-V2 Conn.	53.9 \pm 0.2	62.8\pm0.3	38.7 \pm 0.1	65.0 \pm 0.8	48.4\pm0.2	64.5\pm0.2	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 [34], 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				
		Noise [%]	Blur [%]	Weather [%]	Digital [%]	Clean [%]
EfficientNet-B0	30.3 \pm 0.4	18.7 \pm 0.2	26.6 \pm 0.0	29.1\pm0.3	40.8 \pm 1.2	68.1\pm0.1
EVEfficientNet-B0	34.1\pm0.0	30.7\pm0.2	30.5\pm0.3	24.3 \pm 0.1	45.0\pm0.1	61.4 \pm 0.4
CORnet-Z	18.0 \pm 0.0	6.4 \pm 0.1	17.0 \pm 0.0	12.4\pm0.3	29.1 \pm 0.1	53.2\pm0.1
EVCORnet-Z	21.3\pm0.0	20.0\pm0.0	18.2\pm0.1	11.6 \pm 0.0	30.4\pm0.0	44.7 \pm 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, Stylized₁₆} 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 ₁₆ [%]
EfficientNet-B0	28.9 \pm 0.2	40.2 \pm 0.4	17.1 \pm 0.5	29.9 \pm 0.2	17.3\pm0.5	40.0 \pm 0.7
EVEfficientNet-B0	33.5\pm0.2	49.1\pm0.0	28.2\pm0.2	31.5\pm0.2	16.8 \pm 0.3	41.8\pm1.0
CORnet-Z	19.5 \pm 0.2	30.9 \pm 0.2	12.5 \pm 0.3	21.0 \pm 0.2	9.5\pm0.1	23.8 \pm 1.8
EVCORnet-Z	21.1\pm0.3	32.9\pm0.1	16.2\pm0.2	21.3\pm0.0	9.0 \pm 0.1	26.0\pm1.1

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 ± 0.1	35.0 ± 0.6	2.0 ± 0.1	43.5 ± 0.3	5.1 ± 0.1	36.2 ± 0.2	3.3 ± 0.1
EVEfficientNet-B0	45.6 ± 0.5	53.2 ± 0.5	31.1 ± 0.2	55.8 ± 0.6	40.8 ± 0.5	55.0 ± 0.6	37.7 ± 0.6
CORnet-Z	17.6 ± 0.3	24.4 ± 0.7	0.7 ± 0.0	34.8 ± 0.6	5.4 ± 0.1	34.7 ± 0.6	5.5 ± 0.0
EVCORnet-Z	31.1 ± 0.6	36.7 ± 0.9	19.0 ± 0.8	38.8 ± 0.6	26.9 ± 0.4	39.0 ± 0.6	26.2 ± 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 ± 0.1	20.9 ± 0.1	30.3 ± 0.4	28.9 ± 0.2	68.1± 0.1
EVEfficientNet-B0	39.7± 0.3	45.6± 0.4	34.1± 0.0	33.5± 0.1	61.4 ± 0.4
CORnet-Z	18.4 ± 0.0	17.6 ± 0.3	18.0 ± 0.0	19.5 ± 0.2	53.2± 0.1
EVCORnet-Z	24.5± 0.3	31.1± 0.6	21.3± 0.0	21.1± 0.3	44.7 ± 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.

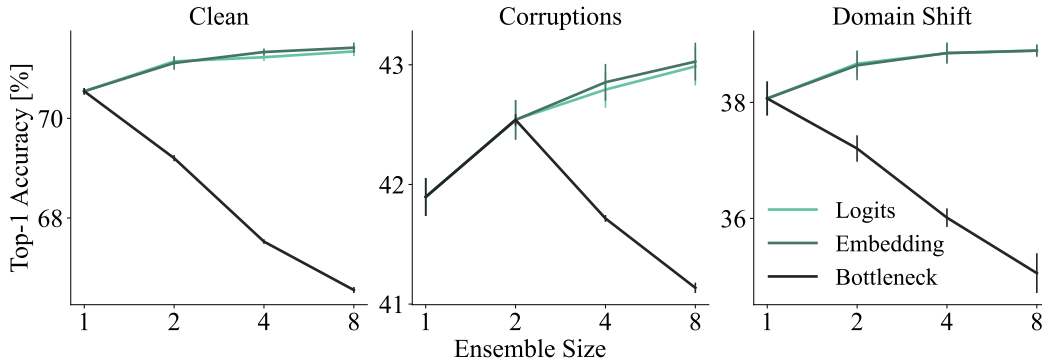


Figure D1: **Averaging EVNet logits or final embeddings yields slight performance improvements, whereas averaging bottleneck activations degrades accuracy.** Accuracy under clean, corruption, and domain-shift images for EVResNet50 ensembles of varying sizes. “Logits” denotes ensembles averaged at the 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] [35]	0.042	0.063	0.041	0.064
Surround Radius [deg] [35]	0.279	0.602	0.289	0.620
Excitation Radius [deg] [35]	0.236	0.289	0.094	0.125
Inhibition Radius [deg] [35]	0.564	0.869	0.226	0.609
Suppression Index [35]	0.808	0.719	0.710	0.610
Saturation Index [36]	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 [35]. Similarly, the bounds for the normalization radius were guided by reported relationships between the suppressive field and the surround Gaussian [37]. The same principle was applied to the contrast sensitivity ratio [38]. 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_{CN}); semisaturation constant (c_{50}); contrast normalization exponent (n). While not obtained through optimization, the kernel sizes used (k_{DoG} and k_{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	[35]
Conv.	r_s [deg]	0.223	0.335	0.224	0.482	0.722	0.534	[35]
	k_s/k_c	-0.068	-0.003	-0.12	-0.037	-0.002	-0.004	[38]
	k_{DoG}	—	—	19	—	—	33	—
Contrast Norm.	r_{CN} [deg]	0.140	0.419	0.419	0.301	0.903	0.902	[37]
	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_{CN}	—	—	43	—	—	69	—

Bayesian optimization. We employed Bayesian optimization using the `gp_minimize` function from the Scipy library [39]. 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 5×5 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 224×224 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 224×224 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 [40] 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 5×10^{-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 [41], 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 [42], 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.

References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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).
- [13] Ankit Rohatgi. *WebPlotDigitizer*. Version 5.2. URL: <https://automeris.io>.
- [14] 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.
- [15] 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).
- [16] 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).
- [17] 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.
- [18] 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.

- [19] 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.
- [20] 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.
- [21] 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).
- [22] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings*. 2014.
- [23] 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.
- [24] 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).
- [25] Dan Hendrycks and Thomas Dietterich. “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations”. In: *International Conference on Learning Representations*. 2019.
- [26] 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.
- [27] Tiago Salvador and Adam M Oberman. “ImageNet-Cartoon and ImageNet-Drawing: two domain shift datasets for ImageNet”. In: *ICML 2022 Shift Happens Workshop*. 2022.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] George A. Miller. “WordNet: a lexical database for English”. In: *Commun. ACM* 38.11 (Nov. 1995), pp. 39–41. ISSN: 0001-0782.
- [34] 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.
- [35] 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.
- [36] 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).

- [37] 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.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] Leslie N. Smith and Nicholay Topin. *Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates*. 2018. arXiv: 1708.07120 [cs.LG].
- [42] 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.