Hybrid Neural Autoencoders for Stimulus Encoding in Visual and Other Sensory Neuroprostheses

Jacob Granley
Department of Computer Science
University of California, Santa Barbara
jgranley@ucsb.edu

Lucas Relic
Department of Computer Science
University of California, Santa Barbara
lucasrelic@ucsb.edu

Michael Beyeler
Department of Computer Science
University of California, Santa Barbara
mbeyeler@ucsb.edu

Abstract

Sensory neuroprostheses are emerging as a promising technology to restore lost sensory function or augment human capabilities. However, sensations elicited by current devices often appear artificial and distorted. Although current models can predict the neural or perceptual response to an electrical stimulus, an optimal stimulation strategy solves the inverse problem: what is the required stimulus to produce a desired response? Here, we frame this as an end-to-end optimization problem, where a deep neural network stimulus encoder is trained to invert a known and fixed forward model that approximates the underlying biological system. As a proof of concept, we demonstrate the effectiveness of this hybrid neural autoencoder (HNA) in visual neuroprostheses. We find that HNA produces high-fidelity patient-specific stimuli representing handwritten digits and segmented images of everyday objects, and significantly outperforms conventional encoding strategies across all simulated patients. Overall this is an important step towards the long-standing challenge of restoring high-quality vision to people living with incurable blindness and may prove a promising solution for a variety of neuroprosthetic technologies.

1 Introduction

Sensory neuroprostheses are emerging as a promising technology to restore lost sensory function or augment human capacities [1, 2]. In such devices, diminished sensory modalities (e.g., hearing [3], vision [4, 5], cutaneous touch [6]) are re-enacted through streams of artificial input to the nervous system. For example, visual neuroprostheses electrically stimulate neurons in the early visual system to elicit neuronal responses that the brain interprets as visual percepts. Such devices have the potential to restore a rudimentary form of vision to millions of people living with incurable blindness.

However, all of these technologies face the challenge of interfacing with a highly nonlinear system of biological neurons whose role in perception is not fully understood. Due to the limited resolution of electrical stimulation, prostheses often create neural response patterns foreign to the brain. Consequently, sensations elicited by current sensory neuroprostheses often appear artificial and distorted [7, 8]. A major outstanding challenge is thus to identify a stimulus encoding that leads to perceptually intelligible sensations. Often there exists a forward model, f (Fig. 1A), constrained by empirical data, that can predict a neuronal or (ideally) perceptual response to the applied stimulus (see [9] for a recent review). To find the stimulus that will elicit a desired response, one essentially needs to find

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

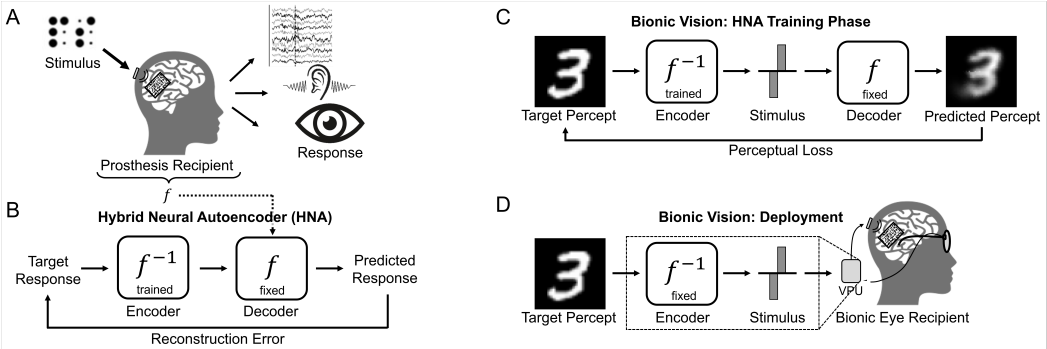

Figure 1: *A)* Sensory neuroprosthesis. A forward model (f) is used to approximate the neuronal or, ideally, perceptual response to electrical stimuli. *B)* Hybrid neural autoencoder (HNA). A deep neural encoder (f^{-1}) is trained to predict the patterns of electrical stimulation that elicit responses closest to the target. *C)* Visual neuroprostheses are one prominent application of HNA, where an encoder can be trained to predict the electrical stimulation needed to elicit a desired visual percept. *D)* The trained encoder is deployed on a vision processing unit (VPU), predicting stimuli in real-time that are decoded by the patient's visual cortex.

the inverse of the forward model, f^{-1}. However, realistic forward models are rarely analytically invertible, making this a challenging open problem for neuroprostheses.

Here we propose to approach this as an end-to-end optimization problem, where a deep neural network (DNN) (*encoder*) is trained to invert a known, fixed forward model (*decoder*, Fig. 1B). The encoder is trained to predict the patterns of electrical stimulation patterns that elicit perception (*e.g.,* vision, audition) or neural responses (*e.g.,* firing rates) closest to the target. This hybrid neural autoencoder (HNA) could in theory be used to optimize stimuli for any open-loop neuroprosthesis with a known forward model that approximates the underlying biological system.

In order to optimize end-to-end, the forward model must be differentiable and computationally efficient. When this is not the case, an alternative approach is to train a surrogate neural network, $\hat{f}$, to approximate the forward model [10–13]. However, even well-trained surrogates may result in errors when included in our end-to-end framework, due to the encoders' ability to learn to exploit the surrogate model. We therefore also evaluate whether a surrogate approach to HNA is suitable.

To this end, we make the following contributions:

- We propose a hybrid neural autoencoder (HNA) consisting of a deep neural encoder trained to invert a fixed, numerical or symbolic forward model (decoder) to optimize stimulus selection. Our framework is general and addresses an important challenge with sensory neuroprostheses.
- As a proof of concept, we demonstrate the utility of HNA for visual neuroprostheses, where we predict electrode activation patterns required to produce a desired visual percept. We show that the HNA is able to produce high-fidelity, patient-specific stimuli representing handwritten digits and segmented images of everyday objects, drastically outperforming conventional approaches.
- We evaluate replacing a computationally expensive or nondifferentiable forward model with a surrogate, highlighting benefits and potential dangers of popular surrogate techniques.

2 Background

Sensory Neuroprostheses Recent advances in neural understanding, wearable electronics, and biocompatible materials have accelerated the development of sensory neuroprostheses to restore perceptual function to people with impaired sensation. Use of neuroprostheses typically requires invasive implants that elicit neural responses via electrical, magnetic, or optogenetic stimulation. Two of the most promising applications are cochlear implants, which stimulate the auditory nerve to elicit sounds [3], and visual implants (see next subsection) to restore vision to the blind. However, a variety of other devices are in development; for instance, to restore touch [6, 14] or motor function [15]. The latter differ from other sensory neuroprostheses in that they generate stimuli using motor

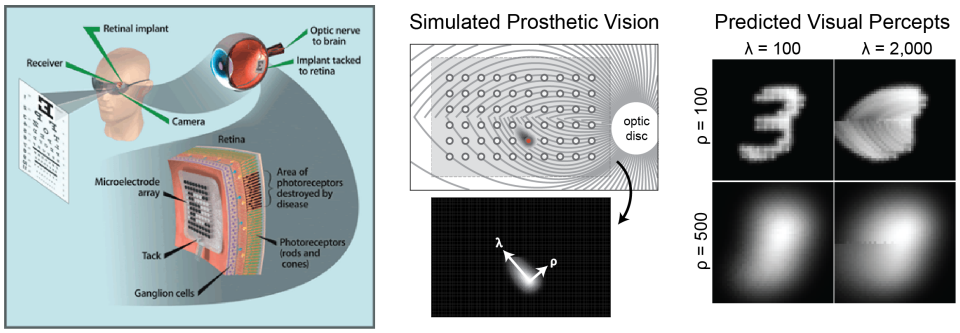

Figure 2: *Left*: Visual prosthesis. Incoming target images are transmitted from a camera to an implant in the retina, which encodes the image as an electrical stimulus pattern. *Center*: Electrical stimulation (red disc) of a nerve fiber bundle (gray lines) leads to elongated tissue activation (gray shaded region) and phosphenes whose shape can be described by two parameters, λ (axonal spread) and ρ (radial spread). *Right*: Predicted percepts for an MNIST digit using varying ρ and λ values.

feedback (*closed loop*) [16, 17]. In the absence of feedback (*open loop*), a proper stimulus encoding is paramount to the success of these devices.

Restoring Vision to the Blind For millions of people who are living with incurable blindness, a visual prostheses (*bionic eye*, Fig. 2, *left*) may be the only treatment option [18]. Analogous to cochlear implants, these devices electrically stimulate surviving cells in the visual pathway to evoke visual percepts (*phosphenes*), which can support simple behavioral tasks [5, 19, 20].

A common misconception is that each electrode in the array can be thought of as a pixel in an image; to generate a complex visual experience, one then simply needs to turn on the right combination of pixels [21]. However, recent evidence suggests that phosphenes often appear distorted (*e.g.*, as lines, wedges, and blobs) and vary drastically across subjects and electrodes [4, 7].

Phosphene appearance has been best characterized in epiretinal implants, where inadvertent activation of nerve fiber bundles (NFBs) in the optic fiber layer of the retina leads to elongated phosphenes [22, 23] (Fig. 2, *center*). To this end, Granley *et. al* [24] developed a forward model to predict phosphene shape as a function of both neuroanatomical parameters (*i.e.*, location of the stimulating electrode) and stimulus parameters (*i.e.*, pulse frequency, amplitude, and duration). With this model, phosphenes are primarily characterized by two main parameters, ρ and λ, which dictate the size and elongation of the elicited phosphene, respectively (Fig. 2, *right*). These parameters can be determined using psychophysical tasks (*e.g.*, drawings, brightness ratings) [22, 24], and although they vary drastically across patients [22], they do not change much over time [25, 26]. Stimulation from multiple electrodes is nonlinearly integrated into a combined perception, and if two electrodes happen to activate the same NFB, they might not generate two distinct phosphenes.

3 Related Work

The conventional 'naive' encoding strategy sets the amplitude of each electrode to the brightness of the corresponding pixel in the target image [5, 27], making the stimulus a down-sampled version of the target. Although simple, this strategy only works with near-linear forward models, cannot account for real phosphene data, and often leads to unrecognizable percepts (Fig. 2, *right*) [7, 22].

Many alternative stimulation strategies have been proposed [28]. Shah *et al.* [29] used a greedy approach to iteratively select the stimuli that best recreate a desired neural activity pattern over a given temporal window, assuming that the brain would integrate them into a coherent visual percept. Ghaffari *et al.* [30] used a neural network surrogate model combined with an interior point algorithm to optimize for localized, circular neural activation patterns for individual electrodes. Fauvel *et al.* [31] used human in-the-loop Bayesian optimization to achieve encodings perceptually favored by the simulated patient. Spencer *et al.* [32] proposed framing stimulus encoding as inversion of a forward model of neural activation patterns, but to approximate the inverse, their approach either requires simplification or is NP-hard [32].

Van Steveninck *et al.* [33] proposed an end-to-end optimization strategy with a fixed phosphene model, similar to HNA. However, their approach crucially differs from ours in its inclusion of a secondary DNN to post-process the predicted phosphenes. This is a critical limitation, because a low reconstruction loss does not reveal whether a high-fidelity encoder was learned or whether the secondary decoder network simply learned to correct for the encoder's mistakes. In addition, they used an unrealistic phosphene model that simply upscales and smooths a binary stimulus pattern. It is therefore not clear whether their results would generalize to real visual prosthesis patients.

Relic *et al.* [10] also utilized the end-to-end approach, but without the secondary decoder network used in [33]. They used a more realistic phosphene model, which accounts for some spatial distortions (*e.g.*, axonal streaks), but not the effects of stimulus parameters. Since including a realistic phosphene model in the loop is not straightforward, they instead trained a surrogate neural network to approximate the forward model. We re-implemented Relic's surrogate approach in this paper as a baseline method to compare against, as described in Section 4.

Taken together, we identified three main limitations of previous work that this study aims to address:

1) **Unrealistic forward models.** Most previous approaches (*e.g.*, [29, 32, 33]) use an overly simplified forward model that cannot account for empirical data [7, 22]. We overcome this limitation by developing (and inverting) a differentiable version of a neurophysiologically informed and psychophysically validated phosphene model [24] that can account for the effects of stimulus amplitude, frequency, and pulse duration on phosphene appearance.
2) **Optimization of neural responses.** Most previous approaches (*e.g.*, [29, 32]) focus on optimizing neural activation patterns in the retina in response to electrical stimulation ("bottom-up"). However, the visual system undergoes extensive remodeling during blinding diseases such as retinitis pigmentosa [34]. Thus the link between neural activity and visual perception is unclear. We overcome this limitation by inverting a phenomenological ("top-down") model constrained by behavioral data that predicts visual perception directly from electrical stimuli [22, 24].
3) **Objective function** Most previous approaches rely on minimizing mean squared error (MSE) between the target and reconstructed image. Although simple and efficient, MSE is known to be a poor measure of perceptual dissimilarity for images [35] and does not align well with human assessments of image quality [36]. We overcome this limitation by proposing a joint perceptual metric that combines mean absolute error (MAE), VGG, and Laplacian smoothing losses.

4 Methods

Problem Formulation We consider a system where there is some known forward process f mapping stimuli to responses $f : \mathcal{S} \mapsto \mathcal{R}, f(\mathcal{S}) \subset \mathcal{R}$. In the case of visual prostheses, f may map stimuli to visual percepts. f may optionally be parameterized by patient-specific parameters ϕ.

Finding the best stimulus for an arbitrary target response $\mathbf{t} \in \mathcal{R}$ is equivalent to finding the inverse of f. However, since not every response can be perfectly reproduced by a stimulus, the true inverse of f is not well defined. We therefore seek the pseudoinverse (still denoted as f^{-1} for simplicity) instead, which outputs the stimuli that produce the closest response to $\mathbf{t}$ under some distance metric d:

$$f^{-1}(\mathbf{t}, \phi) \coloneqq \arg\min_{\mathbf{s} \in \mathcal{S}} d(f(\mathbf{s}; \phi), \mathbf{t}). \tag{1}$$

This problem is straightforward to solve using an autoencoder approach, where a learned encoder f^{-1} is trained to invert the fixed decoder f (i.e., forward model).

Encoder We approximate the pseudoinverse f^{-1} with a DNN encoder $\hat{f}^{-1}(\mathbf{t}, \phi; \theta)$ with weights θ, trained to minimize the distance d between the target image $\mathbf{t}$ and predicted image $\hat{\mathbf{t}}$:

$$\min_{\theta,\, \phi \sim p(\phi)} d(\mathbf{t}, \hat{\mathbf{t}}) \tag{2}$$

$$\hat{\mathbf{t}} = f(\, \hat{f}^{-1}(\mathbf{t}, \phi;\, \theta);\ \phi), \tag{3}$$

where ϕ is sampled from a uniform random distribution spanning the empirically observed range of patient-specific parameters [22, 24].

We use a simple architecture consisting solely of fully connected (FC) and batch normalization (BN) [37] layers (1.4M trainable parameters). First, the target $\mathbf{t}$ is flattened and input to a FC layer. In parallel, the patient parameters ϕ are input to a BN layer and two hidden FC layers. Then, the outputs of these two paths are concatenated, and the combined vector fed through two FC layers, producing an intermediate representation $\mathbf{x}$. Amplitudes are predicted from $\mathbf{x}$ with a FC layer. The amplitudes are then concatenated with $\mathbf{x}$, put through a BN layer, and used to predict frequency and pulse duration, each with a FC layer. The outputs are merged into a stimulus matrix $\hat{\mathbf{s}}$. All layers use leaky ReLU activation, except for stimulus outputs, which use ReLU to enforce nonnegativity.

Decoder The HNA decoder is a differentiable approximation of the underlying biological system, and describes the transform from stimulus to response. For our decoder f, we use a reformulated but equivalent version of the model described in [24]. This model is specific to epiretinal prostheses; analogous models exist for other neuroprostheses (*e.g.*, auditory [38–43], tactile and somatosensory [44–48]), and could potentially be adapted for use with HNA. We use a square 15×15 array of 150μm electrodes, spaced 400μm apart and centered on the fovea. The size and scale of this device are motivated by similar designs in real epiretinal implants.

f takes as input a stimulus matrix $\mathbf{s} \in \mathbb{R}_{\geq 0}^{n_e \mathrm{x} 3}$, where the stimulus on each electrode ($\mathbf{s_e}$) is a biphasic pulse train described by its frequency, amplitude, and pulse duration. A simulated map of retinal NFBs is used to predict phosphene shape. Following [22], each retinal ganglion cells' activation is assumed to be the maximum stimulation intensity along its axon. Axon sensitivity is assumed to decay exponentially with i) distance to the stimulating electrode (radial decay rate, ρ) and distance to the soma along the curved axon (axonal decay rate, λ). To account for stimulus properties [24], ρ, λ, and the per-electrode brightness are scaled by three simple equations $F_{\text{size}}(\mathbf{s_e}, \phi)$, $F_{\text{streak}}(\mathbf{s_e}, \phi)$, and $F_{\text{bright}}(\mathbf{s_e}, \phi)$, respectively.

The brightness of a pixel located at the point $\mathbf{x} \in \mathbb{R}^2$ in the output image is given by

$$f(\mathbf{s}; \phi) = \max_{\mathbf{a} \in A} \sum_{e \in E} F_{\text{bright}}(\mathbf{s_e}, \phi) \exp\left(\frac{-||\mathbf{x} - \mathbf{e}||_2^2}{2\rho^2 F_{\text{size}}(\mathbf{s_e}, \phi)} + \frac{-d_s(\mathbf{x}, \mathbf{a})^2}{2\lambda^2 F_{\text{streak}}(\mathbf{s_e}, \phi)} \right) \tag{4}$$

where A is the cells' axon trajectory, E is the set of electrodes, $\phi = \{\rho, \lambda, ...\}$ is a set of 12 patient-specific parameters, and d_s is the path length along the axon trajectory [49]from $\mathbf{a}$ to $\mathbf{x}$:

$$d_s(\mathbf{x}, \mathbf{a}) = \int_{\mathbf{a}}^{\mathbf{x}} \sqrt{A(\theta)^2 + \left(\frac{dA(\theta)}{d\theta} \right)^2} \, d\theta. \tag{5}$$

This model (f) can be fit to individual patients; however, it is computationally slow and not differentiable. For more details on these equations, see [24]. We therefore considered two approaches:

- **Differentiable Model:** We reformulated equations 4 and 5 into an equivalent set of parallelized matrix operations, implemented in Tensorflow [50]. Significant efforts were put towards developing a model optimized for XLA engines on GPU, resulting in speedups of up to 5000x compared to the model as presented in [24], enabling large-scale gradient descent. To enforce differentiability, we numerically approximated d_s using $|A| = 500$ axon segments per axon.
- **Surrogate Model:** We also implemented the surrogate approach from [10] as a baseline method, where f is approximated with another DNN $\hat{f}_\phi(\mathbf{s}; \theta_f)$ with weights θ_f. To achieve this we generated 50,000 percepts using randomly selected stimuli passed through f and fit a DNN to produce identical images. As f is highly dependent on patient-specific parameters ϕ, we generated new data and fit a separate surrogate model for each ϕ in our experimental set. Specific implementation details of the surrogate are presented in Appendix A. Our implementation improves upon [10] by using the more advanced phosphene model described above, which accounts for effects of stimulus properties and allows for optimization of stimulus frequency in addition to amplitude.

Metrics To measure perceptual similarity, we use a joint perceptual objective consisting of a VGG [51] similarity term, a mean absolute error (MAE) term, and a smoothness regularization term. The MAE term is given by $L_{\text{MAE}} = \frac{1}{|\mathbf{t}|}||\mathbf{t} - \hat{\mathbf{t}}||_1$.

The VGG term aims to capture higher-level differences between images [33, 52]. The target image and reconstructed phosphene are input to VGG-19 pretrained on ImageNet [53], and the MSE between

the activations on a downstream convolutional layer is computed. Let V_l be a function that extracts the activations of the l-th convolutional layer for a given image. The VGG loss is then defined as $L_{\text{VGG}} = \frac{1}{|\mathbf{t}|}||V_l(\mathbf{t}) - V_l(\hat{\mathbf{t}})||_2^2$.

We also include a smoothing regularization term. This term imposes a loss on the second spatial derivative of the predicted image. A low second derivative implies that where the target image does change, it changes slowly. We found this encouraged smoother, more connected phosphenes. To approximate the second derivative, we convolve the image with a Laplacian filter [54] of size k, denoted by $Lap(\cdot, k)$, and compute the mean. The smoothness loss is given by:

$$L_{\text{Smooth}} = \frac{1}{|\hat{\mathbf{t}}|} \sum_i \left(\frac{\partial^2}{dx^2} \hat{\mathbf{t}} \right)_i = \frac{1}{|\hat{\mathbf{t}}|} \sum_i Lap(\hat{\mathbf{t}}, k)_i. \tag{6}$$

Our final objective is the weighted sum of the three individual losses, given by Eq. 7, where α and β are hyperparameters controlling the relative weighting of the three terms.

$$d = L_{\text{MAE}} + \alpha L_{\text{Smooth}} + \beta L_{\text{VGG}}. \tag{7}$$

We also implement a secondary metric to quantify phosphene recognizability, applicable only for the MNIST reconstruction task. We first pre-train a classifier network on the MNIST targets until it reaches 99% test accuracy, and then fix the weights. The relative accuracy (RA) is then defined as the ratio of the classifiers accuracy on the reconstructed images to its accuracy on the targets $RA = ACC/ACC(\mathbf{t})$. A perfect encoder would result in $RA = 100\%$. A similar process was not possible for the COCO task due to the possibility of having multiple objects in each target image.

Training/Optimization We trained using Tensorflow 2.7 [50] on a single NVIDIA RTX 3090 with 24GB memory. Stochastic gradient descent with Nesterov momentum was used to minimize the joint perceptual loss. We used a batch size of 16 due to memory limitations imposed by f. The amplitude, frequency predictions are scaled by 2, 20 respectively, while the pulse duration predictions were shifted by 1e-3 prior to being fed through the decoder. This encourages the initial predictions of the network to be in a reasonable range. The Laplacian filter size k is set to 5. We choose l to be first convolutional layer in the last block using cross validation (see Appendix B). Similarly, we perform cross validation to find the best values for α and β. Instead of using one value, we found that incrementally increasing the weighting of the VGG loss (β) from 0 while simultaneously decreasing the initially high weight on the smoothing constraint (α) was crucial for performance, especially when the range of allowed ϕ values was large (see Appendix B).

Datasets We first evaluated on handwritten digits from MNIST [55], enabling comparison to previous works [10]. Images preprocessing consisted of resizing the target images to the same shape as the output of f (49x49). We also evaluate on more realistic images of common objects from the MS-COCO [56] dataset. We selected a subset of 25 of the MS-COCO object categories deemed more likely to be encountered by blind individuals (*e.g.* people, household objects), and use only images that contain at least one instance of these objects. We further filter out images by various other criteria, such as being too cluttered or too dim. This process results in a total of approximately 47K training images and 12K test images. See Appendix C for a full description of the selection process.

Natural images often contain too much detail to be encoded with prosthetic vision. While scene simplification strategies exist [57], we focus on the encoding algorithm, so we simply used the ground-truth segmentation masks to segment out the objects of interest. The images were then converted to grayscale, and resized to 49×49 pixels.

5 Results

5.1 MNIST

The phosphenes produced from the HNA, surrogate, and naive encoders on the MNIST test set are shown in Fig. 3 and performance is summarized in Table 1. For each MNIST sample, the target image is input to the encoder, which predicts a stimulus. The stimulus is fed through the true forward model f, and the predicted phosphene is shown. Since the surrogate method must be retrained for each ϕ, results are only shown for 4 simulated patients. Our proposed approach outperformed the baselines across all metrics (see Appendix D for a comparison of stimuli).

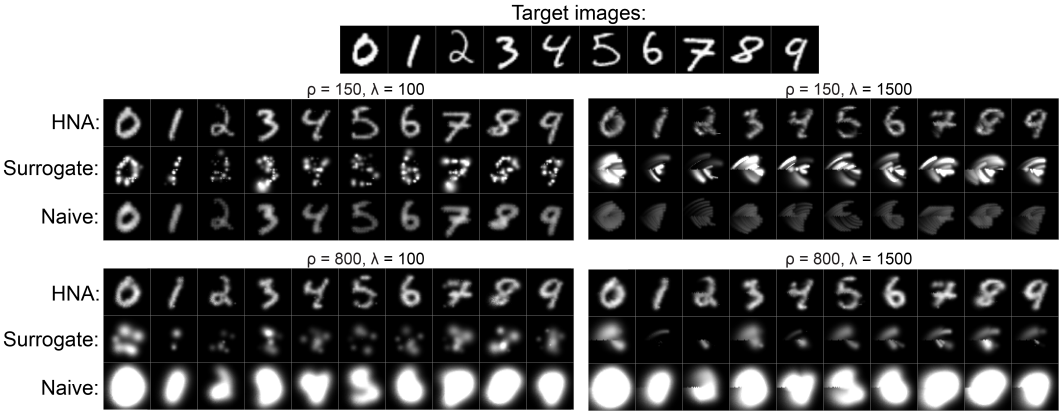

Figure 3: Reconstructed MNIST targets for HNA, surrogate, and naive encoders across 4 specific simulated patients. Note that the brightness of the naive encoder is clipped for display

Table 1: MNIST performance

Encoding	ρ=150 λ=100			ρ=150 λ=1500			ρ=800 λ=100			ρ=800 λ=1500		
	Joint Loss	MAE	RA	Joint Loss	MAE	RA	Joint Loss	MAE	RA	Joint Loss	MAE	RA
Naive	1.161	0.1855	90.3	1.442	0.214	78.1	8.152	1.500	34.8	8.780	1.726	28.8
Surrogate	2.509	0.1351	53.8	3.118	0.2431	30.7	1.692	0.2135	19.9	1.694	0.2237	18.1
HNA	**0.559**	**0.064**	**98.1**	**1.029**	**0.1412**	**89.3**	**0.913**	**0.113**	**95.9**	**0.957**	**0.126**	**94.8**

5.2 COCO

The phosphenes produced by HNA and the naive encoder for the segmented COCO dataset are shown in Fig. 4. We omit the surrogate results due to its poor perceptual performance on MNIST. Averaged across all ϕ, HNA had a joint loss of 0.713 on the test set and MAE of 0.1408, while the naive encoder had a joint loss of 1.873 and MAE of 0.2830.

5.3 Modeling Patient-to-Patient Variations

MNIST encoder performance across simulated patients (ϕ) is shown in Fig. 5. Since the surrogate encoder has to be retrained for each patient, comparison is infeasible. To visualize the effects of changing ρ and λ on the produced phosphenes, Fig. 5A shows the result of encoding two example MNIST digits, both using the naive method and our encoder. As λ increases, the naive phosphenes appear increasingly elongated, and as ρ increases, the phosphenes become increasingly large and blurry. The phosphenes from HNA are slightly too dim and disconnected at low ρ, but are relatively stable across other values of ρ and λ.

To compare performance across the entire dataset, we computed the average test set loss across the same range of ρ and λ (Fig. 5B). The encoder performs well across a wide range of simulated patients, with larger loss only at low ρ. The naive method performs well only on a limited set of ϕ, with small λ and $\rho \approx 200$. The naive loss was higher than the learned encoder at every simulated point. Random sampling of ρ and λ for each image results in a joint loss of 0.921, MAE of 0.120, and RA of 94.0% for HNA, while the naive encoder results in a joint loss of 3.17, MAE of 0.596, and RA of 63.6%. The same analysis yielded similar results on COCO (Appendix E). An analysis across other parameters is presented in Appendix F.

In order for prosthetic vision to be useful, different instances of the same objects would ideally produce similar phosphenes, allowing for consistent perception. To evaluate whether our model achieves this, we cluster the target images and resulting phosphenes using t-SNE [58] shown in Fig. 5C. The ground truth images form clusters corresponding to the digits 0-9. The phosphenes from our encoder roughly form similar, slightly less separated groupings, whereas the naive phosphenes do not. To ensure that this was not the result of bad t-SNE hyperparameters, we repeated the clustering across different perplexities and learning rates, obtaining similar or worse results.

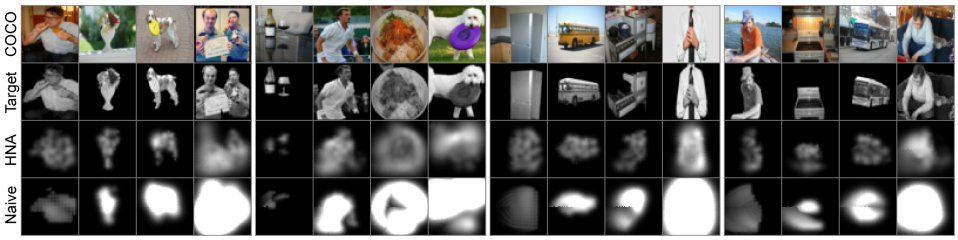

Figure 4: Original (*top row*), segmented (*second row*), and reconstructed targets for the COCO dataset, for both HNA (*third row*) and naive encoders (*bottom row*). Left to right within each block of 4 images, ρ takes values of 200, 400, 600, 800. Left to right across blocks, λ takes values of 250, 750, 1250, 2000. Note that the brightness of the naive method is clipped for display.

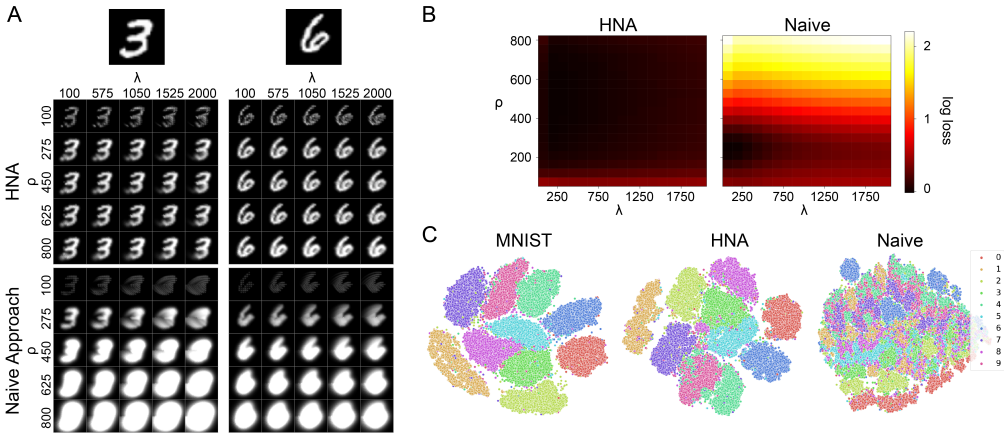

Figure 5: Encoder performance across simulated patients (varying ρ and λ) on the MNIST dataset. ***A***: Target, HNA encoder, and naive encoder phosphenes for two example digits. ***B***: Heatmaps showing the log joint loss across ρ and λ for HNA and naive encoders. ***C***: T-SNE clusterings on original MNIST targets, HNA reconstructed phosphenes, and naive reconstructed phosphenes.

5.4 Joint Perceptual Error Ablation Study

To show that the joint perceptual metric performs better than any of its individual components, we train models using just the VGG loss and just MAE loss. Shown are values for ρ=150 and λ=600. As mentioned previously, encoders trained using just VGG loss fail to converge, thus we pretrain the VGG encoder using MAE and smoothing loss, then transition to using only VGG. We do not consider ablating the smoothing term (Eq. 6) because it is simply a regularization term. Fig. 6 shows the phosphenes produced by HNA trained on the joint, VGG-only, and MAE-only loss.

The VGG encoder had a test VGG loss of 4% lower than the joint model, but its produced phosphenes are oversmoothed and blurry. The MAE encoder had a final test MAE of 9% lower than the joint model, but its produced phosphenes are disconnected and low-quality. The joint model had a RA of 99.0%, the VGG encoder had a RA of 95.9%, and the joint model had a RA of 77.6%

6 Discussion

Visual Prostheses We found that HNA is able to produce high-fidelity stimuli from the MNIST and COCO datasets that outperform conventional encoding strategies across all tested conditions. Importantly, HNA produces phosphenes that are consistent across representations of the same object (Fig. 5C), which is critical to allowing prosthesis users to learn to associate certain visual patterns with specific objects. On the MNIST task, HNA produced high quality reconstructions, nearly matching the targets (Figure 3). On the harder COCO task, HNA significantly outperformed the naive encoder, but was still unable to capture all of the detail in the images. In Appendix G, we demonstrate that this is largely due to the implant's limited spatial resolution and not a fundamental limitation of HNA.

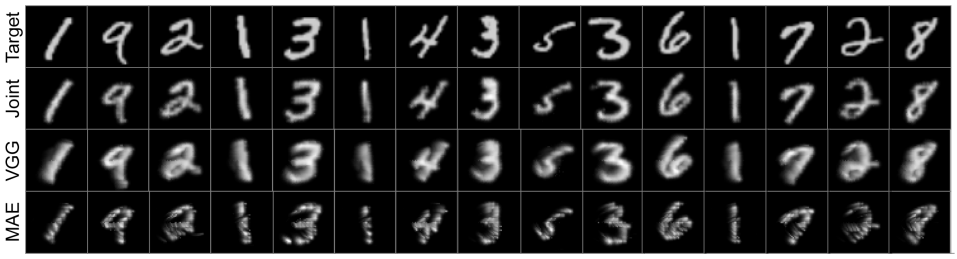

Figure 6: MNIST images for HNA encoders trained using the joint, VGG-only, and MAE-only loss.

Another advantage of the HNA is that it can be trained to predict stimuli across a wide range of patient-specific parameter values ϕ, whereas the conventional naive encoder works well only for small values of ρ and λ. This may be one reason why the naive encoding strategy has been shown to lead to substantial individual differences in visual outcomes [19, 59]. Our results suggest that stimuli produced with HNA may be able to reduce at least some amount of this patient-to-patient variability.

Furthermore, HNA also proved superior to a surrogate forward model. The latter offer an alternative when the forward model is computationally expensive or not differentiable. Understandably, any inaccuracies in the surrogate model will propagate to the learned encoder during training. However, we observed that even for well trained surrogates, the encoder may still learn to exploit the inexact surrogate instead of learning to invert the true model (see Appendix A). It is possible that this exploitation could be mitigated to some extent by adversarially-robust training techniques [60]. We suspect that the surrogate method's inferior performance here compared to [10] can be explained by our larger stimulus search space. Thus, we cannot currently suggest HNA for surrogate forward models, unless the forward model is sufficiently simple or has a small stimulus space.

Deployment HNA encoders must be lightweight enough to be deployed in resource-limited neuroprosthetic environments. Our encoder's single image inference time was 1.2ms on GPU and 4ms on CPU. Future work could reduce these numbers through network pruning, mixed precision, and architecture search. Low-power Edge AI accelerators (*e.g.*, Intel's Neural Compute Stick) and dedicated neuromorphic hardware (*e.g.*, BrainChip's Akida SoC) may provide another solution.

Broader Impacts While our work is presented in the context of visual prostheses, the HNA framework may apply to any sensory neuroprosthesis where stimulus selection can be informed by a numeric or symbolic forward model. For example, HNA could be used in cochlear implants [3] to choose stimuli that result in a desired sound, and in spinal cord implants [15] to find the best way to relay neural signals through a damaged section of the spinal cord. Conveniently, the forward models required by HNA have already been developed for a range of applications [38–48]. However, HNA might not apply to all neural interfaces, such as systems without a clear neural or perceptual target (*e.g.*, deep brain stimulation for the treatment of Parkinson's [61]) or closed-loop systems [16, 62].

Limitations Despite HNA's potential, the current implementation has a number of limitations. First, as presented the HNA encoder only applies to static targets. Hence dynamic targets must be split into individual frames and encoded separately. However, one approach might be to encode entire stimulus sequences (instead of frames) that are optimized to reconstruct the dynamic target sequence.

Second, HNA works best if there is an accurate forward model mapping from stimulus space to perception. However, Appendix H shows that HNA may still give benefits over a naive encoding even when patient-specific parameters are unknown or mis-specified. In general, if a prosthesis elicits similar results across patients, then a non-patient-specific model would suffice.

Third, the current works deals only with simulated patients. The use of a DNN for stimulus encoding in real patients may raise safety concerns. Since we cannot examine the process by which stimuli are chosen, it is possible that HNA might produce harmful stimuli that could lead to serious adverse events (*e.g.*, seizures). However, this concern is mitigated by the fact that most neuroprostheses are equipped with firmware responsible for ensuring stimuli stay within FDA-approved safety limits.

7 Conclusion

In summary, this paper proposes a hybrid autoencoder structure as a general framework for stimulus optimization in sensory neuroprostheses and, as a proof of concept, demonstrates its utility on the prominent example of visual neuroprostheses, drastically outperforming conventional encoding strategies. This constitutes an important step towards the long-standing challenge of restoring high-quality vision to people living with incurable blindness and may prove a promising solution for a variety of neuroprosthetic technologies.

8 Acknowledgements

This work was supported by the National Institutes of Health (NIH R00 EY-029329 to MB).

References

[1] Caterina Cinel, Davide Valeriani, and Riccardo Poli. Neurotechnologies for Human Cognitive Augmentation: Current State of the Art and Future Prospects. *Frontiers in Human Neuroscience*, 13:13, January 2019.

[2] Eduardo Fernandez. Development of visual Neuroprostheses: trends and challenges. *Bioelectronic Medicine*, 4(1):12, August 2018.

[3] Blake S. Wilson, Charles C. Finley, Dewey T. Lawson, Robert D. Wolford, Donald K. Eddington, and William M. Rabinowitz. Better speech recognition with cochlear implants. *Nature*, 352(6332):236–238, July 1991. Number: 6332 Publisher: Nature Publishing Group.

[4] Eduardo Fernández, Arantxa Alfaro, Cristina Soto-Sánchez, Pablo Gonzalez-Lopez, Antonio M. Lozano, Sebastian Peña, Maria Dolores Grima, Alfonso Rodil, Bernardeta Gómez, Xing Chen, Pieter R. Roelfsema, John D. Rolston, Tyler S. Davis, and Richard A. Normann. Visual percepts evoked with an intracortical 96-channel microelectrode array inserted in human occipital cortex. *Journal of Clinical Investigation*, 131(23):e151331, December 2021.

[5] Yvonne Hsu-Lin Luo and Lyndon da Cruz. The Argus® II Retinal Prosthesis System. *Progress in Retinal and Eye Research*, 50:89–107, January 2016.

[6] Daniel W. Tan, Matthew A. Schiefer, Michael W. Keith, James Robert Anderson, Joyce Tyler, and Dustin J. Tyler. A neural interface provides long-term stable natural touch perception. *Science Translational Medicine*, 6(257):257ra138–257ra138, October 2014. Publisher: American Association for the Advancement of Science.

[7] Cordelia Erickson-Davis and Helma Korzybska. What do blind people "see" with retinal prostheses? Observations and qualitative reports of epiretinal implant users. *PLOS ONE*, 16(2):e0229189, February 2021. Publisher: Public Library of Science.

[8] Craig D. Murray. Embodiment and Prosthetics. In Pamela Gallagher, Deirdre Desmond, and Malcolm MacLachlan, editors, *Psychoprosthetics*, pages 119–129. Springer, London, 2008.

[9] Bingni W. Brunton and Michael Beyeler. Data-driven models in human neuroscience and neuroengineering. *Current Opinion in Neurobiology*, 58:21–29, October 2019.

[10] Lucas Relic, Bowen Zhang, Yi-Lin Tuan, and Michael Beyeler. Deep Learning–Based Perceptual Stimulus Encoder for Bionic Vision. In *Augmented Humans 2022*, AHs 2022, pages 323–325, New York, NY, USA, March 2022. Association for Computing Machinery.

[11] David Montes de Oca Zapiain, James A. Stewart, and Rémi Dingreville. Accelerating phase-field-based microstructure evolution predictions via surrogate models trained by machine learning methods. *npj Computational Materials*, 7(1):1–11, January 2021. Number: 1 Publisher: Nature Publishing Group.

[12] Mohammad Amin Nabian and Hadi Meidani. A Deep Neural Network Surrogate for High-Dimensional Random Partial Differential Equations. *Probabilistic Engineering Mechanics*, 57:14–25, July 2019. arXiv:1806.02957 [physics, stat].

[13] Stefanos Nikolopoulos, Ioannis Kalogeris, and Vissarion Papadopoulos. Non-intrusive surrogate modeling for parametrized time-dependent partial differential equations using convolutional autoencoders. *Engineering Applications of Artificial Intelligence*, 109:104652, March 2022.

[14] Gregg A. Tabot, John F. Dammann, Joshua A. Berg, Francesco V. Tenore, Jessica L. Boback, R. Jacob Vogelstein, and Sliman J. Bensmaia. Restoring the sense of touch with a prosthetic hand through a brain interface. *Proceedings of the National Academy of Sciences*, 110(45):18279–18284, November 2013.

[15] Marco Capogrosso, Tomislav Milekovic, David Borton, Fabien Wagner, Eduardo Martin Moraud, Jean-Baptiste Mignardot, Nicolas Buse, Jerome Gandar, Quentin Barraud, David Xing, Elodie Rey, Simone Duis, Yang Jianzhong, Wai Kin D. Ko, Qin Li, Peter Detemple, Tim Denison, Silvestro Micera, Erwan Bezard, Jocelyne Bloch, and Grégoire Courtine. A brain–spine interface alleviating gait deficits after spinal cord injury in primates. *Nature*, 539(7628):284–288, November 2016.

[16] Christopher A. R. Chapman, Noah Goshi, and Erkin Seker. Multifunctional Neural Interfaces for Closed-Loop Control of Neural Activity. *Advanced Functional Materials*, 28(12):1703523, 2018. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adfm.201703523.

[17] Fabien B. Wagner, Jean-Baptiste Mignardot, Camille G. Le Goff-Mignardot, Robin Demesmaeker, Salif Komi, Marco Capogrosso, Andreas Rowald, Ismael Seáñez, Miroslav Caban, Elvira Pirondini, Molywan Vat, Laura A. McCracken, Roman Heimgartner, Isabelle Fodor, Anne Watrin, Perrine Seguin, Edoardo Paoles, Katrien Van Den Keybus, Grégoire Eberle, Brigitte Schurch, Etienne Pralong, Fabio Becce, John Prior, Nicholas Buse, Rik Buschman, Esra Neufeld, Niels Kuster, Stefano Carda, Joachim von Zitzewitz, Vincent Delattre, Tim Denison, Hendrik Lambert, Karen Minassian, Jocelyne Bloch, and Grégoire Courtine. Targeted neurotechnology restores walking in humans with spinal cord injury. *Nature*, 563(7729):65–71, November 2018.

[18] Lauren N. Ayton, Nick Barnes, Gislin Dagnelie, Takashi Fujikado, Georges Goetz, Ralf Hornig, Bryan W. Jones, Mahiul M. K. Muqit, Daniel L. Rathbun, Katarina Stingl, James D. Weiland, and Matthew A. Petoe. An update on retinal prostheses. *Clinical Neurophysiology*, 131(6):1383–1398, June 2020.

[19] Katarina Stingl, Ruth Schippert, Karl U. Bartz-Schmidt, Dorothea Besch, Charles L. Cottriall, Thomas L. Edwards, Florian Gekeler, Udo Greppmaier, Katja Kiel, Assen Koitschev, Laura Kühlewein, Robert E. MacLaren, James D. Ramsden, Johann Roider, Albrecht Rothermel, Helmut Sachs, Greta S. Schröder, Jan Tode, Nicole Troelenberg, and Eberhart Zrenner. Interim Results of a Multicenter Trial with the New Electronic Subretinal Implant Alpha AMS in 15 Patients Blind from Inherited Retinal Degenerations. *Frontiers in Neuroscience*, 11, 2017. Publisher: Frontiers.

[20] Lewis Karapanos, Carla J. Abbott, Lauren N. Ayton, Maria Kolic, Myra B. McGuinness, Elizabeth K. Baglin, Samuel A. Titchener, Jessica Kvansakul, Dean Johnson, William G. Kentler, Nick Barnes, David A. X. Nayagam, Penelope J. Allen, and Matthew A. Petoe. Functional Vision in the Real-World Environment With a Second-Generation (44-Channel) Suprachoroidal Retinal Prosthesis. *Translational Vision Science & Technology*, 10(10):7–7, August 2021. Publisher: The Association for Research in Vision and Ophthalmology.

[21] Wm H. Dobelle. Artificial Vision for the Blind by Connecting a Television Camera to the Visual Cortex. *ASAIO Journal*, 46(1):3–9, February 2000.

[22] Michael Beyeler, Devyani Nanduri, James D. Weiland, Ariel Rokem, Geoffrey M. Boynton, and Ione Fine. A model of ganglion axon pathways accounts for percepts elicited by retinal implants. *Scientific Reports*, 9(1):1–16, June 2019.

[23] J. F. Rizzo, J. Wyatt, J. Loewenstein, S. Kelly, and D. Shire. Perceptual efficacy of electrical stimulation of human retina with a microelectrode array during short-term surgical trials. *Invest Ophthalmol Vis Sci*, 44(12):5362–9, December 2003.

[24] Jacob Granley and Michael Beyeler. A Computational Model of Phosphene Appearance for Epiretinal Prostheses. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pages 4477–4481, November 2021. ISSN: 2694-0604.

[25] Yvonne H-L. Luo, Joe Jiangjian Zhong, Monica Clemo, and Lyndon da Cruz. Long-term Repeatability and Reproducibility of Phosphene Characteristics in Chronically Implanted Argus II Retinal Prosthesis Subjects. *American Journal of Ophthalmology*, 170:100–109, October 2016.

[26] M. Beyeler, A. Rokem, G. M. Boynton, and I. Fine. Learning to see again: biological constraints on cortical plasticity and the implications for sight restoration technologies. *J Neural Eng*, 14(5):051003, June 2017.

[27] S. C. Chen, G. J. Suaning, J. W. Morley, and N. H. Lovell. Simulating prosthetic vision: I. Visual models of phosphenes. *Vision Research*, 49(12):1493–506, June 2009.

[28] Wei Tong, Hamish Meffin, David J Garrett, and Michael R Ibbotson. Stimulation strategies for improving the resolution of retinal prostheses. *Frontiers in neuroscience*, 14:262, 2020.

[29] Nishal P. Shah, Sasidhar Madugula, Lauren Grosberg, Gonzalo Mena, Pulkit Tandon, Pawel Hottowy, Alexander Sher, Alan Litke, Subhasish Mitra, and E.J. Chichilnisky. Optimization of Electrical Stimulation for a High-Fidelity Artificial Retina. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 714–718, March 2019. ISSN: 1948-3554.

[30] Dorsa Haji Ghaffari, Yao-Chuan Chang, Ehsan Mirzakhalili, and James D. Weiland. Closed-loop Optimization of Retinal Ganglion Cell Responses to Epiretinal Stimulation: A Computational Study. In *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 597–600, May 2021. ISSN: 1948-3554.

[31] Tristan Fauvel and Matthew Chalk. Human-in-the-loop optimization of visual prosthetic stimulation. preprint, Neuroscience, November 2021.

[32] Martin J. Spencer, Tatiana Kameneva, David B. Grayden, Hamish Meffin, and Anthony N. Burkitt. Global activity shaping strategies for a retinal implant. *Journal of Neural Engineering*, 16(2):026008, January 2019. Publisher: IOP Publishing.

[33] Jaap de Ruyter van Steveninck, Umut Güçlü, Richard van Wezel, and Marcel van Gerven. End-to-end optimization of prosthetic vision. *Journal of Vision*, 22(2):20, February 2022.

[34] Robert E Marc, Bryan W Jones, Carl B Watt, and Enrica Strettoi. Neural remodeling in retinal degeneration. *Progress in Retinal and Eye Research*, 22(5):607–655, 2003.

[35] Zhou Wang and Alan C. Bovik. Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Processing Magazine*, 26(1):98–117, January 2009. Conference Name: IEEE Signal Processing Magazine.

[36] Guangtao Zhai and Xiongkuo Min. Perceptual image quality assessment: a survey. *Science China Information Sciences*, 63(11):211301, November 2020.

[37] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Technical Report arXiv:1502.03167, arXiv, March 2015. arXiv:1502.03167 [cs] type: article.

[38] Michael F Dorman, Anthony J Spahr, Philipos C Loizou, Cindy J Dana, and Jennifer S Schmidt. Acoustic simulations of combined electric and acoustic hearing (eas). *Ear and Hearing*, 26(4):371–380, 2005.

[39] Mario A Svirsky, Nai Ding, Elad Sagi, Chin-Tuan Tan, Matthew Fitzgerald, E Katelyn Glassman, Keena Seward, and Arlene C Neuman. Validation of acoustic models of auditory neural prostheses. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8629–8633. IEEE, 2013.

[40] MF Dorman, PC Loizou, A Spahr, and CJ Dana. Simulations of combined acoustic/electric hearing. In *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439)*, volume 3, pages 1999–2001. IEEE, 2003.

[41] Michael F Dorman, Philipos C Loizou, and Dawne Rainey. Speech intelligibility as a function of the number of channels of stimulation for signal processors using sine-wave and noise-band outputs. *The Journal of the Acoustical Society of America*, 102(4):2403–2411, 1997.

[42] William B Cooper, Emily Tobey, and Philipos C Loizou. Music perception by cochlear implant and normal hearing listeners as measured by the montreal battery for evaluation of amusia. *Ear and hearing*, 29(4):618, 2008.

[43] Philipos C Loizou, Michael Dorman, Oguz Poroy, and Tony Spahr. Speech recognition by normal-hearing and cochlear implant listeners as a function of intensity resolution. *The Journal of the Acoustical Society of America*, 108(5):2377–2387, 2000.

[44] Hannes P Saal, Benoit P Delhaye, Brandon C Rayhaun, and Sliman J Bensmaia. Simulating tactile signals from the whole hand with millisecond precision. *Proceedings of the National Academy of Sciences*, 114(28):E5693–E5702, 2017.

[45] Elizaveta V Okorokova, Qinpu He, and Sliman J Bensmaia. Biomimetic encoding model for restoring touch in bionic hands through a nerve interface. *Journal of neural engineering*, 15(6):066033, 2018.

[46] Douglas J Weber, Rebecca Friesen, and Lee E Miller. Interfacing the somatosensory system to restore touch and proprioception: essential considerations. *Journal of motor behavior*, 44(6):403–418, 2012.

[47] Sung Soo Kim, Arun P Sripati, and Sliman J Bensmaia. Predicting the timing of spikes evoked by tactile stimulation of the hand. *Journal of neurophysiology*, 104(3):1484–1496, 2010.

[48] Milana P Mileusnic, Ian E Brown, Ning Lan, and Gerald E Loeb. Mathematical models of proprioceptors. i. control and transduction in the muscle spindle. *Journal of neurophysiology*, 96(4):1772–1788, 2006.

[49] N. M. Jansonius, J. Nevalainen, B. Selig, L. M. Zangwill, P. A. Sample, W. M. Budde, J. B. Jonas, W. A. Lagrèze, P. J. Airaksinen, R. Vonthein, L. A. Levin, J. Paetzold, and U. Schiefer. A mathematical description of nerve fiber bundle trajectories and their variability in the human retina. *Vision Research*, 49(17):2157–2163, August 2009.

[50] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[51] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. Technical Report arXiv:1409.1556, arXiv, April 2015. arXiv:1409.1556 [cs] type: article.

[52] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal Style Transfer via Feature Transforms. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[53] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. ISSN: 1063-6919.

[54] Sylvain Paris, Samuel W Hasinoff, and Jan Kautz. Local Laplacian Filters: Edge-aware Image Processing with a Laplacian Pyramid. *Communications of the ACM*, 58:11, 2015.

[55] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[56] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. Technical Report arXiv:1405.0312, arXiv, February 2015. arXiv:1405.0312 [cs] type: article.

[57] Nicole Han, Sudhanshu Srivastava, Aiwen Xu, Devi Klein, and Michael Beyeler. Deep Learning–Based Scene Simplification for Bionic Vision. In *Augmented Humans Conference 2021*, AHs'21, pages 45–54, New York, NY, USA, February 2021. Association for Computing Machinery.

[58] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[59] Eli Peli. Testing Vision Is Not Testing For Vision. *Translational Vision Science & Technology*, 9(13):32–32, December 2020. Publisher: The Association for Research in Vision and Ophthalmology.

[60] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[61] Alim Louis Benabid. Deep brain stimulation for Parkinson's disease. *Current Opinion in Neurobiology*, 13(6):696–706, December 2003.

[62] Bardia Bozorgzadeh, Douglas R. Schuweiler, Martin J. Bobak, Paul A. Garris, and Pedram Mohseni. Neurochemostat: A Neural Interface SoC With Integrated Chemometrics for Closed-Loop Regulation of Brain Dopamine. *IEEE Transactions on Biomedical Circuits and Systems*, 10(3):654–667, June 2016. Conference Name: IEEE Transactions on Biomedical Circuits and Systems.

[63] M. Beyeler, G. M. Boynton, I. Fine, and A. Rokem. pulse2percept: A Python-based simulation framework for bionic vision. In K. Huff, D. Lippa, D. Niederhut, and M. Pacer, editors, *Proceedings of the 16th Science in Python Conference*, pages 81–88, 2017.

[64] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] we highlight both the limitations of our general framework and our specific proof of concept, see Section 6
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6, last paragraph
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] We do not present any theoretical results
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include the trained models, and code to reproduce the main result figures in the supplementary materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] detailed instructions are given in Section 4, in the training/optimization subsection, which describes hyperparameter selection, learning rates, and many other training details. Data splits were predetermined by the datasets (MNIST and COCO already have splits), with some more details given in Appendix C
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We do not train our models multiple times. Once trained, the networks produce deterministic results.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] This is specified in Section 4, training paragraph
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We used models from [10, 22, 24], which are all cited whenever used in the paper.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] we provide saved weights for our final trained models, along with code to display results shown on main results figures.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We do not use any new personal data
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The data used does not contain any identifying information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Surrogate Model

This section covers specific implementation details about the surrogate model as well as observations on its performance.

A.1 Implementation Details

Dataset To create training data for the surrogate model $\hat{f}_\phi$, we used the phosphene model described in [24] and implemented in pulse2percept v0.8 [63]. $50,000$ stimuli were created by first selecting a number of electrodes to stimulate between 1 and 30 randomly chosen electrodes, then randomly selecting an amplitude between 1 and 10 (specified as a multiple of the assumed threshold current) and frequency between 1 and 200 Hz for each electrode. In addition, between 10 and 100 electrodes were chosen to act as "noise" electrodes, where either amplitude or frequency was given a nonzero value, but not both. The purpose of these electrodes was for the surrogate model to learn that both a nonzero amplitude and a nonzero frequency are required to produce a visible percept. We used an 80-20 train-test split. As the surrogate model is highly dependent on patient-specific parameters ϕ, we generated new data and fit a separate surrogate for each of the following ϕ: $((\rho, \lambda) \in \{(150, 100), (150, 1500), (800, 100), (800, 1500)\})$.

Network Architecture The surrogate model $\hat{f}_\phi$ used a fully-connected architecture. The input to the model was a stimulus matrix $\mathbf{s} \in \mathbb{R}_{\geq 0}^{n_e \mathrm{x} 3}$, which was identical to the input to f. The stimulus matrix was split into amplitude and frequency components (pulse duration was not used due to poor model performance), which were fed through a FC layer. The outputs of both FC layers were concatenated and fed through another FC layer. Concurrently, the model computed the element-wise product of the amplitude and frequency components and passed it through a separate FC layer. The outputs of the previous two layers were then concatenated and fed through a final FC layer with output size 49×49.

The model was trained for 45 epochs using AdamW [64] optimizer and MAE loss.

A.2 Approximating the Forward Model

The surrogate model was able to accurately approximate the true phosphene model f. Table 2 shows MAE over the validation set ($10,000$ percepts) for all 4 trained $\hat{f}_\phi$. Visually, the predicted percepts were nearly identical to the ground truth.

Table 2: Surrogate model performance

ϕ	$\rho = 150\ \lambda = 100$	$\rho = 150\ \lambda = 1500$	$\rho = 800\ \lambda = 100$	$\rho = 800\ \lambda = 1500$
MAE	0.0119	0.0189	0.0078	0.0115

A.3 Predicted Stimuli

Despite the low surrogate validation error, training with the surrogate model would often result in the encoder suggesting almost adversarial stimuli; that is, stimuli that if fed through the true forward model f would lead to drastically different percepts than if fed through the surrogate model $\hat{f}$ (see Fig. A.1). With these adversarial-like stimuli, the encoder appears to be performing well under the surrogate model, but performs poorly when the same stimuli are input to the true forward model. We identify this as the primary disadvantage of using a surrogate model and resolving this issue remains an open research problem for end-to-end training with surrogate methods.

We noticed several issues caused by the effects of varying stimulus parameters on phosphene appearance. For example, increasing amplitude increases size and brightness, while increasing frequency increases brightness only. We noticed a larger mismatch between the surrogate and the forward model on the extreme ends of the spectrum (*e.g.* very high frequency, low amplitude),

resulting in the encoder settling into a minimum that does not exist in the true forward model. It is important to note this disparity appears despite a high training accuracy of the surrogate alone. Although these examples are specific to the bionic vision application, we expect surrogate models derived to describe other neuromodulation technologies to suffer from similar limitations.

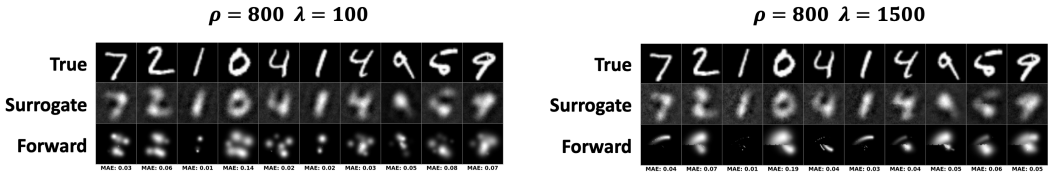

Figure A.1: The encoder would often suggest stimuli that lead to drastically different percepts when fed through the surrogate model ($\hat{f}$, middle row) as compared to the true forward model (f, bottom row). Examples are shown for $\rho = 800$; $\lambda = 100$ (*left*) and $\rho = 800$ $\lambda = 1500$ (*right*).

B Hyperparameter Selection

In this section, we detail how HNA hyperparameters (l, k, α, and β) were chosen.

VGG Loss To choose the layer of the VGG network to use for VGG loss (l) we performed cross validation across a set of candidate layers. Previous studies [52] have shown that the first layer with each of the 5 convolutional blocks perform well for neural style transfer. Thus, we choose these as our candidate layers. For cross validation, we trained HNA for 50 epochs using each candidate layer. The resulting phosphenes are shown in Figure B.1. Using earlier layers, the VGG term performs similarly to MAE, and phosphenes are disconnected. We chose layer 5_1 based on its perceived ability to capture high-level perceptual differences between images, although layer 4_1 also performs similarly.

Figure B.1: Phosphenes produced by HNA encoder with different layers chosen for VGG loss. Layer 5_1 denotes the first layer within the fifth convolutional block.

Laplacian Smoothing We chose to use a kernel size 5 for the Laplacian filter used to estimate the second derivative (k, Eq. 6). The size of the filter controls the scale on which smoothing is applied (*i.e.*, smaller filters sizes only encourage continuity within a small local region, whereas larger filters encourage continuity within a larger region). Size 5 was chosen because larger filters were observed to over-smooth the image, while smaller filters still led to highly disconnected phosphenes.

Joint Perceptual Metric We performed cross validation to find the best values for α and β. Instead of using one value, we found scheduled weighting to be crucial for performance. The scheduler incrementally increased the weight of the VGG loss (β) from 0 while simultaneously decreasing the initially high weight on the smoothing constraint (α). This was motivated by the observation that the VGG loss performed poorly during early iterations when the predicted phosphene was near-random.

Under this scheduled weighting strategy, the loss is dominated early on by the MAE and smoothing terms. This encourages the the model to just output reasonable encodings. As training progresses, the predicted phosphenes become higher quality, causing the VGG loss to perform better, and thus the smoothing term is no longer as important.

Additionally, we found it beneficial to temporarily decrease the learning rate by a factor of 10 for a short 'warm-up' duration following each increase in β, before resetting to 50% of the prior learning rate. This results in the learning rate gradually decreasing throughout training by a factor of around 100. Throughout the paper, we use $\alpha = 0$ and $\beta = 0.00008$ for comparisons of loss values.

C COCO Dataset

For the COCO task (Section 5.2), we used subset of images from the MS-COCO dataset [56]. MS-COCO was chosen due to its selection of common household objects relevant to the daily life of prosthesis users, as well as availability of ground-truth segmentation masks. To select the images suitable for prosthetic vision, we filtered out images according to the following criteria:

1. **Too cluttered.** Any image with greater than 15 total objects was removed. Removed: 15566
2. **Select chosen objects.** Any image that did not have at least 1 object from the selected categories that was larger than 4% of the total image was removed. Removed: 42289
3. **Too many.** Any image with greater than 5 objects meeting criteria 2 was removed. Removed: 1017
4. **Too dim.** Any objects in the image with average pixel brightness less than 50 were discarded. If this resulted in an image having 0 remaining objects, the image was removed. Removed: 434

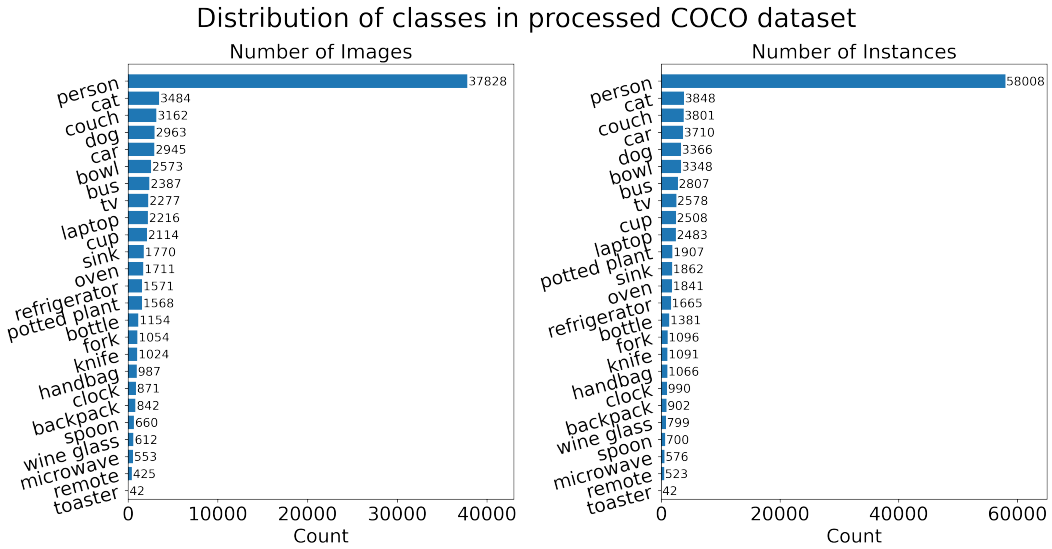

Figure C.1: Number of images (*left*) or instances (*right*) of each category in the processed COCO dataset.

This resulted in a total of 47,532 training images and 11,883 test images (80-20 train-test split). The objects in the remaining images were segmented out using the ground-truth segmentation masks, resized to (49, 49), and converted to grayscale. The distributions of classes used is shown in Figure C.1.

D Predicted Stimuli

Here, we directly examine the stimuli resulting from HNA and naive encoders. Stimuli and their resulting phosphenes for example images from the test set are shown in Figure D.1. The naive encoder produces stimuli with constant frequency (20 Hz) and pulse duration (0.45 ms), which are not shown.

We make the following observations about the predicted stimuli:

- Both encoders activate electrodes corresponding to the shape of the target image. In naive stimuli, the amplitude directly corresponds to the pixel brightness. In HNA stimuli, the distributions of amplitude, frequency, and pulse duration across the electrodes is more complex and harder to characterize, but lead to higher-quality phosphenes.
- HNA uses amplitudes inversely proportional to ρ.
- For small ρ, HNA primarily uses amplitude to control brightness. For large ρ, HNA primarily uses frequency to modulate brightness, keeping amplitudes low to limit phosphene size.
- HNA uses small pulse durations to create lines parallel to the underlying axon NFB (*i.e.*, it utilizes the streaked phosphenes to its advantage), and large pulse durations to create lines perpendicular to the underlying NFB. In other words, HNA was able to exploit application-specific (*i.e.*, neuroanatomical) information that is baked into the forward model.
- On average, HNA uses more electrodes, larger frequencies and pulse durations, and smaller amplitudes than the naive encoder. A large active electrode count and high pulse durations may not be desirable for some prostheses, due to tissue activation and frame rate limits. We found that it was easy to constrain these parameters using regularization on the output stimuli, at the cost of slightly decreased performance.

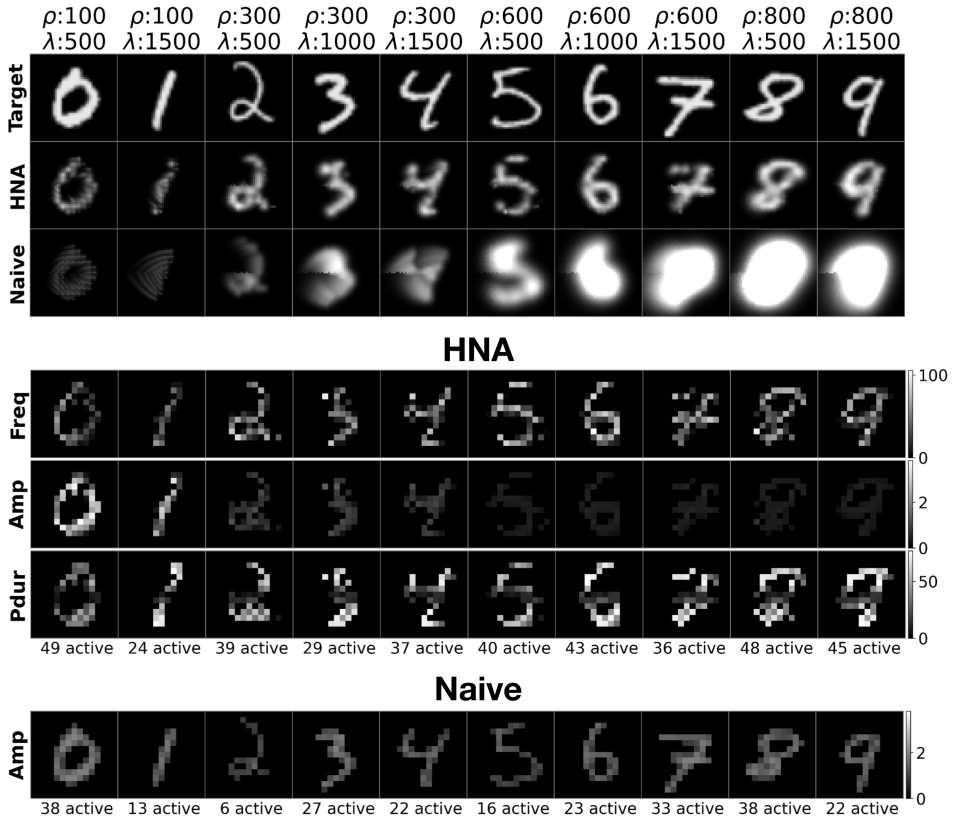

Figure D.1: *Top*: Example MNIST target images, and the phosphenes produced by HNA and naive encoders, encoded at various ρ and λ values. *Center*: The stimuli corresponding to the HNA phosphenes. From top to bottom, stimulus frequency (Hz), amplitude (xTh), and pulse duration (ms) are shown. The number of 'active' electrodes stimulated above threshold levels is given below each stimuli. *Bottom*: Stimuli corresponding to the naive phosphenes.

E COCO Patient-to-Patient Variations

We repeated the analysis presented in Section 5.3 for the COCO dataset. Figure E.1A shows two example COCO images, encoded by both HNA and the naive encoder, across varying ρ and λ values. The heatmaps in Figure E.1B show the log of the joint perceptual loss across simulated patients, for both the naive and HNA encoders. To measure phosphene consistency, we performed T-SNE clustering on a subset of the COCO images which have only 1 object. Unfortunately, T-SNE clustering of the ground-truth COCO images did not form groups corresponding to the object types (Figure E.1C), suggesting that the representation of object instances vary drastically across COCO images. Therefore, it was not meaningful to repeat the analysis presented in Fig. 5C.

Similar to the MNIST results presented in Section 5.3, HNA produced higher-quality representations than the naive encoder, resulting in a lower joint loss for every simulated patient. HNA performed consistently well across all simulated patients (Figure E.1B), with a small increase in loss for small ρ (< 100). Similar to MNIST, the naive encoder only performs well for patients with a mid-to-low ρ (≈ 200) and low λ.

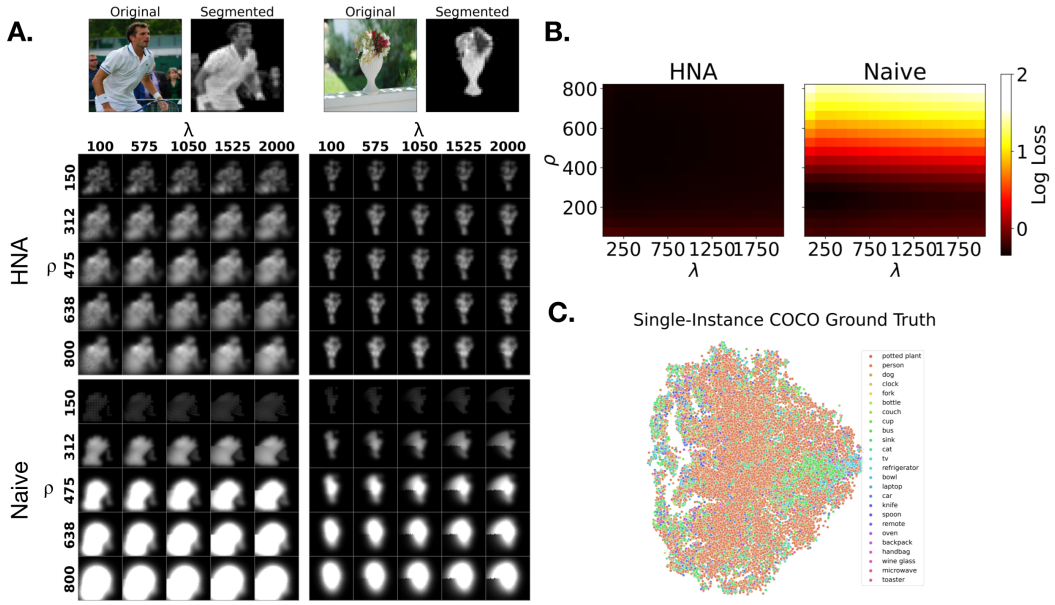

Figure E.1: COCO Encoder performance across simulated patients (varying ρ and λ). **A:** Phosphenes produced by HNA and Naive encoders of two example images. **B:** Heatmaps showing the log joint loss across ρ and λ for HNA and naive encoders. **C.** Ground-truth COCO images cannot be clustered using T-SNE into groups corresponding to the object types. The clustering was performed on COCO images that only contained one object.

F Modeling Other Patient-to-Patient Variations

Previously, results were presented across patient-specific parameters ρ and λ, because these have the greatest impact on phosphene appearance. However, the forward model has a number of other patient-specific parameters, which HNA is also able to adapt to. For full details on all parameters of the forward model, see [24]. Out of the remaining parameters, $a2$, $a3$, and $a5$ are the most impactful on phosphene appearance. $a2$ and $a3$ modulate how much the brightness contribution from each electrode scales with increasing amplitude and frequency, respectively. $a5$ locally scales the global radial current spread ρ based on each electrodes amplitude. Figure F.1 (*left*) illustrates the effect of these parameters on phosphene appearance.

Figure F.1 compares HNA to naive encoder performance across $a2$, $a3$, and $a5$. The ranges for these parameters are based on values empirically observed in retinal prosthesis users [24]. HNA produces relatively consistent phosphenes, and outperforms the naive encoder across all conditions.

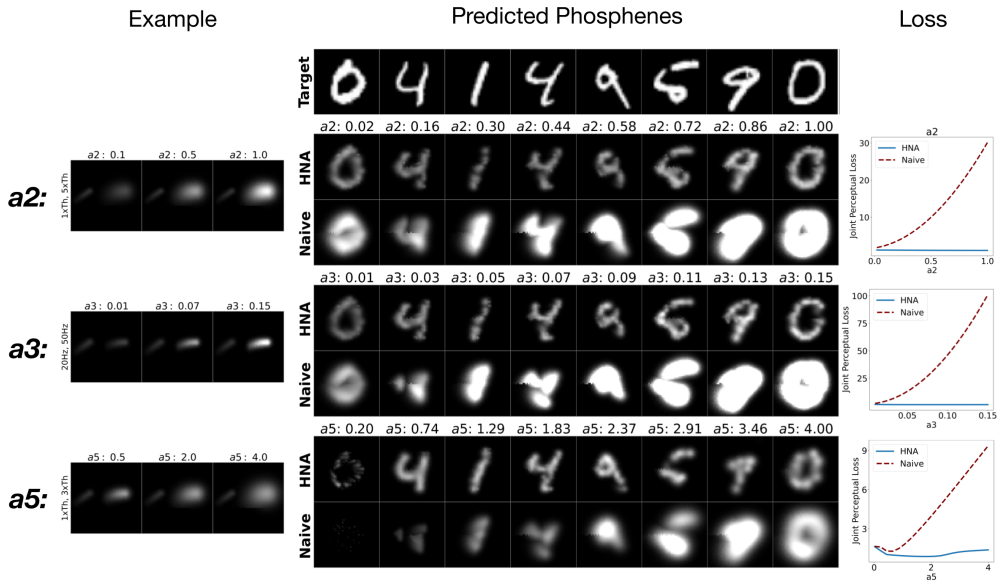

Figure F.1: *Left*: Examples of how a2, a3, and a5 affect single-electrode phosphenes. a2 modulates local brightness scaling with increasing amplitude, a3 modulates local brightness scaling with increasing frequency, and a5 modulates local size scaling with increasing amplitude. *Center*: Phosphenes predicted with HNA and naive encoders for varying a2, a3, and a5, increasing left to right. *Right*: Plot showing the joint loss across a2, a3, and a5 for HNA (solid) and naive encoder (dashed line).

G Simulating Higher-Resolution Implants

On the COCO task, HNA significantly outperformed the naive encoder, but was still unable to capture all of the detail in the images. Two of the main reasons for this are the limited spatial resolution of the implant and the patient-specific distortions from the forward model. Here, we present results from HNAs trained on implants of higher resolution, at small ρ and λ. The chosen implants are illustrated in Figure G.1A. For a fair comparison, each HNA was trained for only 50 epochs.

Phosphenes resulting from the HNA trained on the different implants are shown in Figure G.1C, and the losses across implants is plotted in Figure G.1B. As implant resolution increases, the phosphenes look increasingly similar to the ground truth, and small details (e.g. facial details, textures) start to emerge.

Thus, HNAs initial failure to capture high-frequency details in the image appears to be an application-specific limitation for visual prostheses more so than a limitation of the HNA framework. For visual prostheses, learning to reconstruct the high-frequency features of complex images despite distortions and limited implant resolution remains an open problem.

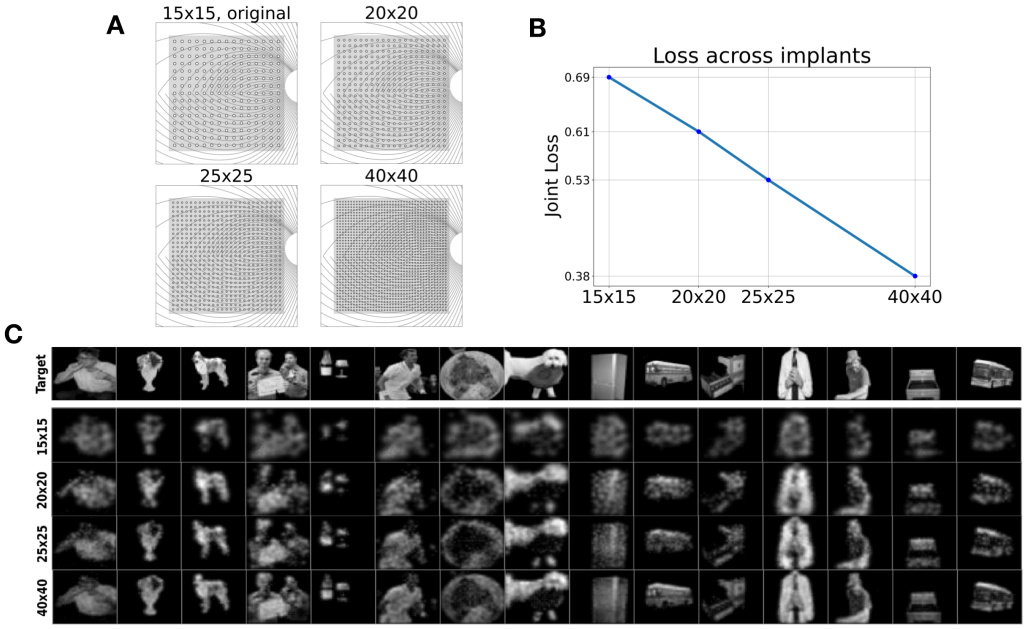

Figure G.1: **A**: The 4 different implants compared. The main text uses the 15×15 implant. **B**: The joint perceptual loss of HNAs trained on the different implants after 50 epochs. **C**: Example images showing the reconstructed phosphenes using each implant

H Mis-Specified Patient-Specific Parameters

Due to noisy or limited patient data, there may be some uncertainty in the measured value of the patient-specific parameters ϕ. Therefore, we conducted an analysis of the consequences of incorrect patient-specific parameters on the encodings produced by HNA. Note that the true patient-specific parameters are not needed during training, so incorrect ϕ will only affect evaluation. A 'mismatch' HNA model was created, where the forward model decoder used the true patient-specific parameters ϕ, and the encoder used another set of patient-specific parameters ϕ'.

In the first experiment, ϕ' was sampled from a uniform random distribution (we again focus on only ρ and λ). The original HNA encoder, naive encoder, and mismatch HNA encoder with random ϕ' were evaluated on the MNIST test set. HNA achieved a joint loss of 0.92, the naive encoder had a joint loss of 3.13, and the mismatch HNA had a joint loss of 1.35 ± 0.003 (mean $\pm$ standard deviation across 10 random ϕ'). Thus, even if the true patient-specific parameters are completely unknown, on average randomly selecting values will still produce higher-quality encodings than the naive method.

In a second experiment, we analyzed whether there were any configurations (ϕ - ϕ' combinations) that resulted in a worse encoding than the naive model. For the 90% of true ϕ, the mismatch model outperformed the naive model regardless of the chosen ϕ'. However, the naive model performs best at $\rho = 250$ and $\lambda = 200$. In Figure H.1A, we hold λ constant at 200 and, for each true ρ, plot the ranges of mis-specified ρ' for which the mismatch HNA still outperforms the naive. Figure H.1B shows a similar plot for varying λ, holding ρ constant at 250. Even for the naive model's ideal patients, HNA still outperforms the naive model for a large proportion of mis-specified ρ and λ.

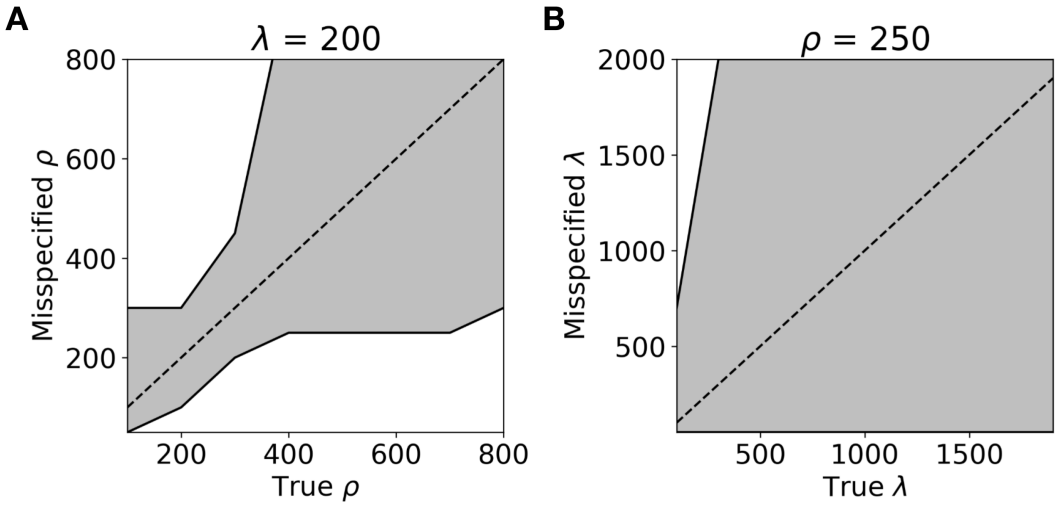

Figure H.1: Plots showing mis-specified HNA performance relative to the naive encoder for varying ρ (panel **A**) and λ (panel **B**). The dashed line marks the correctly specified model, and shaded area between the solid lines shows the region where the mis-specified HNA outperforms the naive encoder. Note that the naive model's ideal patient was used, with λ fixed at 200 and ρ fixed at 250, respectively.