# A PRESCRIPTIVE THEORY FOR BRAIN-LIKE INFERENCE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The Evidence Lower Bound (ELBO) is a widely used objective for training deep generative models, such as Variational Autoencoders (VAEs). In the neuroscience literature, an identical objective is known as the Free Energy Principle (FEP), hinting at a potential unified framework for brain function and machine learning. Despite its utility in interpreting generative models, including diffusion models, ELBO maximization is often seen as too broad to offer prescriptive guidance for specific architectures in neuroscience or machine learning. In this work, we show that maximizing ELBO under Poisson assumptions for general sequences leads to a spiking neural network that performs Bayesian posterior inference through its membrane potential dynamics. The resulting model, the iterative Poisson VAE (i$\mathcal{P}$-VAE), has a closer connection to biological neurons than previous brain-inspired predictive coding models based on Gaussian assumptions. Compared to amortized and iterative VAEs, i$\mathcal{P}$-VAE learns sparser representations and exhibits superior generalization to out-of-distribution samples. These findings suggest that optimizing ELBO, combined with Poisson assumptions, provides a solid foundation for developing prescriptive theories in NeuroAI.

## 1 INTRODUCTION

Optimizing the Evidence Lower Bound (ELBO) serves as a unifying objective for training deep generative models (Hinton et al., 1995; Dayan et al., 1995; Kingma & Welling, 2014; Rezende et al., 2014; Luo, 2022). Even when models don't explicitly reference ELBO, they're often optimizing objectives closely related to it (Luo, 2022; Kingma & Gao, 2023). This is directly paralleled by the Free Energy Principle (FEP) in neuroscience, which absorbs previous theoretical frameworks like Predictive Coding, Bayesian Brain, and Active Learning (Friston, 2005; 2009; 2010). FEP states that a single objective, the minimization of variational free energy, is all that is needed. Because this is equivalent to maximizing ELBO, it suggests a powerful unifying theoretical framework for neuroscience and machine learning (Friston, 2010).

However, in many ways, Free Energy (and by proxy, ELBO) is too general to be useful as a theory (Gershman, 2019; Andrews, 2021). In practice, the specific implementations of FEP predictive coding have been difficult to map directly onto neural circuits (Millidge et al., 2021a; 2022), struggling with negative rates and prediction signals that have not been observed empirically (Walsh et al., 2020; Millidge et al., 2022). Similarly, in machine learning, it is often discovered after the fact that a new objective is actually ELBO maximization (or KL minimization; Hobson (1969)) masquerading as something else (Kingma & Gao, 2023)—and not the other way around. If ELBO is "all you need," then why is ELBO not prescriptive?

One possibility, at least in neuroscience, is that ELBO's lack of prescriptive theory results from incorrect approximating distributions. In fact, most of the difficulty mapping predictive coding onto neural circuits has to do with terms that result from the Gaussian assumption (Millidge et al., 2022). In contrast, biological neurons are largely modeled as conditionally Poisson (Goris et al., 2014).

Recent work provides a potential prescriptive route: replacing Gaussians with Poisson distributions. To this end, Vafaii et al. (2024) introduced a reparameterization algorithm for training Poisson Variational Autoencoders ($\mathcal{P}$-VAE). They observed that replacing Gaussians in ELBO reduces to an amortized version of sparse coding, an influential model inspired by the brain that captures many features of the selectivity in early visual cortex (Olshausen & Field, 1996; 2004). $\mathcal{P}$-VAE learns sparse representations, avoids posterior collapse, and performs better on downstream classification tasks. However, the authors identified a large amortization gap in $\mathcal{P}$-VAE (Vafaii et al., 2024), adding to a growing body of work that highlights limitations of amortized inference Cremer et al. (2018); Kim & Pavlovic (2021). A potential solution is to develop more general iterative inference solutions, or hybrid iterative-amortized ones (Marino et al., 2018; Kim et al., 2018).

Here, we extend the Poisson VAE to include iterative inference ("iterative $\mathcal{P}$-VAE," or i$\mathcal{P}$-VAE). This results in a generalization of predictive coding that maps well onto biological neurons. i$\mathcal{P}$-VAE implements Bayesian posterior inference via private membrane potential dynamics, resembling a spiking version of the Locally Competitive Algorithm (LCA) for sparse coding (Rozell et al., 2008). This solution avoids the major problems with predictive coding: there is no explicit prediction, neurons communicate through spikes, and feedback is modulatory—all consistent with real neurons (Gilbert & Li, 2013; Kandel et al., 2000). But how effective is i$\mathcal{P}$-VAE as a machine learning model?

We evaluate i$\mathcal{P}$-VAE in terms of convergence, reconstruction performance, efficiency, and out-of-distribution (OOD) generalization. We find that i$\mathcal{P}$-VAE converges to sparse posterior representations, outperforming other iterative VAEs (Kim et al., 2018; Marino et al., 2018).

**Contributions.** We introduce a new architecture, i$\mathcal{P}$-VAE, that accomplishes the following:

- Deriving the ELBO for sequences with Poisson-distributed latents results in a neural network that spikes, and performs predictive coding in the dynamics of the membrane potential.
- By reusing the same set of weights across iterations and utilizing sparse, integer spike counts, i$\mathcal{P}$-VAE is well-suited for hardware implementations and energy-efficient deployment.
- i$\mathcal{P}$-VAE demonstrates robust out-of-distribution generalization, excelling in both within-dataset perturbations and cross-dataset generalization.

Taken together, i$\mathcal{P}$-VAE is a powerful brain-inspired architecture that tightly maps onto biological neurons while outperforming much larger models in key objectives such as performance, parameter count, sparsity, and out-of-distribution generalization.

## 2 Background and related work

**Generative models and ELBO.** Generative models learn to represent the data distribution, $p(\boldsymbol{x})$, typically by invoking latent variables $\boldsymbol{z}$, such that $p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$ (Bishop & Nasrabadi, 2006). The key challenge is computing, $p(\boldsymbol{z}|\boldsymbol{x})$, the posterior distribution of these latent variables given the data, which is typically intractable except for simple cases.

Variational inference offers a practical solution by introducing an approximate posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ parameterized by $\phi$ (Blei et al., 2017). The goal is to make this approximation as close as possible to the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$. Ideally, one would minimize the KL divergence between $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z}|\boldsymbol{x})$, but since we cannot compute $p(\boldsymbol{z}|\boldsymbol{x})$ exactly, direct minimization is not feasible.

The Evidence Lower Bound (ELBO) provides a tractable objective that indirectly minimizes the KL divergence between the approximate and true posteriors. Specifically, the relationship is:

$$\log p(\boldsymbol{x}) = \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right]}_{\text{ELBO}} + \mathcal{D}_{\text{KL}}\Big(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,\big\|\, p(\boldsymbol{z}|\boldsymbol{x})\Big) \tag{1}$$

Since $\log p(\boldsymbol{x})$ does not depend on $\phi$ and the KL divergence is non-negative, maximizing the ELBO effectively minimizes the intractable KL divergence (Hinton et al., 1995; Kingma & Welling, 2014; Rezende et al., 2014). Interestingly, even when generative models seem to optimize a different loss function, like diffusion models (Chan, 2024; Ho et al., 2020), they are often still performing KL minimization through the ELBO (Kingma & Gao, 2023; Luo, 2022).

**ELBO in Neuroscience.** The Evidence Lower Bound (ELBO) has an identical formulation in neuroscience, where it is referred to as the Free Energy (Friston, 2005; 2009; 2010). The Free Energy Principle (FEP) extends the framework of perception as inference (Alhazen, 1011–1021 AD;

Von Helmholtz, 1867; Mumford, 1992), drawing concepts from predictive coding (PC; Srinivasan et al. (1982); Rao & Ballard (1999)). Extensive research has explored how PC might be implemented by neurons (Boerlin et al., 2013; Millidge et al., 2021a), and PC has been applied in machine learning for predictive models (Lotter et al., 2017; Wen et al., 2018; Millidge et al., 2024).

Despite their neural inspiration, FEP is challenging to map directly onto neuronal circuits (Kogo & Trengove, 2015; Aitchison & Lengyel, 2017; Millidge et al., 2022). This difficulty results from assuming Gaussian for the approximate posterior and prior (Millidge et al., 2022). The Gaussian assumption results in models with explicit predictions or prediction errors, which have not been observed empirically (Mikulasch et al., 2023). Solutions also struggle with how to avoid negative firing rates due to subtraction operations (Bastos et al., 2012; Keller & Mrsic-Flogel, 2018). While leaky integrate-and-fire (LIF) circuits can be engineered to perform predictive coding (Boerlin et al., 2013), these implementations do not naturally arise from ELBO maximization, making the theory more postdictive than prescriptive. The related framework of sparse coding can be thought of as a form of predictive coding with a sparse prior Olshausen & Field (1996; 2004). A biologically plausible implementation of sparse coding, known as the locally competitive algorithm (LCA; Rozell et al. (2008)), results naturally in a dynamic update rule that resembles neural circuits. However, LCA relies on maximum a posteriori inference, which is restrictive if we aim to sample from the full posterior distribution.

**Bayesian posterior inference: iterative versus amortized.** In contrast to predictive coding, Variational Autoencoders (VAEs) introduced a computationally-efficient solution to maximize ELBO through *amortized* inference (Kingma & Welling, 2014; Rezende et al., 2014). Amortized inference uses a parameterized neural network (the "encoder" or "recognition" network) to produce the parameters of an approximate posterior, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, in one shot. The term "amortized" reflects that the computational cost of inference is paid during training, not at test time, similar to cost distribution in accounting (Gershman & Goodman, 2014). While amortized inference is considered efficient, it can suffer from an *amortization gap*—the discrepancy between the approximate posterior provided by the encoder and the optimal variational parameters—which can be significant (Cremer et al., 2018).

To address the amortization gap, hybrid approaches have been developed that introduce iterative elements into the VAE framework (Marino et al., 2018; Kim et al., 2018; Marino et al., 2021). For example, Marino et al. (2018) proposed a method where the encoder network takes as input both the data sample $\boldsymbol{x}$ and the gradients of the loss with respect to the variational parameters $\nabla_{\boldsymbol{\lambda}}\mathcal{L}$, with $\boldsymbol{\lambda} = \{\boldsymbol{\mu}, \boldsymbol{\sigma}^2\}$. Alternatively, semi-amortized inference (Kim et al., 2018) starts with an amortized initial estimate and refines it using stochastic variational inference (SVI) updates (Hoffman et al., 2013). Our method is closely related to these approaches, and we compare to them in the results.

Although VAEs and predictive coding are related through their optimization of ELBO (Marino, 2022), recent work has made that connection more explicit, demonstrating that classical predictive coding networks can be seen as a subclass of iterative inference in VAEs (Boutin et al., 2020). A key difference between our work and Boutin et al. (2020) is that they show the Rao & Ballard (1999) loss function arises from assuming a delta-function posterior in the ELBO. In our work, predictive coding naturally emerges in the dynamics of the log spike rates, which comes from a fairly general assumption of Poisson distributions.

**Poisson VAE.** A large body of literature in neuroscience has demonstrated that neuron spike counts are well described by a Poisson process over short counting windows (Goris et al., 2014). Building on this, Vafaii et al. (2024) introduced the Poisson Variational Autoencoder (P-VAE), which performs posterior inference using discrete spike counts. They developed a Poisson reparameterization trick and derived the ELBO for Poisson-distributed VAEs ($\mathcal{P}$-VAE).

In $\mathcal{P}$-VAE, the KL term penalizes firing rates, similar to sparse coding, and the ELBO, when paired with a linear generative model, reduces to amortized sparse coding. When trained on natural image patches, $\mathcal{P}$-VAE learns sparse solutions with Gabor-like basis vectors and latent sparsity, similar to sparse coding. While $\mathcal{P}$-VAE outperformed Gaussian VAEs in sparsity and downstream classification, the authors noted a significant performance gap with traditional sparse coding, likely arising from an amortization gap due to the lack of iterative updates. Our work builds upon $\mathcal{P}$-VAE, suggesting that Poisson is the right choice for parameterizing the distributions in ELBO (see Appendix B for a discussion).

# 3 INTRODUCING THE ITERATIVE POISSON VAE (IP-VAE)

In this section, we derive the ELBO for sequences with Poisson distributions. We show the resulting architecture (i$\mathcal{P}$-VAE) implements iterative Bayesian posterior inference with dynamics on the log rates. We relate this directly to membrane potential dynamics in a spiking neural network and show that it solves many of the implementation limitations of classic predictive coding.

**General setup.** We conceptualize iterative inference by starting with the more general framework of inference over a sequence (Chung et al., 2015). From there, we can treat iterative inference for images as a sequences of the same image repeated at all time points. This approach is appealing because dynamics emerge necessarily, and it builds a foundation for future work on dynamic sequences.

Consider a sequence of $T+1$ observed data points, $\vec{\boldsymbol{x}} = \{\boldsymbol{x}_t : t = 0, \ldots, T\}$ where $\boldsymbol{x}_t \in \mathbb{R}^M$, and corresponding latent variables, $\vec{\boldsymbol{z}} = \{\boldsymbol{z}_t : t = 0, \ldots, T\}$, where each $\boldsymbol{z}_t$ is $K$-variate. We denote the full probabilistic generative model as the joint distribution, $p(\vec{\boldsymbol{x}}, \vec{\boldsymbol{z}})$. A reasonable starting assumption for modeling the physical world is Markovian dependence between consecutive data points (Van Kampen, 1992), resulting in the marginal distribution:

$$p(\vec{\boldsymbol{x}}) = \int p(\vec{\boldsymbol{x}}, \vec{\boldsymbol{z}})\, d\vec{\boldsymbol{z}} = p(\boldsymbol{x}_0) \prod_{t=1}^{T} p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}), \tag{2}$$

where $p(\boldsymbol{x}_0) = \int p(\boldsymbol{x}_0|\boldsymbol{z}_0) p(\boldsymbol{z}_0) d\boldsymbol{z}_0$, and $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{z}_t) p(\boldsymbol{z}_t|\boldsymbol{x}_{t-1}) d\boldsymbol{z}_t$. For our sequence data, the ELBO can be written as follows:

$$\begin{aligned} \log p_\theta(\vec{\boldsymbol{x}}) &\geq \mathbb{E}_{q_\phi(\vec{\boldsymbol{z}}|\vec{\boldsymbol{x}})} \left[ \log \frac{p_\theta(\vec{\boldsymbol{x}}, \vec{\boldsymbol{z}})}{q_\phi(\vec{\boldsymbol{z}}|\vec{\boldsymbol{x}})} \right] \\ &= \mathbb{E}_{q_\phi(\vec{\boldsymbol{z}}|\vec{\boldsymbol{x}})} \left[ \log p_\theta(\vec{\boldsymbol{x}}|\vec{\boldsymbol{z}}) \right] - \mathcal{D}_{\text{KL}}\Big( q_\phi(\vec{\boldsymbol{z}}|\vec{\boldsymbol{x}}) \,\big\|\, p_\theta(\vec{\boldsymbol{z}}) \Big) \\ &= \mathcal{L}_{\text{ELBO}}(\vec{\boldsymbol{x}}; \theta, \phi), \end{aligned} \tag{3}$$

where $p_\theta(\vec{\boldsymbol{z}})$ is a prior (either learned or fixed) over latents, and $p_\theta(\vec{\boldsymbol{x}}|\vec{\boldsymbol{z}})$ is the conditional likelihood distribution, which is computed via a decoder network. Model parameters, $(\phi, \theta)$—corresponding to the encoder and decoder networks of a VAE, respectively—are jointly optimized. Below we will express the ELBO for sequences when using the Poisson Variational Autoencoder framework.

**Iterative Poisson VAE.** To extend the $\mathcal{P}$-VAE to sequences, i$\mathcal{P}$-VAE needs to make explicit how the prior and posterior distributions update with each sample. The simplest starting point is assuming stationarity, implying that the posterior over the previous stimulus should act as a prior for the current one (although future extensions could extend to nonstationary signals such as videos with a more sophisticated update rule). Because of the Markovian assumption, the prior, $p(\vec{\boldsymbol{z}})$, then factorizes into the initial prior, $p(\boldsymbol{z}_0)$ and a product over all time steps:

$$p(\vec{\boldsymbol{z}}) = p(\boldsymbol{z}_0) \prod_{t=1}^{T} p(\boldsymbol{z}_t | \boldsymbol{x}_{t-1}) \tag{4}$$

The initial prior, $p(\boldsymbol{z}_0) = \mathcal{P}\text{ois}(\boldsymbol{z}_0; \boldsymbol{r}_0)$, is Poisson with learned prior rates, $\boldsymbol{r}_0 \in \mathbb{R}^K_{>0}$. Subsequent time steps have prior rates that depend on the stimulus from the previous time step, $p(\boldsymbol{z}_t|\boldsymbol{x}_{t-1}) = \mathcal{P}\text{ois}(\boldsymbol{z}_t; \boldsymbol{r}_t(\boldsymbol{x}_{t-1}))$. The approximate posterior factorizes as well:

$$q(\vec{\boldsymbol{z}}|\vec{\boldsymbol{x}}) = q(\boldsymbol{z}_0|\boldsymbol{x}_0) \prod_{t=1}^{T} q(\boldsymbol{z}_t | \boldsymbol{x}_t, \boldsymbol{x}_{t-1}), \tag{5}$$

with initial posterior, $q(\boldsymbol{z}_0|\boldsymbol{x}_0) = \mathcal{P}\text{ois}(\boldsymbol{z}_0; \boldsymbol{r}_0 \odot \boldsymbol{\delta r}(\boldsymbol{x}_0))$, and time-dependent posterior, $q(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) = \mathcal{P}\text{ois}(\boldsymbol{z}_t; \boldsymbol{r}_t(\boldsymbol{x}_{t-1}) \odot \boldsymbol{\delta r}(\boldsymbol{x}_t))$, both parameterized as Poisson distributions. We follow the formulation in Vafaii et al. (2024), and define the posterior rates via an element-wise multiplicative interaction between $\boldsymbol{r}$ and some gain modulator, $\boldsymbol{\delta r} \in \mathbb{R}^K_{>0}$. This is a natural choice because rates must be positive, and without loss of generality, the relationship between two positive variables can be written in terms of a base rate, and a multiplicative gain on that base rate.

The conditional log-likelihood for i$\mathcal{P}$-VAE factorizes into a sum over individual sample likelihoods $\log p(\vec{\boldsymbol{x}}|\vec{\boldsymbol{z}}) = \sum_{t=0}^{T} \log p(\boldsymbol{x}_t|\boldsymbol{z}_t)$. The KL-term of the ELBO (eq. (3)) also factorizes:

$$\begin{aligned}\mathcal{D}_{\mathrm{KL}}\Big(q(\vec{\boldsymbol{z}}|\vec{\boldsymbol{x}}) \,\Big\|\, p(\vec{\boldsymbol{z}})\Big) &= \mathcal{D}_{\mathrm{KL}}\Big(q(\boldsymbol{z}_0|\boldsymbol{x}_0) \,\Big\|\, p(\boldsymbol{z}_0)\Big) + \sum_{t=1}^{T} \mathcal{D}_{\mathrm{KL}}\Big(q(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) \,\Big\|\, p(\boldsymbol{z}_t|\boldsymbol{x}_{t-1})\Big) \\ &= \boldsymbol{r}_0 \cdot f(\boldsymbol{\delta r}(\boldsymbol{x}_0)) + \sum_{t=1}^{T} \boldsymbol{r}_t(\boldsymbol{x}_{t-1}) \cdot f(\boldsymbol{\delta r}(\boldsymbol{x}_t)),\end{aligned} \tag{6}$$

where $\cdot$ represents a vector dot product, and $f(y) = 1 - y + y \log y$ is applied element-wise. Because rates are positive, the KL term penalizes large rates, acting like a sparsity penalty (Vafaii et al., 2024). The remaining sections describe how we specify the multiplicative gain, $\boldsymbol{\delta r}$, which results in adaptive Bayesian posterior updating in the dynamics of the model.

**Bayesian posterior updates using membrane potential dynamics** Because rates are positive and prior and posterior rates interact multiplicatively, it is difficult to implement dynamic updates directly on rates. A natural solution is to define updates on log rates, $\boldsymbol{u}(t) \coloneqq \log \boldsymbol{r}(t)$, with $\mathbb{R}^K$ as our state space for a $K$-dimensional latent space.

Dynamic updates on log-rates is both a mathematical convenience and biologically realistic. Because of internal noise, the spike threshold of real neurons is best modeled as an expansive nonlinearity like an exponential (Priebe et al., 2004; Fourcaud-Trocmé et al., 2003). Further, synapses have a compressive nonlinearity for incoming spikes because of synaptic depression (Abbott et al., 1997). Here, we take $\log(x)$ to be the synaptic nonlinearity and $\exp(x)$ to be the spiking nonlinearity. For the aforementioned reasons, $\boldsymbol{u}(t)$ can be interpreted quite literally as membrane potentials.

We define the model updates as $\boldsymbol{u}_{t+1} = \boldsymbol{u}_t + \boldsymbol{\delta u}_t$, with $\boldsymbol{r}_t = \exp(\boldsymbol{u}_t)$ acting as the corresponding prior rates at time $t$, and $\boldsymbol{r}_t \odot \boldsymbol{\delta r} = \exp(\boldsymbol{u}_{t+1})$, as the posterior rates at time $t$. When processing the next input in the sequence, we take the previous posterior and use it as our current prior. This works, because in the present paper, we restrict ourselves to stationary inputs comprised of the same image presented multiple times.

A natural choice for $\boldsymbol{\delta u}$ is the gradient of the loss with respect to $\boldsymbol{u}$, through the samples $\boldsymbol{z}$. However, the KL term results in high order terms, which for this implementation we approximate as the following dynamics (See appendix D for a detailed derivation):

$$\boldsymbol{\delta u}_t = \boldsymbol{J}_\theta \cdot \Delta_t = \frac{\partial f_\theta(\boldsymbol{z})}{\partial \boldsymbol{z}}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_t} \cdot \big(\boldsymbol{x}_t - f_\theta(\boldsymbol{z}_t)\big), \tag{7}$$

where $\boldsymbol{J}_\theta$ is the Jacobian of the decoder, $f_\theta$, which is a function of sampled spike counts $\boldsymbol{z}$.

Importantly, this form aligns with real neuronal properties for several reasons. Since the comparison, $\boldsymbol{x}_t - f_\theta(\boldsymbol{z}_t)$, is based on spikes, each neuron's update does not directly depend on the internal states of other neurons, which matches how real neurons function (Kandel et al., 2000). Additionally, because the comparison happens on membrane potential (log rates), feedback will appear as a modulatory signal on rate, which is also consistent with neuroscience literature (Gilbert & Li, 2013). Finally, this update (eq. (7)) resembles a generalization of Rao & Ballard (1999) for nonlinear generative models and avoids hacky solutions to keep rates positive, after subtracting them.

It is straightforward to see how this is an SNN for linear decoder networks. If $f_\theta(\boldsymbol{z}) = \boldsymbol{\Phi}\boldsymbol{z}$, then

$$\begin{aligned}\boldsymbol{\delta u}_t &= \boldsymbol{\Phi}^T(\boldsymbol{x}_t - \boldsymbol{\Phi}\boldsymbol{z}) \\ &= \boldsymbol{\Phi}^T\boldsymbol{x} - \boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{z} \\ &= \boldsymbol{\Phi}^T\boldsymbol{x} - \boldsymbol{W}\boldsymbol{z},\end{aligned} \tag{8}$$

where the first term is the feedforward receptive fields (the input current) and the second term, $\boldsymbol{W}$, are the recurrent weights between neurons, implementing lateral competition. Note that they only communicate with each other through spikes, $\boldsymbol{z}$. Thus for linear generative models, i$\mathcal{P}$-VAE closely resembles the locally competitive algorithm for sparse coding (LCA; Rozell et al. (2008)), except that it is explicitly spiking and does not have a leak term (although this could included by replacing the diagonal of the recurrent term with a leak rather than having neurons operate on their own spikes).

In this section, we showed how following some fairly general assumptions for optimizing ELBO with Poisson distribution, led us to a spiking neural network that implements Bayesian posterior updates via predictive coding in the membrane potential dynamics. In the next section, we evaluate i$\mathcal{P}$-VAE and compare it to amortized $\mathcal{P}$-VAE, as well as iterative Gaussian VAEs.

## 4 EXPERIMENTS

We performed empirical analyses of i$\mathcal{P}$-VAE and alternative iterative VAE models. In section 4.1, we test the general performance and stability of inference dynamics, including generalization to longer sequence lengths. Section 4.2 shows i$\mathcal{P}$-VAE closes the gap with sparse coding. Section 4.3 demonstrates robustness to out-of-distribution (OOD) samples by evaluating models trained on MNIST (LeCun et al., 2010) with perturbed samples (e.g., rotated MNIST). We then evaluate OOD generalization from MNIST to other character-based datasets in section 4.3. Finally, in section 4.4, we visualize the learned weights of i$\mathcal{P}$-VAE, revealing their compositional nature, which is consistent with i$\mathcal{P}$-VAE's strong generalization capabilities. We push the limits of MNIST-trained models by testing their performance on natural images.

**Architecture notation.** We experimented with both convolutional and multi-layer perceptron (MLP) architectures. We highlight the encoder and decoder networks using red and blue, respectively. We use the $\langle \texttt{enc}|\texttt{dec} \rangle$ convention to clearly specify which type was used. For example $\langle \texttt{mlp}|\texttt{mlp} \rangle$ means both encoder and decoder networks were mlp. We use the notation $\langle \texttt{jacob}|\texttt{mlp} \rangle$ to denote our fully iterative (non-amortized) i$\mathcal{P}$-VAE. We chose symmetrical architectures, such that $\langle \texttt{mlp}|\texttt{mlp} \rangle$ has exactly twice as many parameters as $\langle \texttt{jacob}|\texttt{mlp} \rangle$.

**Datasets.** For the generalization results, we use MNIST, extended MNIST (EMNIST; Cohen et al. (2017)), Omniglot (Lake et al., 2015) and Imagenet32 (Chrabaszcz et al., 2017). We resize Omniglot and Imagenet32 to $28 \times 28$ for more straightforward comparisons. We also replicated the sparsity analysis in Fig. 3 of Vafaii et al. (2024) in our Table 1, using the van Hateren natural images dataset with whitened, contrast normalized $16 \times 16$ patches.

**Alternative models.** We compare our iterative $\mathcal{P}$-VAE (i$\mathcal{P}$-VAE) to $\mathcal{P}$-VAE. The main difference between their two architectures is that the latter independently parameterizes an encoder, whereas the former constructs its encoder adaptively by inverting the decoder. We also compare to state-of-the-art methods that combine iterative with amortized inference. These include iterative amortized VAE (ia-VAE; Marino et al. (2018)), and semi-amortized VAE (sa-VAE; Kim et al. (2018)). Since ia-VAE comes with both hierarchical (h) and single-level (s) variants, we compare to each of these.

**Number of iterations.** For i$\mathcal{P}$-VAE, we experimented with different numbers of training iterations, $T_{\text{train}}$. During training, we differentiate through the entire sequence of iterations, which can lead to qualitatively different dynamics. We report results for $T_{\text{train}} = 4, 16, 32, 64$. For generalization results, we use a model with $T_{\text{train}} = 64$. At test time, we report results using $T_{\text{test}} = 1{,}000$ iterations, unless stated otherwise. For semi-amortized models, we use their default number of train and test iterations found in their code, unless stated otherwise (sa-VAE: $T_{\text{train}} = T_{\text{test}} = 20$; ia-VAE: $T_{\text{train}} = T_{\text{test}} = 5$).

### 4.1 STABILITY BEYOND THE TRAINING REGIME AND CONVERGENCE.

An algorithm with strong generalization potential should learn how to perform inference that extends beyond the training regime. We evaluated this by training models on MNIST under different numbers of training iterations, $T_{\text{train}} = 4, 16, 32$, and $64$. We used both $\langle \texttt{jacob}|\texttt{mlp} \rangle$ and $\langle \texttt{jacob}|\texttt{conv} \rangle$ architectures and then tested each model on its ability to keep improving beyond the training number of iterations. In Fig. 1a, we show that i$\mathcal{P}$-VAE converges. Even with as few as 4 iterations, i$\mathcal{P}$-VAE learns to keep improving. We also observe that increasing the number of training iterations has an interesting effect: i$\mathcal{P}$-VAE trained with a larger number of iterations starts from worse performance, but converge to better solutions (Fig. 1a). This suggests i$\mathcal{P}$-VAE learns dynamics that depend on the training sequence length, but generalizes beyond the training set in all cases.

In contrast, the two hybrid models (sa-VAE and ia-VAE) start with strong amortized initial guesses, but plateau rapidly (Fig. 1a, right), and converge to a much higher MSE than i$\mathcal{P}$-VAE models, which have a fraction of the parameters. The authors of sa-VAE were aware of issues regarding dominance of the iterative part of the algorithm for Omniglot, and reported using tricks like gradient clipping to mitigate it, which we suspect is the source for our observations on MNIST (see footnote 6 in Kim et al. (2018)). We also see that ia-VAE (single-level) starts to diverge outside its training regime. [1]

Overall, i$\mathcal{P}$-VAE achieves the best reconstruction performance and continues to improve outside the training regime, unlike other models. This shows the first sign of OOD generalization in i$\mathcal{P}$-VAE: temporal generalization. In later sections, we test whether i$\mathcal{P}$-VAE can generalize OOD in vision tasks, but first, we evaluate the performance and sparsity on natural images as in Vafaii et al. (2024).

### 4.2 IP-VAE CLOSES THE GAP WITH SPARSE CODING

One of the limitations of previous work with $\mathcal{P}$-VAE, was that the authors identified a large performance gap between $\mathcal{P}$-VAE and LCA sparse coding (Vafaii et al., 2024). Here, we evaluated i$\mathcal{P}$-VAE and compared models on their ability to reconstruct whitened natural image patches (table 1). Unlike $\mathcal{P}$-VAE, i$\mathcal{P}$-VAE performs as well as LCA with similar sparsity levels. $\mathcal{P}$-VAE, and the two hybrid approaches, have many more parameters and achieve much worse performance. [2]

[1] It's worth noting that in our hands, ia-VAE (s) often resulted in nans at test time upon going beyond $T_{\text{train}}$.

[2] The performance ia-VAE and sa-VAE might be modestly improved by tuning the tradeoff between reconstruction and the KL term.

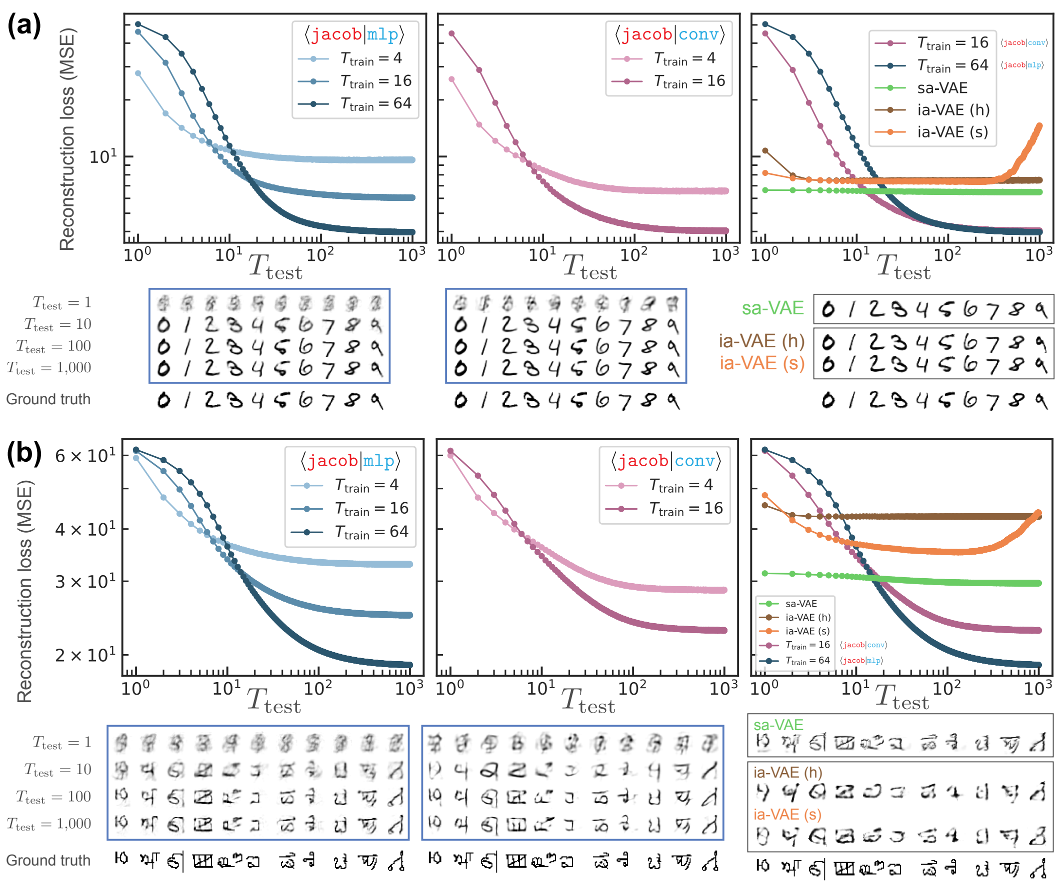


Figure 1: i$\mathcal{P}$-VAE learns to learn. **(a)** Training i$\mathcal{P}$-VAE on as few as $T_{\text{train}} = 4$ time steps allows it to generalize and keep improving its inference beyond the training domain. This holds true irrespective of the i$\mathcal{P}$-VAE architecture; left, ⟨jacob|mlp⟩; middle, ⟨jacob|conv⟩. In contrast, hybrid amortized/iterative models do not improve, and either remain flat or diverge (right). **(b)** i$\mathcal{P}$-VAE trained on MNIST generalizes to Omniglot at test time. All models in this figure were trained on MNIST, and tested either on MNIST (a), or Omniglot (b).

Table 1: Model performance and efficiency. We prefer lightweight models that achieve low reconstruction loss using sparse representations and fewer parameters. We reported results on natural image patches extracted from the van Hateren dataset (Van Hateren & van der Schaaf, 1998). All models have $K = 512$ dimensional latent space. For the i$\mathcal{P}$-VAE models, we scaled the $\beta$ parameter proportional to the number of training inference iterations. Specifically, we chose $\beta = 3/8 * T_{\text{train}}$. We found that i$\mathcal{P}$-VAE results were robust to variations in $\beta$. Entries formatted as mean$_{\pm\text{std}}$.

| Model | $\beta$ | Architecture | # params ↓ | MSE ↓ | Sparsity ↑ | | # iters | |
|---|---|---|---|---|---|---|---|---|
| | | | | | lifetime | % | train | test |
| i$\mathcal{P}$-VAE | 24.00 | ⟨jacob\|lin⟩ | 0.13 $M$ | **12.0**$_{\pm 2.6}$ | 0.79$_{\pm.03}$ | 60.0 | 64 | 1$K$ |
| i$\mathcal{P}$-VAE | 3.00 | ⟨jacob\|lin⟩ | 0.13 $M$ | 27.5$_{\pm 7.1}$ | 0.85$_{\pm.02}$ | 73.2 | 8 | 1$K$ |
| i$\mathcal{P}$-VAE | 1.50 | ⟨jacob\|lin⟩ | 0.13 $M$ | 50.4$_{\pm 15.5}$ | **0.90**$_{\pm.03}$ | **83.3** | 4 | 1$K$ |
| $\mathcal{P}$-VAE | 0.50 | ⟨conv\|lin⟩ | 3.44 $M$ | 101.9$_{\pm 25.3}$ | 0.76$_{\pm.16}$ | 65.9 | 1 | 1 |
| $\mathcal{P}$-VAE | 0.75 | ⟨conv\|lin⟩ | 3.44 $M$ | 119.4$_{\pm 26.4}$ | 0.83$_{\pm.09}$ | 77.7 | 1 | 1 |
| $\mathcal{P}$-VAE | 1.00 | ⟨conv\|lin⟩ | 3.44 $M$ | 131.8$_{\pm 31.2}$ | **0.90**$_{\pm.08}$ | **84.1** | 1 | 1 |
| LCA | 0.28 | - | 0.13 $M$ | **16.1**$_{\pm 8.1}$ | 0.79$_{\pm.02}$ | 65.6 | 1$K$ | 1$K$ |
| LCA | 0.44 | - | 0.13 $M$ | 28.5$_{\pm 14.1}$ | 0.86$_{\pm.02}$ | 73.9 | 1$K$ | 1$K$ |
| LCA | 0.70 | - | 0.13 $M$ | 50.1$_{\pm 25.2}$ | **0.92**$_{\pm.01}$ | **83.4** | 1$K$ | 1$K$ |
| ia-VAE (s) | 1.00 | ⟨mlp\|mlp⟩ | 39.55 $M$ | 80.08$_{\pm 21.06}$ | 0.36$_{\pm.00}$ | ∼0.0 | 5 | 10 |
| sa-VAE | 1.00 | ⟨conv\|conv⟩ | 1.67 $M$ | 97.74$_{\pm 38.97}$ | 0.36$_{\pm.00}$ | ∼0.0 | 20 | 20 |

### 4.3 OUT-OF-DISTRIBUTION GENERALIZATION.

In this section, we evaluate whether MNIST-trained models generalize to OOD perturbations and dataset. First, we tested whether MNIST-trained models generalize to Omniglot (see Fig. 1b). We found that i$\mathcal{P}$-VAE improves over iterations and outperforms alternative models in terms of reconstruction quality. In this section, we evaluate two levels of generalization tasks: (1) within-dataset perturbations; and, (2) across similar datasets (i.e., digits to characters).

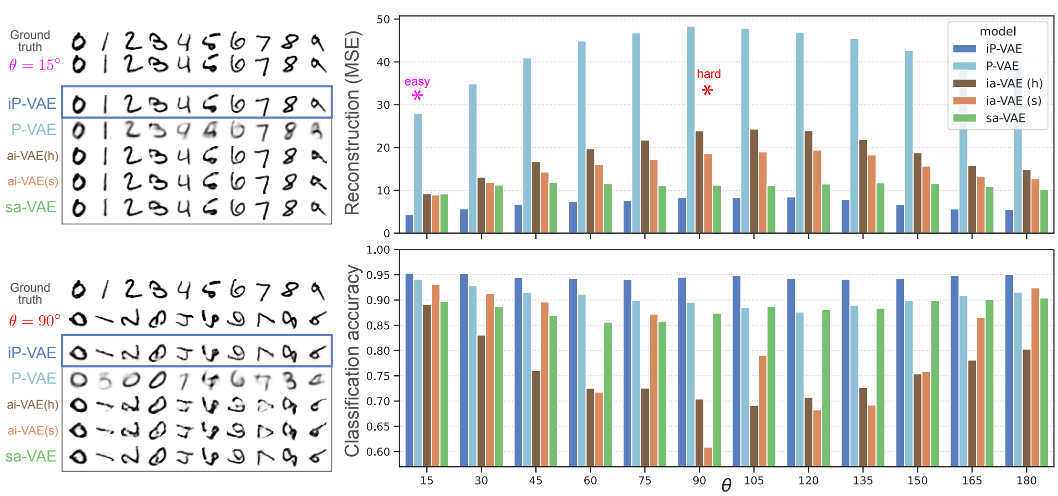


Figure 2: Robustness to training set perturbation. We rotated MNIST digits and evaluated model performance in both reconstruction of the perturbed inputs, and classification accuracy. On the left, we show reconstructed samples for easy ($\theta = 15°$) and hard ($\theta = 90°$) tasks across different models. On the right, we visualize the average reconstruction loss and classification accuracies over different rotations. Both visualy and quantitavely, i$\mathcal{P}$-VAE maintains a high performance regardless of the rotation, and outperforms alternative models.

**OOD generalization to within-dataset perturbation.** We tested whether models trained on standard MNIST generalized to rotated MNIST digits. We rotate MNIST between 0 and 180 degrees, with incremental steps of 15 degrees. We then test (a) whether models are capable of reconstructing the rotated digits, and (b) whether the representations of rotated digits can be used to classify them (Fig. 2). i$\mathcal{P}$-VAE and sa-VAE demonstrated consistent performance across angles, both in terms of reconstruction loss and classification accuracy. Amortized $\mathcal{P}$-VAE shows worse reconstruction performance than all iterative models, but its classification accuracy is remarkably consistent across angles, beating or matching all models except for i$\mathcal{P}$-VAE. ia-VAE variants were greatly affected by the rotation, with significant falloff in both their classification score and reconstruction. Overall, i$\mathcal{P}$-VAE maintains stable performance across rotations at levels above alternative models.

**OOD generalization across similar datasets.** If a model learns compositional features, and if it employs an effective inference algorithm that leverages those features, it should be able to represent datasets that are within the same distributional vicinity as the training set. To test this, we evaluated MNIST-trained models on EMNIST and Omniglot. We report both mean squared error (MSE) of reconstruction and classification accuracy [3].

Again, i$\mathcal{P}$-VAE exhibited superior reconstruction performance over other models, both visually and MSE (Fig. 3). It also had substantially higher classification accuracy, suggesting it learns a compositional code and has strong generalization potential.

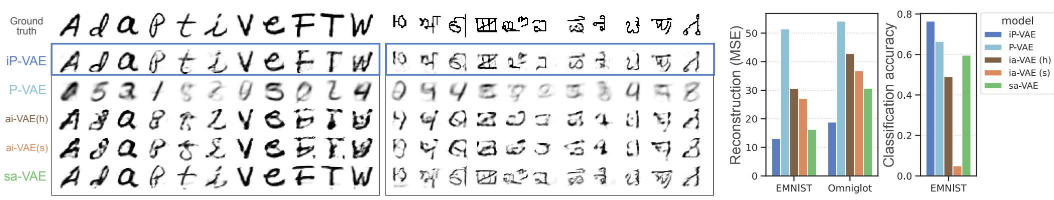


Figure 3: Evaluating generalization from models trained on MNIST digits to novel character datasets (EMNIST and Omniglot) at test time. The right panel shows the average classification performance on latent representations for EMNIST. The middle-right panel compares the reconstruction performance on EMNIST and Omniglot. The left two panels visualize the reconstructions on EMNIST and Omniglot, respectively. In both metrics, i$\mathcal{P}$-VAE maintains high performance compared to alternative models.

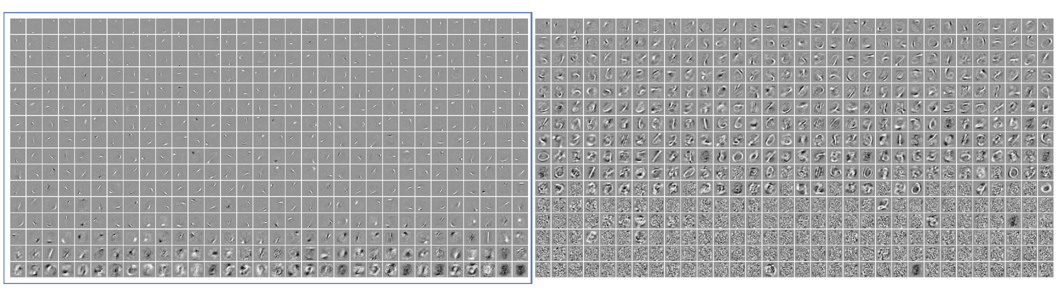

Figure 4: i$\mathcal{P}$-VAE learns a compositional set of features for the last layer's weights, enabling its generalization capacity. Left, i$\mathcal{P}$-VAE with a $\langle$jacob|mlp$\rangle$ architecture; right, $\mathcal{P}$-VAE with an $\langle$mlp|mlp$\rangle$ architecture. Both models were trained on MNIST, but only i$\mathcal{P}$-VAE develops Gabor-like features. In contrast, the non-iterative, amortized $\mathcal{P}$-VAE clearly overfits to MNIST. Features are ordered in ascending order of their weight distribution kurtosis to highlight the sparse nature of i$\mathcal{P}$-VAE feature space. Best viewed when zoomed in.

### 4.4 A COMPOSITIONAL CODE THAT GENERALIZES ACROSS DOMAINS .

Using the $\langle$jacob|mlp$\rangle$ variant of i$\mathcal{P}$-VAE, we visualized the 512 learned features of the last layer of the mlp decoder. In Fig. 4, we show the features learned by i$\mathcal{P}$-VAE trained on MNIST and contrast

[3] We omit classification accuracy for Omniglot due to its large number of classes (over 1,000)

them to features learned by $\mathcal{P}$-VAE, also trained on MNIST. We see a stark contrast. i$\mathcal{P}$-VAE features are Gabor-like, while $\mathcal{P}$-VAE features look like digits or strokes of the digits. While previous work highlighted strokes as the compositional subcomponents of digits (Lee et al., 2007), i$\mathcal{P}$-VAE learns an even more general code that generalized to cropped, grey scaled natural images (Fig. 5).

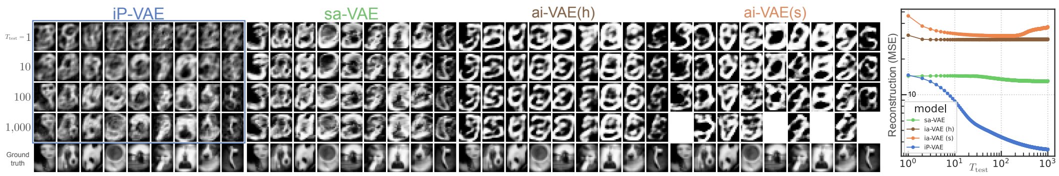


Figure 5: Evaluating generalization from models trained on MNIST digits to cropped, gray scaled natural images (ImageNet32) at test time. The right panel shows average reconstruction performance over inference iterations for the entire dataset. The left panels visualizes selected ground truth images compared with model reconstructions. The ai-VAE variants are unable to adapt to the new domain, whereas sa-VAE can capture more details. i$\mathcal{P}$-VAE outperforms the alternatives, and its reconstructions are shown to maintain the semantic information of ground truth images.

Since both i$\mathcal{P}$-VAE and $\mathcal{P}$-VAE are spiking models, this result suggests that the difference lies in the inference algorithm: i$\mathcal{P}$-VAE is iterative and adaptive; whereas, $\mathcal{P}$-VAE is one-shot amortized. Overall, our experiments provide strong evidence for the utility of iterative algorithms in practical settings.

## 5 Discussion and Conclusions

In this work, we introduce the i$\mathcal{P}$-VAE, which is a spiking neural network that maximizes ELBO, while performing Bayesian posterior updates through membrane potential dynamics. Empirically, i$\mathcal{P}$-VAE exhibits outstanding adaptability and robustness to OOD samples, while being able to dynamically trade off compute and performance. It outperforms amortized versions and recent iterative inference VAEs on every task we tested while using substantially fewer parameters.

i$\mathcal{P}$-VAE results directly from the choice of Poisson in the ELBO and it avoids many of the problems with predictive coding. First, there is no population-wide prediction signal, only a feedforward receptive field and recurrent terms. Second, neurons only communicate through spikes and all dynamics are private on the membrane potential. And finally, additive terms in the membrane potential appear as gains in the spike rate, which avoids negative rates, and is more consistent with real neurons Gilbert & Li (2013).

We believe i$\mathcal{P}$-VAE is well positioned for a neuromorphic implementation. The recent rise of neuromorphic hardware as an avenue for performance improvements requires new algorithms that can make use of its architecture (Schuman et al., 2022). We found that i$\mathcal{P}$-VAE with a linear decoder reduces to a spiking LCA, addressing the performance gap noted by Vafaii et al. (2024). Both algorithms share key features: sparsity, recurrence, and parameter efficiency. Since LCA has been implemented as an SNN (Zylberberg et al., 2011) and on neuromorphic hardware (Du et al., 2024), we expect the same for i$\mathcal{P}$-VAE.

In summary, the choice of Poisson in the ELBO results in a spiking neural network, i$\mathcal{P}$-VAE, that performs iterative Bayesian inference. This lays the groundwork for a prescriptive theoretical framework for building brain-like generative models that can leverage neuromorphic hardware.

**Limitations and future work.** In our experiments, we tested the simplest version of i$\mathcal{P}$-VAE, showing the practical benefits of the derived theory. There are a few avenues that we did not test, and we think are exciting for future work. The design of a hierarchical model is a natural extension for brain-like algorithm, especially given evidence that hierarchical VAE are more aligned to the brain (Vafaii et al., 2023). In addition, training and evaluating on nonstationary sequences like videos would be a straightforward extension, as we derived the theory with this in mind. When attempting to use such sequences, it may also be beneficial to explore more sophisticated forward-predictive models that "evolve" current posteriors to future priors.

# REFERENCES

Larry F Abbott, JA Varela, Kamal Sen, and SB Nelson. Synaptic depression and cortical gain control. *Science*, 275(5297):221–224, 1997.

Laurence Aitchison and Máté Lengyel. With or without you: predictive coding and bayesian inference in the brain. *Current opinion in neurobiology*, 46:219–227, 2017.

Alhazen. *Book of optics (Kitab Al-Manazir)*. 1011–1021 AD.

Christina Allen and Charles F Stevens. An evaluation of causes for unreliability of synaptic transmission. *Proceedings of the National Academy of Sciences*, 91(22):10380–10383, 1994.

Mel Andrews. The math is not the territory: navigating the free energy principle. *Biology & Philosophy*, 36(3):30, 2021.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.

Bruno B Averbeck, Peter E Latham, and Alexandre Pouget. Neural correlations, population coding and computation. *Nature reviews neuroscience*, 7(5):358–366, 2006.

Jimmy Ba, Geoffrey E Hinton, Volodymyr Mnih, Joel Z Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past. *Advances in neural information processing systems*, 29, 2016.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/01386bd6d8e091c2ab4c7c7de644d37b-Paper.pdf`.

Shaojie Bai, Vladlen Koltun, and J. Zico Kolter. Multiscale deep equilibrium models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5238–5250. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/3812f9a59b634c2a9c574610eaba5bed-Paper.pdf`.

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S. Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise, 2022.

Andre M Bastos, W Martin Usrey, Rick A Adams, George R Mangun, Pascal Fries, and Karl J Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, 2012. doi: 10.1016/j.neuron.2012.10.038.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Martin Boerlin, Christian K Machens, and Sophie Denève. Predictive coding of dynamical variables in balanced spiking networks. *PLoS computational biology*, 9(11):e1003258, 2013.

Victor Boutin, Aimen Zerroug, Minju Jung, and Thomas Serre. Iterative vae as a predictive brain model for out-of-distribution generalization. *arXiv preprint arXiv:2012.00557*, 2020.

Daniel A Butts, Yuwei Cui, and Alexander RR Casti. Nonlinear computations shaping temporal processing of precortical vision. *Journal of Neurophysiology*, 116(3):1344–1357, 2016.

William H Calvin and CHARLES F Stevens. Synaptic noise and other sources of randomness in motoneuron interspike intervals. *Journal of neurophysiology*, 31(4):574–587, 1968.

Matteo Carandini. Amplification of trial-to-trial response variability by neurons in visual cortex. *PLoS biology*, 2(9):e264, 2004.

Stanley H. Chan. Tutorial on diffusion models for imaging and vision. 2024. URL https://arxiv.org/abs/2403.18103.

Michael Chang, Thomas L. Griffiths, and Sergey Levine. Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=-5rFUTO2NWe.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/b618c3210e934362ac261db280128c22-Paper.pdf.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pp. 1078–1086. PMLR, 2018.

Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press, 2005.

Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural Computation*, 7(5):889–904, 1995. doi: 10.1162/neco.1995.7.5.889.

AF Dean. The variability of discharge of simple cells in the cat striate cortex. *Experimental Brain Research*, 44(4):437–440, 1981.

Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration, 2024. URL https://arxiv.org/abs/2303.11435.

Xuexing Du, Zhong-qi K Tian, Songting Li, and Douglas Zhou. A generalized spiking locally competitive algorithm for multiple optimization problems. *arXiv preprint arXiv:2407.03930*, 2024.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Nicolas Fourcaud-Trocmé, David Hansel, Carl Van Vreeswijk, and Nicolas Brunel. How spike generation mechanisms determine the neuronal response to fluctuating inputs. *Journal of neuroscience*, 23(37):11628–11640, 2003.

Karl Friston. A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological Sciences*, 360(1456):815–836, 2005. doi: 10.1098/rstb.2005.1622.

Karl Friston. The free-energy principle: a rough guide to the brain? *Trends in cognitive sciences*, 13 (7):293–301, 2009.

Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2): 127–138, 2010. doi: 10.1038/nrn2787.

Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014. URL `https://escholarship.org/uc/item/34j1h7k5`.

Samuel J Gershman. What does the free energy principle tell us about the brain? *arXiv preprint arXiv:1901.07945*, 2019.

Charles D Gilbert and Wu Li. Top-down influences on visual processing. *Nature Reviews Neuroscience*, 14(5):350–363, 2013.

Robbe LT Goris, J Anthony Movshon, and Eero P Simoncelli. Partitioning neuronal variability. *Nature neuroscience*, 17(6):858–865, 2014.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Geoffrey E Hinton and David C Plaut. Using fast weights to deblur old memories. In *Proceedings of the ninth annual conference of the Cognitive Science Society*, pp. 177–186, 1987.

Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Arthur Hobson. A new theorem of information theory. *Journal of Statistical Physics*, 1:383–391, 1969.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90020-8. URL `https://www.sciencedirect.com/science/article/pii/0893608089900208`.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5149–5169, 2021.

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. Going beyond linear transformers with recurrent fast weight programmers. *Advances in neural information processing systems*, 34:7703–7717, 2021.

Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and St'ephane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representation. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=ANvmVS2Yr0`.

R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960. ISSN 0021-9223. doi: 10.1115/1.3662552. URL `https://doi.org/10.1115/1.3662552`.

Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, A James Hudspeth, Sarah Mack, et al. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.

Georg B Keller and Thomas D Mrsic-Flogel. Predictive processing: a canonical cortical computation. *Neuron*, 100(2):424–435, 2018.

Minyoung Kim and Vladimir Pavlovic. Reducing the amortization gap in variational autoencoders: A bayesian random function approach. *arXiv preprint arXiv:2102.03151*, 2021.

Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, pp. 2678–2687. PMLR, 2018.

Diederik P Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=NnMEadcdyD.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. 2014.

Naoki Kogo and Chris Trengove. Is predictive coding theory articulated enough to be testable?, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi: 10.1038/nature14539.

Honglak Lee, Chaitanya Ekanadham, and Andrew Ng. Sparse deep belief net model for visual area v2. *Advances in neural information processing systems*, 20, 2007.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11525–11538. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8511df98c02ab60aea1b2356c013bc0f-Paper.pdf.

William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ewdt9xe.

Calvin Luo. Understanding diffusion models: A unified perspective. arxiv 2022. *arXiv preprint arXiv:2208.11970*, 2022.

Laurin Luttmann and Paolo Mercorelli. Comparison of backpropagation and kalman filter-based training for neural networks. In *2021 25th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 234–241, 2021. doi: 10.1109/ICSTCC52150.2021.9607274.

Zachary F Mainen and Terrence J Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, 268(5216):1503–1506, 1995.

Joe Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3403–3412. PMLR, 7 2018. URL https://proceedings.mlr.press/v80/marino18a.html.

Joseph Marino. Predictive coding, variational autoencoders, and biological connections. *Neural Computation*, 34(1):1–44, 2022. doi: 10.1162/neco_a_01458.

Joseph Marino, Alexandre Piché, Alessandro Davide Ialongo, and Yisong Yue. Iterative amortized policy optimization. *Advances in Neural Information Processing Systems*, 34:15667–15681, 2021.

Fabian A Mikulasch, Lucas Rudelt, Michael Wibral, and Viola Priesemann. Where is the error? hierarchical predictive coding through dendritic error computation. *Trends in Neurosciences*, 46 (1):45–59, 2023.

Beren Millidge, Anil K. Seth, and Christopher L. Buckley. Predictive coding: a theoretical and experimental review. *CoRR*, abs/2107.12979, 2021a. URL https://arxiv.org/abs/2107.12979.

Beren Millidge, Alexander Tschantz, Anil Seth, and Christopher Buckley. Neural kalman filtering, 2021b. URL https://arxiv.org/abs/2102.10021.

Beren Millidge, Tommaso Salvatori, Yuhang Song, Rafał Bogacz, and Thomas Lukasiewicz. Predictive coding: Towards a future of deep learning beyond backpropagation? In *International Joint Conference on Artificial Intelligence*, 2022. doi: 10.24963/ijcai.2022/774.

Beren Millidge, Mufeng Tang, Mahyar Osanlouy, Nicol S Harper, and Rafal Bogacz. Predictive coding networks for temporal prediction. *PLOS Computational Biology*, 20(4):e1011183, 2024.

Sreyas Mohan, Joshua L Vincent, Ramon Manzorro, Peter Crozier, Carlos Fernandez-Granda, and Eero Simoncelli. Adaptive denoising via gaintuning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 23727–23740. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/c7558e9d1f956b016d1fdba7ea132378-Paper.pdf.

David Mumford. On the computational architecture of the neocortex: Ii the role of cortico-cortical loops. *Biological Cybernetics*, 66(3):241–251, 1992. doi: 10.1007/BF00198477.

Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. doi: 10.1038/381607a0.

Bruno A Olshausen and David J Field. Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–487, 2004. doi: 10.1016/j.conb.2004.07.007.

Nicholas J Priebe, Ferenc Mechler, Matteo Carandini, and David Ferster. The contribution of spike threshold to the dichotomy of cortical simple and complex cells. *Nature neuroscience*, 7(10): 1113–1122, 2004. doi: 10.1038/nn1310.

Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, 1999. doi: 10.1038/4580.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286. PMLR, 2014. URL https://proceedings.mlr.press/v32/rezende14.html.

Fred Rieke, David Warland, Rob de Ruyter Van Steveninck, and William Bialek. *Spikes: exploring the neural code*. MIT press, 1999.

Christopher J Rozell, Don H Johnson, Richard G Baraniuk, and Bruno A Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural Computation*, 20(10):2526–2563, 2008. doi: 10.1162/neco.2008.03-07-486.

Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, 2016.

Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

C. D. Schuman, S. R. Kulkarni, M. Parsa, J. P. Mitchell, P. Date, and B. Kay. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2022. doi: 10.1038/s43588-022-00223-2.

Terrence J Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117(48):30033–30038, 2020.

Michael N Shadlen and William T Newsome. The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *Journal of neuroscience*, 18(10):3870–3896, 1998.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Mandyam Veerambudi Srinivasan, Simon Barry Laughlin, and Andreas Dubs. Predictive coding: a fresh view of inhibition in the retina. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 216(1205):427–459, 1982. doi: 10.1098/rspb.1982.0085.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): Rnns with expressive hidden states, 2024. URL https://arxiv.org/abs/2407.04620.

Malvin C Teich. Fractal character of the auditory neural spike train. *IEEE Transactions on Biomedical Engineering*, 36(1):150–160, 1989.

Michael Teti. Lca-pytorch. [Computer Software] https://doi.org/10.11578/dc.20230728.4, jun 2023. URL https://doi.org/10.11578/dc.20230728.4.

David J Tolhurst, J Anthony Movshon, and Andrew F Dean. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision research*, 23(8):775–785, 1983.

Margaret Trautner, Gabriel Margolis, and Sai Ravela. Informative neural ensemble kalman learning, 2020. URL https://arxiv.org/abs/2008.09915.

Wilson Truccolo, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery N Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, 2005.

Hadi Vafaii, Jacob L. Yates, and Daniel A. Butts. Hierarchical VAEs provide a normative account of motion processing in the primate brain. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1wOkHN9JK8.

Hadi Vafaii, Dekel Galor, and Jacob L. Yates. Poisson variational autoencoder. 2024. URL https://arxiv.org/abs/2405.14473.

J Hans Van Hateren and Arjen van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265(1394):359–366, 1998.

Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Hermann Von Helmholtz. *Handbuch der physiologischen Optik*, volume 9. Voss, 1867. URL https://archive.org/details/handbuchderphysi00helm.

Kevin S Walsh, David P McGovern, Andy Clark, and Redmond G O'Connell. Evaluating the neurophysiological evidence for predictive processing as a model of perception. *Annals of the new York Academy of Sciences*, 1464(1):242–268, 2020.

Alison I Weber and Jonathan W Pillow. Capturing the dynamical repertoire of single neurons with generalized linear models. *Neural computation*, 29(12):3260–3289, 2017.

Haiguang Wen, Kuan Han, Junxing Shi, Yizhen Zhang, Eugenio Culurciello, and Zhongming Liu. Deep predictive coding network for object recognition. In *International conference on machine learning*, pp. 5266–5275. PMLR, 2018.

B. Widrow. *Adaptive "adaline" Neuron Using Chemical "memistors."*. 1960. URL https://books.google.com/books?id=Yc4EAAAAIAAJ.

Bernard Widrow and Samuel D. Stearns. *Adaptive Signal Processing*. Prentice-Hall PTR, 1985.

Robert Wilson and Leif Finkel. A neural implementation of the kalman filter. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper_files/paper/2009/file/6d0f846348a856321729a2f36734d1a7-Paper.pdf.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications, 2024. URL https://arxiv.org/abs/2209.00796.

Han Yu, Jiashuo Liu, Xingxuan Zhang, Jiayun Wu, and Peng Cui. A survey on evaluation of out-of-distribution generalization. *arXiv preprint arXiv:2403.01874*, 2024.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2022.

Joel Zylberberg, Jason Timothy Murphy, and Michael Robert DeWeese. A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of v1 simple cell receptive fields. *PLoS computational biology*, 7(10):e1002250, 2011.

## A Experiment details

In our comparisons to previous work, we utilized the code accompanied with sa-VAE (Kim et al. (2018)), ai-VAE (Marino et al. (2018)), and $\mathcal{P}$-VAE Vafaii et al. (2024). Across models where code was provided, we trained using the same train/validation split, and without changing the parameters in the code unless we specify otherwise. For the locally competitive algorithm (LCA) baseline, we used the library lca-pytorch (Teti, 2023) to replicate the analysis from Vafaii et al. (2024).

Since the code for sa-VAE was limited to a Bernoulli observation model, we adapted it for compatibility to Gaussian by removing the sigmoid in the decoder and replacing its reconstruction loss with MSE (for the van Hateren dataset). For sa-VAE, only Omniglot parameters were provided, with default batch size of 50, and default number of epochs of 100. We trained it on Omniglot with default parameters, on van Hateren for 100 epochs and batch size 200, on MNIST for 32 epochs and batch size 50, and EMNIST for 16 epochs and batch size 50, adjusting for the size and complexity of datasets.

The codebase for ai-VAE included parameters for both Bernoulli and Gaussian observation models, and we use them accordingly. We used their MNIST configuration for MNIST, EMNIST, and Omniglot. We used their CIFAR configuration for van Hateren, except for increasing batch size to 200 (van Hateren is much smaller spatially). For training the ai-VAE single-level model on van Hateren, we matched the latent dimension to all other van Hateren models (512 dims instead of 1024 from the CIFAR configuration). The number of epochs in the ai-VAE code base is hardcoded to 2000, but we stopped the models between 780 and 2000 epochs when the loss converged. We found that the training code occasionally resulted in nans, requiring rerunning the training from the checkpoint. In one case, the hierarchical van Hateren model, the training was unable to proceed past 61 epochs without stopping due to nans.

We obtained the $\mathcal{P}$-VAE code upon request from the authors and used the default parameters as described in the appendix of Vafaii et al. (2024).

# B ARE REAL NEURONS TRULY POISSON?

In this section, we discuss empirical and theoretical observations from neuroscience that support our Poisson assumption.

"Poisson-like" noise in neuroscience has a long history. It begins with observations that neurons do not fire the same sequence of spikes to repeated presentations of the same input and that the variance is proportional to the mean (Tolhurst et al., 1983; Dean, 1981) and was followed by the observation that for short counting windows, that proportion is 1 (Teich, 1989; Shadlen & Newsome, 1998; Averbeck et al., 2006; Rieke et al., 1999; Dayan & Abbott, 2005). Larger windows and higher visual areas are notably super-Poisson, but that can be attributed to a modulation of the rate of an inhomogeneous Poisson process (Goris et al., 2014). In other words, neurons are conditionally Poisson, not marginally Poisson (Truccolo et al., 2005).

Spike-generation, it is argued, is not noisy (Mainen & Sejnowski, 1995; Calvin & Stevens, 1968), but synaptic noise (Allen & Stevens, 1994) or noise on the membrane potential can create a Poisson-like distributions of spikes (Carandini, 2004). An important caveat is that the most famous examples of precision in spike generation, Mainen & Sejnowski (1995), is well captured well by a Poisson-process Generalized linear model (Weber & Pillow, 2017), although that precision depends on the Bernoulli approximation to a Poisson process in the limit where only 0 or 1 spikes are possible. There is a widely-held misconception that precise timing cannot be produced by spike-rate models, but inhomogeneous rate models can operate at high time resolution and produce precise spiking (Butts et al., 2016).

Importantly, to maximize the ELBO, one has to choose an approximate posterior and prior. Because spike counts are integer and cannot be negative, Poisson is a more natural choice than Gaussian without knowing anything about neural firing statistics. Here, we found that Poisson assumption produced a prescriptive theory for neural coding. Future work might interpret this assumption at higher time resolution using inhomeneous Poisson processes in the limit of binary spiking.

# C EXTENDED RELATED WORKS

## C.1 DIFFUSION MODELS

Diffusion models have recently gained significant traction in various generative tasks, demonstrating impressive performance across applications (Yang et al., 2024; Chan, 2024). Originally introduced by Sohl-Dickstein et al. (2015), these models iteratively restore data structure by learning a reverse diffusion process. Despite the dominance of one-shot feedforward methods, the success of diffusion models highlights the ongoing relevance of iterative approaches. Several studies have sought to explain why these models perform so well in tasks like image generation. In this section, we highlight three key findings.

First, Delbracio & Milanfar (2024) and Bansal et al. (2022) showed that fully deterministic iterative restoration methods, without diffusion theory, can match the performance of conditional diffusion models. This suggests that the strength of diffusion models lies, at least partially, in their iterative nature.

Second, Kingma & Gao (2023) revealed that despite their distinct loss functions, diffusion models essentially optimized the ELBO objective (identical under certain conditions), particularly in noise-perturbed data settings. This adds further support to the idea that diffusion models succeed not because of their diffusion-specific properties, but because they are iterative, aligning them closely with i$\mathcal{P}$-VAE, which also optimizes an ELBO-like objective through iterative processes.

Finally, Kadkhodaie et al. (2024) found that diffusion models operate by applying a shrinkage operation on an adaptive basis, a fundamental concept in signal processing. In methods like sparse coding, this is represented by an $L1$ regularization term. Similarly, an $L1$-like term appears in i$\mathcal{P}$-VAE, which also uses integer representations to zero out small values. These similarities suggest a strong connection between i$\mathcal{P}$-VAE and diffusion models, presenting an exciting direction for future research.

### C.2 Adaptive Filters

Adaptive filters are a widely used class of algorithms capable of modeling signals with varying statistics (Widrow & Stearns (1985)). Their applications are highly diverse, including communications, control and robotics, weather prediction, and inverse problems such as denoising. Two of the most popular adaptive filter classes, the Kalman filter (Kalman (1960)) and the Least mean squares (LMS) filter (Widrow & Stearns (1985)), have close connections to machine learning. The LMS filter was originally based on research aiming to train neural networks (Widrow (1960)). Backpropagation can be understood as a generalization of the LMS filter when applied to multi-layer networks. Although the Kalman filter has not had much use as a learning algorithm, a recent line of work shows that there is a lot of potential benefits in doing so (Trautner et al. (2020); Luttmann & Mercorelli (2021)). Both algorithms, when used in dynamic settings, encode the prediction residual (like i$\mathcal{P}$-VAE), and can be interpreted from the framework of predictive coding. More concretely, Millidge et al. (2021a) showed predictive coding in the linear case corresponds to Kalman filtering, and also showed the relationship between backpropagation (extension of LMS) and predictive coding. Later, Millidge et al. (2021b) showed that predictive coding and Kalman filtering, although not identical in general, optimize the same objective. In addition, they show a neurally plausible implementation of the Kalman filter (see Wilson & Finkel (2009) for an earlier paper in this line of work).

In future work, it would be interesting to incorporate additional ideas from the rich literature of Kalman filters. Particularly, extensions of Kalman filtering, such as the ensemble Kalman filtering, tend to be better suited for nonlinear and nongaussian applications (albeit with the loss of guarantees).

### C.3 Test-time optimization

There has been a recent surge of work showing that incorporating test-time optimization leads to improved performance. One notable line of work is known as Test-Time-Training (TTT), introduced by Sun et al. (2020). TTT is a general approach for updating model parameters in test time using self-supervised learning, demonstrating increased performance and robustness. Around the same time Quan et al. (2020) introduced Self2Self, a denoising method that is only trained during test time. A follow-up to Self2Self instead optimized a per-layer gain value of a trained model Mohan et al. (2021). In a recent paper, Sun et al. (2024) extended the TTT framework to language modeling, introducing an architecture that outperforms transformers (Vaswani et al., 2017) and Mamba (Gu & Dao, 2023). The authors also showed that theoretically, transformers can be understood as a special case of their TTT algorithm. In this work, we found that i$\mathcal{P}$-VAE can also be understood within the TTT framework. Overall, our results reveal a novel grounding of TTT within well-established theoretical concepts in neuroscience.

### C.4 Feedforward versus iterative computation

Deep learning is currently the dominant paradigm in artificial intelligence (AI) research, driven largely by the success of feedforward neural networks (LeCun et al., 2015; Sejnowski, 2020). The deep learning era invoked the universal approximation theorem (Hornik et al., 1989) and emphasized parallelization of training (Krizhevsky et al., 2012; Vaswani et al., 2017) leading to an over-reliance on models that perform one-shot inference. This “unrolling” of inference diverged from the classic AI literature, which recognized the importance of iterative algorithms (Russell & Norvig, 2016). Although feedforward models initially achieved remarkable results, their limitations became increasingly apparent as they struggled to generalize beyond their training distributions (Zhou et al., 2022; Yu et al., 2024). To counter this limitation, iterative computation at test time has recently resurfaced as a promising direction (Sun et al., 2020; 2024).

Unlike feedforward models, iterative algorithms refine their predictions over multiple steps, allowing them to adapt dynamically to new inputs. Examples include iterative amortized inference techniques Marino et al. (2018); Kim et al. (2018), diffusion models Sohl-Dickstein et al. (2015); Ho et al. (2020); Song & Ermon (2019), energy based models (Du & Mordatch, 2019; LeCun et al., 2006), test-time training Sun et al. (2020; 2024), meta-learning algorithms (Andrychowicz et al., 2016; Finn et al., 2017; Hospedales et al., 2021), neural ordinary differential equations (Chen et al., 2018), deep equilibrium models (Bai et al., 2019; 2020), object-centric models (Locatello et al., 2020; Chang et al., 2022), and many more. These methods have demonstrated that a dynamic, multi-step inference process can help overcome many of the challenges faced by static models.

### C.5 Fast weights

In the late 1980s and early 1990s, Hinton & Plaut (1987) and Schmidhuber (1992) introduced the concept of "fast weights" as a way to enhance the adaptability of neural networks through dynamic memory. These innovations laid the foundation for modern models like transformers and recurrent neural networks, significantly influencing memory-augmented architectures and iterative inference methods. Fast weights are particularly relevant in iterative inference, where dynamic updates align with the goal of flexible, adaptive neural computation (Ba et al., 2016; Irie et al., 2021). In our work, the adaptive Bayesian posterior updates in $\boldsymbol{u}(t)$—the membrane potential state of i$\mathcal{P}$-VAE—closely parallel the concept of fast weights.

## D Dynamics

In this section, we will go through the derivation of the dynamics of i$\mathcal{P}$-VAE ( eq. (7) in the main paper). Our goal is to define membrane potential updates in a way that the resulting dynamics will minimize the ELBO loss.

We begin with the general definition of the ELBO, $\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right]$, and consider its Monte Carlo estimate using a single sample, $\boldsymbol{z}$, drawn from the approximate posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$:

$$
\begin{aligned}
\ell(\boldsymbol{x},\boldsymbol{z}) &\coloneqq \log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \\
&= \log \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \\
&= \log p(\boldsymbol{x}|\boldsymbol{z}) + \log \frac{p(\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \\
&= -\mathrm{MSE}(\boldsymbol{x},\boldsymbol{z}) + \boldsymbol{r} \odot (\exp(\boldsymbol{\delta u}) - 1) - \boldsymbol{z} \odot \boldsymbol{\delta u}.
\end{aligned} \tag{9}
$$

In the last line of eq. (9), we inserted our specific choice of Gaussian conditional density, resulting in $\log p(\boldsymbol{x}|\boldsymbol{z}) = -\mathrm{MSE}(\boldsymbol{x},\boldsymbol{z}) = -\left\|\boldsymbol{x} - f_\theta(\boldsymbol{z})\right\|^2$. We also expressed the log ratio between the prior and approximate posterior distributions, both modeled as Poisson, as in the case in i$\mathcal{P}$-VAE.

Next, we take the partial derivative of $\ell(\boldsymbol{x},\boldsymbol{z})$ w.r.t the samples $\boldsymbol{z}$ and keep only the first order terms. This results in:

$$
\frac{\partial}{\partial \boldsymbol{z}}\ell(\boldsymbol{x},\boldsymbol{z}) \approx -\frac{\partial}{\partial \boldsymbol{z}}\mathrm{MSE}(\boldsymbol{x},\boldsymbol{z}) - \boldsymbol{\delta u}. \tag{10}
$$

If we define our posterior updates, $\boldsymbol{\delta u}$, to be proportional to the gradient of $\ell(\boldsymbol{x},\boldsymbol{z})$ w.r.t the state variable, $\boldsymbol{u}$, we get:

$$\begin{aligned}\boldsymbol{\delta u} &\coloneqq \alpha \nabla_{\boldsymbol{u}} \ell(\boldsymbol{x}, \boldsymbol{z}) \\ &= \alpha \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{u}} \frac{\partial}{\partial \boldsymbol{z}} \ell(\boldsymbol{x}, \boldsymbol{z}) \\ &\approx -\alpha \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{u}} \left[ \frac{\partial}{\partial \boldsymbol{z}} \mathrm{MSE}(\boldsymbol{x}, \boldsymbol{z}) + \boldsymbol{\delta u} \right],\end{aligned} \tag{11}$$

where $\alpha$ is a proportionality constant. We rearrange some terms to get the following update rule:

$$\boldsymbol{\delta u} = -\left( \frac{\alpha \partial \boldsymbol{z} / \partial \boldsymbol{u}}{1 + \alpha \partial \boldsymbol{z} / \partial \boldsymbol{u}} \right) \frac{\partial}{\partial \boldsymbol{z}} \mathrm{MSE}(\boldsymbol{x}, \boldsymbol{z}). \tag{12}$$

The stochastic samples, $\boldsymbol{z}$, depend to the state variable, $\boldsymbol{u}$, through firing rates, $\boldsymbol{r} = \exp(\boldsymbol{u})$. Therefore, we have $\partial \boldsymbol{z} / \partial \boldsymbol{u} = (\partial \boldsymbol{z} / \partial \boldsymbol{r}) (\partial \boldsymbol{r} / \partial \boldsymbol{u})$. But $\partial \boldsymbol{r} / \partial \boldsymbol{u}$ is just $\boldsymbol{r}$, and if we approximate $\partial \boldsymbol{z} / \partial \boldsymbol{r}$ using the straight-through estimator, we will have $\partial \boldsymbol{z} / \partial \boldsymbol{u} \approx \boldsymbol{r}$. Plug this back into eq. (12) to get:

$$\boldsymbol{\delta u} \approx -\left( \frac{\alpha \boldsymbol{r}}{1 + \alpha \boldsymbol{r}} \right) \frac{\partial}{\partial \boldsymbol{z}} \mathrm{MSE}(\boldsymbol{x}, \boldsymbol{z}). \tag{13}$$

The proportionality coefficient, $\alpha \boldsymbol{r} / (1 + \alpha \boldsymbol{r})$, can be interpreted as an adaptive learning rate that depends on the instantaneous firing rate of neurons. While this result is intriguing, in the present work we simplified our update rule by removing the proportionality coefficient. Instead, we simply used the gradient of the MSE to compute $\boldsymbol{\delta u}$:

$$\begin{aligned}\boldsymbol{\delta u} &\propto -\frac{\partial}{\partial \boldsymbol{z}} \mathrm{MSE}(\boldsymbol{x}, \boldsymbol{z}) \\ &= -\frac{\partial}{\partial \boldsymbol{z}} \left\| \boldsymbol{x} - f_\theta(\boldsymbol{x}) \right\|^2 \\ &\propto \frac{\partial f_\theta(\boldsymbol{z})}{\partial \boldsymbol{z}} \cdot \left( \boldsymbol{x}_t - f_\theta(\boldsymbol{z}_t) \right) \\ &= \boldsymbol{J}_\theta \cdot \Delta_t.\end{aligned} \tag{14}$$

This concludes our derivation of eq. (7).