INFERENCE AND LEARNING IN SPARSE AUTOENCODERS AS NATURAL GRADIENT FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoders are widely used to uncover interpretable features in neural networks, yet reliable recovery remains difficult when features overlap or activate infrequently. These challenges involve both inferring which features explain an input and learning the dictionary that represents them. Here, we unify inference and dictionary learning as natural-gradient flows on a shared variational free energy. We instantiate this framework as BeFOND, an encoder-free sparse coding model with closed-form inference and learning dynamics. We show how recurrent explaining away reduces interference between overlapping features, while Fisher preconditioning can compensate for the slow learning of rare features. On synthetic data, BeFOND improves dictionary recovery and rare-feature detection, with a growing advantage over amortized baselines as superposition increases. On language-model activations, it improves single-feature concept detection and selective intervention, outperforming pretrained reference SAEs with substantially less training data. Its feature quality continues to improve with dictionary width, whereas the evaluated baselines largely plateau. Together, these results show how improving inference and learning within a unified probabilistic framework can make better use of data and dictionary capacity to interpret and intervene on neural representations.

1 INTRODUCTION

Sparse autoencoders (SAEs) infer sparse representations of data in terms of a learned, overcomplete dictionary of features. Their roots lie in sparse coding (Olshausen & Field, 1996; Lewicki & Sejnowski, 2000), motivated by the organization of sensory representations in biological systems (Olshausen & Field, 1997; 2004). SAEs have since become central in interpreting neural network representations (reviewed in Klindt et al., 2026), especially those of language models (Yun et al., 2021; Cunningham et al., 2024; Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2025).

Despite these successes, SAEs face challenges in both inference and dictionary learning. Standard SAEs use a trained neural network to perform one-pass inference—known as *amortized* inference—which can struggle to disentangle overlapping features in superposition (O'Neill et al., 2025). But improving inference alone is insufficient. In controlled experiments, Barin-Pacela et al. (2026) identified dictionary learning as the bottleneck: replacing an SAE's encoder with iterative inference on the same learned dictionary failed to close the feature-recovery gap. These problems are coupled during training: the dictionary determines which explanations of the data are available, while inferred feature activations guide dictionary updates. Therefore, poor inference can compromise dictionary learning, and a poor dictionary can limit what even accurate inference recovers.

Unlike standard SAEs, classical sparse coding models infer representations through iterative, per-sample refinement (Rozell et al., 2008). More recently in this lineage, Vafaii et al. (2025) introduced the *Free-energy Online Natural-gradient Dynamics* (FOND) framework, which formulates inference as natural-gradient flow on variational free energy. FOND's guiding principle is to respect the intrinsic geometry of the statistical manifold: updates are measured by their effect on the represented probability distribution through the Fisher information metric (Amari, 1998; Martens, 2020; Khan & Rue, 2023). However, FOND did not address learning through the same geometric principle. Its generative model parameters were learned by backpropagating through the inference trajectory, with the associated computational and memory costs.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

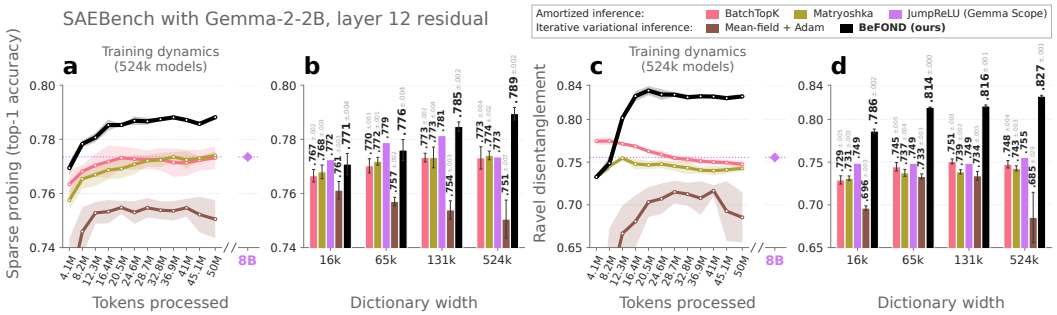


Figure 1: **Feature quality and sample efficiency on Gemma-2-2B activations.** **a,c**: Training dynamics at 524k width. **b,d**: Final performance across dictionary widths. Details in section 5.

Here, we extend this principle to learning. Just as inference follows natural-gradient flow on free energy with respect to variational parameters, we define learning as natural-gradient flow on the same objective with respect to generative parameters. Both processes thus respect the statistical geometry of the distributions they update. We instantiate this extended FOND framework as BeFOND: an encoder-free iterative SAE with Bernoulli latents, a factorized prior, and a linear Gaussian decoder. We derive closed-form expressions for its inference and learning dynamics, allowing the dictionary to be learned without backpropagating through the inference trajectory.

On the inference side, BeFOND implements *explaining away* through recurrent interactions between overlapping features, enabling lateral competition leading to sparse, disentangled representations. We isolate the contribution of these interactions and show that they become increasingly important as feature superposition increases. On the learning side, we prove that Fisher preconditioning makes local dictionary-learning rates independent of feature frequency. Consistent with this prediction, BeFOND recovers more accurate dictionaries and learns to detect rare features with few misses or false positives, while requiring orders-of-magnitude fewer samples relative to state-of-the-art SAEs.

We evaluate BeFOND on synthetic data with known ground-truth features using `SynthSAEBench` (Chanin & Garriga-Alonso, 2026), and on language-model representations using `SAEBench` (Karvonen et al., 2025). Baselines include amortized SAEs—BatchTopK (Bussmann et al., 2024), Matryoshka BatchTopK (Bussmann et al., 2025), and JumpReLU (Rajamanoharan et al., 2024b)—and an iterative mean-field variational baseline with Adam decoder learning (Kingma & Ba, 2015). On Gemma-2-2B layer-12 activations, BeFOND outperforms all evaluated baselines on both sparse probing (Kantamneni et al., 2025) and RAVEL disentanglement (Huang et al., 2024). BeFOND's performance continues to improve as we increase dictionary width from 16k to 524k latents, whereas the baselines plateau (Fig. 1b,d). These gains also emerge with substantially less training data: at a dictionary width of 524k, BeFOND matches the evaluated Gemma Scope JumpReLU checkpoints (Lieberum et al., 2024) in sparse probing after just ∼6M training tokens, compared with the 8B tokens used to train those checkpoints, over ∼1,000× reduction in training tokens (Fig. 1a).

Summary of our contributions and paper outline: **(1)** We extend FOND (Vafaii et al., 2025) to unify inference and learning as natural-gradient flows on variational free energy, instantiate it as BeFOND, and derive the resulting dynamics in closed-form (section 3 and section B).

(2) We analyze BeFOND's recurrent "explaining away" (section 4.1) and use ablations to establish its contribution to feature recovery under increasing superposition (section C.3.1).

(3) We characterize how Fisher preconditioning in BeFOND's learning rule compensates for feature rarity and test its consequences for dictionary learning through synthetic experiments and optimizer ablations (sections C.3.2, C.4 and 4.2).

(4) We evaluate BeFOND on language-model representations, examining feature quality, sample efficiency, and scaling with dictionary width (section 5).

2 Background and Related Work

Notation. Bold lowercase and uppercase symbols denote vectors and matrices. Inference and learning quantities appear in red and blue, respectively.

Sparse coding. Classical sparse coding (Olshausen & Field, 1996; 1997; Lewicki & Sejnowski, 2000; Aharon et al., 2006; Rozell et al., 2008; Gregor & LeCun, 2010; Mairal et al., 2010) models each input $\boldsymbol{x} \in \mathbb{R}^M$ as a linear combination of dictionary atoms with sparse coefficients $\boldsymbol{z}$:

$$\boldsymbol{x} \;=\; \boldsymbol{\Phi}\boldsymbol{z} + \boldsymbol{\epsilon}\,, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_M)\,, \tag{1}$$

The *dictionary* $\boldsymbol{\Phi} \in \mathbb{R}^{M\times K}$ has feature directions $\boldsymbol{\Phi}_{\cdot i} \equiv \boldsymbol{\phi}^{(i)}$ as its columns. We consider overcomplete dictionaries ($K > M$) with unit isotropic Gaussian noise. Learning estimates the dictionary from data, while inference finds sparse codes that explain each input. These processes are coupled because inference uses the current dictionary, and dictionary updates depend on the inferred codes.

Amortized versus iterative inference. Standard SAEs learn an encoder $\boldsymbol{z} = f_{\boldsymbol{\phi}}(\boldsymbol{x})$ that predicts a sparse code in one forward pass. These include ReLU (Bricken et al., 2023; Cunningham et al., 2024; Templeton et al., 2024), JumpReLU (Rajamanoharan et al., 2024b), gated (Rajamanoharan et al., 2024a), TopK (Makhzani & Frey, 2014; Gao et al., 2025), BatchTopK (Bussmann et al., 2024), and Matryoshka BatchTopK (Bussmann et al., 2025) SAEs. Training this mapping across samples *amortizes* inference, replacing a separate optimization for each input.

Iterative sparse coding instead refines the code for each observation (Rozell et al., 2008; Beck & Teboulle, 2009; Fang et al., 2022). In continuous time, this takes the schematic form

$$\dot{\boldsymbol{z}} \;=\; g(\boldsymbol{z}, \boldsymbol{\Phi}, \boldsymbol{x})\,. \tag{2}$$

Iterative inference has yielded mixed results on language-model representations (Chanin, 2025; Sun et al., 2026); see Appendix A.2 for further discussion.

Variational inference and free energy. A generative model combines a prior with a likelihood:

$$\underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}_{\text{joint distribution}} \;=\; \underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})}_{\text{conditional likelihood}}\; \underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{z})}_{\text{prior}}\,, \qquad p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) \;=\; \underbrace{\mathcal{N}(\boldsymbol{x}; \boldsymbol{\Phi}\boldsymbol{z}, \mathbf{I}_M)}_{\text{probabilistic form of eq. (1)}}\,. \tag{3}$$

Bayes' rule gives $p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})/p_{\boldsymbol{\theta}}(\boldsymbol{x})$. Computing the evidence $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ requires summing or integrating over latent configurations, which is generally intractable for large overcomplete models. *Variational inference* (Jordan et al., 1999; Blei et al., 2017) turns this problem into optimization: choose an approximate posterior $q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x})$ and optimize its parameters $\boldsymbol{u}$, such as Gaussian means and variances or Bernoulli logits (section A.3). Define the *variational free energy*:

$$\mathcal{F}(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{\theta}) \;\coloneqq\; \mathbb{E}_{q_{\boldsymbol{u}}}\!\left[-\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x})}\right] \;=\; -\mathbb{E}_{q_{\boldsymbol{u}}}\!\Big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\Big] + \mathcal{D}_{\mathrm{KL}}\Big(q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x}) \,\Big\|\, p_{\boldsymbol{\theta}}(\boldsymbol{z})\Big)\,. \tag{4}$$

Free energy is the negative evidence lower bound, $\mathcal{F} = -\mathrm{ELBO}$. Its terms balance explaining the observation against departing from the prior. The identity

$$\mathcal{F}(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{\theta}) \;=\; \mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x}) \,\|\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})) \;-\; \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \tag{5}$$

shows that, with $\boldsymbol{\theta}$ fixed, reducing $\mathcal{F}$ reduces the KL divergence to the true posterior because the evidence is independent of $\boldsymbol{u}$ (section A.3). An amortized encoder can predict $\boldsymbol{u}$, as in VAEs (Kingma & Welling, 2014), or inference can optimize it separately for each input:

$$\boldsymbol{u}^*(\boldsymbol{x}) \;=\; \operatorname*{argmin}_{\boldsymbol{u}} \mathcal{F}(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{\theta})\,. \tag{6}$$

Holding $\boldsymbol{x}$ and $\boldsymbol{\theta}$ fixed, write the objective as $\mathcal{F}(\boldsymbol{u})$. Starting from $\boldsymbol{u}_0$, gradient descent with step size $\eta > 0$ gives:

$$\boldsymbol{u}_{t+1} \;=\; \boldsymbol{u}_t \;-\; \eta\, \nabla_{\boldsymbol{u}} \mathcal{F}(\boldsymbol{u}_t)\,. \tag{7}$$

This update measures movement using Euclidean distance in parameter space (section A.4). Equal-sized parameter changes can produce different changes in beliefs, and reparameterizing the same distributions can change the descent path. Measuring movement between distributions instead leads to the geometry of their *statistical manifold* (Amari & Nagaoka, 2000).

From gradient descent to natural gradients. Using the Fisher metric gives the *natural-gradient* update:

$$\boxed{\boldsymbol{u}_{t+1} \;=\; \boldsymbol{u}_t \;-\; \eta\, \mathbf{G}_u^{-1}(\boldsymbol{u}_t)\nabla_{\boldsymbol{u}}\mathcal{F}(\boldsymbol{u}_t)\,.} \tag{8}$$

where $\mathbf{G}_u$ is the Fisher information matrix of the variational family:

$$\mathbf{G}_u(\boldsymbol{u}) \;=\; \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{u}}(\cdot\mid\boldsymbol{x})}\Big[\big(\nabla_{\boldsymbol{u}}\log q_{\boldsymbol{u}}(\boldsymbol{z}\mid\boldsymbol{x})\big)\big(\nabla_{\boldsymbol{u}}\log q_{\boldsymbol{u}}(\boldsymbol{z}\mid\boldsymbol{x})\big)^{\top}\Big]\,. \tag{9}$$

The Fisher matrix measures local statistical distinguishability, so the update accounts for how parameter changes affect the represented distribution (Amari, 1998; Amari & Nagaoka, 2000; Amari, 2016). Natural gradients have a long history in variational inference, underpinning stochastic variational inference (Hoffman et al., 2013), the Bayesian learning rule (Khan & Rue, 2023), and Bayesian Online Natural Gradient (Jones et al., 2024). Closed-form natural gradients need not be available, and computing the Fisher and applying its inverse can be expensive, motivating approximations (Martens & Grosse, 2015; Martens, 2020). Natural Reweighted Wake–Sleep uses natural gradients for generative and recognition networks (Várady et al., 2022). Recent TopK SAE work preconditions decoder updates with a damped empirical code Gram matrix (Picozzi, 2026). Both retain learned encoders, whereas recurrent variational inference optimizes posterior parameters for each input. Vafaii et al. (2025) introduced FOND by prescribing natural-gradient flow of $\mathcal{F}$ with respect to $\boldsymbol{u}$. A forward Euler discretization yields eq. (8).

3 FOND UNIFIES INFERENCE AND LEARNING AS NATURAL GRADIENT FLOW

FOND (Vafaii et al., 2025) formulated inference as natural gradient flow on variational free energy (eq. (8)). But it stopped short of prescribing learning dynamics through the same principle. The geometric argument applies equally to learning: just as $\boldsymbol{u}$ changes the approximate posterior $q_{\boldsymbol{u}}$, the generative parameters $\boldsymbol{\theta}$ change the joint distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})$. This motivates extending FOND by defining learning as natural-gradient flow on the same free energy with respect to $\boldsymbol{\theta}$. Thus, inference and learning are both treated as dynamical processes within a unified framework:

FOND: Free-energy Online Natural-gradient Dynamics

Flexible choices: Choose a generative model $p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})$, an inference model $q_{\boldsymbol{u}}(\boldsymbol{z}\mid\boldsymbol{x})$, and calculate the variational free energy $\mathcal{F}(\boldsymbol{u},\boldsymbol{\theta})$.

FOND's prescription: Define inference and learning as natural gradient flows on $\mathcal{F}$:

Inference: $\dot{\boldsymbol{u}} \coloneqq -\mathbf{G}_u^{-1}(\boldsymbol{u})\,\dfrac{\partial\mathcal{F}(\boldsymbol{u},\boldsymbol{\theta})}{\partial\boldsymbol{u}}$, **Learning:** $\dot{\boldsymbol{\theta}} \coloneqq -\mathbf{G}_{\theta}^{-1}(\boldsymbol{\theta})\,\dfrac{\partial\mathcal{F}(\boldsymbol{u},\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$.

3.1 BeFOND: A BERNOULLI-DISTRIBUTED ITERATIVE SPARSE CODING MODEL

We instantiate our extended FOND by following its three steps: choose the distributions, compute the free energy, and derive the dynamics.

Distribution Specifications. A generative model requires a prior and a conditional likelihood (eq. (3)). We choose a factorized Bernoulli prior and a linear Gaussian decoder, together with a factorized Bernoulli variational posterior:

Likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid\boldsymbol{z})$	*Prior* $p_{\boldsymbol{\theta}}(\boldsymbol{z})$	*Posterior* $q_{\boldsymbol{u}}(\boldsymbol{z}\mid\boldsymbol{x})$
$\mathcal{N}(\boldsymbol{x};\boldsymbol{\Phi}\boldsymbol{z},\boldsymbol{\Sigma})$ (10) $\boldsymbol{\Sigma}$ diagonal and positive	$\prod_{i=1}^{K}\text{Bernoulli}(z_i;\pi_{0,i})$ (11) $\boldsymbol{\pi}_0=\sigma(\boldsymbol{u}_0)$	$\prod_{i=1}^{K}\text{Bernoulli}(z_i;r_i)$ (12) $\boldsymbol{r}=\sigma(\boldsymbol{u})$

Here $\boldsymbol{z} \in \{0,1\}^K$, and $\sigma(\cdot)$ is the logistic sigmoid. The generative parameters are $\boldsymbol{\theta} \equiv (\boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{u}_0)$. Binary Sparse Coding already combines binary latents with a linear Gaussian decoder, learning its parameters through truncated variational EM (Henniges et al., 2010). Here we use a factorized posterior with natural-gradient inference and learning.

Free energy. Define $\boldsymbol{e} \coloneqq \boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{r}$ as the mean reconstruction residual, $\mathbf{W} \coloneqq \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi}$ as the precision-weighted Gram matrix, and $\mathbf{C} \coloneqq \mathrm{diag}[\boldsymbol{r} \odot (\mathbf{1} - \boldsymbol{r})]$ as the posterior covariance. We weight the prior KL by $\beta > 0$, with $\beta = 1$ recovering the usual variational free energy. The expectations can be evaluated exactly (section B.1):

$$\mathcal{F}(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{e}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{e} + \frac{1}{2}\mathrm{Tr}(\mathbf{W}\mathbf{C}) + \frac{1}{2}\log\det\boldsymbol{\Sigma} + \beta \sum_{i=1}^{K}\left[r_i \log\frac{r_i}{\pi_{0,i}} + (1-r_i)\log\frac{1-r_i}{1-\pi_{0,i}}\right]. \tag{13}$$

Dynamics. Applying FOND's natural-gradient prescription, using the Fisher metrics of the variational posterior and the generative joint distribution, yields closed-form inference and learning dynamics (section B.2.1):

$$\textit{Inference} \quad \dot{\boldsymbol{u}} = \underbrace{\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x}}_{\text{feedforward drive}} - \underbrace{\mathbf{W}\boldsymbol{r}}_{\substack{\text{recurrent}\\ \text{interactions}}} - \underbrace{\tfrac{1}{2}\,\mathrm{diag}(\mathbf{W}) \odot (\mathbf{1} - 2\boldsymbol{r})}_{\text{variance penalty}} - \underbrace{\beta(\boldsymbol{u} - \boldsymbol{u}_0)}_{\text{homeostatic leak}} \tag{14}$$

$$\textit{Learning} \quad \dot{\boldsymbol{\Phi}} = \eta_\Phi\big(\boldsymbol{x}\boldsymbol{r}^\top - \boldsymbol{\Phi}\mathbf{E}\big)\mathbf{E}_p^{-1} \tag{15} \qquad \dot{\boldsymbol{\Sigma}} = \eta_\Sigma(\mathbf{V} - \boldsymbol{\Sigma}) \tag{16} \qquad \dot{\boldsymbol{\pi}}_0 = \eta_0\beta(\boldsymbol{r} - \boldsymbol{\pi}_0) \tag{17}$$

Here $\eta_\Phi, \eta_\Sigma, \eta_0 > 0$ set the learning timescales, and prior learning is expressed in mean coordinates $\boldsymbol{\pi}_0 = \sigma(\boldsymbol{u}_0)$. The posterior and prior second moments are

$$\mathbf{E} = \mathbf{C} + \boldsymbol{r}\boldsymbol{r}^\top, \qquad \mathbf{E}_p = \mathrm{diag}[\boldsymbol{\pi}_0 \odot (\mathbf{1} - \boldsymbol{\pi}_0)] + \boldsymbol{\pi}_0\boldsymbol{\pi}_0^\top, \tag{18}$$

and $\mathbf{V}$ is diagonal, with $V_{mm} = e_m^2 + (\boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^\top)_{mm}$, the expected squared reconstruction error in each observed dimension. Learning holds the inferred posterior fixed. For minibatches, we average $\boldsymbol{x}\boldsymbol{r}^\top$, $\mathbf{E}$, $\mathbf{V}$, and $\boldsymbol{r}$ before applying the learning rules. These equations keep the prior fixed during inference. We use a rolling-reference variant in our implementation, described in section B.1.1 and Algorithm 1.

4 Theoretical Analyses, Synthetic Experiments, and Ablations

We study inference and learning using theory and synthetic data with known features and activations. Ablations test lateral interactions and compare separately tuned natural and Euclidean recipes.

4.1 BeFOND's Recurrent Term Enables Explaining Away

Superposition mixes evidence from overlapping features (Elhage et al., 2022; Klindt et al., 2026), making inference of sparse codes difficult (section A.1). Here, we show how BeFOND's iterative inference mitigates this interference through a phenomenon known as *explaining away*. As an example, wet grass could be explained by rain or a sprinkler. Observing a wet roof supports rain without changing the sprinkler's direct evidence from the grass. As rain explains more of the wetness, the sprinkler becomes less likely. This reduction in belief in one explanation through evidence for another is known as *explaining away* (Murphy, 2012). The challenge is that the influence of one piece of evidence depends on the others. An encoder that compresses the input to one fixed projection cannot generally capture the full dependence, as the following proposition establishes.

Proposition 4.1 (A limitation of single-layer inference)**.** *Suppose the data are generated by the Bernoulli–Gaussian model:*

$$\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{z} + \boldsymbol{\epsilon}, \qquad z_j \overset{\text{ind}}{\sim} \mathrm{Bernoulli}(\pi_j), \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I),$$

where $0 < \pi_j < 1$ for every j, and the noise is independent of $\boldsymbol{z}$.

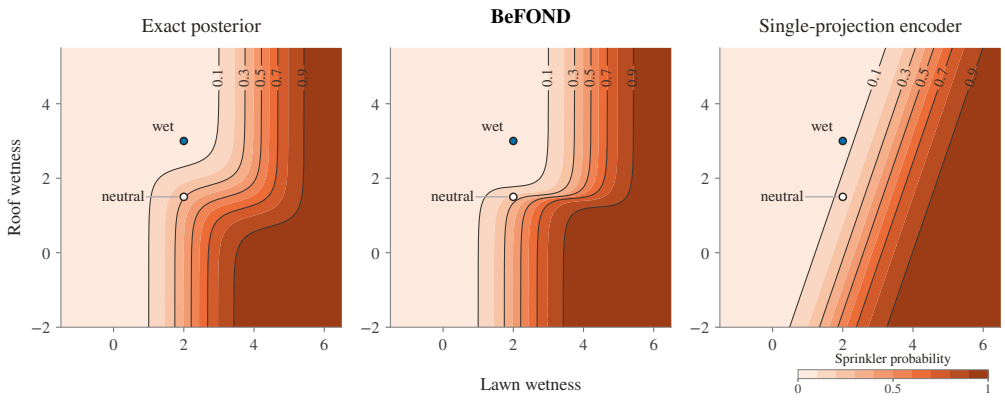


Figure 2: Heatmaps show sprinkler probabilities under exact inference, converged BeFOND with a fixed prior, and a numerically optimized single-projection encoder. The marked observations have identical lawn wetness but different roof evidence: neutral (white, halfway dry / wet) or wet (blue). Fixed-prior inference and encoder fitting are detailed in Appendix C.2.3.

Suppose a dictionary atom $\phi^{(i)}$ is neither orthogonal nor parallel to another atom $\phi^{(j)}$. Then there are no weights w_i, bias b_i, and scalar function f_i such that

$$f_i(w_i^\top x + b_i) = p(z_i = 1 \mid x) = \mathbb{E}[z_i \mid x] \qquad \textit{for every } x \in \mathbb{R}^d.$$

For almost every dictionary with $K, M \geq 2$, no latent falls under the exception, so for generic dictionaries a feature's exact posterior probability cannot be computed from one fixed input projection followed by a scalar nonlinearity. The proof shows that overlapping, nonparallel atoms make the posterior log-odds gradient change direction with the input, which is impossible for a function of one fixed projection (Appendix C.2.1). This result generalizes O'Neill et al. (2025) and covers ReLU (Bricken et al., 2023; Cunningham et al., 2024), JumpReLU (including Gemma Scope) (Rajamanoharan et al., 2024b; Lieberum et al., 2024), weight-sharing gated (Rajamanoharan et al., 2024a), and fixed-threshold BatchTopK and Matryoshka BatchTopK SAEs (Bussmann et al., 2024; 2025), all of which compute each latent from a single projection of the input (Appendix C.2.1). TopK SAEs fall outside this class, since whether a latent fires depends on every other pre-activation.

BeFOND's inference dynamics (eq. (14)) couple explanations through the off-diagonal entries of $-(\Phi^\top \Phi)r$: positively overlapping features reduce one another's activation drive. A wet roof strengthens rain, which explains the lawn's wetness and suppresses the sprinkler. Repeated updates make this competition depend on the full observation (Fig. 2).

Appendix C.2.2 gives an approximate-inference counterpart to Proposition 4.1 and sufficient conditions under which BeFOND can achieve strictly lower expected posterior KL than the best possible single-projection amortized encoder. We now measure the effects of explaining away on a full-scale experiment.

Explaining away under superposition. `SynthSAEBench` (Chanin & Garriga-Alonso, 2026) makes this problem measurable by providing known feature directions and activations, together with feature correlations and hierarchy. Superposition is measured as the mean maximum (mm) absolute cosine similarity:

$$\rho_{mm} = \frac{1}{K_\star} \sum_{i=1}^{K_\star} \max_{j \neq i} \frac{\left| (\phi_\star^{(i)})^\top \phi_\star^{(j)} \right|}{\|\phi_\star^{(i)}\| \, \|\phi_\star^{(j)}\|},$$

where $\phi_\star^{(i)}$ are the $K_\star$ ground-truth feature directions. Larger ρ_{mm} means that each feature has, on average, a more strongly overlapping alternative. Reducing the input dimension at fixed feature count increases this overlap, modeling the interference created when many features share a real network's activation space.

As overlap increases, a strong detection score becomes less diagnostic of a feature's presence: another feature may explain the same evidence. Explaining away addresses this ambiguity by discounting evidence already accounted for by competing causes.

We measure dictionary recovery using the *mean correlation coefficient* (MCC), which quantifies alignment between learned and ground-truth feature directions. BeFOND achieves the highest MCC throughout the sweep, declining only modestly as superposition increases, while maintaining zero dead latents (Fig. 3).

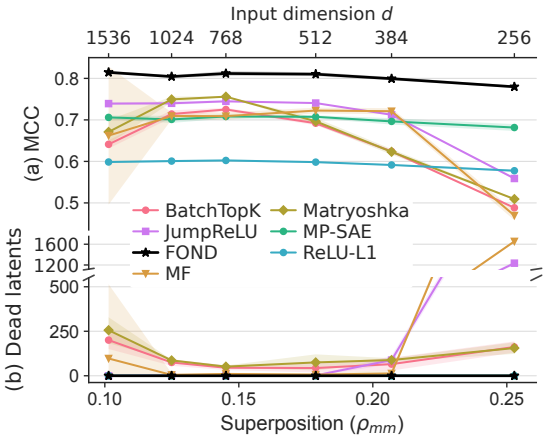


Figure 3: BeFOND maintains the highest recovery, and without dead latents ($K = 4{,}096$). Means and 95% Student-t intervals over 3 seeds. Exact values: Appendix C.3.1, Table 3.

Ablating lateral interactions. Removing lateral interactions reduces dictionary recovery as feature overlap increases. Restoring suppressive interactions substantially improves recovery, whereas excitatory-only inference activates nearly the entire dictionary without comparable recovery (Fig. 6; Appendix C.3.1).

4.2 BeFOND's natural gradient flow improves rare feature learning

Features extracted from language models vary widely in activation frequency (Gao et al., 2025, Appendix E.4), and larger SAE dictionaries recover increasingly rare concepts (Bricken et al., 2023; Templeton et al., 2024). Scaling dictionary width therefore places greater demands on learning from rare examples: additional neurons must capture infrequent features to expand the model's coverage.

Learning an infrequently active feature. For an isolated feature active with probability ρ, averaging over inputs scales the decoder gradient by ρ. The same decoder error therefore produces a smaller update for a rarer feature. Matched Fisher preconditioning divides out this frequency factor.

Theorem 4.2 (Exact rarity compensation)**.** *For $p(\boldsymbol{x} \mid \boldsymbol{z}) = \mathcal{N}(\boldsymbol{\Phi}\boldsymbol{z}, \mathbf{I})$, fix a training minibatch and hold its inferred distributions and prior references fixed. Let $\boldsymbol{\Phi}^*$ minimize $\mathcal{F}$. If the averaged second moments satisfy $\mathbf{E}_p = \mathbf{E} \succ 0$, then*

$$\textit{Euclidean:} \quad \frac{d\boldsymbol{\Phi}}{d\tau} = -\eta_\Phi(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*)\mathbf{E},$$

$$\textit{Natural:} \quad \frac{d\boldsymbol{\Phi}}{d\tau} = -\eta_\Phi \mathbf{G}_\Phi^{-1}\nabla_\Phi \mathcal{F} = -\eta_\Phi(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*),$$

where $\mathbf{G}_\Phi = \mathbf{E}_p \otimes \mathbf{I}$ and $\eta_\Phi > 0$. Natural flow reduces every decoder error by $e^{-\eta_\Phi \Delta t}$ over an interval $\Delta t \geq 0$. For the same fractional error reduction, Euclidean flow takes $\kappa(\mathbf{E}) \geq \rho_{\max}/\rho_{\min}$ times as long in its slowest eigendirection as in its fastest, where $\rho_i = \mathbf{E}_{ii}$ is the minibatch-averaged posterior activation probability.

The rate difference becomes more pronounced as the frequency range widens. If $\rho_i = c\, i^{-\alpha}$, with $\alpha > 0$ and $i = 1, \ldots, K$, the slowest Euclidean direction takes at least K^α times as long as the fastest to achieve the same fractional error reduction. Under the theorem's matched-moment assumptions, natural flow gives every direction the same rate. For the proof and the extension to unequal prior and posterior second moments, see Appendix C.4. Appendix C.4.2 shows that, with known codes and frozen Adam normalization, an optimer used by many common SAE baselines (Bussmann et al., 2024; Rajamanoharan et al., 2024b), Zipfian feature frequencies leave a local condition number of at least $K^{\alpha/2}$, whereas matched Fisher preconditioning reduces it to one.

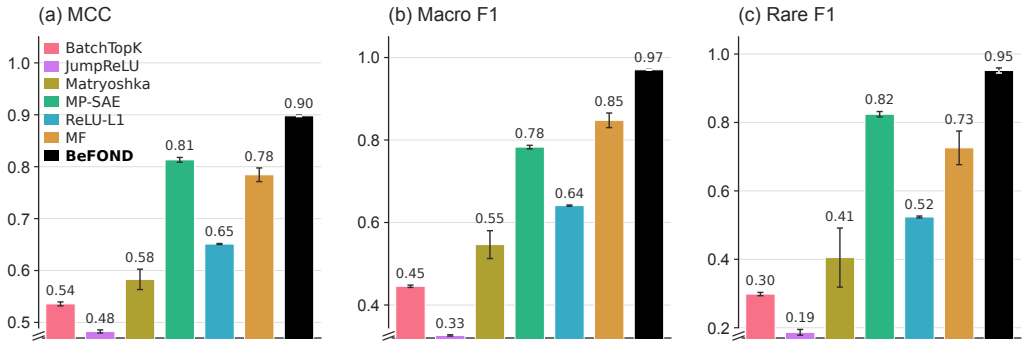


Figure 4: BeFOND outperforms all evaluated baselines on MCC, Macro F1, and Rare F1 in `SynthSAEBench` ($d = 768$, $K = K_\star = 16{,}384$). Rare F1 covers the least frequent quarter of features. Bars show means over 3 initialization seeds with 95% Student-t confidence intervals, evaluated on 10^6 held-out samples. Table 4 gives individual-seed results.

Recovering rare features. The *full-width* SynthSAEBench setting (Chanin & Garriga-Alonso, 2026) tests whether decoder capacity recovers the rare tail, with 16,384 learned latents for 16,384 generating features. This supplies one decoder slot per true feature, so complete recovery requires representing rare features alongside common ones. MCC measures recovery of the feature directions, while Rare F1 measures detection of the 25% least frequent generating features, averaging their activation F1 scores. BeFOND leads all metrics, exceeding the evaluated baselines (Fig. 4; Table 4).

Ablating inference and learning geometry. The natural-inference/natural-learning recipe reaches stronger recovery with fewer decoder updates than the separately tuned Euclidean variants (Fig. 7; Appendix C.3.2). Appendix C.5.2 reports learning curves against the other baselines, and Table 4 reports dead-latent counts alongside recovery.

5 Evaluating BeFOND on language-model activations

We evaluate whether BeFOND's inference and learning benefits studied in section 4 translate to improved feature quality in language-model representations. We focus on three questions: (1) whether BeFOND outperforms widely used SAE baselines on Gemma 2B scale activations, (2) whether these benefits increase with dictionary width, and (3) how early they emerge during training.

Experimental setup. We evaluate feature quality under a 50M-token training budget, alongside released Gemma Scope JumpReLU checkpoints trained on 4B tokens at 16k width and 8B tokens at widths 32k–524k (Lieberum et al., 2024; Rajamanoharan et al., 2024b). Our primary comparison trains BeFOND, BatchTopK (Bussmann et al., 2024), Matryoshka BatchTopK (Bussmann et al., 2025), and a mean-field variational baseline with Adam decoder learning (Kingma & Ba, 2015). These models use whitened Gemma-2-2B (Gemma Team, 2024) layer-12 residual-stream activations ($d = 2{,}304$) extracted from Pile Uncopyrighted (monology, 2023), with dictionary widths from 16,384 to 524,288 latents. We report means and sample standard deviations over three BeFOND training seeds and five baseline training seeds, using $T = 10$ inference steps for BeFOND and $T = 20$ for mean-field at evaluation. Amortized baselines use the selected target sparsities in Table 5. Gemma Scope checkpoints retain their original preprocessing. Appendix D provides further details.

Evaluation metrics. Reconstruction quality alone does not establish whether an SAE has learned useful features (Gao et al., 2025; Karvonen et al., 2025). We therefore evaluate two complementary capabilities using `SAEBench` (Karvonen et al., 2025). *Sparse probing* tests concept detection using a logistic probe restricted to $k = 1$ or $k = 5$ selected SAE features (Kantamneni et al., 2025). Here, k counts input features, not predicted class labels. We average task-level accuracy, ROC-AUC, and F1 equally across 113 classification tasks. Accuracy measures correct predictions, ROC-AUC measures the ranking of positive versus negative examples, and F1 summarizes the balance between precision

and recall. *RAVEL* tests selective intervention on represented attributes (Huang et al., 2024). Its cause score measures successful changes to a target attribute, while isolation measures preservation of other attributes; disentanglement is the mean of these two scores. Higher scores are better for all reported metrics.

Full numerical results are reported in Table 5 in Appendix D.

Scaling with dictionary width. BeFOND achieves the highest mean RAVEL disentanglement at every tested width, including comparisons with the evaluated Gemma Scope checkpoints (Fig. 1d). At 524k latents, its score is 0.827 ± 0.001, compared with 0.748 ± 0.004 for BatchTopK, the strongest amortized baseline trained on 50M tokens (Table 5). This 7.9 percentage-point advantage reflects improvements in both cause and isolation: BeFOND scores 0.800 and 0.854, respectively, versus 0.719 and 0.776 for BatchTopK.

Single-feature probing also favors BeFOND at large widths. At 524k, its accuracy is 0.789 ± 0.002, versus 0.774 ± 0.002 for Matryoshka BatchTopK, an improvement of 1.5 percentage points. Across widths from 16k to 524k, BeFOND's single-feature accuracy increases from 0.771 to 0.789, and its RAVEL score increases from 0.786 to 0.827 (Fig. 1b,d). Over the same range, BatchTopK and Matryoshka each improve by approximately 0.6 percentage points in single-feature accuracy. Thus, increasing dictionary capacity yields larger gains for BeFOND under the evaluated training recipes.

The advantage is metric-dependent. Five-feature probing accuracy differs less between methods, and BeFOND does not lead every AUC or F1 comparison (Table 8).

High feature quality emerges early in training. At 524k latents, BeFOND reaches the evaluated Gemma Scope single-feature probing score by 8.2M training tokens (Fig. 1a). The corresponding Gemma Scope checkpoints were trained on 8B tokens, a training-budget ratio of approximately 976. By 50M tokens, BeFOND exceeds their single-feature accuracy (0.789 versus 0.7735) and RAVEL disentanglement (0.827 versus 0.7555), using 160 times fewer training tokens. These comparisons establish that the reported scores are attainable with substantially smaller token budgets.

Iterative inference alone does not reproduce the gains. The mean-field baseline performs iterative variational inference and learns its decoder using Adam. At 524k, it reaches single-feature accuracy of 0.751 ± 0.007 and RAVEL disentanglement of 0.685 ± 0.029, below BeFOND on both metrics (Table 5). The evaluated iterative Adam recipe therefore does not reproduce BeFOND's benefits. This comparison complements the synthetic analyses in section 4; it does not isolate learning geometry from other differences between the Gemma training recipes.

6 Conclusion and Discussion

We formulate inference and dictionary learning as natural-gradient flows on a shared variational free energy (the FOND framework). Our theory establishes limitations of single-projection encoders and shows how Fisher preconditioning compensates for infrequent activation under frozen, matched moments. We introduce BeFOND, which improves synthetic feature recovery and supports increasing feature quality at larger dictionary widths on Gemma.

Runtime remains a limitation: repeated inference steps and numerical decoder updates add computational cost (Appendices B.8 and D.3). Future work will focus on improving computational efficiency and extending the factorized variational family with pairwise and higher-order interactions to represent dependencies between latent features.

Our results on layer 12 of Gemma-2-2B suggest that BeFOND improves concept detection and enables more selective interventions. The gains grow with dictionary width while competing methods plateau, suggesting that BeFOND makes better use of additional capacity. Together with the synthetic experiments, these findings position BeFOND as a competitive, scalable alternative to standard SAEs, grounded in a principled probabilistic framework.

AI USE STATEMENT

We used generative AI tools to assist with software implementation and debugging, experimental analysis and visualization, literature searches, mathematical derivations, and manuscript drafting and editing. The authors are responsible for the final text, results, and claims.

REFERENCES

M. Aharon, M. Elad, and A. Bruckstein. rmK-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. doi: 10.1109/tsp.2006.881199. URL https://doi.org/10.1109/tsp.2006.881199.

Awad H. Al-Mohy and Nicholas J. Higham. Computing the Action of the Matrix Exponential, with an Application to Exponential Integrators. *SIAM Journal on Scientific Computing*, 33(2):488–511, 2011. doi: 10.1137/100788860. URL https://doi.org/10.1137/100788860.

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A. Saurous, and Kevin Murphy. Fixing a Broken ELBO. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 159–168. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/alemi18a.html.

Shun-ichi Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276, 1998. doi: 10.1162/089976698300017746. URL https://doi.org/10.1162/089976698300017746.

Shun-ichi Amari. *Information Geometry and Its Applications*. Springer Japan, 2016. ISBN 9784431559771. doi: 10.1007/978-4-431-55978-8. URL https://doi.org/10.1007/978-4-431-55978-8.

Shun-ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical Monographs*. American Mathematical Society, 2000. ISBN 9780821805312. doi: 10.1090/mmono/191. URL https://doi.org/10.1090/mmono/191.

Lukas Balles and Philipp Hennig. Dissecting Adam: The sign, magnitude and variance of stochastic gradients. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 404–413. PMLR, 2018. URL https://proceedings.mlr.press/v80/balles18a.html.

Vitória Barin-Pacela, Shruti Joshi, Isabela Camacho, Simon Lacoste-Julien, and David Klindt. Stop Probing, Start Coding: Why Linear Probes and Sparse Autoencoders Fail at Compositional Generalisation, 2026. URL https://arxiv.org/abs/2603.28744.

Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. doi: 10.1137/080716542. URL https://doi.org/10.1137/080716542.

P. G. Bissiri, C. C. Holmes, and S. G. Walker. A General Framework for Updating Belief Distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):1103–1130, 2016. doi: 10.1111/rssb.12158. URL https://doi.org/10.1111/rssb.12158.

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. doi: 10.1080/01621459.2017.1285773. URL https://doi.org/10.1080/01621459.2017.1285773.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. URL https://web.stanford.edu/~boyd/cvxbook/.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen,

Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/.

Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK Sparse Autoencoders, 2024. URL https://arxiv.org/abs/2412.06410.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning Multi-Level Features with Matryoshka Sparse Autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 6077–6101. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/bussmann25a.html.

David Chanin. Training Matching Pursuit SAEs on LLMs. LessWrong, December 2025. URL https://www.lesswrong.com/posts/rE43EfHigXcjTmomJ/training-matching-pursuit-saes-on-llms.

David Chanin and Adrià Garriga-Alonso. SynthSAEBench: Evaluating Sparse Autoencoders on Scalable Realistic Synthetic Data, 2026. URL https://arxiv.org/abs/2602.14687.

chanind-goodfire. SynthSAEBench Full-Width SAE Sweep. Hugging Face model and training-recipe release, 2026. URL https://huggingface.co/chanind-goodfire/synthsaebench-full-width-sweep.

Valérie Costa, Thomas Fel, Ekdeep S Lubana, Bahareh Tolooshams, and Demba Ba. From Flat to Hierarchical: Extracting Sparse Representations with Matching Pursuit. In *Advances in Neural Information Processing Systems 38*, pp. 38625–38668. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi: 10.52202/085713-1154. URL https://doi.org/10.52202/085713-1154.

S.M. Cox and P.C. Matthews. Exponential Time Differencing for Stiff Systems. *Journal of Computational Physics*, 176(2):430–455, 2002. doi: 10.1006/jcph.2002.6995. URL https://doi.org/10.1006/jcph.2002.6995.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Paper-Conference.pdf.

I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57 (11):1413–1457, 2004. doi: 10.1002/cpa.20042. URL https://doi.org/10.1002/cpa.20042.

James G. Dowty. Chentsov's theorem for exponential families. *Information Geometry*, 1: 117–135, 2018. doi: 10.1007/s41884-018-0006-4. URL https://doi.org/10.1007/s41884-018-0006-4.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Michael Y.-S. Fang, Mayur Mudigonda, Ryan Zarcone, Amir Khosrowshahi, and Bruno A. Olshausen. Learning and Inference in Sparse Coding Models With Langevin Dynamics. *Neural Computation*, 34(8):1676–1700, 2022. doi: 10.1162/neco_a_01505. URL https://doi.org/10.1162/neco_a_01505.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, volume 2025, pp. 26721–26754, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/42ef3308c230942d223c411adf182c88-Paper-Conference.pdf.

Gemma Team. Gemma 2: Improving Open Language Models at a Practical Size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 399–406, 2010. URL https://icml.cc/Conferences/2010/papers/449.pdf.

William W. Hager. Updating the Inverse of a Matrix. *SIAM Review*, 31(2):221–239, 1989. URL https://people.clas.ufl.edu/hager/files/update-1.pdf.

Marc Henniges, Gervasio Puertas, Jörg Bornschein, Julian Eggert, and Jörg Lücke. Binary Sparse Coding. In *Latent Variable Analysis and Signal Separation*, pp. 450–457. Springer Berlin Heidelberg, 2010. ISBN 9783642159947. doi: 10.1007/978-3-642-15995-4_56. URL https://doi.org/10.1007/978-3-642-15995-4_56.

Arthur Hobson. A new theorem of information theory. *Journal of Statistical Physics*, 1(3):383–391, 1969. doi: 10.1007/bf01106578. URL https://doi.org/10.1007/bf01106578.

Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19: 209–286, 2010. doi: 10.1017/s0962492910000048. URL https://doi.org/10.1017/s0962492910000048.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic Variational Inference. *Journal of Machine Learning Research*, 14(40):1303–1347, 2013. URL http://jmlr.org/papers/v14/hoffman13a.html.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. RAVEL: Evaluating Interpretability Methods on Disentangling Language Model Representations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8669–8687, Bangkok, Thailand, aug 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.470. URL https://aclanthology.org/2024.acl-long.470/.

Adam Jermyn and Adly Templeton. Ghost Grads: An Improvement on Resampling. *Transformer Circuits Thread*, January 2024. URL https://transformer-circuits.pub/2024/jan-update/index.html#dict-learning-resampling. Circuits Updates, January 2024.

Matt Jones, Peter Chang, and Kevin Murphy. Bayesian Online Natural Gradient (BONG). In *Advances in Neural Information Processing Systems 37*, pp. 131104–131153. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-4167. URL https://doi.org/10.52202/079017-4167.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37(2):183–233, 1999. doi: 10.1023/a:1007665907178. URL https://doi.org/10.1023/a:1007665907178.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are Sparse Autoencoders Useful? A Case Study in Sparse Probing. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29018–29049. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kantamneni25a.html.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum Stuart Mcdougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A Comprehensive Benchmark for Sparse Autoencoders in Language Model Interpretability. In *Proceedings of the 42nd International*

Conference on Machine Learning, volume 267 of *Proceedings of Machine Learning Research*, pp. 29223–29264. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/karvonen25a.html.

Mohammad Emtiyaz Khan and Håvard Rue. The Bayesian Learning Rule. *Journal of Machine Learning Research*, 24(281):1–46, 2023. URL http://jmlr.org/papers/v24/22-0291.html.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6114.

David Klindt, Charles O'Neill, Patrik Reizinger, Harald Maurer, and Nina Miolane. A unifying framework from neural superposition to sparse interpretable codes. *Nature Machine Intelligence*, 8 (7):1025–1037, 2026. doi: 10.1038/s42256-026-01259-z. URL https://www.nature.com/articles/s42256-026-01259-z.

Michael S. Lewicki and Terrence J. Sejnowski. Learning Overcomplete Representations. *Neural Computation*, 12(2):337–365, 2000. doi: 10.1162/089976600300015826. URL https://doi.org/10.1162/089976600300015826.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 278–300. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.blackboxnlp-1.19. URL https://doi.org/10.18653/v1/2024.blackboxnlp-1.19.

Haihao Lu, Robert M. Freund, and Yurii Nesterov. Relatively Smooth Convex Optimization by First-Order Methods, and Applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018. doi: 10.1137/16m1099546. URL https://doi.org/10.1137/16m1099546.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online Learning for Matrix Factorization and Sparse Coding. *Journal of Machine Learning Research*, 11(2):19–60, 2010. URL http://jmlr.org/papers/v11/mairal10a.html.

Alireza Makhzani and Brendan Frey. k-Sparse Autoencoders. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.5663.

S.G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. doi: 10.1109/78.258082. URL https://doi.org/10.1109/78.258082.

James Martens. New Insights and Perspectives on the Natural Gradient Method. *Journal of Machine Learning Research*, 21(146):1–76, 2020. URL http://jmlr.org/papers/v21/17-678.html.

James Martens and Roger Grosse. Optimizing Neural Networks with Kronecker-factored Approximate Curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/martens15.html.

monology. Pile Uncopyrighted. Hugging Face dataset, 2023. URL https://huggingface.co/datasets/monology/pile-uncopyrighted.

Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012. URL https://probml.github.io/pml-book/book0.html.

Radford M. Neal and Geoffrey E. Hinton. A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants. In Michael I. Jordan (ed.), *Learning in Graphical Models*, pp. 355–368. Springer Netherlands, 1998. ISBN 9789401061049. doi: 10.1007/978-94-011-5014-9_12. URL https://doi.org/10.1007/978-94-011-5014-9_12.

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. URL https://doi.org/10.1038/381607a0.

Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997. doi: 10.1016/s0042-6989(97)00169-7. URL https://doi.org/10.1016/s0042-6989(97)00169-7.

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. URL https://doi.org/10.1016/j.conb.2004.07.007.

Charles O'Neill, Alim Gumran, and David Klindt. Compute Optimal Inference and Provable Amortisation Gap in Sparse Autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 46877–46896. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/o-neill25a.html.

Dario Picozzi. The information geometry of large language models is shared, learned, and controllable, 2026. URL https://arxiv.org/abs/2609.11063.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving Sparse Decomposition of Language Model Activations with Gated Sparse Autoencoders. In *Advances in Neural Information Processing Systems*, volume 37, pp. 775–818. Curran Associates, Inc., 2024a. doi: 10.52202/079017-0024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/01772a8b0420baec00c4d59fe2fbace6-Paper-Conference.pdf.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping Ahead: Improving Reconstruction Fidelity with JumpReLU Sparse Autoencoders, 2024b. URL https://arxiv.org/abs/2407.14435.

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. URL https://doi.org/10.1162/neco.2008.03-07-486.

Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2 edition, 2003. URL https://www-users.cse.umn.edu/~saad/IterMethBook_2ndEd.pdf.

Simo Särkkä and Lennart Svensson. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2 edition, 2023. doi: 10.1017/9781108917407. URL https://doi.org/10.1017/9781108917407.

Wenjie Sun, Di Wang, and Lijie Hu. The Price of Amortized inference in Sparse Autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 72686–72710, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/76358e88c3ebd38ba58af9d0d84b71e4-Paper-Conference.pdf.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/.

Lloyd N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Society for Industrial and Applied Mathematics, 2019. URL https://people.maths.ox.ac.uk/trefethen/ATAP/.

Hadi Vafaii, Dekel Galor, and Jacob Yates. Brain-like Variational Inference. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 163700–163778. Curran Associates, Inc., 2025. doi: 10.52202/085713-5465. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/efa0939caa8a906fd0a61a9c60e46e8b-Paper-Conference.pdf.

Csongor Várady, Riccardo Volpi, Luigi Malagò, and Nihat Ay. Natural Reweighted Wake–Sleep. *Neural Networks*, 155:574–591, 2022. doi: 10.1016/j.neunet.2022.09.006. URL https://doi.org/10.1016/j.neunet.2022.09.006.

Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008. doi: 10.1561/2200000001. URL https://doi.org/10.1561/2200000001.

Zeyu Yun, Yubei Chen, Bruno Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 1–10, Online, jun 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.deelio-1.1. URL https://aclanthology.org/2021.deelio-1.1/.

APPENDIX: INFERENCE AND LEARNING IN SPARSE AUTOENCODERS AS NATURAL GRADIENT FLOW

A Extended background **17**

A.1 Superposition and inference 17

A.2 Iterative sparse encoders. 17

A.3 Variational inference and the free energy objective 18

A.4 Trust regions and natural-gradient inference 19

B Extended FOND model and dynamics **21**

B.1 Model and objective 21

B.2 Natural-gradient dynamics 25

B.3 Decoder bias 26

B.4 Bernoulli uncertainty and sparse coding 27

B.5 Activation threshold under a rolling prior 27

B.6 Numerical integration and matrix-free updates 27

B.7 Algorithm 28

B.8 Computational complexity 29

C Extended theoretical analyses, synthetic experiments, and ablations **30**

C.1 Experimental setup 30

C.2 Inference proofs 31

C.3 Inference and learning ablations 35

C.4 Exact and approximate rarity compensation 38

C.5 Learning and rare-feature recovery 41

D Extended Gemma experiments **45**

D.1 Experimental setup 45

D.2 Results 46

D.3 Wall-clock training times on H200 GPUs 47

A Extended background

A.1 Superposition and inference

Inferring a sparse code can be difficult even with a perfectly learned dictionary. To isolate the inference problem, consider the noiseless generative process $\boldsymbol{x} = \boldsymbol{\Phi}^* \boldsymbol{z}$, where the ground truth dictionary $\boldsymbol{\Phi}^* \in \mathbb{R}^{M \times K}$ has unit-norm columns, and suppose we have learned the true dictionary $\boldsymbol{\Phi} = \boldsymbol{\Phi}^*$. Projecting the observation onto each dictionary atom gives:

$$\left(\boldsymbol{\Phi}^\top \boldsymbol{x}\right)_i \;=\; z_i \;+\; \underbrace{\sum_{j \neq i} \left(\phi^{(i)}\right)^\top \phi^{(j)} z_j}_{\text{interference from other features}} \quad . \tag{19}$$

The projection combines the feature's own activity with contributions from other active features, weighted by their overlap with its dictionary atom. Even when only one feature j is active, an inactive feature i can receive a nonzero projection. When several features are active, their contributions can reinforce or cancel one another, making projection scores less reliable indicators of individual feature activity (Elhage et al., 2022).

For an orthonormal dictionary, all interference terms vanish, and inference reduces to a single projection: $\boldsymbol{z} = \boldsymbol{\Phi}^\top \boldsymbol{x}$. However, an overcomplete dictionary ($K > M$) cannot have mutually orthogonal columns, so this interference cannot be eliminated for all feature pairs.

Therefore, inference should distinguish a feature's own contribution from evidence attributable to other active features. This motivates *explaining away*: support for a candidate feature decreases when other features already explain the evidence that initially supported it (Murphy, 2012). Such interactions allow inference to account for overlap between dictionary atoms rather than interpreting each projection independently (Rozell et al., 2008).

A.2 Iterative sparse encoders.

Iterative encoders refine sparse representations through successive updates rather than a single feed-forward prediction. ISTA, FISTA, and soft-thresholding LCA optimize ℓ_1-regularized reconstruction (Daubechies et al., 2004; Beck & Teboulle, 2009; Rozell et al., 2008), with LCA implementing lateral competition; LISTA learns a finite-step approximation (Gregor & LeCun, 2010), while matching pursuit (MP) greedily selects atoms to explain the remaining residual (Mallat & Zhang, 1993).

With a fixed pretrained dictionary, better inference remains limited by the learned feature directions. Additional ISTA refinement improves reconstruction with little change in BatchTopK probe scores; non-amortized ISTA worsens splitting for Gemma TopK and increases dead latents for Pythia Standard SAEs (Sun et al., 2026). FISTA applied to fixed SAE dictionaries also fails to close the feature-recovery gap in controlled experiments, implicating the learned dictionary as a bottleneck (Barin-Pacela et al., 2026). On Pythia-160M, however, Sun et al. report improved targeted-probe selectivity and generally higher generative intervention scores with semi- and non-amortized inference (Sun et al., 2026, Appendix C).

Approaches that also learn the dictionary have shown promising results: MP-SAE captures hierarchical features through residual-guided selection and supports different inference-time sparsity levels (Costa et al., 2025), while LocA-SAE seeks similar feature-quality benefits through grouped amortized encoders rather than explicit iterative refinement (Sun et al., 2026). But better reconstruction has not consistently translated into better interpretability tools. On Gemma-2-2B, MP-SAEs improve reconstruction but underperform the tested feed-forward baselines on sparse probing, with substantially longer training; the evaluator therefore recommends against them as a first choice for practical interpretability (Chanin, 2025). LocA-SAE eliminates dead latents on Gemma but scores below the Standard SAE on the same proxy probes, so improved utilization need not yield more predictive features (Sun et al., 2026). Across these evaluations, reconstruction, probing, and intervention metrics need not improve together, so the practical benefit of iterative refinement remains task-dependent.

A.3 VARIATIONAL INFERENCE AND THE FREE ENERGY OBJECTIVE

We provide a pedagogical introduction to variational inference and derive the free energy objective. In this derivation, we distinguish two KL divergences: one measures how well we approximate the true posterior, while the other measures departure from the prior. They play different roles in the same exact identity.

Finally, we explain why minimizing free energy improves inference using a few lines of algebra.

Variational posterior inference. Consider an observation $\boldsymbol{x} \in \mathbb{R}^M$ and K-dimensional latent variables $\boldsymbol{z}$. For Bernoulli latents, $\boldsymbol{z} \in \{0,1\}^K$. A generative model specifies a prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ and a likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})$, whose product is the joint distribution. Bayes' rule gives:

$$p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x}) \;=\; \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})}\,, \qquad p_{\boldsymbol{\theta}}(\boldsymbol{x}) \;=\; \sum_{\boldsymbol{z} \in \{0,1\}^K} p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\, p_{\boldsymbol{\theta}}(\boldsymbol{z})\,. \tag{20}$$

The denominator is the *model evidence*: the probability density of the observation after summing over all latent explanations. This sum contains 2^K terms, making direct evaluation generally infeasible for large K. The "true posterior" here means the exact posterior under the chosen generative model.

Variational inference is a powerful approximation method that replaces this computation with an optimization problem (Jordan et al., 1999; Blei et al., 2017). In variational inference, we introduce a tractable approximate posterior $q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x})$ and adjust its variational parameters $\boldsymbol{u}$ to minimize

$$\mathcal{D}_{\mathrm{KL}}\Big(q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x}) \,\Big\|\, p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})\Big) \;\coloneqq\; \sum_{\boldsymbol{z}} q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x}) \log \frac{q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x})}\,. \tag{21}$$

This KL divergence is nonnegative and vanishes exactly when the two distributions agree (Blei et al., 2017). But its value still depends on the intractable posterior, so we can't directly compute and optimize it. The next step isolates that dependence, leaving an objective we can optimize without evaluating the evidence.

Free-energy derivation. Fix $\boldsymbol{x}$ and $\boldsymbol{\theta}$ throughout the derivation. To keep the algebra readable, write $q(\boldsymbol{z}) \equiv q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x})$ and suppress the parameter subscript on p. All expectations below are over $\boldsymbol{z} \sim q$. For finite Bernoulli logits and a positive Gaussian covariance, the distributions are strictly positive, so the following ratios are well defined.

Since q is normalized, $\sum_{\boldsymbol{z}} q(\boldsymbol{z}) = 1$. Multiply the log evidence by this sum, apply Bayes' rule, and insert $q(\boldsymbol{z})/q(\boldsymbol{z}) = 1$:

$$\begin{aligned}
\log p(\boldsymbol{x}) \;&=\; \log p(\boldsymbol{x}) \underbrace{\sum_{\boldsymbol{z}} q(\boldsymbol{z})}_{=1} \;=\; \mathbb{E}_q[\log p(\boldsymbol{x})] \\
&=\; \mathbb{E}_q\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{z} \mid \boldsymbol{x})}\right] \\
&=\; \mathbb{E}_q\left[\log \left[\frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{z} \mid \boldsymbol{x})} \frac{q(\boldsymbol{z})}{q(\boldsymbol{z})}\right]\right] \\
&=\; \mathbb{E}_q\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z})}\right] \;+\; \mathbb{E}_q\left[\log \frac{q(\boldsymbol{z})}{p(\boldsymbol{z} \mid \boldsymbol{x})}\right] \\
&=\; \underbrace{\mathbb{E}_q\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z})}\right]}_{\mathrm{ELBO} \,=\, -\mathcal{F}} \;+\; \mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{z}) \,\|\, p(\boldsymbol{z} \mid \boldsymbol{x}))\,.
\end{aligned} \tag{22}$$

All we did was multiply by one twice, then rearrange logarithms. No inequality was needed: we have an exact decomposition of the log evidence. Nonnegativity of the final KL gives the *evidence lower bound*, $\mathrm{ELBO} \le \log p(\boldsymbol{x})$. Its negative is the variational free energy, $\mathcal{F} \equiv -\mathrm{ELBO}$ (Neal & Hinton, 1998; Kingma & Welling, 2014).

Posterior and prior KL divergences. Expanding $p_{\theta}(\boldsymbol{x}, \boldsymbol{z}) = p_{\theta}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\theta}(\boldsymbol{z})$ inside the ELBO reveals a second KL (Kingma & Welling, 2014). With $\mathcal{F} \equiv \mathcal{F}(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{\theta})$, we have:

$$\begin{aligned} \underbrace{\log p_{\theta}(\boldsymbol{x})}_{\text{log evidence}} &= \underbrace{-\mathcal{F}}_{\text{ELBO}} + \underbrace{\mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x}) \,\|\, p_{\theta}(\boldsymbol{z} \mid \boldsymbol{x}))}_{\substack{\text{inference KL: original objective} \\ \text{(gap to the true posterior)}}} \\ \mathcal{F} &= \underbrace{\mathbb{E}_{q_{\boldsymbol{u}}}[-\log p_{\theta}(\boldsymbol{x} \mid \boldsymbol{z})]}_{\text{reconstruction term (accuracy)}} + \underbrace{\mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x}) \,\|\, p_{\theta}(\boldsymbol{z}))}_{\text{prior KL (coding rate)}} . \end{aligned} \tag{23}$$

The *inference KL* compares the approximate posterior with the true posterior, $p_{\theta}(\boldsymbol{z} \mid \boldsymbol{x})$. This is the original objective in eq. (21). The *prior KL*, by contrast, compares the approximate posterior with $p_{\theta}(\boldsymbol{z})$, before conditioning on the observation. It is a component of free energy, and can be interpreted as a *coding rate*: the cost of representing the input-dependent distribution relative to the prior (Alemi et al., 2018). We use natural logarithms, so this cost is measured in nats.

These quantities need not decrease together. If inference starts at the prior, the prior KL is initially zero. Explaining an informative observation generally requires moving away from that prior, increasing the coding rate while reducing the inference KL. Free energy balances this cost against the improvement in expected log likelihood. Minimizing the prior KL alone would simply return the prior and ignore the observation.

Free energy and posterior approximation. Rearranging the first line of eq. (23) yields

$$\mathcal{F}(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{\theta}) = \mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x}) \,\|\, p_{\theta}(\boldsymbol{z} \mid \boldsymbol{x})) - \log p_{\theta}(\boldsymbol{x}) . \tag{24}$$

The evidence depends on the generative model and observation, but not on the variational parameters. Therefore, with $\boldsymbol{x}$ and $\boldsymbol{\theta}$ fixed, every decrease in $\mathcal{F}$ is an equal decrease in the inference KL (Blei et al., 2017). Equivalently,

$$\nabla_{\boldsymbol{u}} \mathcal{F} = \nabla_{\boldsymbol{u}} \mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x}) \,\|\, p_{\theta}(\boldsymbol{z} \mid \boldsymbol{x})) . \tag{25}$$

This is the sense in which free energy minimization improves inference. It holds regardless of whether the variational parameters are produced by an amortized encoder or refined iteratively. The bound becomes tight, $\mathcal{F} = -\log p_{\theta}(\boldsymbol{x})$, only when the approximate posterior equals the true posterior. A restricted variational family may not contain that posterior (Wainwright & Jordan, 2008; Blei et al., 2017); for example, a factorized Bernoulli distribution cannot represent all the dependencies induced by overlapping dictionary atoms. Optimization can reduce the KL without eliminating this approximation gap.

A.4 Trust regions and natural-gradient inference

Inference updates the approximate posterior $q_{\boldsymbol{u}}$ to reduce $\mathcal{F}(\boldsymbol{u})$, with the observation, generative parameters, and prior reference held fixed. The gradient tells us how the objective changes locally, but turning it into an update also requires a measure of the size of a step. We develop this choice from ordinary gradient descent to the Fisher geometry used by FOND (Vafaii et al., 2025).

Local quadratic approximation. At the current iterate $\boldsymbol{u}_t$, write $\mathbf{g}_t := \nabla_{\boldsymbol{u}} \mathcal{F}(\boldsymbol{u}_t)$. For a small displacement $\Delta \boldsymbol{u}$,

$$\mathcal{F}(\boldsymbol{u}_t + \Delta \boldsymbol{u}) = \mathcal{F}(\boldsymbol{u}_t) + \mathbf{g}_t^{\top} \Delta \boldsymbol{u} + o(\|\Delta \boldsymbol{u}\|). \tag{26}$$

Minimizing the linear term alone gives no finite step when $\mathbf{g}_t \neq 0$: it keeps decreasing along $-\mathbf{g}_t$. Adding a quadratic cost for movement produces the well-defined problem

$$\Delta \boldsymbol{u} = \arg\min_{\boldsymbol{\delta}} \left\{ \mathbf{g}_t^{\top} \boldsymbol{\delta} + \frac{1}{2\eta} \|\boldsymbol{\delta}\|_2^2 \right\} = -\eta \mathbf{g}_t, \qquad \eta > 0. \tag{27}$$

The last equality follows by differentiating and setting $\mathbf{g}_t + \boldsymbol{\delta}/\eta = 0$. Thus ordinary gradient descent solves a local surrogate that balances predicted loss reduction against squared Euclidean displacement (Martens, 2020). The quadratic term controls how far we move from the point where the linear approximation was made; it is not an additional term in the inference objective.

Trust-region formulation. Instead of penalizing movement, we can constrain it (Boyd & Vandenberghe, 2004, Sec. 9.4):

$$\min_{\boldsymbol{\delta}} \ \mathbf{g}_t^\top \boldsymbol{\delta} \quad \text{subject to} \quad \|\boldsymbol{\delta}\|_2^2 \le \epsilon. \tag{28}$$

For $\epsilon > 0$ and $\mathbf{g}_t \neq 0$, the optimum is $\Delta \boldsymbol{u} = -\sqrt{\epsilon}\,\mathbf{g}_t / \|\mathbf{g}_t\|_2$. It agrees with eq. (27) when $\epsilon = \eta^2 \|\mathbf{g}_t\|_2^2$. The learning rate therefore determines an implicit movement budget, which also depends on the local gradient. A fixed learning rate does not imply a fixed trust-region radius. Both formulations ask which small movement most reduces the linearized objective.

Metric-dependent updates. A positive-definite matrix $\mathbf{M}_t$ defines the local squared length $\boldsymbol{\delta}^\top \mathbf{M}_t \boldsymbol{\delta}$. Replacing the Euclidean penalty gives

$$\Delta \boldsymbol{u} = \arg\min_{\boldsymbol{\delta}} \left\{ \mathbf{g}_t^\top \boldsymbol{\delta} + \frac{1}{2\eta} \boldsymbol{\delta}^\top \mathbf{M}_t \boldsymbol{\delta} \right\} = -\eta \mathbf{M}_t^{-1} \mathbf{g}_t. \tag{29}$$

The corresponding constraint $\boldsymbol{\delta}^\top \mathbf{M}_t \boldsymbol{\delta} \le \epsilon$ has solution

$$\Delta \boldsymbol{u} = -\sqrt{\frac{\epsilon}{\mathbf{g}_t^\top \mathbf{M}_t^{-1} \mathbf{g}_t}}\, \mathbf{M}_t^{-1} \mathbf{g}_t, \qquad \mathbf{g}_t \neq 0. \tag{30}$$

The Euclidean ball becomes an ellipsoid: directions assigned a larger movement cost receive smaller updates. Steepest descent consequently depends on the chosen geometry (Amari, 1998; Boyd & Vandenberghe, 2004). Here the metric is evaluated at the current iterate and held fixed within the local problem. Its quadratic form measures infinitesimal length, rather than an exact distance between widely separated points.

KL divergence and statistical distance. The parameters $\boldsymbol{u}$ specify a distribution. A step that barely changes this distribution should have a small statistical cost even if its parameter displacement is large. Conversely, a numerically small step can be costly if it substantially changes the represented beliefs. This motivates a movement budget based on

$$D_{\mathrm{KL}}(q_{\boldsymbol{u}+\Delta \boldsymbol{u}} \,\|\, q_{\boldsymbol{u}}) = \mathbb{E}_{q_{\boldsymbol{u}+\Delta \boldsymbol{u}}} \left[\log \frac{q_{\boldsymbol{u}+\Delta \boldsymbol{u}}(\boldsymbol{z})}{q_{\boldsymbol{u}}(\boldsymbol{z})} \right]. \tag{31}$$

Hobson's axiomatic characterization provides one motivation for KL as an information measure for finite-outcome belief revision (Hobson, 1969). In the optimization problem above, it measures the change between successive approximate posteriors. The KL term in the free energy instead compares the approximate posterior with its prior reference.

For a smooth family with common positive support and sufficient regularity to differentiate under the expectation, define the score $\mathbf{s}_{\boldsymbol{u}}(\boldsymbol{z}) := \nabla_{\boldsymbol{u}} \log q_{\boldsymbol{u}}(\boldsymbol{z})$. Normalization gives $\mathbb{E}_q[\mathbf{s}_{\boldsymbol{u}}] = 0$. The KL divergence and its first derivative vanish at zero displacement, while its Hessian there is the Fisher information:

$$\mathbf{G}_u(\boldsymbol{u}) = \mathbb{E}_{q_{\boldsymbol{u}}} \left[\mathbf{s}_{\boldsymbol{u}}(\boldsymbol{z}) \mathbf{s}_{\boldsymbol{u}}(\boldsymbol{z})^\top \right], \tag{32}$$

$$D_{\mathrm{KL}}(q_{\boldsymbol{u}+\Delta \boldsymbol{u}} \,\|\, q_{\boldsymbol{u}}) = \frac{1}{2} \Delta \boldsymbol{u}^\top \mathbf{G}_u(\boldsymbol{u}) \Delta \boldsymbol{u} + o(\|\Delta \boldsymbol{u}\|^2). \tag{33}$$

Reversing the KL arguments gives the same quadratic term (Martens, 2020). Thus, although KL is asymmetric, both orientations induce the same local metric.

The Fisher metric also has a statistical characterization beyond this Taylor expansion. For regular minimal exponential families, Dowty's Chentsov-type theorem characterizes it up to a positive constant under preservation by canonical sufficient statistics, the appropriate scaling for independent replication, and additional compatibility and continuity assumptions (Dowty, 2018). Sufficient statistics retain the information about the parameter, so this requirement asks that such a reduction preserve statistical lengths. These statistical conditions, rather than coordinate invariance alone, distinguish the Fisher metric.

The natural-gradient update. Using the quadratic KL approximation in the movement penalty yields

$$\boxed{\boldsymbol{u}_{t+1} = \boldsymbol{u}_t - \eta\, \mathbf{G}_u(\boldsymbol{u}_t)^{-1} \nabla_{\boldsymbol{u}} \mathcal{F}(\boldsymbol{u}_t).} \tag{34}$$

This is natural-gradient descent wherever the Fisher is nonsingular (Amari, 1998; Martens, 2020). The associated local trust-region problem constrains $D_{\mathrm{KL}}(q_{\boldsymbol{u}+\Delta\boldsymbol{u}}\|q_{\boldsymbol{u}}) \le \epsilon/2$. After replacing KL by its quadratic expansion, the normalized step is given by eq. (30) with $\mathbf{M}_t = \mathbf{G}_u(\boldsymbol{u}_t)$. For the continuous flow $\dot{\boldsymbol{u}} = -\mathbf{G}_u^{-1}\nabla_{\boldsymbol{u}}\mathcal{F}$,

$$\frac{d\mathcal{F}}{dt} = -\big(\nabla_{\boldsymbol{u}}\mathcal{F}\big)^\top \mathbf{G}_u^{-1}\nabla_{\boldsymbol{u}}\mathcal{F} \le 0. \tag{35}$$

A finite update still uses a local approximation: an arbitrary step size need not decrease the original objective.

A Gaussian inference example. Consider a Gaussian target posterior $p(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \mathbf{m}, \mathbf{S})$ and the variational family $q_{\boldsymbol{u}}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{u}, \mathbf{S})$, with fixed $\mathbf{S} \succ 0$. In this example $\boldsymbol{u}$ denotes the posterior mean. For the ordinary variational free energy,

$$\mathcal{F}(\boldsymbol{u}) - \mathcal{F}(\mathbf{m}) = D_{\mathrm{KL}}(q_{\boldsymbol{u}}\|p) = \frac{1}{2}(\boldsymbol{u}-\mathbf{m})^\top \mathbf{S}^{-1}(\boldsymbol{u}-\mathbf{m}). \tag{36}$$

The score is $\mathbf{S}^{-1}(\boldsymbol{z}-\boldsymbol{u})$, giving $\mathbf{G}_u = \mathbf{S}^{-1}$. Hence

$$\nabla_{\boldsymbol{u}}\mathcal{F} = \mathbf{S}^{-1}(\boldsymbol{u}-\mathbf{m}), \qquad \mathbf{G}_u^{-1}\nabla_{\boldsymbol{u}}\mathcal{F} = \boldsymbol{u}-\mathbf{m}. \tag{37}$$

Along an eigenvector of $\mathbf{S}$ with eigenvalue λ_j, a Euclidean step multiplies the mean error by $1-\eta/\lambda_j$; a natural-gradient step multiplies every component by $1-\eta$. Euclidean convergence therefore requires $0 < \eta < 2\lambda_{\min}(\mathbf{S})$, whereas natural-gradient convergence requires $0 < \eta < 2$ in this example. A narrow direction can force small Euclidean steps while broad directions converge slowly. Fisher preconditioning equalizes these contraction factors; with $\eta = 1$ it reaches the target mean in one step for this mean-only quadratic problem.

Reparameterization invariance. Let $\boldsymbol{v} = f(\boldsymbol{u})$ be a smooth invertible reparameterization with Jacobian $\mathbf{J} = \partial\boldsymbol{v}/\partial\boldsymbol{u}$. The chain rule and the score definition give

$$\nabla_{\boldsymbol{v}}\mathcal{F} = \mathbf{J}^{-\top}\nabla_{\boldsymbol{u}}\mathcal{F}, \qquad \mathbf{G}_v = \mathbf{J}^{-\top}\mathbf{G}_u\mathbf{J}^{-1}, \qquad \mathbf{G}_v^{-1}\nabla_{\boldsymbol{v}}\mathcal{F} = \mathbf{J}\mathbf{G}_u^{-1}\nabla_{\boldsymbol{u}}\mathcal{F}. \tag{38}$$

The natural-gradient velocity transforms exactly as a velocity should: $\dot{\boldsymbol{v}} = \mathbf{J}\dot{\boldsymbol{u}}$. Consequently, wherever the flow is well defined, the two coordinate systems describe the same evolution of distributions (Amari, 1998; Martens, 2020). A finite Euler step has this equivalence exactly for affine coordinate changes, but generally only to first order for nonlinear changes (Martens, 2020).

B Extended FOND model and dynamics

B.1 Model and objective

Bernoulli instantiation of FOND. Recovering a feature dictionary requires two coupled operations: infer which features explain each observation, and use those explanations to improve the dictionary. FOND derives the first operation as recurrent natural-gradient descent on variational free energy (Vafaii et al., 2025). Its original Poisson instantiation learns the decoder by backpropagating through the inference trajectory. Here we extend the natural-gradient prescription to learning itself. Inference changes the approximate posterior $q_{\boldsymbol{u}}$; learning changes the generative distribution $p_{\boldsymbol{\theta}}$. The two natural-gradient flows use their respective Fisher metrics to measure local changes in these distributions (Amari, 1998; Martens, 2020). Appendix A.4 develops this geometric motivation.

Bernoulli features and a Gaussian decoder. In BeFOND, each latent indicates whether a feature is present, and its decoder atom specifies that feature's contribution to the observation. The posterior mean r_i therefore expresses confidence in feature presence. For K binary latents and an M-dimensional observation, the local inference model is

$$\begin{aligned} p(\boldsymbol{z}) &= \prod_{i=1}^{K} \mathrm{Bernoulli}(z_i; \pi_{\mathrm{ref},i}), & q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x}) &= \prod_{i=1}^{K} \mathrm{Bernoulli}(z_i; r_i), \\ \boldsymbol{r} &= \sigma(\boldsymbol{u}), & p(\boldsymbol{x} \mid \boldsymbol{z}) &= \mathcal{N}(\boldsymbol{x}; \boldsymbol{\Phi}\boldsymbol{z}, \boldsymbol{\Sigma}), \end{aligned} \tag{39}$$

where $\boldsymbol{\Sigma}$ is diagonal and positive. Equivalently, $\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{z} + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$: the noise describes variation not explained by the selected atoms, while the atom itself sets the amplitude and direction of a present feature. The learned initial prior $\boldsymbol{\pi}_0$ supplies the first reference; the rolling rule in Appendix B.1.1 updates that reference as inference proceeds. A learned decoder bias accounts for a nonzero observation mean by replacing $\boldsymbol{x}$ with $\boldsymbol{x} - \boldsymbol{b}$; Appendix B.3 gives its update and inference-response correction.

Conditioning on the observation couples otherwise independent latent features: the likelihood contains cross-terms $W_{ij} z_i z_j$, where $W_{ij} = \boldsymbol{\phi}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\phi}_j$. When atoms overlap, the evidence for one feature depends on which others explain the input. Factorizing q restricts the represented posterior dependencies, but each optimized mean can still depend on the other means through the shared reconstruction (Appendix B.2).

The Poisson and Bernoulli choices both admit analytic moments and Fisher metrics (Wainwright & Jordan, 2008). In canonical coordinates,

latent family	support	r_i	$G_{u,ii} = \mathrm{Var}_q(z_i)$
Poisson	$\{0, 1, 2, \ldots\}$	e^{u_i}	r_i
Bernoulli	$\{0, 1\}$	$\sigma(u_i)$	$r_i(1 - r_i)$

(40)

The variance in the last column enters both the expected reconstruction loss and the geometry of inference. This is the connection that allows the distributional assumptions to determine the dynamics.

Free-energy decomposition. Appendix A.3 derives the fixed-prior ELBO identity. Here we evaluate its two terms for the Bernoulli–Gaussian model, holding the current reference fixed. A distribution over codes assigns probability to several possible reconstructions. Write $\boldsymbol{\delta} = \boldsymbol{z} - \boldsymbol{r}$ and $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{r}$. Since $\mathbb{E}_q[\boldsymbol{\delta}] = 0$ and $\mathbb{E}_q[\boldsymbol{\delta}\boldsymbol{\delta}^\top] = \mathbf{C}$, expanding the square gives

$$\begin{aligned}\mathbb{E}_q\left[\|\boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{z}\|^2_{\boldsymbol{\Sigma}^{-1}}\right] &= \boldsymbol{e}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{e} - 2\boldsymbol{e}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi} \mathbb{E}_q[\boldsymbol{\delta}] + \mathbb{E}_q[\boldsymbol{\delta}^\top \mathbf{W} \boldsymbol{\delta}] \\ &= \|\boldsymbol{e}\|^2_{\boldsymbol{\Sigma}^{-1}} + \mathrm{tr}(\mathbf{W}\mathbf{C}),\end{aligned} \tag{41}$$

where $\mathbf{W} = \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi}$ and $\mathbf{C} = \mathrm{diag}[\boldsymbol{r} \odot (\mathbf{1} - \boldsymbol{r})]$. The cross-term vanishes because the centered code has zero mean. For the KL term, factorization turns the log ratio of products into a sum:

$$\begin{aligned}D_{\mathrm{KL}}(q\|p) &= \sum_i \mathbb{E}_q\left[z_i \log \frac{r_i}{\pi_{\mathrm{ref},i}} + (1 - z_i) \log \frac{1 - r_i}{1 - \pi_{\mathrm{ref},i}}\right] \\ &= \sum_i \left[r_i \log \frac{r_i}{\pi_{\mathrm{ref},i}} + (1 - r_i) \log \frac{1 - r_i}{1 - \pi_{\mathrm{ref},i}}\right].\end{aligned} \tag{42}$$

A decoder that reconstructs $\boldsymbol{x}$ from $\boldsymbol{r}$ can therefore still have a large expected error if the individual latent configurations produce different observations. Including this uncertainty cost and the KL penalty gives

$$\begin{aligned}\mathcal{F} = \tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{r})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{r}) + \tfrac{1}{2}\sum_i W_{ii} r_i(1 - r_i) + \tfrac{1}{2}\log\det \boldsymbol{\Sigma} \\ + \beta \sum_i \left[r_i \log \frac{r_i}{\pi_{\mathrm{ref},i}} + (1 - r_i) \log \frac{1 - r_i}{1 - \pi_{\mathrm{ref},i}}\right] + \mathrm{const},\end{aligned} \tag{43}$$

with $\mathbf{W} = \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi}$. The log determinant normalizes the Gaussian likelihood, and β weights departure from the current prior reference. At $\beta = 1$ and a fixed generative prior, this is the negative evidence lower bound. The Bernoulli uncertainty term also cancels the quadratic self-interaction in each coordinate, leaving reconstruction affine in r_i when the other means are fixed (Appendix B.4). Overlapping atoms still couple different coordinates, so a factorized posterior does not imply independent inference updates.

The moments needed for inference and learning are available without sampling latent codes:

$$\begin{aligned}\mathbf{C} &= \mathrm{diag}[\boldsymbol{r} \odot (\mathbf{1} - \boldsymbol{r})], & \mathbf{E} &= \mathbf{C} + \boldsymbol{r}\boldsymbol{r}^\top, \\ \mathbf{E}_p &= \mathrm{diag}[\boldsymbol{\pi}_{\mathrm{ref}} \odot (\mathbf{1} - \boldsymbol{\pi}_{\mathrm{ref}})] + \boldsymbol{\pi}_{\mathrm{ref}}\boldsymbol{\pi}_{\mathrm{ref}}^\top, & V_{mm} &= (\boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{r})^2_m + (\boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^\top)_{mm}.\end{aligned} \tag{44}$$

Here $\mathbf{E}$ and $\mathbf{E}_p$ are the posterior and prior second moments, and V_{mm} is the posterior-expected squared residual in observed dimension m.

Bernoulli inference in BeFOND. For our factorized Bernoulli posterior, $\boldsymbol{u}$ contains logits and $r_i = \sigma(u_i)$. The score is $\boldsymbol{z} - \boldsymbol{r}$, so

$$\mathbf{G}_u = \mathrm{diag}(\boldsymbol{r} \odot (\mathbf{1} - \boldsymbol{r})), \qquad \Delta\boldsymbol{u}^\top \mathbf{G}_u \Delta\boldsymbol{u} = \sum_i r_i(1 - r_i)(\Delta u_i)^2. \tag{45}$$

For a fixed small logit displacement, the distribution changes less when a Bernoulli probability is near zero or one. In probability coordinates, the same infinitesimal length is $\sum_i (dr_i)^2/[r_i(1-r_i)]$: a fixed probability displacement is more costly near the boundary. For finite logits, all these variances are positive. Applying the inverse Fisher cancels the variance factor in the Euclidean logit gradient, yielding the inference dynamics derived in section B.2. The update therefore accounts for how much a parameter movement changes the inferred beliefs.

Natural-gradient inference and learning. For Bernoulli logits, the Fisher is $\mathbf{G}_u = \mathbf{C}$. The same factor $r_i(1-r_i)$ multiplies the ordinary logit gradient, so applying $\mathbf{G}_u^{-1}$ removes this attenuation from the inference drive. The remaining terms combine evidence from the input, recurrent interactions through the dictionary, and relaxation toward the reference. Appendix B.2 derives these terms and their cancellation explicitly.

Learning then holds the inferred moments fixed. The decoder gradient compares the observed input–latent moment $\langle \boldsymbol{x}\boldsymbol{r}^\top \rangle$ with the model prediction $\mathbf{\Phi E}$. Its Fisher uses the prior second moment, $\mathbf{G}_\Phi = \mathbf{E}_p \otimes \mathbf{\Sigma}^{-1}$, giving a correction by $\mathbf{E}_p^{-1}$ rather than the posterior's diagonal Fisher. This correction is what can compensate for weak learning signals in rarely active directions; exact compensation requires matching the minibatch-averaged second moments (Appendix C.4). The covariance update tracks expected squared residuals, while the learned initial prior tracks detached posterior means across inputs. These decoder, covariance, and initial-prior updates use the inferred statistics directly, without backpropagating through the inference trajectory. Their explicit flows and derivations appear in Appendices B.2 and B.2.1. The decoder evolves in learning time τ over an interval Δt; h denotes the separate inference integration interval.

B.1.1 Online prior and working memory

Consider observing a scene whose underlying features persist. After the first observation, the inferred distribution contains information about that particular scene that the population prior did not contain. The next update should start from this informed belief. FOND's rolling rule makes it the new prior reference (Vafaii et al., 2025): the KL penalty now measures departure from the current prediction, while the likelihood supplies evidence for revising it. The learned initial prior starts the sequence; the rolling reference carries its evolving, input-specific belief forward. Fig. 5 shows this predict–update sequence and its fixed-input specialization. Appendix B.5 derives its activation threshold, and Appendix C.3.3 reports the rolling-prior experiments varying β and the initial-prior cap.

Prior prediction. Let q_t be the inferred distribution after observation $\boldsymbol{x}_t$. To form the next reference, a state-transition model predicts which latent configurations remain plausible. For a transition kernel $\mathcal{K}(\boldsymbol{z}' \mid \boldsymbol{z})$, prediction averages over the preceding belief (Särkkä & Svensson, 2023, Chapter 6):

$$p_{\mathrm{ref},t}(\boldsymbol{z}') = \sum_{\boldsymbol{z}} \mathcal{K}(\boldsymbol{z}' \mid \boldsymbol{z}) q_{t-1}(\boldsymbol{z}). \tag{46}$$

If the latent state persists without process noise, then $\mathcal{K}(\boldsymbol{z}' \mid \boldsymbol{z}) = \mathbf{1}[\boldsymbol{z}' = \boldsymbol{z}]$, so prediction simply copies the belief:

$$p_{\mathrm{ref},0} = p_0, \qquad p_{\mathrm{ref},t} = q_{t-1} \quad (t \geq 1), \qquad p_0(\boldsymbol{z}) = \prod_i \mathrm{Bernoulli}(z_i; \pi_{0,i}). \tag{47}$$

The identity transition means that the causes persist, not that we know which causes are present. Copying q_{t-1} therefore retains both its feature preferences and its uncertainty. With no new likelihood contribution, the belief is preserved unchanged; this is the working memory supplied by the rolling reference. A new independent input starts again from the learned initial prior.

Posterior update. The observation can support the prediction or require it to change. Expected reconstruction error rewards explanations of the current input, while the KL penalty controls departure

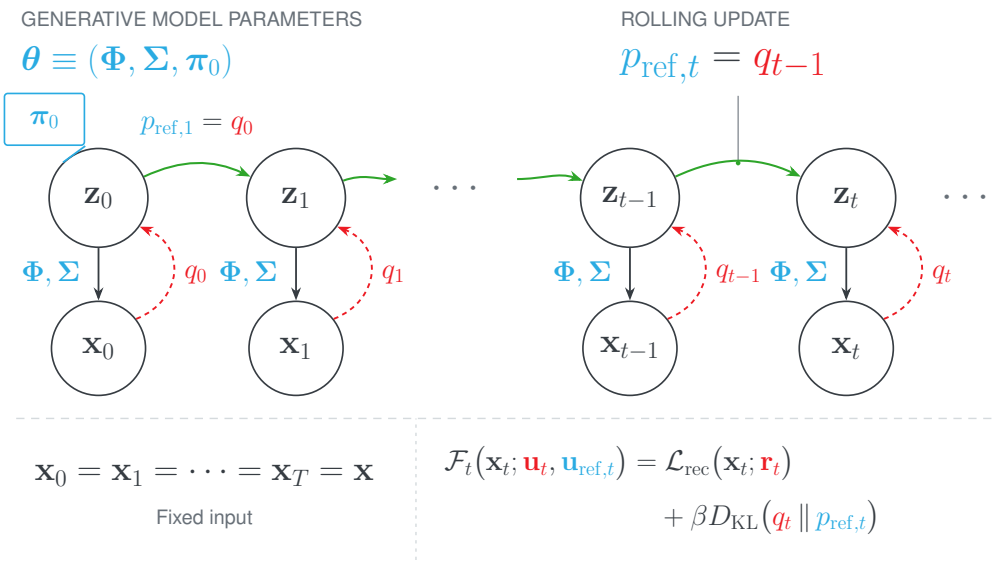


Figure 5: Online inference with a rolling prior. The learned initial prior initializes the first belief. Under identity state dynamics, prediction carries each posterior into the next reference (green arrows); the observation model is shown by black arrows and inference by red dashed arrows. The fixed-input case presents the same observation at each step.

from the prediction. For a Bernoulli reference with means $\pi_{\mathrm{ref},t}$ and logits $u_{\mathrm{ref},t} = \mathrm{logit}(\pi_{\mathrm{ref},t})$, the local objective is

$$\mathcal{F}_t(\boldsymbol{u}) = \underbrace{-\mathbb{E}_{q_{\boldsymbol{u}}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{z})}_{\mathcal{L}_{\mathrm{rec}}(\boldsymbol{x}_t; \boldsymbol{r})} + \beta D_{\mathrm{KL}}(q_{\boldsymbol{u}} \,\|\, p_{\mathrm{ref},t}), \qquad \boldsymbol{r} = \sigma(\boldsymbol{u}). \tag{48}$$

The decoder and reference remain fixed during this local update. At $\beta = 1$, minimizing over unrestricted distributions gives the Bayesian posterior for the reference and observation. For general $\beta > 0$, it gives a distribution proportional to $p_{\mathrm{ref},t}(\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{z})^{1/\beta}$ (Bissiri et al., 2016). The factorized Bernoulli family gives tractable expectations and local updates while retaining marginal uncertainty and approximating posterior dependencies.

Recurrent belief revision. Write the negative mean-coordinate reconstruction gradient as $\boldsymbol{a}_t(\boldsymbol{r})$. Including the Bernoulli uncertainty term, its ith coordinate is

$$[\boldsymbol{a}_t(\boldsymbol{r})]_i = (\phi^{(i)})^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_t - \tfrac{1}{2} W_{ii} - \sum_{j \neq i} W_{ij} r_j. \tag{49}$$

The first term supplies evidence for feature i; the last adjusts it for other active explanations. When $W_{ij} > 0$, increasing belief in feature j reduces the drive for i. Repeated updates thus let explanations revise one another even though q is factorized. Applying the posterior Fisher inverse gives

$$\dot{\boldsymbol{u}} = \boldsymbol{a}_t(\boldsymbol{r}) - \beta(\boldsymbol{u} - \boldsymbol{u}_{\mathrm{ref},t}). \tag{50}$$

Initialize at the predictive logits. The KL force is initially zero, so evidence can change the belief before the fixed reference exerts a restoring force. One ETD1 step with interval h integrates this relaxation while freezing the reconstruction drive (Appendix B.6):

$$\boldsymbol{u}_t = \boldsymbol{u}_{\mathrm{ref},t} + \frac{1 - e^{-\beta h}}{\beta} \boldsymbol{a}_t(\boldsymbol{\pi}_{\mathrm{ref},t}), \qquad \beta > 0. \tag{51}$$

The resulting posterior becomes the reference for the next outer step. The next update therefore retains the inferred changes instead of pulling them back toward the original population prior. If several inner steps are taken, their reference stays fixed and the restoring force generally becomes nonzero. The parameter β controls resistance to local belief change, and h controls how far the local flow is integrated.

Evidence accumulation and sparsity. Rolling also changes the influence of the initial prior as a sequence proceeds. To see this, suppose the initial prior and likelihood are positive and each local objective is minimized exactly over unrestricted distributions. Repeated substitution of the posterior for the next reference gives

$$q_t(\boldsymbol{z}) \;\propto\; p_0(\boldsymbol{z}) \prod_{j=0}^{t} p_{\boldsymbol{\theta}}(\boldsymbol{x}_j \mid \boldsymbol{z})^{1/\beta}. \tag{52}$$

Each observation contributes a likelihood factor, while the initial prior enters once. For a fixed repeated input, the likelihood power is $(t+1)/\beta$: the initial prior becomes weaker relative to reconstruction evidence. Replaying one stored observation supplies no new measurement, so this computation is not ordinary conditioning on one observation. Finite mean-field ETD1 steps approximate the local updates and need not attain this unrestricted endpoint.

This explains why a sparse initial prior need not enforce the same sparsity throughout a rollout. Its low probabilities set a high initial activation threshold, but the rolling reference can retain increases supported by reconstruction evidence. A cap on mean initial-prior activity therefore constrains the starting belief, not final posterior activity.

B.2 Natural-gradient dynamics

For the free energy in Appendix B.1, inference changes the posterior while holding the model fixed. Learning then changes the model with the inferred statistics fixed:

$$\dot{\boldsymbol{u}} = \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - \mathbf{W}\boldsymbol{r} - \tfrac{1}{2}\,\mathrm{diag}(\mathbf{W}) \odot (\mathbf{1} - 2\boldsymbol{r}) - \beta(\boldsymbol{u} - \boldsymbol{u}_{\mathrm{ref}}), \tag{53}$$

$$\dot{\boldsymbol{\Phi}} = \eta_\Phi(\boldsymbol{x}\boldsymbol{r}^\top - \boldsymbol{\Phi}\mathbf{E})\mathbf{E}_p^{-1}, \tag{54}$$

$$\dot{\boldsymbol{\Sigma}} = \eta_\Sigma(\mathbf{V} - \boldsymbol{\Sigma}), \tag{55}$$

$$\dot{\boldsymbol{\pi}}_0 = \eta_0 \beta(\boldsymbol{r} - \boldsymbol{\pi}_0). \tag{56}$$

Here $\boldsymbol{u}_{\mathrm{ref}} = \mathrm{logit}(\boldsymbol{\pi}_{\mathrm{ref}})$. The initial prior $\boldsymbol{\pi}_0$ tracks detached posterior means; the rolling reference is refreshed within inference (Appendix B.1.1). For minibatches, average $\boldsymbol{x}\boldsymbol{r}^\top$, $\mathbf{E}$, $\mathbf{E}_p$, and $\mathbf{V}$ before applying the updates. Derivations appear in Appendix B.2.1.

Fisher preconditioning for inference. For the Bernoulli posterior, $\partial \log q / \partial u_i = z_i - r_i$. Independence under q gives

$$\mathbf{G}_u = \mathbb{E}_q[(\boldsymbol{z} - \boldsymbol{r})(\boldsymbol{z} - \boldsymbol{r})^\top] = \mathbf{C}, \tag{57}$$

$$\nabla_{\boldsymbol{u}} \mathcal{F} = \mathbf{C}\Big[\mathbf{W}\boldsymbol{r} - \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \tfrac{1}{2}\,\mathrm{diag}(\mathbf{W}) \odot (\mathbf{1} - 2\boldsymbol{r})\Big] + \mathbf{C}\beta(\boldsymbol{u} - \boldsymbol{u}_{\mathrm{ref}}). \tag{58}$$

Multiplication by $-\mathbf{G}_u^{-1}$ cancels $\mathbf{C}$ and yields Equation (53) whenever $0 < r_i < 1$. Ordinary logit-gradient descent instead multiplies the same drive by $r_i(1 - r_i)$, making it small near either boundary. The natural flow removes this factor from the logit update; the probability still changes according to $\dot{\boldsymbol{r}} = \mathbf{C}\dot{\boldsymbol{u}}$. This distinction matters when interpreting the revival of a nearly inactive latent.

Decoder Fisher information. Let $\langle\cdot\rangle$ denote the minibatch average. The Gaussian decoder score is $\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{z})\boldsymbol{z}^\top$. Averaging its outer product over the complete-data model, with the chosen prior references held fixed, gives

$$\mathbf{G}_\Phi = \mathbf{E}_p \otimes \boldsymbol{\Sigma}^{-1}, \qquad \nabla_\Phi \mathcal{F} = \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\Phi}\mathbf{E} - \langle \boldsymbol{x}\boldsymbol{r}^\top \rangle\right), \tag{59}$$

where both second moments are averaged across the minibatch before the Fisher action. Vectorization of decoder entries is implicit. The resulting matrix-form flow is Equation (54), with $\boldsymbol{x}\boldsymbol{r}^\top$ replaced by its minibatch average. This is the complete-data model Fisher, rather than the observed-data Fisher or an outer product of empirical loss gradients (Amari, 1998; Martens, 2020).

The distinction between $\mathbf{E}_p$ and $\mathbf{E}$ is essential: the former supplies the metric, and the latter the curvature of the frozen reconstruction objective. Averaging per-example inverses would give a

different update. Likewise, matching average firing rates alone does not match second moments. For example, a shared factorized prior with mean $\langle \boldsymbol{r} \rangle$ has off-diagonal moments $\langle r_i \rangle \langle r_j \rangle$, whereas the averaged posterior has $\langle r_i r_j \rangle$. Their difference is the across-input covariance of the inferred means. Appendix C.4 states the exact matching condition and the unequal-moment extension.

B.2.1 Model updates

Proof of Eqs. (53)–(56). For Bernoulli inference, differentiating Eq. (43) gives

$$\nabla_{\boldsymbol{r}} \mathcal{F} = -\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x} + \mathbf{W} \boldsymbol{r} + \tfrac{1}{2} \operatorname{diag}(\mathbf{W}) \odot (\mathbf{1} - 2\boldsymbol{r}) + \beta(\boldsymbol{u} - \boldsymbol{u}_{\mathrm{ref}}).$$

The posterior Fisher $\mathbf{C}$ cancels the logit Jacobian, yielding Eq. (53). For the decoder, the complete-data Fisher block is $\mathbf{E}_p \otimes \boldsymbol{\Sigma}^{-1}$, which cancels the precision in the gradient and gives Eq. (54).

For each diagonal variance and each initial-prior probability,

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \Sigma_{mm}} &= \frac{\Sigma_{mm} - V_{mm}}{2\Sigma_{mm}^2}, & G_{\Sigma_{mm}} &= \frac{1}{2\Sigma_{mm}^2}, \\ \frac{\partial [\beta \mathcal{D}_{\mathrm{KL}}(q \,\|\, p_0)]}{\partial \pi_{0,i}} &= \beta \frac{\pi_{0,i} - r_i}{\pi_{0,i}(1 - \pi_{0,i})}, & G_{\pi_{0,i}} &= \frac{1}{\pi_{0,i}(1 - \pi_{0,i})}. \end{aligned} \tag{60}$$

Preconditioning proves Eqs. (55)–(56). With frozen targets, integration yields

$$\begin{aligned} \boldsymbol{\Sigma}(\Delta t) &= \mathbf{V} + [\boldsymbol{\Sigma}(0) - \mathbf{V}] e^{-\eta_\Sigma \Delta t}, \\ \boldsymbol{\pi}_0(\Delta t) &= \boldsymbol{r} + [\boldsymbol{\pi}_0(0) - \boldsymbol{r}] e^{-\eta_0 \beta \Delta t}. \end{aligned} \tag{61}$$

□

Decoder-bias correction.

Proof of Eq. (62). Hold the incoming reference and covariance fixed during one unclamped inference step. Only the feedforward drive depends on the bias, so

$$\frac{\partial \boldsymbol{u}^{\mathrm{new}}}{\partial \boldsymbol{b}} = -\frac{1 - e^{-\beta h}}{\beta} \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1}, \qquad \nabla_{\boldsymbol{u}} \mathcal{F} = -\mathbf{C} \dot{\boldsymbol{u}}.$$

Applying the chain rule at the updated posterior gives

$$\nabla_{\boldsymbol{b}} \mathcal{F} = -\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{b} - \boldsymbol{\Phi} \boldsymbol{r}) + \frac{1 - e^{-\beta h}}{\beta} \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi} \mathbf{C} \dot{\boldsymbol{u}}.$$

Multiplication by the inverse bias Fisher block $\boldsymbol{\Sigma}$ and minibatch averaging gives Eq. (62). □

B.3 Decoder bias

A learned bias gives $p(\boldsymbol{x} \mid \boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{b} + \boldsymbol{\Phi} \boldsymbol{z}, \boldsymbol{\Sigma})$; inference uses $\boldsymbol{x} - \boldsymbol{b}$. With a fixed posterior, the bias fits the mean residual. Differentiating through one unclamped ETD1 inference step adds a response term:

$$\dot{\boldsymbol{b}} = \eta_\Phi \mathbb{E}_{\mathrm{batch}} \left[\underbrace{\boldsymbol{x} - \boldsymbol{b} - \boldsymbol{\Phi} \boldsymbol{r}}_{\text{reconstruction residual}} \; \underbrace{- \frac{1 - e^{-\beta h}}{\beta} \boldsymbol{\Phi} \mathbf{C} \dot{\boldsymbol{u}}}_{\text{inference response}} \right]. \tag{62}$$

The incoming reference is fixed; $\mathbf{C}$ and $\dot{\boldsymbol{u}}$ are evaluated after that step. This expression assumes one shared covariance across the batch. Appendix B.2.1 derives the correction. The full-width synthetic experiment freezes the bias.

B.4 Bernoulli uncertainty and sparse coding

The Bernoulli posterior variance is $\mathrm{Var}_q(z_i) = r_i(1-r_i)$, which vanishes at 0 and 1 and is maximal at $1/2$. In isolation, its cost favors confident absence or presence. In the full expected reconstruction loss, it cancels the quadratic self-term:

$$\underbrace{\tfrac{1}{2}W_{ii}r_i^2}_{\text{mean reconstruction}} + \underbrace{\tfrac{1}{2}W_{ii}r_i(1-r_i)}_{\text{posterior uncertainty}} = \tfrac{1}{2}W_{ii}r_i. \tag{63}$$

Consequently, with the other coordinates fixed, reconstruction is affine in r_i and the KL term is strictly convex for $\beta > 0$. The full objective has an interior coordinate optimum at finite evidence; the variance term alone does not force binary posteriors.

B.5 Activation threshold under a rolling prior

Appendix B.1.1 motivates the rolling rule. Here we make its sparsity effect explicit. Starting one ETD1 step at the reference, the gain in Equation (51) tends to $1/\beta$ as $h \to \infty$. Since $\sigma(u) > 1/2$ exactly when $u > 0$, for $\beta > 0$ we obtain

$$r_i^{\text{new}} > \tfrac{1}{2} \iff (\phi^{(i)})^\top \Sigma^{-1}\boldsymbol{x} - \tfrac{1}{2}W_{ii} - \sum_{j\neq i} W_{ij}\pi_{\text{ref},j} > \beta \log \frac{1-\pi_{\text{ref},i}}{\pi_{\text{ref},i}}. \tag{64}$$

The right side is positive for a reference probability below $1/2$ and grows as that probability decreases. It sets the reconstruction evidence needed to activate the feature, after accounting for the other features' contributions on the left. Updating the reference changes this threshold at the next outer step. The prior cap and inference depth can therefore affect posterior sparsity through different parts of the rollout (Appendix C.3.3).

B.6 Numerical integration and matrix-free updates

Stiffness and exponential integration. Natural gradients specify the continuous-time evolution of beliefs and model parameters, but implementing these dynamics requires a numerical update. A difficulty with explicit Euler is that stable continuous dynamics can become unstable after discretization. For the simple relaxation $\dot{v} = -\beta v$, Euler multiplies v by $1-\beta h$, whereas exact integration multiplies it by $e^{-\beta h}$. Euler therefore requires $0 < \beta h < 2$ for decay. When fast relaxation accompanies slower dynamics, this stability restriction can force small steps even when the solution changes slowly. Exponential integrators address this source of stiffness by integrating the linear component exactly (Cox & Matthews, 2002; Hochbruck & Ostermann, 2010).

Inference integration. Write Equation (53) as $\dot{\boldsymbol{u}} = \boldsymbol{a}(\boldsymbol{r}) - \beta(\boldsymbol{u} - \boldsymbol{u}_{\text{ref}})$, where $\boldsymbol{a}$ contains the reconstruction and Bernoulli uncertainty terms. Holding the reference fixed and freezing $\boldsymbol{a}$ during one interval gives the first-order exponential time-differencing update, ETD1 (Cox & Matthews, 2002; Hochbruck & Ostermann, 2010):

$$\boldsymbol{u}_{\text{new}} = \boldsymbol{u}_{\text{ref}} + e^{-\beta h}(\boldsymbol{u} - \boldsymbol{u}_{\text{ref}}) + \frac{1-e^{-\beta h}}{\beta}\boldsymbol{a}(\boldsymbol{r}), \qquad \beta > 0. \tag{65}$$

The reference relaxation is thus preserved exactly. Here ETD1 follows the same natural-gradient direction as Euler, with effective step $(1-e^{-\beta h})/\beta$, bounded by $1/\beta$. Starting from the reference removes the middle term; Appendix B.1.1 describes how that reference is passed between inference steps. The nonlinear drive remains approximate: the $h \to \infty$ limit fully relaxes the frozen subproblem but does not guarantee convergence of recurrent inference. Finite inference depth therefore does not assume that the posterior has reached an optimum before every decoder update.

Frozen-statistics learning. The same principle applies to learning. With minibatch statistics, prior references, and covariance fixed, and $\mathbf{E}, \mathbf{E}_p \succ 0$, the decoder flow has the exact solution (Appendix C.4)

$$\boldsymbol{\Phi}(\Delta t) - \boldsymbol{\Phi}^* = \big(\boldsymbol{\Phi}(0) - \boldsymbol{\Phi}^*\big)\exp\!\big[-\eta_\Phi \Delta t\, \mathbf{E}\mathbf{E}_p^{-1}\big]. \tag{66}$$

Euler must accommodate the fastest relaxation mode; exponential integration instead applies each mode's exact decay over the chosen interval. When $\mathbf{E} = \mathbf{E}_p$, every mode contracts by the same factor, recovering exact rarity compensation.

The configured decoder solver, `matrix_free_phi1_auto`, evaluates a matrix-function action on the decoder residual. Let $s = \eta_\Phi \Delta t$, $\mathbf{E}_{p,\lambda} = \mathbf{E}_p + \lambda \mathbf{I}$ with Fisher damping $\lambda \geq 0$, and $\mathbf{A} = \mathbf{E}\mathbf{E}_{p,\lambda}^{-1}$. The corresponding frozen update is

$$\begin{aligned} \mathbf{\Phi}(\Delta t) = \mathbf{\Phi}(0) \\ + s\big(\langle \boldsymbol{x}\boldsymbol{r}^\top \rangle - \mathbf{\Phi}(0)\mathbf{E}\big)\mathbf{E}_{p,\lambda}^{-1}\varphi_1(-s\mathbf{A}), \end{aligned} \tag{67}$$

where brackets denote minibatch averages and $\varphi_1(\mathbf{Z}) = \sum_{m=0}^{\infty} \mathbf{Z}^m/(m+1)!$ is the entire matrix function used in exponential integration (Hochbruck & Ostermann, 2010). For $\lambda = 0$, this recovers the exact frozen-statistics solution above. The residual form avoids explicitly computing $\mathbf{\Phi}^*$ or subtracting a near-identity exponential from the identity. The implementation applies this function without constructing a dense $K \times K$ matrix. For a matrix $\mathbf{R}$ whose rows are minibatch posterior means,

$$\mathbf{E}\boldsymbol{v} = \mathrm{diag}(\langle \boldsymbol{r} \odot (\mathbf{1} - \boldsymbol{r}) \rangle)\boldsymbol{v} + \frac{1}{B}\mathbf{R}^\top(\mathbf{R}\boldsymbol{v}), \tag{68}$$

and the same representation applies to $\mathbf{E}_p$. Low-rank solves use the Woodbury identity (Hager, 1989) when the rank is at most 1,024; larger ranks use Jacobi-preconditioned conjugate gradients (Saad, 2003, Section 9.2). Thus the minibatch rank and conditioning also affect cost (Appendix B.8).

The solver combines scaled Taylor evaluation (Al-Mohy & Higham, 2011) with Chebyshev polynomial approximation (Trefethen, 2019, Chapters 3–4). It selects Taylor evaluation when the predicted work is within 32 operator applications, and otherwise uses a Chebyshev recurrence with degree chosen from the coefficient tail. Each operator application requires a posterior-moment product and a solve with the damped prior moment, so the polynomial degree and linear-solve iterations contribute to decoder-update cost. The selected synthetic recipes and audited whitened Gemma configurations request Chebyshev tolerance 10^{-5} and maximum degree 2,048 (512 for the raw 16K Gemma run); these control the matrix-function approximation rather than a guarantee on final feature recovery. Solver failures are reported instead of being treated as converged updates. The exponential solution is exact for the ideal frozen-statistics subproblem; numerical approximation, Fisher damping, and decoder norm constraints modify that ideal update.

The variance and initial-prior flows likewise integrate to exponential averages of their frozen targets (Appendix B.2.1). These updates exploit solvable components of the derived dynamics while leaving approximation to the remaining nonlinear terms and numerical matrix-function evaluation. Parameter floors, a mean initial-prior cap, and decoder constraints are applied as specified by each recipe. They are implementation constraints, not additional consequences of the unconstrained rarity-compensation theorem. All inferred statistics are detached during learning, so the decoder update does not backpropagate through the inference trajectory.

B.7 Algorithm

Algorithm 1 gives the Bernoulli inference and moment updates. During inference, the generative parameters remain fixed. Learning uses the inferred minibatch moments.

Algorithm 1 FOND with a rolling inference reference

Require: Decoder $\boldsymbol{\Phi}$, diagonal covariance $\boldsymbol{\Sigma} \succ 0$, initial prior $\boldsymbol{\pi}_0 \in (0,1)^K$; inference depths, $\beta > 0$, integration parameter h, and learning schedules.

for each training minibatch **do**
  Center observations by the decoder bias, if learned.
  $\boldsymbol{u} \leftarrow \operatorname{logit}(\boldsymbol{\pi}_0)$, independently for each observation.
  for each outer step up to the sampled training horizon **do**
    $\boldsymbol{u}_{\mathrm{ref}} \leftarrow \boldsymbol{u}$, $\boldsymbol{\pi}_{\mathrm{ref}} \leftarrow \sigma(\boldsymbol{u}_{\mathrm{ref}})$.
    $\mathbf{E}_p \leftarrow \mathbb{E}_{\mathrm{batch}}[\operatorname{diag}(\boldsymbol{\pi}_{\mathrm{ref}} \odot (\mathbf{1} - \boldsymbol{\pi}_{\mathrm{ref}})) + \boldsymbol{\pi}_{\mathrm{ref}} \boldsymbol{\pi}_{\mathrm{ref}}^\top]$.
    for each inner inference step **do**
      $\boldsymbol{r} \leftarrow \sigma(\boldsymbol{u})$, $\mathbf{W} \leftarrow \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi}$.
      $\dot{\boldsymbol{u}} \leftarrow \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - \mathbf{W}\boldsymbol{r} - \frac{1}{2} \operatorname{diag}(\mathbf{W}) \odot (\mathbf{1} - 2\boldsymbol{r}) - \beta(\boldsymbol{u} - \boldsymbol{u}_{\mathrm{ref}})$.
      $\boldsymbol{u} \leftarrow \boldsymbol{u} + \dfrac{1 - e^{-\beta h}}{\beta} \dot{\boldsymbol{u}}$.
    end for
    if this is the final outer step **then**
      $\boldsymbol{r} \leftarrow \sigma(\boldsymbol{u})$, $\mathbf{C} \leftarrow \operatorname{diag}[\boldsymbol{r} \odot (\mathbf{1} - \boldsymbol{r})]$, $\mathbf{E} \leftarrow \mathbb{E}_{\mathrm{batch}}[\mathbf{C} + \boldsymbol{r}\boldsymbol{r}^\top]$.
      $V_{mm} \leftarrow \mathbb{E}_{\mathrm{batch}}[(\boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{r})_m^2 + (\boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^\top)_{mm}]$.
      $\boldsymbol{\Phi}^* \leftarrow \mathbb{E}_{\mathrm{batch}}[\boldsymbol{x}\boldsymbol{r}^\top]\mathbf{E}^{-1}$.
      $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Phi}^* + (\boldsymbol{\Phi} - \boldsymbol{\Phi}^*) \exp[-\eta_\Phi \Delta t \, \mathbf{E}\mathbf{E}_p^{-1}]$.
      Update enabled variance and bias parameters by Eqs. (61) and (62); apply the stated decoder constraints.
    end if
  end for
  if an initial-prior update is scheduled **then**
    Infer $\boldsymbol{r}$ on the prior-update batch, then detach it.
    $\boldsymbol{\pi}_0 \leftarrow \mathbb{E}_{\mathrm{batch}}[\boldsymbol{r}] + (\boldsymbol{\pi}_0 - \mathbb{E}_{\mathrm{batch}}[\boldsymbol{r}])e^{-\eta_0 \beta \Delta t}$.
    Apply the specified prior floor and mean-probability cap.
  end if
end for

The decoder line is the frozen-moment matrix-exponential update; implementations can apply its matrix functions without forming dense inverses. ETD1 uses gain $1/\beta$ at $h = \infty$. Evaluation resets the initial prior for each input and performs inference without learning.

B.8 Computational complexity

Let B be minibatch size, d input dimension, K dictionary width, and T the total number of inner inference steps for an input. A dense SAE encoder costs $O(BdK)$; its training step has the same order, with a larger constant for decoding and backpropagation. BeFOND applies the lateral operator as $\boldsymbol{\Phi}^\top[\boldsymbol{\Phi}\boldsymbol{r}]$ (with diagonal observation precision inserted when used), so it does not form a dense $K \times K$ Gram matrix. Its costs are

$$\begin{aligned} \text{BeFOND inference:} \quad & O(TBdK), \\ \text{BeFOND training step:} \quad & O(TBdK + C_{\mathrm{NGD}} + C_{\mathrm{prior}}), \end{aligned} \tag{69}$$

where C_{NGD} includes the moment products and linear solves used by the decoder update, and C_{prior} is the amortized cost of the separate initial-prior fits. Solver work depends on minibatch rank, conditioning, and numerical tolerance; Appendix B.6 describes the matrix-free evaluation. No gradient is propagated through the T inference steps, avoiding storage of that trajectory for backpropagation.

If a conventional SAE update uses c units of dense matrix-multiplication work, its cost is $O(cBdK)$. With $T = O(c)$, BeFOND's inference contribution has that same order. This asymptotic comparison has not yet translated into comparable training throughput in our implementation (Appendix D.3).

C Extended theoretical analyses, synthetic experiments, and ablations

C.1 Experimental setup

C.1.1 Datasets

SynthSAEBench. We measure dictionary and support recovery using the known generating features of SynthSAEBench (Chanin & Garriga-Alonso, 2026). The full-width experiment uses $d = 768$ and $K = K_\star = 16{,}384$. The superposition experiment fixes $K = 4{,}096$, $K_\star = 16{,}384$, and varies d from 256 to 1,536. Inputs are unwhitened, with parent scaling disabled (`scale_children_by_parent=false`). Feature probabilities follow a Zipf law with exponent 0.5 over $[0.0005, 0.4]$. The generator uses 128 roots, branching factor 4, depth 3, and probability compensation.

C.1.2 Training

BeFOND alternates inference and learning as in Algorithm 1. Table 1 groups the synthetic settings. Prior-update examples are counted separately from decoder-update examples. Geometry arms are separately tuned; their early checkpoints include recipe-matched reproductions.

Table 1: SynthSAEBench training settings. Decoder and prior examples are counted separately. Full-width runs use $d = 768$, $K = K_\star = 16{,}384$; superposition runs use $K = 4{,}096$.

Experiment	Settings
BeFOND best / continuous	Seed 0; batch 2,048; 88,779 updates; 181.82M decoder and 18.18M prior examples; linear LR $0.02 \to 10^{-5}$, no warmup; $\beta = 1.6$
Inference / decoder	Bernoulli; ETD1 $h = \infty$; one inner step; uniform training depth 1–10, test depth 200. Prior Fisher, current moments, no damping; $\Delta t = 1$; bias and variance fixed
Initialization / constraints	Gaussian scale 0.1443375673; prior logits -4.7538364662; soft decoder norm 3.5 with coefficient 100; prior mean cap 3×10^{-4} and floor 10^{-5}
Prior / matrix functions	Prior update every 10 batches on 2,048 separate examples; depth ≤ 200, KL threshold 0.01, rejection threshold 1, window 10. Taylor limit 32 applications; Chebyshev tolerance 10^{-5}, maximum degree 2,048
Original baselines	BatchTopK/JumpReLU/Matryoshka: target $L_0 = 35/45/20$, batch 1,024, LR 3×10^{-4}, 200M decoder examples. MP/ReLU-L1: target $L_0 = 35$, 200,000,512 decoder plus 19M monitoring examples. MF: 20 inference steps in training and testing, batch 1,024, 195,313 updates (200,000,512 examples), cosine schedule with dictionary / other-parameter LR $0.001/0.003$, damping 1 on the first step then 0.2, decoder norm cap 4
Modified baselines	Target $L_0 = 45/40/35$ for BatchTopK/JumpReLU/Matryoshka; MinFire or resampling and learning-rate decay
Training streams	Model seed 0; BeFOND data seed 1; plain amortized baseline data seed 100101; native MP/ReLU-L1 use the global PyTorch and NumPy streams initialized with seed 0, without a separate training-stream seed; MF uses data seed 0
Geometry comparison	Batch 2,048; 100,000 updates; 150,000-step cosine schedule stopped at 100,000; bias and variance fixed. Tuples below are $(\beta, \text{norm}, \text{prior cap}, \text{peak LR})$
N/E; E/E; N/N	$(1.125, 2.5, 0.001, 0.1)$; $(0.8, 3.5, 0.01, 0.1)$; $(1.125, 3.5, 0.0002, 0.006)$. Natural inference: ETD1 $h = \infty$; Euclidean inference: Euler $h = 128$
Superposition / lateral	Baseline comparison: initialization seeds 0, 1, 2. Lateral ablations: seed 0. Batch 1,024. Recipe differences are specified in Section C.3.1

C.1.3 Evaluation

Synthetic recovery. MCC is dictionary mean correlation coefficient. Feature-macro F1 uses positive-cosine one-to-one matching to the generating dictionary; Rare F1 averages over its least

frequent quartile. A latent is dead when it is inactive on every test example. Activity is $r_i > 0.5$ for Bernoulli BeFOND/MF and positive native output for the SAE baselines. Mean L_0 counts active latents per example. Tests use 10^6 examples with seed 49100, including the MF reevaluation reported in Table 4; the independent frequency reference uses 5×10^5 examples with seed 315031. BeFOND/MF use 200/20 test inference steps. Geometry validation uses 32,768 examples, seed 813011, and 100 inference steps.

C.1.4 Implementation

Training horizon. Each optimizer step draws one inference horizon for the whole minibatch, resets the posterior, runs that many outer steps, and updates the decoder only at the final step. The exponential sampler assigns

$$\Pr(t = j) = \frac{e^{-\lambda(j-1)}}{\sum_{s=1}^{t_{\text{outer}}} e^{-\lambda(s-1)}}, \qquad j = 1, \dots, t_{\text{outer}}, \tag{70}$$

where λ is solved numerically so the *truncated discrete* distribution has the configured mean. Probabilities use FP64; Brent's root solver uses tolerance 10^{-14}. At mean 1 the distribution is a point mass at 1; at mean $(t_{\text{outer}} + 1)/2$ it is uniform. Sampling uses NumPy's `default_rng` seeded by the model seed and zero-based optimizer step, identically on every device and after resumption. The draw does not advance the data or latent-sampling generators.

K1 uses a fixed horizon of one and one inner step. Its initial-prior fit and test inference can still use longer horizons. Uniform sampling on 1–10 has mean 5.5, $\Pr(t = 1) = 0.1$, and 95th percentile 10. Exponential sampling on 1–50 with the same mean has $\Pr(t = 1) = 0.181753$ and 95th percentile 15.

Synthetic baseline implementations use SAE Lens 6.51.0 (BatchTopK/JumpReLU/Matryoshka) or 6.37.0 (native MP/ReLU-L1).

C.2 Inference proofs

C.2.1 Single-layer inference

Proposition C.1 (Two-atom case)**.** *Let $z_i \in \{0, 1\}$ with $z_i \sim \text{Bernoulli}(\pi_i)$ independently, with $0 < \pi_i < 1$, and let $\boldsymbol{x} = \boldsymbol{\phi}^{(1)} z_1 + \boldsymbol{\phi}^{(2)} z_2 + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_M)$ independently of $\boldsymbol{z}$. If the dictionary atoms are linearly independent and $W_{12} = (\boldsymbol{\phi}^{(1)})^\top \boldsymbol{\phi}^{(2)} \neq 0$, then no encoder $\hat{z}_1(\boldsymbol{x}) = f(\boldsymbol{w}^\top \boldsymbol{x} + b)$, for any scalar function f, can equal $p(z_1 = 1 \mid \boldsymbol{x})$ for every input $\boldsymbol{x}$.*

Assume independent Bernoulli latents with interior prior probabilities and unit observation covariance. Set

$$a_i(\boldsymbol{x}) = \text{logit}(\pi_i) + (\boldsymbol{\phi}^{(i)})^\top \boldsymbol{x} - \tfrac{1}{2}\|\boldsymbol{\phi}^{(i)}\|^2, \qquad W_{ij} = (\boldsymbol{\phi}^{(i)})^\top \boldsymbol{\phi}^{(j)}.$$

Proof of Proposition C.1. Marginalizing z_2 gives

$$\begin{aligned} \text{logit}\, p(z_1 = 1 \mid \boldsymbol{x}) &= a_1(\boldsymbol{x}) + \log \frac{1 + e^{a_2(\boldsymbol{x}) - W_{12}}}{1 + e^{a_2(\boldsymbol{x})}}, \\ \nabla_{\boldsymbol{x}} \text{logit}\, p(z_1 = 1 \mid \boldsymbol{x}) &= \boldsymbol{\phi}^{(1)} + [\sigma(a_2 - W_{12}) - \sigma(a_2)]\boldsymbol{\phi}^{(2)}. \end{aligned} \tag{71}$$

When $W_{12} \neq 0$, the bracket is nonconstant. Linear independence of the atoms therefore makes the gradient direction vary with $\boldsymbol{x}$. A posterior of the form $f(\boldsymbol{w}^\top \boldsymbol{x} + b)$ is constant along every direction orthogonal to $\boldsymbol{w}$, so its log-odds gradient must be parallel to $\boldsymbol{w}$ everywhere. This is a contradiction, even if f is not assumed differentiable. □

Multiple atoms. By assumption, some atom $\boldsymbol{\phi}^{(i)}$, $i \geq 2$, is neither orthogonal to nor a multiple of $\boldsymbol{\phi}^{(1)}$.

Proof of Proposition 4.1. Suppose $f(\boldsymbol{w}^\top \boldsymbol{x} + b) = p(z_1 = 1 \mid \boldsymbol{x})$ for every $\boldsymbol{x}$, and write $L(\boldsymbol{x}) = \text{logit}\, p(z_1 = 1 \mid \boldsymbol{x})$. As in the proof of Proposition C.1, ∇L is then parallel to $\boldsymbol{w}$ everywhere.

Differentiation of the finite partition functions gives

$$\nabla L = \phi^{(1)} + \sum_{k\geq 2} \Delta_k \phi^{(k)}, \qquad \Delta_k = \mathbb{E}[z_k \mid z_1 = 1, \boldsymbol{x}] - \mathbb{E}[z_k \mid z_1 = 0, \boldsymbol{x}]. \tag{72}$$

Along $\boldsymbol{x} = t\boldsymbol{d}$, with $\boldsymbol{d}$ orthogonal to no atom, the fields of all other bits diverge while their couplings remain bounded. Each bit becomes certain independently of z_1; hence $\Delta_k \to 0$ and $\nabla L \to \phi^{(1)}$. Therefore, $\boldsymbol{w} \parallel \phi^{(1)}$, and $\nabla L^\top \boldsymbol{v} = 0$ everywhere for every $\boldsymbol{v} \perp \phi^{(1)}$.

Let Q contain the atoms parallel to the chosen atom i, writing $\phi^{(j)} = \lambda_j \phi^{(i)}$. Choose $\boldsymbol{v} \perp \phi^{(1)}$ with $(\phi^{(i)})^\top \boldsymbol{v} \neq 0$, and choose $\boldsymbol{d} \perp \phi^{(i)}$ orthogonal to no atom outside Q. Such a $\boldsymbol{d}$ exists because only finitely many proper hyperplanes are excluded. Along $\boldsymbol{x} = \boldsymbol{x}_0 + t\boldsymbol{d}$, all bits outside $Q \cup \{1\}$ freeze, leaving finite conditional partition functions $Z_0(s), Z_1(s)$ with $s = (\phi^{(i)})^\top \boldsymbol{x}_0$. Thus

$$\nabla L^\top \boldsymbol{v} \longrightarrow ((\phi^{(i)})^\top \boldsymbol{v}) D'(s), \qquad D(s) = \log \frac{Z_1(s)}{Z_0(s)}.$$

As s moves from $-\infty$ to $+\infty$, every bit in Q switches according to the sign of λ_j. Since $W_{1j} = \lambda_j W_{1i}$,

$$D(+\infty) - D(-\infty) = -W_{1i} \sum_{j\in Q} |\lambda_j| \neq 0.$$

Therefore D' is nonzero somewhere, contradicting $\nabla L^\top \boldsymbol{v} = 0$ for a single-projection encoder. □

Orthogonality and parallelism of two atoms are polynomial conditions on $\boldsymbol{\Phi}$ that fail for almost every dictionary when $M \geq 2$, so for almost every dictionary with $K \geq 2$ no latent falls under the exception. If no two atoms are parallel, exceptions must be orthogonal to every other atom; for $K > M$, at most $M - 1$ atoms can have that property. Whitening extends both results to any fixed positive-definite $\boldsymbol{\Sigma}$, with $\mathbf{W} = \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi}$.

C.2.2 Iterative inference versus single projection encoders

Proposition C.2 (Fixed-prior inference versus single-projection encoders)**.** *Consider the Bernoulli–Gaussian model of Proposition 4.1 with unit observation covariance. Let*

$$\mathbf{W} = \boldsymbol{\Phi}^\top \boldsymbol{\Phi}, \qquad \boldsymbol{J} = \mathbf{W} - \operatorname{diag}(\mathbf{W}),$$

and suppose that

$$\lambda_{\min}(\boldsymbol{J}) > -4$$

and that some pair of atoms is neither orthogonal nor parallel. Let $\mathcal{A}$ contain all encoders

$$q_f(\boldsymbol{z} \mid \boldsymbol{x}) = \prod_i \operatorname{Bernoulli}\big(z_i;\ f_i(\boldsymbol{w}_i^\top \boldsymbol{x} + b_i)\big)$$

with arbitrary weights, biases, and measurable $f_i : \mathbb{R} \to [0, 1]$, and let

$$R(q) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\mathcal{D}_{\mathrm{KL}}\Big(q(\cdot \mid \boldsymbol{x}) \,\Big\|\, p(\cdot \mid \boldsymbol{x})\Big)\right].$$

Let q_t be the factorized posterior with means

$$\boldsymbol{r}(t) = \sigma(\boldsymbol{u}(t)),$$

where $\boldsymbol{u}(t)$ follows Equation (14) with $\beta = 1$ and the prior fixed at the generating one,

$$\boldsymbol{u}_0 = \operatorname{logit} \boldsymbol{\pi}, \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0.$$

Then the optimal factorized posterior

$$q^*(\cdot \mid \boldsymbol{x}) = \operatorname*{arg\,min}_{q \text{ factorized}} \mathcal{D}_{\mathrm{KL}}\Big(q \,\Big\|\, p(\cdot \mid \boldsymbol{x})\Big)$$

is unique and

$$\lim_{t\to\infty} R(q_t) = R(q^*) < \inf_{q_f \in \mathcal{A}} R(q_f).$$

Consequently, for all sufficiently large finite t,

$$R(q_t) < \inf_{q_f \in \mathcal{A}} R(q_f).$$

Proof. Zero atoms decouple, since their rows of J vanish, so we drop them. Write $a_i(x) = \mathrm{logit}(\pi_i) + (\phi^{(i)})^\top x - \frac{1}{2}W_{ii}$ and $\psi(r) = \sum_i \big[r_i \log r_i + (1-r_i)\log(1-r_i)\big]$ for the negative Bernoulli entropy. For a factorized q with means $r \in [0,1]^K$, using $z_i^2 = z_i$, the divergence $\mathcal{D}_{\mathrm{KL}}\Big(q \,\Big\|\, p(\cdot \mid x)\Big)$ equals, up to a term independent of r,

$$F_x(r) = \psi(r) - a(x)^\top r + \tfrac{1}{2} r^\top J r.$$

Its Hessian $\mathrm{diag}\big(1/(r_i(1-r_i))\big) + J$ is smallest at $r = \frac{1}{2}\mathbf{1}$, where it equals $4\mathbf{I} + J$, so $\nabla^2 F_x \succeq 4m\,\mathbf{I}$ with $m = 1 + \lambda_{\min}(J)/4 > 0$. Hence F_x has a unique minimizer $r^*(x) \in (0,1)^K$, which defines q^* and satisfies $u^* := \mathrm{logit}\, r^* = a - J r^*$.

Convergence. With $\beta = 1$ and $u_0 = \mathrm{logit}\,\pi$, Equation (14) reads $\dot{u} = a - Jr - u = -\nabla F_x(r)$, a smooth, globally Lipschitz flow in u. Let $V(t) = \mathcal{D}_{\mathrm{KL}}\Big(q_{r^*} \,\Big\|\, q_{r(t)}\Big)$, the Bregman divergence of ψ. Then $\dot{V} = -\langle \nabla F_x(r), r - r^*\rangle \le -\big[F_x(r) - F_x(r^*)\big]$ by convexity, while $\frac{d}{dt}F_x(r) = -\nabla F_x^\top \mathrm{diag}\big(r_i(1-r_i)\big)\nabla F_x \le 0$. Hence

$$t\big[F_x(r(t)) - F_x(r^*)\big] \le \int_0^t \big[F_x(r(s)) - F_x(r^*)\big] ds \le V(0) \le \sum_i -\log\min\{\pi_i, 1-\pi_i\}$$

for every x. Since $R(q_t) - R(q^*) = \mathbb{E}\big[F_x(r(t)) - F_x(r^*)\big]$, this gives $R(q_t) \to R(q^*)$ at rate $O(1/t)$.

r_i^* is not a single-projection function. Choose i, j with $W_{ij} \neq 0$ and $\phi^{(i)}, \phi^{(j)}$ not parallel. Implicit differentiation of $\nabla F_x(r^*) = \mathbf{0}$ gives $\partial r^*/\partial x = H_x^{-1}\Phi^\top$ with $H_x = \nabla^2 F_x(r^*) \succeq 4m\,\mathbf{I}$, and $\partial u^*/\partial x = \Phi^\top - J H_x^{-1}\Phi^\top$; both are bounded. Along a ray $x = td$ with d orthogonal to no atom, $u^* = a - Jr^*$ forces every coordinate to saturate, so $\partial r^*/\partial x = \mathrm{diag}\big(r_k^*(1-r_k^*)\big)\,\partial u^*/\partial x \to \mathbf{0}$ and $\nabla_x u_i^* \to \phi^{(i)}$. Now let $Q = \{k : \phi^{(k)} = \lambda_k \phi^{(j)}\}$ and take a ray $x = tv$ with $v \perp \phi^{(j)}$ orthogonal to no atom outside Q. The coordinates outside Q saturate, while r_Q^* converges to the minimizer $s \in (0,1)^Q$ of the reduced problem, which is strongly convex because $\lambda_{\min}(J_{QQ}) \ge \lambda_{\min}(J)$. With $H_Q = \mathrm{diag}\big(1/(s_k(1-s_k))\big) + J_{QQ} \succ 0$, the limiting equations give $\partial r_Q^*/\partial x \to H_Q^{-1}\lambda(\phi^{(j)})^\top$, and since $W_{ik} = \lambda_k W_{ij}$ for $k \in Q$,

$$\nabla_x u_i^*(tv) \longrightarrow \phi^{(i)} - W_{ij}\left(\lambda^\top H_Q^{-1}\lambda\right)\phi^{(j)}.$$

The coefficient is nonzero, so this limit is not parallel to $\phi^{(i)}$. The gradient of u_i^* therefore has no fixed direction, and neither u_i^* nor r_i^* is a function of a single projection.

Positive approximation gap. Let $\delta = \inf_{f,w,b} \mathbb{E}\big[(f(w^\top x + b) - r_i^*(x))^2\big]$ over measurable $f : \mathbb{R} \to [0,1]$. If $\delta = 0$, take $g_n = f_n(w_n^\top x + b_n) \to r_i^*$ in $L^2(p)$, with unit w_n (absorbing scale into f_n; constant functions can take any direction) and, along a subsequence, $w_n \to w$. Since $p(x)$ is continuous and positive, the convergence also holds in L^2_{loc}. Each g_n is invariant along $w_n^\perp$, so $\int g_n\, \partial_{v_n}\psi_0 = 0$ for every test function ψ_0 and $v_n \perp w_n$. Taking $v_n = (\mathbf{I} - w_n w_n^\top)v \to v$ for $v \perp w$ and passing to the limit gives $\int r_i^*\, \partial_v \psi_0 = 0$, so the smooth function r_i^* is constant along $w^\perp$, contradicting the previous paragraph. Hence $\delta > 0$.

Gap in expected divergence. For $q_f \in \mathcal{A}$ with means r_f, $4m$-strong convexity and the optimality of r^* give

$$R(q_f) - R(q^*) = \mathbb{E}\big[F_x(r_f(x)) - F_x(r^*(x))\big] \ge 2m\,\mathbb{E}\|r_f - r^*\|_2^2 \ge 2m\,\delta > 0.$$

Together with $R(q_t) \to R(q^*)$, this proves the claim. □

The fixed-reference specialization of Algorithm 1 discretizes this flow. With $\beta = 1$ and the reference at the prior, its inner step is $\mathrm{logit}\, r \leftarrow \mathrm{logit}\, r - (1 - e^{-h})\nabla F_x(r)$, that is, mirror descent with mirror map ψ. Since $\nabla^2 F_x \preceq \big(1 + \lambda_{\max}(J)/4\big)\nabla^2\psi$, the same conclusion holds after finitely many inner steps, at a geometric rate, whenever $1 - e^{-h} \le 4/(4 + \lambda_{\max}(J))$ (Lu et al., 2018). This discrete guarantee requires both $\beta = 1$ and the stated step-size bound; it does not automatically cover the experimental step sizes.

The condition $\lambda_{\min}(\boldsymbol{J}) > -4$ is exactly the condition for the mean-field objective to be strongly convex, so it has a single optimum that BeFOND approaches; the argument does not rely on solving a hard optimization problem. A width-independent sufficient condition follows for atoms of equal norm c under unit covariance: $\lambda_{\min}(\boldsymbol{J}) \geq -c^2$, with equality when $K > M$, so it reduces to $c < 2$ in units of the noise standard deviation. For stronger coupling the objective can have several local minima, one per competing explanation, and the proposition does not apply.

Rolling-reference inference (Appendix B.1.1) allows the inference budget to be allocated across reference updates. It includes fixed-reference inference by assigning the entire budget to the first inner block and returning its output before refreshing the reference. The proposition establishes a capability within this family; it does not prove the same guarantee for the fixed `t_inner` allocation used in our experiments. With a general positive-definite $\boldsymbol{\Sigma}$, whitening gives the same statement with $\mathbf{W} = \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Phi}$ and $\boldsymbol{J} = \mathbf{W} - \operatorname{diag}(\mathbf{W})$. The strong-convexity condition and discrete step-size bound use this $\boldsymbol{J}$; the sufficient norm bound applies to whitened atoms $\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\phi}^{(i)}$.

C.2.3 Rain–sprinkler example

Fig. 2 illustrates the two-atom case with independent rain and sprinkler latents, each with prior probability 0.1. The coordinates of $\boldsymbol{x}$ are lawn and roof wetness, with

$$\boldsymbol{\Phi} = \begin{pmatrix} 2 & 2 \\ 3 & 0 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = (1.05)^2 I_2.$$

Rain contributes to both coordinates, while the sprinkler contributes only to lawn wetness. The atoms are neither orthogonal nor parallel, so Proposition 4.1 applies after whitening. Moreover, $\lambda_{\min}(\boldsymbol{J}) = -4/1.05^2 > -4$, satisfying Proposition C.2.

The exact posterior is computed by enumerating the four latent configurations.

Fixed-prior inference. The BeFOND panel shows the unique optimal factorized Bernoulli posterior. We solve the stationary equations from Proposition C.2 by fixed-point iteration, initialized at $\boldsymbol{r} = (0.1, 0.1)$, with $\beta = 1$ and the generating prior held fixed. The update map has contraction factor $1/1.05^2 < 1$; iteration continues until the fixed-point residual divided by $1 - 1/1.05^2$ is below 2×10^{-13}. Thus the panel represents the converged fixed-prior inference dynamics of eq. (14), with no decoder learning or reference refresh.

Single-projection encoder. We numerically fit $r_i(\boldsymbol{x}) = f_i(\boldsymbol{w}_i^\top \boldsymbol{x})$ to minimize the same model-averaged reverse KL to the exact posterior. Each unit direction is parameterized by an angle. A 12×12 angular grid over $[0, \pi)^2$ supplies five starting points for local refinement. For each direction pair, the scalar response functions are solved from the conditional stationarity equations of this KL objective and represented by piecewise-linear probabilities. Final fits use 2,001 grid points per function, extending eight noise standard deviations beyond the projected component means, with constant extrapolation. Conditional expectations use 160-point Gauss–Hermite quadrature, and the final expected KL uses 192 points per input coordinate and mixture component. The fitted rain and sprinkler directions are $65.80°$ and $-20.02°$, respectively, giving expected KL 0.021554 nats.

The white point has roof wetness 1.5, halfway between the dry and wet conditional means, and therefore equal roof likelihoods with and without rain. Raising roof wetness to 3 while holding lawn wetness fixed strengthens rain and suppresses the sprinkler. Table 2 reports the exact and factorized probabilities at these points.

Table 2: Posterior activation probabilities at the marked observations in Figure 2, rounded to six decimal places. Neutral and wet correspond to $\boldsymbol{x} = (2, 1.5)^\top$ and $\boldsymbol{x} = (2, 3)^\top$, respectively.

	Neutral roof		Wet roof	
Method	Rain	Sprinkler	Rain	Sprinkler
Exact posterior	0.292138	0.292138	0.960707	0.033017
BeFOND	0.228996	0.228996	0.974111	0.019506

C.3 Inference and learning ablations

C.3.1 Inference ablation: lateral interactions

Figures 3 and 6 use $K = 4{,}096$. The baseline comparison uses initialization seeds 0, 1, and 2, with individual results in Table 3. The lateral ablations use seed 0. Metric definitions are in Appendix C.1.3. The lateral ablations use matched learning rates, β, and decoder norms. Their full BeFOND control differs from the selected BeFOND recipe in the baseline comparison at $d = 512, 384, 256$: the ablations use $(\mathrm{LR}, \beta) = (0.001, 1), (0.001, 1), (0.001, 2)$, whereas the baseline comparison uses $(0.0003, 0.5), (0.0003, 1), (0.0003, 1.5)$. Excitatory-only runs at $d = 1{,}024$ and $1{,}536$ stopped early. At $d = 256$, suppressive interactions raise MCC from 0.2810 to 0.6862 within the matched ablation recipes. Excitatory-only inference activates almost all 4,096 latents but reaches MCC 0.2825. Suppressive-only inference has one dead latent at $d = 1{,}536$ and zero at the other dimensions; both full and zero-off-diagonal inference have zero throughout. Thus, activity alone does not explain the recovery differences.

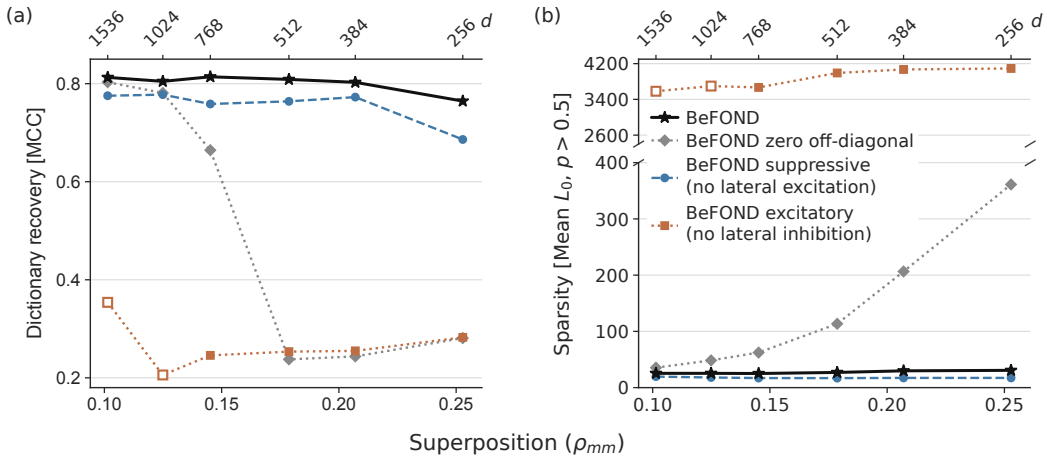


Figure 6: Lateral suppression supports feature recovery under superposition. Hollow markers denote early-stopped excitatory runs ($K = 4{,}096$).

C.3.2 Learning ablation: inference and learning geometry

The Euclidean ablations also reveal a substantial gain in training efficiency (Fig. 7). Natural inference and learning (N/N) reach MCC 0.500 and Rare F1 0.280 after just 1,000 updates, exceeding both Euclidean variants after 100,000 updates on their respectively tuned runs.

Table 3: Individual-seed results for Figure 3 ($K = 4{,}096$, $K_\star = 16{,}384$). MCC values retain the recorded precision. Dead latents are counted on 10^6 held-out examples using each model's native activity rule ($r_i > 0.5$ for BeFOND and MF).

		MCC			Dead latents		
Method	d	Seed 0	Seed 1	Seed 2	0	1	2
BatchTopK	1536	0.6447949409484863	0.6418958902359009	0.6367465853691101	178	217	206
	1024	0.7172077894210815	0.7101552486419678	0.715070366859436	78	80	66
	768	0.7237797379493713	0.7247610688209534	0.7268125414848328	49	47	39
	512	0.6935576796531677	0.6925675868988037	0.6897946000099182	37	38	55
	384	0.6238665580749512	0.6256771087646484	0.6227874159812927	56	82	58
	256	0.4887199401855469	0.48809993267059326	0.4880616068840027	152	176	157
JumpReLU	1536	0.7402234673500061	0.7397291660308838	0.7378072738647461	0	0	0
	1024	0.741813063621521	0.739402711391449	0.7388435006141663	3	2	2
	768	0.7435916662216187	0.7454241514205933	0.7452358603477478	2	0	1
	512	0.7406731843948364	0.7403227090835571	0.7406840324401855	3	2	1
	384	0.7123968005180359	0.7126669883728027	0.712004542350769	78	90	96
	256	0.5576815605163574	0.55924391746521	0.5596640110015869	1249	1232	1224
Matryoshka	1536	0.6665480136871338	0.6840691566467285	0.6623464822769165	266	223	279
	1024	0.7525583505630493	0.7481846213340759	0.7474654912948608	88	86	85
	768	0.7555773258209229	0.7561160326004028	0.7561805248260498	54	51	49
	512	0.6980507969856262	0.6989551186561584	0.6928752660751343	54	87	84
	384	0.6256262063980103	0.6234501600265503	0.6201913356781006	93	87	85
	256	0.5091139078140259	0.5098989605903625	0.5087519288063049	170	153	144
MP-SAE	1536	0.7041427493095398	0.7060015201568604	0.7078589200973511	0	0	0
	1024	0.7029024362564087	0.7033072113990784	0.6972198486328125	0	0	0
	768	0.7092912197113037	0.7096635699272156	0.705768346786499	0	0	0
	512	0.7090048789978027	0.7061803340911865	0.7071677446365356	0	0	0
	384	0.6933020353317261	0.6994995474815369	0.6963708400726318	0	0	0
	256	0.6802412867546082	0.679160475730896	0.6853885054588318	0	0	0
ReLU-L1	1536	0.597491443157196	0.5992552042007446	0.598943829536438	0	0	0
	1024	0.59971684217453	0.6019572019577026	0.6006413698196411	0	0	0
	768	0.6030566692352295	0.6011125445365906	0.6026996374130249	0	0	0
	512	0.5990475416183472	0.5983516573905945	0.5974198579788208	0	0	0
	384	0.5921348333358765	0.5912116765975952	0.590924084186554	0	0	0
	256	0.5765519738197327	0.578854224334717	0.5773016214370728	0	0	0
MF	1536	0.6974799036979675	0.7041385769844055	0.584987998008728	0	0	291
	1024	0.7075880169868469	0.7071225643157959	0.7132526636123657	3	9	2
	768	0.7050071358680725	0.7113231420516968	0.7108327746391296	8	12	11
	512	0.719290018081665	0.7244423627853394	0.7237823009490967	10	3	7
	384	0.7211331129074097	0.7238309979438782	0.7178806662559509	14	5	20
	256	0.46739497780799866	0.4664973020553589	0.47486913204193115	1645	1655	1664
BeFOND	1536	0.8128377795219421	0.8148884773254395	0.8158183097839355	0	0	0
	1024	0.8047415018081665	0.8055942058563232	0.8029000759124756	0	0	0
	768	0.8140117526054382	0.8091920614242554	0.8121296167373657	0	0	0
	512	0.812289297580719	0.8088412880897522	0.8094125986099243	0	0	0
	384	0.799635648727417	0.8007169365882874	0.7975541353225708	0	0	0
	256	0.7783401012420654	0.7805794477462769	0.7804576754570007	0	0	0

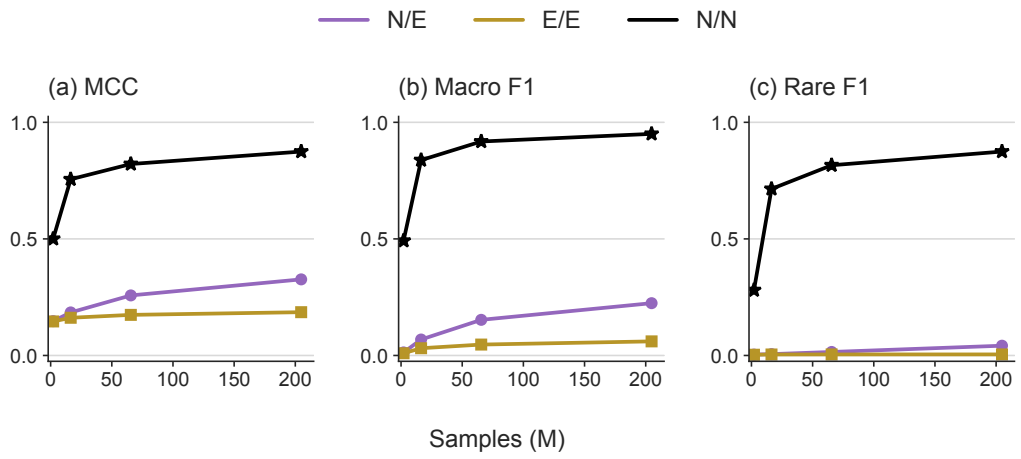


Figure 7: The natural-inference/natural-learning recipe reaches higher recovery with fewer samples. Panels show MCC, Macro F1, and Rare F1 on SynthSAEBench ($d = 768$, $K = K_\star = 16{,}384$). N/E denotes inference/learning geometry: natural (N) or Euclidean (E). Recipes are tuned separately (Appendix C.1.2).

The natural-inference/natural-learning recipe reaches stronger recovery in fewer decoder updates. The recipes also differ in learning rate, β, decoder norms, prior caps, and integration settings (Table 1), so this comparison does not isolate geometry's causal contribution. Theorem 4.2 predicts equal contraction for frozen, matched second moments. Moment matching was not measured in these runs; Appendix C.4 gives the proof and unequal-moment extension.

C.3.3 Rolling prior

Fig. 8 varies β and the cap on the mean initial prior in 72 completed fits: six values of each hyperparameter and seeds 0 and 1. Increasing β reduces final activity in this sweep, but also reduces recovery at the larger values. Several looser caps produce overlapping curves because the learned prior lies below those caps. The initial cap therefore does not directly specify the final L_0, consistent with the rolling-reference dynamics in Appendix B.5. All fits use unwhitened inputs, batch size

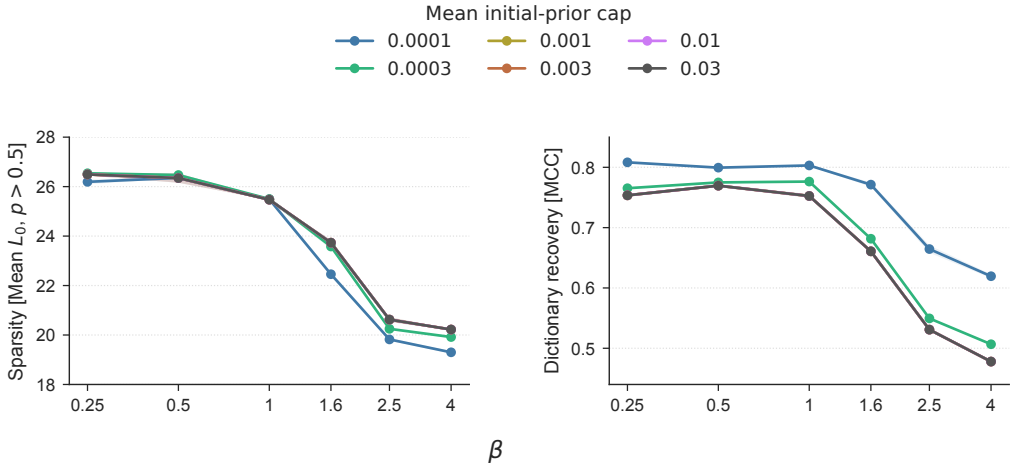


Figure 8: Sparsity and recovery versus β on SynthSAEBench ($d = 768$, $K = 4{,}096$, $K_\star = 16{,}384$). Colors give the mean initial-prior cap; lines show the mean over two seeds and shading their range. Each fit uses 12,000 updates and is evaluated on 8,192 examples after 5,000 inference steps.

1,024, a learning rate decaying from 0.001 to 10^{-5}, one inner step with $h = \infty$, and an exponential training horizon with mean 5.5 capped at 50. The decoder starts at norm 2 and is clipped at norm 4; the covariance is fixed, decoder bias is learned, and Fisher damping is 10^{-5}. Initial-prior fits use 100 inference steps on 1,024 examples every 500 updates, with no positive prior floor. Total exposure is 12.288M main plus 24,576 prior examples per fit.

C.4 Exact and approximate rarity compensation

Fix a training minibatch and hold its inferred distributions, prior references, and covariance $\boldsymbol{\Sigma} \succ 0$ fixed during the decoder update. Throughout this subsection, $\mathbf{E} \succ 0$ and $\mathbf{E}_p \succ 0$ denote the minibatch averages of the posterior and prior second moments in Eq. (44), and $\mathcal{F}$ denotes the resulting minibatch-averaged free energy. In Fisher-preconditioned decoder updates, the matrix gradient is implicitly vectorized columnwise before multiplication by $\mathbf{G}_\Phi^{-1}$, and the result is reshaped to the decoder matrix.

Proof of Theorem 4.2. For the unit-covariance likelihood in the theorem, the decoder-dependent terms are

$$\mathcal{F}(\boldsymbol{\Phi}) = \tfrac{1}{2}\mathbb{E}_{\text{batch}}\mathbb{E}_q[\|\boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{z}\|_2^2] + \text{const.}$$

The posterior and prior references are fixed, so the KL term is constant in $\boldsymbol{\Phi}$. Expanding the quadratic gives

$$\begin{aligned} \nabla_{\boldsymbol{\Phi}}\mathcal{F} &= \boldsymbol{\Phi}\mathbf{E} - \mathbb{E}_{\text{batch}}[\boldsymbol{x}\boldsymbol{r}^\top], \\ \nabla^2_{\text{vec}(\boldsymbol{\Phi})}\mathcal{F} &= \mathbf{E} \otimes \mathbf{I} \succ 0. \end{aligned} \tag{73}$$

Hence the minimizer is unique, $\boldsymbol{\Phi}^* = \mathbb{E}_{\text{batch}}[\boldsymbol{x}\boldsymbol{r}^\top]\mathbf{E}^{-1}$, and $\nabla_{\boldsymbol{\Phi}}\mathcal{F} = (\boldsymbol{\Phi} - \boldsymbol{\Phi}^*)\mathbf{E}$.

For a model draw $\tilde{\boldsymbol{x}} \mid \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\Phi}\boldsymbol{z}, \mathbf{I})$, the decoder score is

$$\nabla_{\boldsymbol{\Phi}} \log p_{\boldsymbol{\Phi}}(\tilde{\boldsymbol{x}} \mid \boldsymbol{z}) = (\tilde{\boldsymbol{x}} - \boldsymbol{\Phi}\boldsymbol{z})\boldsymbol{z}^\top.$$

The conditional residual has second moment $\mathbf{I}$. Averaging the vectorized score's outer product over these model draws and the fixed prior references therefore gives $\mathbf{G}_\Phi = \mathbf{E}_p \otimes \mathbf{I}$. Its inverse acts on a matrix gradient by right multiplication with $\mathbf{E}_p^{-1}$. Thus, when $\mathbf{E}_p = \mathbf{E}$,

$$\begin{aligned} \text{Euclidean:} \quad & \frac{d}{d\tau}(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*) = -\eta_\Phi(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*)\mathbf{E}, \\ \text{Natural:} \quad & \frac{d}{d\tau}(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*) = -\eta_\Phi(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*). \end{aligned}$$

Integrating the natural flow yields

$$\boldsymbol{\Phi}(\Delta t) - \boldsymbol{\Phi}^* = e^{-\eta_\Phi \Delta t}(\boldsymbol{\Phi}(0) - \boldsymbol{\Phi}^*). \tag{74}$$

Matching is required only between the minibatch-averaged second moments, not separately for every example.

To compare relaxation times, let $\mathbf{E}\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j$. Since $\mathbf{E} \succ 0$, its eigenvectors form an orthonormal basis and $\lambda_j > 0$. Along the Euclidean trajectory,

$$(\boldsymbol{\Phi}(\Delta t) - \boldsymbol{\Phi}^*)\boldsymbol{v}_j = e^{-\eta_\Phi \lambda_j \Delta t}(\boldsymbol{\Phi}(0) - \boldsymbol{\Phi}^*)\boldsymbol{v}_j.$$

Its modal relaxation times are $1/(\eta_\Phi \lambda_j)$, so the ratio of the slowest to the fastest is $\lambda_{\max}(\mathbf{E})/\lambda_{\min}(\mathbf{E}) = \kappa(\mathbf{E})$. Natural flow instead has relaxation time $1/\eta_\Phi$ in every direction. Since $z_i^2 = z_i$ for Bernoulli latents,

$$\rho_i = \mathbf{E}_{ii} = \mathbb{E}_{\text{batch}}\mathbb{E}_q[z_i] = \mathbb{E}_{\text{batch}}[r_i].$$

The Rayleigh quotient bounds each diagonal entry between the smallest and largest eigenvalues, giving

$$\lambda_{\max}(\mathbf{E}) \geq \rho_{\max}, \qquad \lambda_{\min}(\mathbf{E}) \leq \rho_{\min}, \qquad \kappa(\mathbf{E}) \geq \frac{\rho_{\max}}{\rho_{\min}}.$$

For one feature with known activity, $\mathbf{E} = p$, recovering the Euclidean rate $\eta_\Phi p$ and natural rate η_Φ used in the scalar example. □

Corollary C.3 (Power-law feature frequencies)**.** *Under Theorem 4.2, suppose the averaged posterior activation probabilities satisfy $\rho_i = c\, i^{-\alpha}$ for $i = 1, \dots, K$, with $c > 0$ and $\alpha > 0$. The ratio of the slowest to the fastest Euclidean relaxation time is at least K^α; for the matched natural-gradient flow it is one.*

Proof. The largest and smallest frequencies are c and $cK^{-\alpha}$. Substituting their ratio into the theorem gives $\kappa(\mathbf{E}) \geq K^\alpha$. Equation (74) gives the common natural-gradient relaxation time. □

The result equalizes local contraction rates. It does not equalize the arrival rate of rare examples or prove global dictionary recovery.

General observation covariance. For fixed $\boldsymbol{\Sigma} \succ 0$, the corresponding gradient, Hessian, and decoder Fisher are

$$\nabla_{\boldsymbol{\Phi}} \mathcal{F} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*)\mathbf{E},$$
$$\nabla^2_{\mathrm{vec}(\boldsymbol{\Phi})} \mathcal{F} = \mathbf{E} \otimes \boldsymbol{\Sigma}^{-1}, \qquad \mathbf{G}_{\boldsymbol{\Phi}} = \mathbf{E}_p \otimes \boldsymbol{\Sigma}^{-1}. \tag{75}$$

The optimum is unchanged. Euclidean flow now obeys $\dot{\boldsymbol{\Phi}} = -\eta_{\boldsymbol{\Phi}} \boldsymbol{\Sigma}^{-1}(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*)\mathbf{E}$; its Hessian has condition number $\kappa(\mathbf{E})\kappa(\boldsymbol{\Sigma})$ because the Kronecker eigenvalues are products. Natural preconditioning cancels the observation precision and gives

$$\frac{d}{d\tau}(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*) = -\eta_{\boldsymbol{\Phi}}(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*)\mathbf{E}\mathbf{E}_p^{-1}.$$

The coefficients are fixed during the update, hence

$$\boldsymbol{\Phi}(\Delta t) - \boldsymbol{\Phi}^* = (\boldsymbol{\Phi}(0) - \boldsymbol{\Phi}^*) \exp[-\eta_{\boldsymbol{\Phi}} \Delta t\, \mathbf{E}\mathbf{E}_p^{-1}]. \tag{76}$$

When $\mathbf{E}_p = \mathbf{E}$, this again reduces to Equation (74), independently of the fixed observation covariance. The following extension treats unequal moments.

Proposition C.4 (Rarity compensation with unequal moments)**.** *Under the same frozen-substep assumptions, let $\mathbf{E} \succ 0$, $\mathbf{E}_p \succ 0$, and $\eta_{\boldsymbol{\Phi}} > 0$, without requiring $\mathbf{E} = \mathbf{E}_p$. The decoder error obeys Eq. (76). Define*

$$\lambda_* := \lambda_{\min}\left(\mathbf{E}_p^{-1/2}\mathbf{E}\mathbf{E}_p^{-1/2}\right) > 0.$$

Then, for $\Delta t \geq 0$,

$$\|(\boldsymbol{\Phi}(\Delta t) - \boldsymbol{\Phi}^*)\mathbf{E}_p^{1/2}\|_F \leq e^{-\eta_{\boldsymbol{\Phi}}\lambda_* \Delta t}\|(\boldsymbol{\Phi}(0) - \boldsymbol{\Phi}^*)\mathbf{E}_p^{1/2}\|_F. \tag{77}$$

In particular, if

$$(1 - \epsilon)\mathbf{E}_p \preceq \mathbf{E} \preceq (1 + \epsilon)\mathbf{E}_p, \qquad 0 \leq \epsilon < 1,$$

then

$$\|(\boldsymbol{\Phi}(\Delta t) - \boldsymbol{\Phi}^*)\mathbf{E}_p^{1/2}\|_F \leq e^{-\eta_{\boldsymbol{\Phi}}(1-\epsilon)\Delta t}\|(\boldsymbol{\Phi}(0) - \boldsymbol{\Phi}^*)\mathbf{E}_p^{1/2}\|_F. \tag{78}$$

Proof. Equation (76) follows from the quadratic decoder gradient derived above and does not require moment matching. Multiplying its error dynamics on the right by $\mathbf{E}_p^{1/2}$ gives

$$\frac{d}{d\tau}\left[(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*)\mathbf{E}_p^{1/2}\right] = -\eta_{\boldsymbol{\Phi}}\left[(\boldsymbol{\Phi} - \boldsymbol{\Phi}^*)\mathbf{E}_p^{1/2}\right]\left(\mathbf{E}_p^{-1/2}\mathbf{E}\mathbf{E}_p^{-1/2}\right).$$

The matrix in the final parentheses is symmetric positive definite, so its decay exponential has spectral norm $e^{-\eta_{\boldsymbol{\Phi}}\lambda_* \Delta t}$. Applying $\|AB\|_F \leq \|A\|_F\|B\|_2$ proves Eq. (77). Under the stated moment bounds, this matrix has eigenvalues in $[1 - \epsilon, 1 + \epsilon]$, hence $\lambda_* \geq 1 - \epsilon$, which proves Eq. (78). □

C.4.1 ACTIVITY-NORMALIZED INFERENCE

Theorem C.5 (Activity-normalized inference). *Fix the observation, decoder, covariance, and current prior reference during inference. For the factorized Bernoulli posterior, let $r_i = \sigma(u_i) \in (0,1)$. At the same posterior state, the unit-rate Euclidean and natural-gradient flows in membrane-potential (logit) coordinates satisfy*

$$\boxed{\dot{u}_i^{\mathrm{NGD}} = -\frac{1}{r_i(1-r_i)}\frac{\partial \mathcal{F}}{\partial u_i}, \qquad \dot{u}_i^{\mathrm{GD}} = r_i(1-r_i)\,\dot{u}_i^{\mathrm{NGD}}.}$$

Thus natural-gradient inference removes the multiplicative Bernoulli variance factor from each membrane-potential update. This identity requires no decoder moment-matching assumption.

Proof of Theorem C.5. For factorized Bernoulli latents, $r_i = \sigma(u_i) \in (0,1)$, so

$$\frac{\partial r_i}{\partial u_i} = r_i(1-r_i), \qquad \mathbf{C} = \mathrm{diag}[\boldsymbol{r} \odot (\mathbf{1} - \boldsymbol{r})].$$

The score is $\partial \log q_{\boldsymbol{u}}(\boldsymbol{z} \mid \boldsymbol{x})/\partial u_i = z_i - r_i$. Independence therefore gives the posterior Fisher $\boldsymbol{G}(\boldsymbol{u}) = \mathbf{C}$. The chain rule yields $\nabla_{\boldsymbol{u}}\mathcal{F} = \mathbf{C}\nabla_{\boldsymbol{r}}\mathcal{F}$. Natural preconditioning cancels this factor:

$$\dot{u}_i^{\mathrm{NGD}} = -\frac{1}{r_i(1-r_i)}\frac{\partial \mathcal{F}}{\partial u_i} = -\frac{\partial \mathcal{F}}{\partial r_i}, \qquad \dot{u}_i^{\mathrm{GD}} = r_i(1-r_i)\dot{u}_i^{\mathrm{NGD}}.$$

□

When the inference drive favors activation, this cancellation removes an additional suppression of weakly active latents' logit updates. It does not guarantee reactivation or uniform convergence of activation probabilities: they still obey $\dot{r}_i = r_i(1-r_i)\dot{u}_i$.

C.4.2 LOCAL CONDITIONING UNDER ADAM NORMALIZATION

Adam normalizes updates using coordinatewise estimates of gradient second moments (Kingma & Ba, 2015; Balles & Hennig, 2018). The following calculation isolates the effect of this normalization on decoder conditioning.

Proposition C.6 (Local conditioning under frozen Adam normalization). *Let $\boldsymbol{x} = \boldsymbol{\Phi}_* \boldsymbol{z} + \boldsymbol{\xi}$, where the known codes have independent coordinates $z_i \sim \mathrm{Bernoulli}(\rho_i)$ with $0 < \rho_i < 1$, and $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ is independent of $\boldsymbol{z}$. For the unconstrained population loss*

$$L(\boldsymbol{\Phi}) = \tfrac{1}{2}\mathbb{E}\|\boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{z}\|_2^2, \qquad \mathbf{E} = \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top], \qquad \boldsymbol{H} = \mathbf{E} \otimes \mathbf{I}_d,$$

consider gradients averaged over B independent samples. Freeze the Adam denominator at the square root of its expected second-moment target at $\boldsymbol{\Phi}_*$, with stabilizer $\epsilon > 0$. The resulting diagonal preconditioner is

$$\mathbf{D}_\epsilon = \mathrm{diag}_{i=1}^{K}\left[(\epsilon + \sqrt{\rho_i/B})^{-1}\right] \otimes \mathbf{I}_d.$$

Writing $\mathbf{A}_\epsilon = \mathbf{D}_\epsilon^{1/2}\boldsymbol{H}\mathbf{D}_\epsilon^{1/2}$, its condition number satisfies

$$\kappa(\mathbf{A}_\epsilon) \geq \frac{\rho_{\max}/(\epsilon + \sqrt{\rho_{\max}/B})}{\rho_{\min}/(\epsilon + \sqrt{\rho_{\min}/B})} \geq \sqrt{\frac{\rho_{\max}}{\rho_{\min}}}. \tag{79}$$

Thus $\rho_i = c\,i^{-\alpha}$, with $0 < c < 1$ and $\alpha > 0$, gives $\kappa(\mathbf{A}_\epsilon) \geq K^{\alpha/2}$. Matched complete-data Fisher preconditioning instead gives $\boldsymbol{H}^{-1/2}\boldsymbol{H}\boldsymbol{H}^{-1/2} = \mathbf{I}$ and condition number one.

Proof. The Hessian is $\boldsymbol{H}$, and $\mathbf{E} = \mathrm{diag}(\rho_i(1-\rho_i)) + \boldsymbol{\rho}\boldsymbol{\rho}^\top \succ 0$. At the optimum, a single-sample gradient is $-\boldsymbol{\xi}\boldsymbol{z}^\top$, with mean zero and vectorized second moment $\boldsymbol{H}$. Independence gives second moment $\boldsymbol{H}/B$ for the averaged gradient, yielding $\mathbf{D}_\epsilon$ above. Each diagonal entry of $\mathbf{A}_\epsilon$ is $a_i = \rho_i/(\epsilon + \sqrt{\rho_i/B})$, repeated d times. Rayleigh quotients give $\kappa(\mathbf{A}_\epsilon) \geq \max_i a_i / \min_i a_i$. Since a_i increases with ρ_i, the first bound follows. For $s = \sqrt{\rho_{\max}/\rho_{\min}} \geq 1$ and $t = \sqrt{\rho_{\min}/B}$, the ratio is $s^2(\epsilon + t)/(\epsilon + st) \geq s$. Substitution gives the width bound. The complete-data Fisher equals $\boldsymbol{H}$ for this model, proving the final claim. □

This is a known-code calculation with Adam's second-moment normalization frozen at its expected target, before applying the square root. It is not a convergence theorem for full Adam or the SAE baselines: momentum and fluctuating moment estimates (Kingma & Ba, 2015), changing inferred codes, encoder learning, and auxiliary losses are absent. Gradient variance affects Adam's normalization (Balles & Hennig, 2018), whereas the model Fisher measures distributional sensitivity (Martens, 2020). The bound concerns local conditioning and does not establish a training-time or sample-efficiency separation. Appendix B.8 discusses the computational cost of the implemented Fisher updates.

C.5 Learning and rare-feature recovery

C.5.1 Resampling and auxiliary-loss variants

Resampling and auxiliary losses aim to revive inactive features (Bricken et al., 2023; Jermyn & Templeton, 2024). Fig. 9 and Table 4 compare the original baselines with modified recipes using MinFire (BatchTopK), resampling (JumpReLU and Matryoshka), learning-rate decay, and retuned sparsity. These modifications substantially improve baseline recovery, but BeFOND retains the highest MCC, Macro F1, and Rare F1. The comparison evaluates complete recipes with different target sparsities; measured L_0 is reported in the table.

The modified recipes follow the released full-width sweep (chanind-goodfire, 2026), using seed 0, 200M samples, batch size 1,024, Adam with learning rate 3×10^{-4} and moments $(0.9, 0.999)$, BF16 autocast, gradient-norm clipping at 1, and learning-rate decay during the final 20% of training. Target L_0 is 45 for BatchTopK, 40 for JumpReLU, and 35 for Matryoshka.

MinFire (the sweep implementation's `MinFireGate` in `gates.py`) reserves each latent's strongest positive activation before filling the remaining batch-wide TopK budget. The selected BatchTopK recipe applies this gate every batch for the first 90% of training, then uses ordinary BatchTopK. JumpReLU uses an input straight-through estimator and a base L_0 coefficient of 0.1 with an adaptive multiplier. JumpReLU and Matryoshka resample every 2,500 updates during the first 80% of training, with inactivity windows of 1,000 and 100 updates, respectively. Resampled decoder atoms follow reconstruction-residual directions, and their optimizer moments are reset. BatchTopK and Matryoshka retain auxiliary-loss coefficient 1; Matryoshka uses prefixes 128, 512, 2,048, and 16,384 with per-level auxiliary losses. Gemma Scope uses reconstruction and L_0 sparsity losses, without a separate dead-latent auxiliary loss, alongside straight-through estimators, tied initialization, learning-rate and sparsity warmup, and decoder normalization (Lieberum et al., 2024, Sections 2.2 and 3.2).

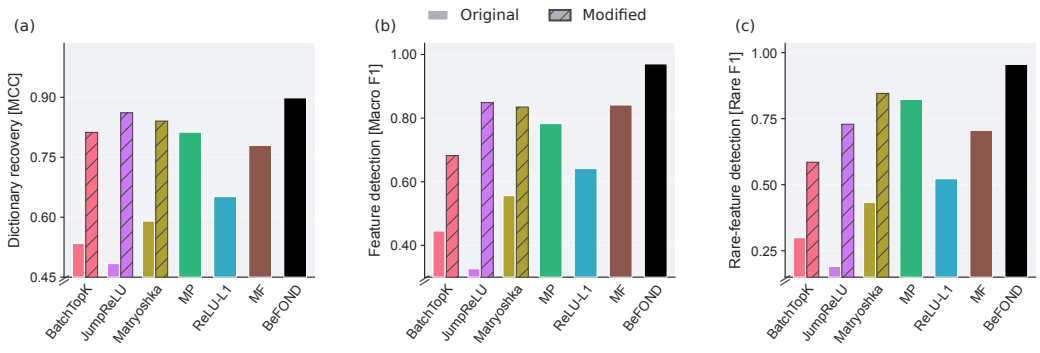


Figure 9: BeFOND retains the highest MCC, Macro F1, and Rare F1 after baseline modifications aimed at preventing inactive features. Hatched bars denote modified recipes ($d = 768$, $K = K_\star = 16{,}384$, seed 0); numerical results appear in Table 4.

Recovery across feature frequencies. Fig. 10 resolves the comparison into frequency quintiles. The BeFOND checkpoint is the best run in Table 4 (MCC 0.8986), and the original and modified baseline checkpoints are those in Fig. 9. Improvements from the modified recipes extend into the rare tail, while BeFOND retains higher coverage and F1 in every quintile. Coverage measures dictionary

Table 4: Full-width SynthSAEBench results ($d = 768$, $K = K_\star = 16{,}384$). Individual rows report initialization seeds 0–2; mean rows give pointwise 95% Student-t confidence intervals for the three recovery metrics, as in Figure 4. Each evaluation uses 10^6 held-out examples. L_0 and dead counts in mean rows are arithmetic means. The lower block retains the modified recipes (+) and continuous-run checkpoint, each at seed 0. The BeFOND runs use 181.82M decoder-update plus 18.18M prior examples.

Method	Seed	MCC ↑	Macro F1 ↑	Rare F1 ↑	L_0	Dead ↓
BatchTopK	0	0.534523	0.445173	0.299436	34.833798	1,577
	1	0.535311	0.444452	0.299666	35.031285	1,733
	2	0.537442	0.446642	0.296203	34.948331	1,707
	Mean	0.5358 ± 0.0038	0.4454 ± 0.0028	0.2984 ± 0.0048	34.94	1672.33
JumpReLU	0	0.483953	0.326567	0.190439	45.148172	8,069
	1	0.481728	0.326672	0.184864	44.826161	8,090
	2	0.482102	0.325518	0.184310	45.098772	8,117
	Mean	0.4826 ± 0.0030	0.3263 ± 0.0016	0.1865 ± 0.0084	45.02	8092.00
Matryoshka	0	0.590345	0.556314	0.433018	19.978349	4,923
	1	0.574629	0.530946	0.366127	19.991146	5,505
	2	0.583862	0.552021	0.415913	19.989287	5,049
	Mean	0.5829 ± 0.0196	0.5464 ± 0.0337	0.4050 ± 0.0863	19.99	5159.00
MP-SAE	0	0.812752	0.782551	0.822595	29.646241	2
	1	0.811610	0.781264	0.821854	29.648383	1
	2	0.815241	0.784698	0.827665	29.688899	3
	Mean	0.8132 ± 0.0046	0.7828 ± 0.0043	0.8240 ± 0.0079	29.66	2.00
ReLU-L1	0	0.651464	0.641205	0.522534	35.089073	174
	1	0.650650	0.640259	0.524647	34.972258	135
	2	0.650945	0.641409	0.524073	34.976428	137
	Mean	0.6510 ± 0.0010	0.6410 ± 0.0015	0.5238 ± 0.0027	35.01	148.67
MF	0	0.779715	0.841050	0.704307	45.107146	514
	1	0.783435	0.846995	0.743286	48.116016	640
	2	0.790151	0.855224	0.729628	68.019993	556
	Mean	0.7844 ± 0.0131	0.8478 ± 0.0177	0.7257 ± 0.0491	53.75	570.00
BeFOND	0	0.898559	0.970345	0.955392	36.033549	1
	1	0.898064	0.970567	0.949700	35.679190	0
	2	0.897350	0.969745	0.950726	35.485085	0
	Mean	$\mathbf{0.8980} \pm 0.0015$	$\mathbf{0.9702} \pm 0.0011$	$\mathbf{0.9519} \pm 0.0075$	35.73	0.33
Additional seed-0 checkpoints						
BatchTopK+	0	0.8123	0.6826	0.5851	45.01	504
JumpReLU+	0	0.8612	0.8487	0.7288	39.96	367
Matryoshka+	0	0.8405	0.8348	0.8457	34.95	495
BeFOND	0	0.8974	0.9701	0.9482	35.70	0

alignment; F1 measures detection of the matched feature's activity, so the two need not coincide. These fifths differ from Rare F1's least-frequent quarter.

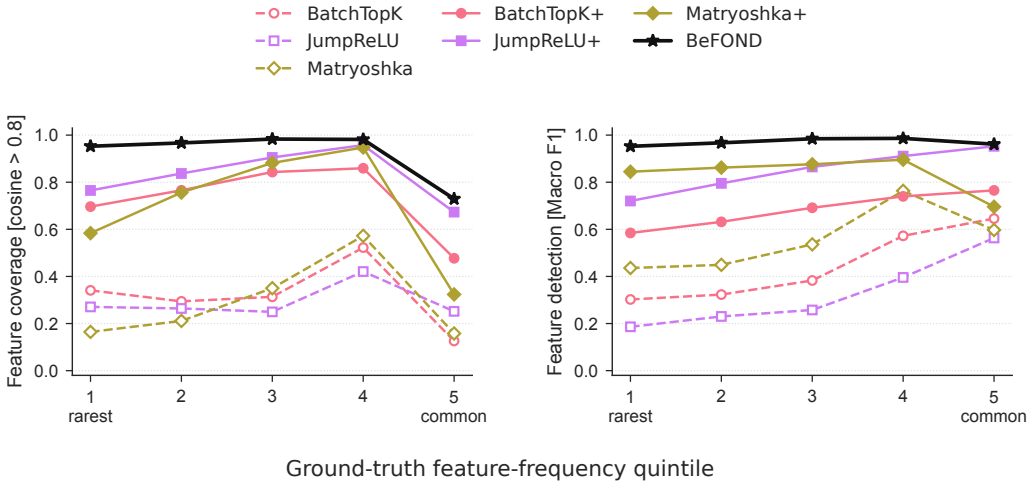


Figure 10: Recovery across feature-frequency quintiles ($d = 768$, $K = K_\star = 16{,}384$, seed 0). Dashed lines show original baselines; "+" denotes their modified recipes. Coverage is the fraction of features matched above cosine 0.8. Quintiles use each run's 500K-example frequency reference; F1 uses 1M held-out examples. Random draws differ between some runs.

Dictionary capacity. Table 4 combines recovery, sparsity, and dead-latent counts. Main-text Fig. 4 reports three-seed means: MCC 0.8980 and Rare F1 0.9519, with 0.33 dead latents on average. Its seed-0 run reaches MCC 0.8986 and Rare F1 0.9554, with one dead latent. The separate continuous run below reaches MCC 0.8974 and Rare F1 0.9482, with zero. MP has two dead latents but lower recovery, so keeping latents active is not sufficient for accurate feature recovery.

C.5.2 Training sample efficiency

Fig. 11 reports MCC, Macro F1, and Rare F1 throughout training in the full-width setting. Its BeFOND curve follows one continuous run, whose endpoint is identified separately in Table 4.

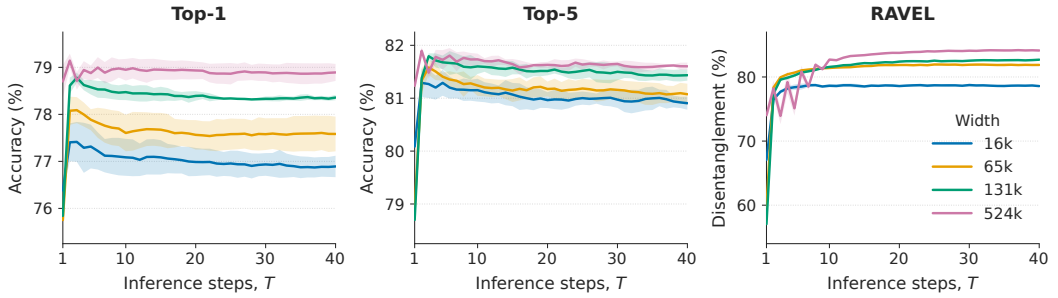


Figure 12: **Performance as a function of inference depth.** Sparse probing top-1 accuracy (left), top-5 accuracy (middle), and RAVEL disentanglement (right) for BeFOND on Gemma-2-2B layer-12 activations, evaluated over $T = 1, \ldots, 40$ inference steps at dictionary widths of 16k, 65k, 131k, and 524k. All evaluations use checkpoint 12,207, corresponding to 49,999,872 training tokens. Lines show means and shaded regions indicate ± 1 sample standard deviation across three training seeds. Sparse probing peaks within the first few steps, whereas RAVEL generally benefits from longer inference.

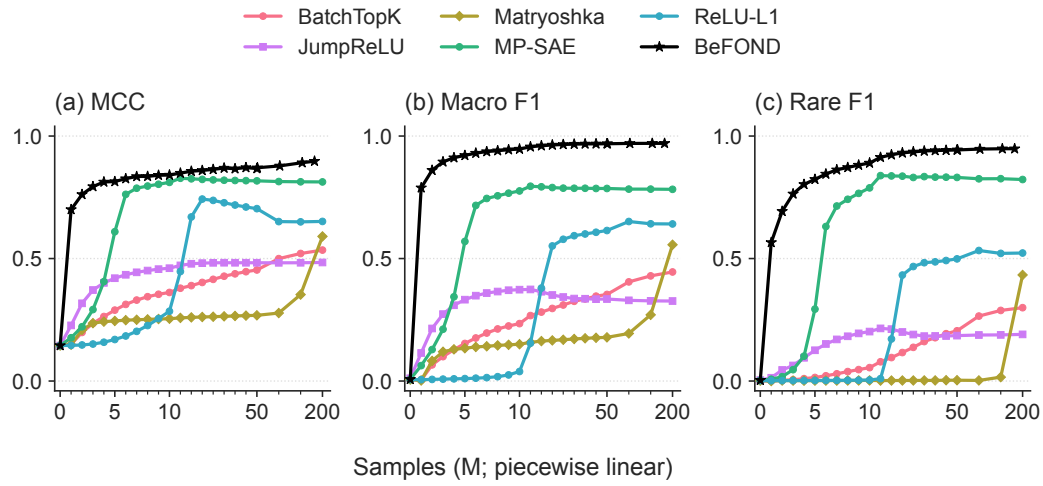


Figure 11: Feature recovery throughout training on unwhitened SynthSAEBench ($d = 768$; 16,384 learned and ground-truth features), using plain baseline recipes and seed 0. (a) Dictionary mean correlation coefficient (MCC). (b) Macro F1 over ground-truth features after positive-cosine one-to-one matching. (c) Rare F1 over the least frequent quartile, defined by a shared independent 500K-example frequency reference (seed 315031). Every checkpoint is evaluated on 1M held-out examples (seed 49100). BeFOND uses 200 inference steps with activity $p > 0.5$; baselines use positive native activations. The x-axis counts decoder-update examples and is piecewise linear: 1M examples at 0–10M, 5M at 10–50M, and 25M beyond 50M occupy equal horizontal distances. Cross-region slopes therefore do not have a common per-sample interpretation. Baseline points through 10M, after 10M through 50M, and beyond 50M come from separate matching-recipe runs with their original full-budget learning-rate schedules; they do not form a single continuous trajectory. Overlapping checkpoints are selected by these fixed windows, not by score. BeFOND uses one continuous run. Samples are rounded up to full batches. Baselines finish at approximately 200M decoder-update examples and BeFOND at 181.82M plus 18.18M separate prior-update examples, which are excluded from the x-axis. Baseline auxiliary exposure is not counted on this axis. Final measured L_0 values are BatchTopK: 34.83, JumpReLU: 45.15, Matryoshka: 19.98, MP-SAE: 29.65, ReLU-L1: 35.09, BeFOND: 35.70. One seed is shown; no uncertainty intervals are estimated.

Table 5: SAEBench results on whitened Gemma-2-2B layer-12 activations after 49,999,872 training tokens. Entries are mean ± sample standard deviation over three training seeds for BeFOND and five for each baseline. For SAE baselines, we selected target L_0 values by mean top-1 accuracy, and then used it for all probing columns, and the model selected by mean disentanglement supplies all RAVEL columns. Parentheses give both values when they differ. Sparse probing measures how well $k = 1$ or $k = 5$ features predict concepts, averaged equally across 113 tasks: **accuracy** is the fraction of correct predictions; **AUC** measures how well positives rank above negatives across thresholds; **F1** balances precision (correctness of positive predictions) and recall (coverage of positives). RAVEL measures whether interventions change a targeted attribute (**cause**) while preserving other attributes (**isolation**), and **disentanglement** is their mean. Higher is better throughout. Within each width and metric, bold marks the highest mean, selected before rounding.

Model	Sparse probing						RAVEL		
	$k = 1$			$k = 5$					
	Acc. ↑	AUC ↑	F1 ↑	Acc. ↑	AUC ↑	F1 ↑	Disentanglement ↑	Cause ↑	Isolation ↑
Dictionary width: 16k									
BatchTopK (target $L_0 = 400$)	.767$_{\pm.002}$	**.798**$_{\pm.003}$	.761$_{\pm.004}$	**.814**$_{\pm.002}$	**.862**$_{\pm.002}$	**.812**$_{\pm.002}$	.729$_{\pm.005}$	.692$_{\pm.019}$	.767$_{\pm.017}$
Matryoshka BatchTopK ($L_0 = 200$ probing; 400 RAVEL)	.768$_{\pm.003}$	.794$_{\pm.003}$	.762$_{\pm.002}$	.810$_{\pm.002}$	.858$_{\pm.002}$	.808$_{\pm.002}$	.731$_{\pm.003}$	.695$_{\pm.013}$	.767$_{\pm.015}$
Mean-field + Adam ($T = 20$ inference steps)	.761$_{\pm.003}$	.761$_{\pm.003}$	.749$_{\pm.004}$	.801$_{\pm.002}$	.842$_{\pm.001}$	.798$_{\pm.002}$	.696$_{\pm.003}$	.639$_{\pm.006}$	.753$_{\pm.006}$
BeFOND ($T = 10$ inference steps)	**.771**$_{\pm.004}$	.781$_{\pm.004}$	**.765**$_{\pm.005}$	.812$_{\pm.002}$	.856$_{\pm.001}$	.811$_{\pm.002}$	**.786**$_{\pm.002}$	**.759**$_{\pm.002}$	**.813**$_{\pm.003}$
Dictionary width: 65k									
BatchTopK ($L_0 = 400$ probing; 800 RAVEL)	.770$_{\pm.003}$	.800$_{\pm.002}$	.765$_{\pm.003}$	**.814**$_{\pm.003}$	**.862**$_{\pm.002}$	**.812**$_{\pm.003}$	.745$_{\pm.005}$	.712$_{\pm.006}$	.777$_{\pm.008}$
Matryoshka BatchTopK (target $L_0 = 400$)	.772$_{\pm.001}$	**.801**$_{\pm.001}$	.766$_{\pm.002}$	.811$_{\pm.001}$	.858$_{\pm.001}$	.809$_{\pm.002}$	.737$_{\pm.004}$	.704$_{\pm.007}$	.771$_{\pm.006}$
Mean-field + Adam ($T = 20$ inference steps)	.757$_{\pm.002}$	.757$_{\pm.002}$	.744$_{\pm.002}$	.794$_{\pm.001}$	.832$_{\pm.001}$	.790$_{\pm.002}$	.733$_{\pm.003}$	.697$_{\pm.007}$	.769$_{\pm.006}$
BeFOND ($T = 10$ inference steps)	**.776**$_{\pm.004}$	.794$_{\pm.004}$	**.769**$_{\pm.006}$	.812$_{\pm.002}$	.856$_{\pm.001}$	.811$_{\pm.002}$	**.814**$_{\pm.000}$	**.789**$_{\pm.001}$	**.840**$_{\pm.001}$
Dictionary width: 131k									
BatchTopK (target $L_0 = 600$)	.773$_{\pm.002}$	**.804**$_{\pm.002}$	.768$_{\pm.003}$	.814$_{\pm.002}$	**.862**$_{\pm.001}$	.813$_{\pm.002}$	.751$_{\pm.003}$	.720$_{\pm.008}$	.782$_{\pm.013}$
Matryoshka BatchTopK ($L_0 = 200$ probing; 400 RAVEL)	.773$_{\pm.004}$	.799$_{\pm.004}$	.768$_{\pm.004}$	.814$_{\pm.001}$	.860$_{\pm.002}$	.812$_{\pm.001}$	.739$_{\pm.002}$	.713$_{\pm.008}$	.764$_{\pm.006}$
Mean-field + Adam ($T = 20$ inference steps)	.754$_{\pm.003}$	.754$_{\pm.003}$	.740$_{\pm.005}$	.791$_{\pm.002}$	.830$_{\pm.003}$	.787$_{\pm.002}$	.734$_{\pm.005}$	.694$_{\pm.007}$	.774$_{\pm.004}$
BeFOND ($T = 10$ inference steps)	**.785**$_{\pm.002}$	.802$_{\pm.003}$	**.778**$_{\pm.003}$	**.816**$_{\pm.001}$	.857$_{\pm.002}$	**.814**$_{\pm.002}$	**.816**$_{\pm.001}$	**.790**$_{\pm.002}$	**.842**$_{\pm.003}$
Dictionary width: 524k									
BatchTopK (target $L_0 = 800$)	.773$_{\pm.004}$	.803$_{\pm.004}$	.768$_{\pm.006}$	.814$_{\pm.003}$	**.860**$_{\pm.002}$	.813$_{\pm.003}$	.748$_{\pm.004}$	.719$_{\pm.010}$	.776$_{\pm.010}$
Matryoshka BatchTopK ($L_0 = 200$ probing; 400 RAVEL)	.774$_{\pm.002}$	.799$_{\pm.002}$	.768$_{\pm.002}$	.813$_{\pm.001}$	.859$_{\pm.001}$	.812$_{\pm.000}$	.743$_{\pm.003}$	.708$_{\pm.013}$	.778$_{\pm.009}$
Mean-field + Adam ($T = 20$ inference steps)	.751$_{\pm.007}$	.751$_{\pm.007}$	.735$_{\pm.009}$	.786$_{\pm.009}$	.825$_{\pm.010}$	.782$_{\pm.009}$	.685$_{\pm.029}$	.622$_{\pm.047}$	.748$_{\pm.013}$
BeFOND ($T = 10$ inference steps)	**.789**$_{\pm.002}$	**.805**$_{\pm.004}$	**.784**$_{\pm.004}$	**.817**$_{\pm.001}$	.859$_{\pm.001}$	**.816**$_{\pm.001}$	**.827**$_{\pm.001}$	**.800**$_{\pm.001}$	**.854**$_{\pm.002}$

D Extended Gemma experiments

D.1 Experimental setup

Dataset. We extract layer-12 post-residual activations from Gemma-2-2B (Gemma Team, 2024) ($d = 2{,}304$). Documents from Pile Uncopyrighted (monology, 2023) are tokenized separately with BOS and truncated to 1,024 tokens; BOS and padding positions are excluded. Validation, calibration, and training use disjoint document ranges. Raw and whitened activations are evaluated separately. Validation and calibration each use 65,536 tokens, with up to 500M training tokens. Extraction uses BF16, batch 4, and no document packing. Whitening statistics were fitted on 500M training tokens.

Training. Table 6 summarizes the training and evaluation settings. The selected BeFOND results in Table 5 use seeds 0–2 and checkpoint 12,207, corresponding to 49,999,872 decoder-training tokens. Evaluation uses ten outer inference steps with one inner step per reference update. The readout retains posterior means above 0.5 and sets the remainder to zero, followed by decoder-norm rescaling.

At widths 16K, 65K, and 131K, the recipes use $\beta = 1$, ETD1 step $h = 1.5$, reference retention 0.999, and decoder-norm targets 4, 1.5, and 0.75, respectively. At 524K they use $\beta = 1.5$, $h = \infty$, full reference retention, and norm target 0.5. All use a global batch of 4,096 and a linear learning-rate schedule with peak 10^{-3} and minimum 10^{-5}; warmup occupies 5% of training at 16K–131K and 1% at 524K.

Table 6: Gemma training and evaluation settings. Preprocessing, checkpoint, inference depth, and readout identify an evaluation configuration. Prefixes of 500M-token runs retain their original schedule.

Method / metric	Settings
BeFOND	Widths 16,384–524,288; seeds 0–2; FP32 inference/parameters; prior Fisher with current moments and no damping; $\Delta t = 1$; decoder solver `matrix_free_phi1_auto`
MF20	Seeds 0–4; batch 2,048; 20 inference steps; norm cap 4; dictionary LR 0.0003; other-parameter LR 0.01; damping 1 then 0.2
MF20+clip	Whitened inputs, 50M tokens, five seeds, 20 inference steps, readout $p > 0.5$. After 2,000 updates, clip the dictionary gradient at three times its running-mean norm and skip batches whose negative ELBO exceeds five times its running mean, capped at approximately 20% of recent batches
BatchTopK / Matryoshka	Whitened inputs; seeds 1–5; batch 4,096; 12,207 updates (49,999,872 tokens); constant LR 3×10^{-4}; FP32 SAEs, BF16 Gemma; SAELens 6.51.1. ZCA fitted on 500M training tokens. Target L_0 values in Table 7 were selected on a separate seed-0 pilot
Baseline initialization / losses	Decoder norm initialized to 0.1; decoder bias subtracted before encoding; activations rescaled by decoder norm; auxiliary-loss coefficient 1; threshold EMA rate 0.01
Matryoshka levels	4,096, 16,384, and the full width (16,384 appears once for the smallest model); Matryoshka auxiliary loss enabled
Gemma Scope	Release `google/gemma-scope-2b-pt-res`, checkpoint paths `layer_12/width_{w}/average_l0_{s}`, with (w, s) given by each row of Table 8. Training: 4B tokens at 16K, 8B at 32K–524K, and 16B at 1M (Lieberum et al., 2024)
Sparse probing	113 tasks; final non-padding token; context 1,024; 1/2/5 selected features; L1 logistic regression; equal task averaging; seed 42
Probe selection	Rank absolute normalized class-mean difference; 10 inverse-penalty values from 10^5 to 10^{-5}; validation ROC-AUC; refit on training data
RAVEL	Average cause and isolation over attributes within each entity, then equally over entities; disentanglement is their mean. Aggregation requires completed results for every configured entity
Reconstruction / sparsity	OpenWebText; context 128; 3,200 / 32,000 prompts

Evaluation. Sparse probing reports mean test accuracy across 113 tasks, using the best 1 or 5 features selected on training data. RAVEL reports cause, isolation, and their mean (Huang et al., 2024). Table 6 specifies the evaluation settings (Karvonen et al., 2025).

Implementation. The Gemma reference environment uses Python 3.11.14, PyTorch 2.7.1+cu126, TransformerLens 2.16.1, Transformers 4.57.3, and SAE Lens 6.24.6.

D.2 Results

The whitened baseline comparison reports sparse probing and RAVEL in Table 7; Table 8 reports the Gemma Scope checkpoints. These evaluations retain their respective preprocessing, training budgets, and seed aggregation specified in the captions.

Table 7: SAEBench results for SAEs trained on 50M tokens of whitened Gemma-2-2B layer-12 activations. Entries report mean ± sample standard deviation over five training seeds (1–5), evaluated at the final checkpoint (49,999,872 tokens). Sparse-probing metrics are macro-averaged over 113 tasks; RAVEL overall is the disentanglement score. All scores are fractions, and higher is better. L_0 denotes target training sparsity, fixed using a separate seed-0 pilot. Bold indicates the highest mean across L_0 values within each width–architecture group, selected before rounding.

Width	Target L_0	Sparse probing (Kantamneni et al., 2025)						RAVEL (Huang et al., 2024)		
		$k = 1$			$k = 5$					
		Acc. ↑	AUC ↑	F1 ↑	Acc. ↑	AUC ↑	F1 ↑	Overall ↑	Cause ↑	Isolation ↑
BatchTopK (16k)										
16k	200	.765$_{\pm.004}$	.791$_{\pm.004}$	.758$_{\pm.005}$	.810$_{\pm.002}$	.859$_{\pm.002}$	.808$_{\pm.003}$	.717$_{\pm.009}$	**.698**$_{\pm.018}$	.736$_{\pm.021}$
16k	400	**.767**$_{\pm.002}$	**.798**$_{\pm.003}$	**.761**$_{\pm.004}$	**.814**$_{\pm.002}$	**.862**$_{\pm.002}$	**.812**$_{\pm.002}$	**.729**$_{\pm.005}$	.692$_{\pm.019}$	**.767**$_{\pm.017}$
Matryoshka BatchTopK (16k)										
16k	200	**.768**$_{\pm.003}$	.794$_{\pm.003}$	**.762**$_{\pm.002}$	**.810**$_{\pm.002}$	**.858**$_{\pm.002}$	**.808**$_{\pm.002}$	.718$_{\pm.004}$	.688$_{\pm.018}$	.747$_{\pm.013}$
16k	400	.767$_{\pm.001}$	**.797**$_{\pm.001}$	.761$_{\pm.002}$	.809$_{\pm.002}$	.857$_{\pm.001}$	.808$_{\pm.002}$	**.731**$_{\pm.003}$	**.695**$_{\pm.013}$	**.767**$_{\pm.015}$
BatchTopK (65k)										
65k	400	**.770**$_{\pm.003}$	.800$_{\pm.002}$	**.765**$_{\pm.003}$	**.814**$_{\pm.003}$	**.862**$_{\pm.002}$	**.812**$_{\pm.003}$	.740$_{\pm.003}$	.700$_{\pm.015}$	**.781**$_{\pm.012}$
65k	800	.768$_{\pm.002}$	**.803**$_{\pm.002}$	.764$_{\pm.002}$	.813$_{\pm.001}$	.860$_{\pm.001}$	.812$_{\pm.001}$	**.745**$_{\pm.005}$	**.712**$_{\pm.006}$	.777$_{\pm.008}$
Matryoshka BatchTopK (65k)										
65k	200	.772$_{\pm.002}$	.798$_{\pm.002}$	**.766**$_{\pm.002}$	**.813**$_{\pm.001}$	**.861**$_{\pm.002}$	**.811**$_{\pm.001}$	.734$_{\pm.009}$	**.707**$_{\pm.015}$	.761$_{\pm.010}$
65k	400	**.772**$_{\pm.001}$	**.801**$_{\pm.001}$	.766$_{\pm.002}$	.811$_{\pm.001}$	.858$_{\pm.001}$	.809$_{\pm.002}$	**.737**$_{\pm.004}$	.704$_{\pm.007}$	**.771**$_{\pm.006}$
BatchTopK (131k)										
131k	600	**.773**$_{\pm.002}$	**.804**$_{\pm.002}$	**.768**$_{\pm.003}$	**.814**$_{\pm.002}$	**.862**$_{\pm.001}$	**.813**$_{\pm.002}$	**.751**$_{\pm.003}$	**.720**$_{\pm.008}$	**.782**$_{\pm.013}$
131k	800	.770$_{\pm.003}$	.803$_{\pm.003}$	.765$_{\pm.003}$	.814$_{\pm.002}$	.859$_{\pm.002}$	.812$_{\pm.002}$	.746$_{\pm.004}$	.720$_{\pm.011}$	.771$_{\pm.005}$
Matryoshka BatchTopK (131k)										
131k	200	**.773**$_{\pm.004}$	.799$_{\pm.004}$	**.768**$_{\pm.004}$	**.814**$_{\pm.001}$	**.860**$_{\pm.002}$	**.812**$_{\pm.001}$	.739$_{\pm.008}$	**.723**$_{\pm.019}$	.754$_{\pm.006}$
131k	400	.773$_{\pm.001}$	**.801**$_{\pm.001}$	.767$_{\pm.002}$	.812$_{\pm.001}$	.858$_{\pm.001}$	.810$_{\pm.002}$	**.739**$_{\pm.002}$	.713$_{\pm.008}$	**.764**$_{\pm.006}$
BatchTopK (524k)										
524k	600	.772$_{\pm.004}$	.801$_{\pm.004}$	.766$_{\pm.004}$	.813$_{\pm.002}$	**.861**$_{\pm.002}$	.812$_{\pm.003}$	.747$_{\pm.006}$	.719$_{\pm.011}$	.775$_{\pm.007}$
524k	800	**.773**$_{\pm.004}$	**.803**$_{\pm.004}$	**.768**$_{\pm.006}$	**.814**$_{\pm.003}$	.860$_{\pm.002}$	**.813**$_{\pm.003}$	**.748**$_{\pm.004}$	**.719**$_{\pm.010}$	**.776**$_{\pm.010}$
Matryoshka BatchTopK (524k)										
524k	200	**.774**$_{\pm.002}$	.799$_{\pm.002}$	**.768**$_{\pm.002}$	**.813**$_{\pm.001}$	**.859**$_{\pm.001}$	**.812**$_{\pm.000}$	.734$_{\pm.005}$	.705$_{\pm.011}$	.764$_{\pm.011}$
524k	400	.771$_{\pm.002}$	**.800**$_{\pm.002}$	.767$_{\pm.003}$	.810$_{\pm.001}$	.858$_{\pm.001}$	.809$_{\pm.002}$	**.743**$_{\pm.003}$	**.708**$_{\pm.013}$	**.778**$_{\pm.009}$

D.3 Wall-clock training times on H200 GPUs

Table 9 compares wall-clock training times on eight NVIDIA H200 GPUs, with one model measured at a time and seed 0 throughout. Each measurement covers the first 999,424 main-training tokens of the original 50M-token schedule. Global batch size is 4,096 for BeFOND, BatchTopK, and Matryoshka, and 2,048 for MF20+clip.

Inputs are loaded, shuffled, whitened, and placed in GPU memory before timing. Initialization and CUDA/JIT warmup are excluded, and the complete initial training state is restored after warmup. The timed interval includes inference, losses, parameter updates, normalization, required communication, and BeFOND's prior updates with convergence checks. Evaluation, checkpoint writing, and logging are excluded. PyTorch timings use the maximum CUDA-synchronized elapsed time across ranks, while JAX waits for all output shards on all devices.

BeFOND retains its original stochastic training horizons, capped at 100 with one inner step, and warms the complete measured prefix to cover adaptive solver paths. MF uses 20 inference steps. Its clipping and loss-based skipping begin after 2,000 updates and are inactive during this prefix. Model tensors are FP32, with native FP64 intermediates retained in BeFOND's moment reductions and Woodbury calculations. BeFOND and the amortized baselines use PyTorch, while MF uses JAX. The distributed baseline implementations preserve the global TopK budget and synchronize gradients, thresholds, and dead-feature histories.

Table 8: SAEBench evaluations of released Gemma Scope layer-12 checkpoints (Lieberum et al., 2024; Karvonen et al., 2025), organized by dictionary width and average sparsity L_0.

Width	L_0	Sparse probing									RAVEL		
		$k=1$			$k=2$			$k=5$			Disentanglement ↑	Cause ↑	Isolation ↑
		Acc. ↑	AUC ↑	F1 ↑	Acc. ↑	AUC ↑	F1 ↑	Acc. ↑	AUC ↑	F1 ↑			
16k	22	0.7337	0.7476	0.7159	0.7529	0.7861	0.7441	0.7771	0.8243	0.7717	0.5989	0.5121	0.6857
16k	41	0.7519	0.7710	0.7366	0.7686	0.8056	0.7601	0.7893	0.8370	0.7835	0.6644	0.6163	0.7126
16k	82	0.7627	0.7857	0.7545	0.7821	0.8207	0.7768	0.8047	0.8523	0.8016	0.6960	0.6489	0.7431
16k	176	0.7679	0.7938	0.7602	0.7890	0.8262	0.7851	0.8129	0.8587	0.8119	0.7206	0.6508	**0.7904**
16k	445	**0.7725**	**0.8033**	**0.7687**	**0.7942**	**0.8363**	**0.7921**	**0.8171**	**0.8632**	**0.8162**	**0.7488**	**0.7205**	0.7771
32k	12	0.7179	0.7264	0.6960	0.7350	0.7665	0.7181	0.7570	0.8046	0.7489	0.6050	0.5461	0.6639
32k	22	0.7379	0.7498	0.7221	0.7535	0.7824	0.7418	0.7758	0.8187	0.7711	0.6416	0.6251	0.6582
32k	40	0.7520	0.7695	0.7422	0.7662	0.7995	0.7590	0.7872	0.8327	0.7842	0.6755	0.6200	0.7311
32k	76	0.7607	0.7815	0.7534	0.7761	0.8138	0.7721	0.7990	0.8466	0.7958	0.7075	0.6714	0.7436
32k	155	0.7675	0.7907	0.7609	0.7865	0.8255	0.7833	0.8054	0.8528	0.8025	0.7249	0.6881	0.7616
32k	360	**0.7749**	**0.8049**	**0.7713**	**0.7897**	**0.8306**	**0.7866**	**0.8140**	**0.8601**	**0.8129**	**0.7450**	**0.7105**	**0.7794**
65k	21	0.7361	0.7459	0.7139	0.7507	0.7785	0.7359	0.7743	0.8186	0.7686	0.6469	0.6053	0.6885
65k	38	0.7501	0.7656	0.7399	0.7660	0.7962	0.7589	0.7874	0.8290	0.7834	0.6767	0.6345	0.7188
65k	72	0.7647	0.7816	0.7552	0.7810	0.8128	0.7751	0.7987	0.8451	0.7965	0.7041	0.6596	0.7487
65k	141	0.7717	0.7930	0.7653	0.7858	0.8231	0.7814	0.8087	0.8541	0.8069	0.7346	**0.7158**	0.7535
65k	297	**0.7788**	**0.8032**	**0.7731**	**0.7947**	**0.8343**	**0.7925**	**0.8178**	**0.8611**	**0.8169**	**0.7487**	0.7016	**0.7959**
131k	12	0.7108	0.7198	0.6870	0.7311	0.7585	0.7146	0.7525	0.7952	0.7443	0.6004	0.5764	0.6245
131k	20	0.7279	0.7362	0.7074	0.7483	0.7752	0.7335	0.7698	0.8132	0.7621	0.6408	0.5950	0.6866
131k	36	0.7379	0.7513	0.7222	0.7565	0.7843	0.7471	0.7810	0.8226	0.7767	0.6828	0.6451	0.7205
131k	67	0.7554	0.7713	0.7444	0.7693	0.8019	0.7622	0.7913	0.8347	0.7876	0.7083	0.6734	0.7432
131k	129	0.7677	0.7865	0.7608	0.7837	0.8173	0.7794	0.8027	0.8475	0.8011	0.7265	0.6561	**0.7969**
131k	264	**0.7814**	**0.8033**	**0.7773**	**0.7941**	**0.8290**	**0.7914**	**0.8138**	**0.8575**	**0.8125**	**0.7486**	**0.7281**	0.7691
262k	11	0.7068	0.7168	0.6786	0.7232	0.7503	0.7054	0.7430	0.7872	0.7310	0.6166	0.5987	0.6345
262k	21	0.7240	0.7355	0.7037	0.7443	0.7732	0.7318	0.7657	0.8078	0.7607	0.6355	0.6169	0.6540
262k	36	0.7375	0.7533	0.7252	0.7573	0.7879	0.7479	0.7772	0.8195	0.7723	0.6795	0.6567	0.7023
262k	67	0.7510	0.7684	0.7427	0.7666	0.7960	0.7596	0.7911	0.8325	0.7883	0.7039	0.6652	0.7426
262k	121	0.7656	0.7854	0.7572	0.7818	0.8155	0.7772	0.8029	0.8480	0.8008	0.7158	0.6737	0.7579
262k	243	**0.7733**	**0.7951**	**0.7655**	**0.7928**	**0.8291**	**0.7906**	**0.8119**	**0.8561**	**0.8105**	**0.7523**	**0.7219**	**0.7827**
524k	22	0.7129	0.7276	0.6976	0.7218	0.7529	0.7104	0.7453	0.7893	0.7384	0.6138	0.5800	0.6475
524k	29	0.7258	0.7421	0.7139	0.7400	0.7725	0.7297	0.7629	0.8076	0.7573	0.6451	0.6002	0.6900
524k	46	0.7387	0.7552	0.7264	0.7535	0.7872	0.7457	0.7770	0.8211	0.7731	0.6803	0.6602	0.7005
524k	65	0.7523	0.7688	0.7412	0.7663	0.7981	0.7589	0.7879	0.8303	0.7856	0.6958	0.6583	0.7332
524k	115	0.7637	0.7815	0.7550	0.7791	0.8110	0.7734	0.8004	0.8419	0.7981	0.7251	0.6760	0.7741
524k	227	**0.7735**	**0.7955**	**0.7682**	**0.7927**	**0.8268**	**0.7890**	**0.8106**	**0.8520**	**0.8094**	**0.7555**	**0.7330**	**0.7781**
1m	19	0.7028	0.7182	0.6875	0.7165	0.7497	0.7081	0.7396	0.7847	0.7350	0.5910	0.5570	0.6250
1m	26	0.7234	0.7388	0.7047	0.7357	0.7693	0.7273	0.7587	0.8084	0.7523	0.6371	0.6120	0.6622
1m	58	0.7354	0.7548	0.7239	0.7550	0.7876	0.7499	0.7755	0.8193	0.7737	0.6722	0.6560	0.6884
1m	73	0.7526	0.7714	0.7435	0.7671	0.8013	0.7633	0.7887	0.8295	0.7866	0.6933	0.6533	0.7334
1m	107	0.7633	0.7797	0.7545	0.7737	0.8030	0.7684	0.7959	0.8385	0.7937	0.7201	0.6857	0.7545
1m	207	**0.7788**	**0.7983**	**0.7726**	**0.7919**	**0.8264**	**0.7881**	**0.8095**	**0.8514**	**0.8081**	**0.7420**	**0.6860**	**0.7979**

Table 9: Elapsed training time in seconds for 999,424 training tokens on eight NVIDIA H200 GPUs (lower is better). Each model is measured once at seed 0. Timings include BeFOND's prior updates and exclude evaluation, initialization, warmup, data preparation, checkpointing, and logging. Matryoshka denotes Matryoshka BatchTopK.

Width K	BeFOND	BatchTopK	Matryoshka	MF20+clip
16,384	33.125	2.719	2.991	2.714
65,536	118.959	9.202	10.737	10.157
131,072	234.665	17.935	20.075	19.914
524,288	989.321	70.121	77.382	78.331

Across these widths, BeFOND takes approximately 12–14× as long as BatchTopK for the measured training prefix. These measurements do not determine full-run time or time to a common recovery score.