INVEXPLAIN: AN AGENTIC INTERPRETATION FRAMEWORK FOR SPARSE FEATURES VIA TRAINING-FREE DIFFUSION INVERSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have achieved remarkable capabilities, yet their internal representations remain difficult to understand. Sparse autoencoders (SAEs) offer a promising approach by decomposing these representations into more interpretable features. However, automatically interpreting individual SAE features remains challenging. Existing automated pipelines primarily infer explanations from corpus-retrieved activating samples, limiting their evidence to activation patterns represented in a fixed corpus. Feature inversion can expand evidence beyond the corpus, but learned inverse models require additional training and remain tied to their training distributions. To overcome these limitations, we introduce InvExplain, an agentic interpretation framework driven by training-free diffusion posterior sampling (DPS). By leveraging a frozen diffusion language model prior, InvExplain actively acquires new, activation-verified evidence beyond the retrieval corpus without additional training. An LLM agent combines generated and corpus-retrieved evidence to formulate and test competing hypotheses, uses DPS to repair failed positive tests, and refines the explanation. Theoretically, we characterize the prior-dependent distortion of training-based inversion and show that diffusion posterior sampling can be viewed as approximating an activation-constrained distribution under a pretrained language prior. Across three models, InvExplain outperforms corpus-reliant baselines on a shared test of whether a feature activates on a text and, if so, where its activation peaks.

1 INTRODUCTION

Large language models (LLMs) have demonstrated strong capabilities in language understanding, reasoning, and generation, but the internal mechanisms underlying these capabilities remain difficult to understand. Mechanistic interpretability aims to make the internal workings of neural networks understandable to humans (Bereska & Gavves, 2024).

Among existing approaches, sparse autoencoders (SAEs) decompose complex model activations into sparse features, providing more manageable units for analyzing internal representations (Bricken et al., 2023; Dunefsky et al., 2024). Empirical studies have identified features associated with lexical, structural, and semantic patterns (Bricken et al., 2023; Gao et al., 2025). SAE features have also been used in downstream applications including data selection (Rathi & Radford, 2026), data synthesis (Li et al., 2026), and interpreting scientific foundation models (Simon & Zou, 2025; Rosenfeld & Sonnewald, 2026).

However, training an SAE does not automatically make its features understandable. Existing explanation methods often retrieve high-activation samples from a fixed corpus and derive explanations from them (Bricken et al., 2023; Lin, 2023; Han et al., 2026). This restricts the available evidence to patterns represented in the corpus: activating patterns absent from the corpus cannot be observed, while particular genres or constructions may be overrepresented among retrieved samples, leading to biased explanations.

Feature inversion offers a way to acquire evidence beyond a fixed corpus by reversing the usual input-to-activation direction and generating texts guided by a target feature activation strength. From a distributional perspective, this can be viewed as conditional sampling, enabling the discovery

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.

of activating samples not present in the retrieval corpus. Recent training-based approaches learn mappings from internal feature representations to language (Luo et al., 2025; Meng et al., 2026), but require additional training and may inherit biases from their training distributions. Our controlled toy experiment illustrates this risk: even an inverse trained on samples from the natural prior exhibits non-negligible approximation error, while training on a shifted prior introduces further deviation from the natural inversion target (Section 2.4, Fig. 1).

Diffusion posterior sampling (DPS) provides a training-free approach for guiding samples from a pretrained diffusion prior toward an external objective (Chung et al., 2023). Recent progress in diffusion language models (DLMs) further provides pretrained natural-language priors (Nie et al., 2025; 2026), while related work has extended posterior sampling and reward guidance to discrete diffusion models (Lovelace et al., 2024; Tejaswi et al., 2026; Venkatraman et al., 2024).

Recently, EnergyDPS (Zhou et al., 2026) demonstrated training-free SAE feature inversion in the visual domain by using a pretrained diffusion model as a natural-image prior. However, to our knowledge, an analogous diffusion-inversion pipeline for language-model SAE features remains unexplored. Discrete and highly stochastic language generation introduces additional challenges in this setting. Motivated by recent advances in DLMs, we develop a training-free diffusion inversion method for SAE features in LLMs.

Building on this inversion method, we introduce InvExplain, an agentic interpretation framework that actively acquires and tests evidence for SAE feature explanation (Figure 2). Specifically, we formulate feature inversion as posterior sampling under a pretrained DLM prior, where target feature activation guides the generation of new activating samples without training a dedicated inverse model. InvExplain combines high-activation corpus samples with activation-verified generated samples to form testable hypotheses. The agent evaluates these hypotheses with targeted tests of when the feature should or should not activate, invokes inversion when an expected activating sample fails, and iteratively refines the explanation to characterize the feature's activation patterns and boundaries.

The main contributions of this work are summarized as follows:

- We propose a training-free feature inversion method that uses a pretrained DLM to generate language samples guided by target SAE-feature activation. To our knowledge, this is the first application of training-free DPS to SAE feature inversion in LLMs.
- We develop InvExplain, an agentic interpretation framework that combines corpus evidence with inversion-generated evidence for hypothesis formation, targeted testing, and iterative explanation refinement.
- We characterize training-domain bias in feature inversion and establish the activation-constrained target of diffusion inversion. Across 110 SAE features from three LLMs, InvExplain achieves 75.58% joint activation-and-localization accuracy, outperforming SAGE (Han et al., 2026) (53.64%) and a Neuronpedia-style baseline (Lin, 2025) (56.03%).

2 METHOD

2.1 SPARSE AUTOENCODERS AND FEATURE ACTIVATIONS

SAEs are designed to decompose the dense, high-dimensional representations $x \in \mathbb{R}^{d_{\text{in}}}$ of large models (typically neuron activations) into sparse, potentially interpretable features $f \in \mathbb{R}^{d_{\text{sae}}}$, where $d_{\text{sae}} \gg d_{\text{in}}$. An SAE maps x to this overcomplete space and reconstructs the original activation as

$$u = W_{\text{enc}}x + b_{\text{enc}}, \qquad f = \phi(u), \qquad \hat{x} = W_{\text{dec}}f + b_{\text{dec}}, \tag{1}$$

where $W_{\text{enc}} \in \mathbb{R}^{d_{\text{sae}} \times d_{\text{in}}}$ and $b_{\text{enc}} \in \mathbb{R}^{d_{\text{sae}}}$ are the encoder weight and bias, while $W_{\text{dec}} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{sae}}}$ and $b_{\text{dec}} \in \mathbb{R}^{d_{\text{in}}}$ form the decoder. For the Top-K SAE considered here, $\phi = \text{TopK}_k$ retains the k strongest feature responses. Each coordinate of f corresponds to a sparse feature with an associated decoder direction (Huben et al., 2024; Gao et al., 2025). Transcoders (Dunefsky et al., 2024) provide a related formulation that reconstructs downstream representations instead of the input representation. Our analysis depends only on encoder-side feature responses and therefore applies to both SAEs and transcoders.

Since SAE training uses only a reconstruction objective without semantic supervision, explanations of individual features must be inferred post hoc. For a text input $t \in \mathcal{X}$, let $x_\ell(t) \in \mathbb{R}^{d_{\text{in}}}$ denote the

target model representation at scored token position ℓ. For feature $j \in \{1, \ldots, d_{\text{sae}}\}$, we define its pre-TopK response and feature activation strength as

$$u_{j,\ell}(t) = [W_{\text{enc}}]_{j,:} x_\ell(t) + [b_{\text{enc}}]_j, \qquad s_j(t) = \max_{1 \le \ell \le n(t)} u_{j,\ell}(t), \tag{2}$$

where $n(t)$ is the number of scored token positions. We suppress the feature index j below and write $s(t)$ for the activation strength of the target feature.

2.2 Feature Interpretation

The feature activation strength $s(t)$ quantifies how strongly the target feature responds to a text t, but this scalar alone does not reveal which properties of the text are associated with its activation. Feature interpretation therefore aims to summarize the activation behavior characterized by s as a natural-language explanation that is understandable to humans.

Classical automated interpretation methods obtain evidence by evaluating the target feature on a finite corpus and retrieving samples with the greatest activation strengths (Bills et al., 2023; Paulo et al., 2025; Han et al., 2026). Let $\mathcal{D} = \{t_1, \ldots, t_M\} \subset \mathcal{X}$ be a corpus of M text samples. Reorder these samples such that $s\big(t_{(1)}\big) \ge s\big(t_{(2)}\big) \ge \cdots \ge s\big(t_{(M)}\big)$, and define the retrieved evidence set as $\mathcal{C}_N(\mathcal{D}, s) := \big\{\big(t_{(i)}, s(t_{(i)})\big)\big\}_{i=1}^{N}$. An automated interpreter can then be represented as a map $\mathscr{A}$ from the evidence set to an explanation $e_{\mathcal{D}} := \mathscr{A}(\mathcal{C}_N(\mathcal{D}, s)), \qquad e_{\mathcal{D}} \in \mathcal{E}$, where $\mathcal{E}$ denotes the space of natural language explanations. In this view, the evidence set connects feature activation strength $s(t)$ to language patterns that the interpreter can describe.

The resulting explanation depends on the corpus through $\mathcal{C}_N(\mathcal{D}, s)$. A finite corpus covers only a limited portion of natural language. If a linguistic or semantic pattern is absent from $\mathcal{D}$, the retrieved evidence cannot contain it, even if unseen samples exhibiting that pattern would strongly activate the feature. The corpus distribution also shapes which activating patterns are available for retrieval, potentially overrepresenting particular contexts or co-occurring properties. Consequently, $e_{\mathcal{D}}$ can describe the feature only through patterns represented in $\mathcal{D}$. This dependence motivates an evidence acquisition mechanism that is not restricted to retrieval from a fixed corpus.

2.3 Inversion for Feature Interpretation

To expand the evidence available to the interpreter $\mathscr{A}$, we seek new text samples guided by the target feature's activation strength. Feature inversion provides this capability by reversing the direction of the mapping $t \mapsto s(t)$. While corpus retrieval evaluates existing texts and selects those with large activation strengths, inversion uses feature activation strength to guide the generation of texts beyond $\mathcal{D}$. Since s is generally many to one, the same activation strength can correspond to texts with different wording, sentence structures, and meanings. We therefore model inversion as a sampling process that can provide multiple samples for explanation.

To serve as evidence for $\mathscr{A}$, these generated samples should satisfy two requirements. They should be understandable to humans, so that their text patterns can serve as evidence. They should also be faithful to the target feature, so that these patterns are associated with its measured activation strength. We encode the first requirement through a natural language prior and the second through the feature activation strength function s. Together, they determine the distribution from which inversion should sample.

Following the KL-minimal soft-constraint principle of Zhou et al. (2026), we express these requirements through a distributional optimization problem. Let p be a natural-language prior on $\mathcal{X}$. We seek a text distribution that remains close to p while assigning greater probability to samples with stronger feature activation.

Definition 1 (Feature inversion distribution)**.** Let $\beta \ge 0$ control the strength of feature guidance, and assume that $Z_p(\beta) = \mathbb{E}_p[e^{\beta s(T)}]$ is finite. The feature inversion distribution induced by p, s, and β is

$$q_p^\beta \in \arg\min_{q \ll p} \{D_{\text{KL}}(q \| p) - \beta \mathbb{E}_q[s(T)]\}, \tag{3}$$

where $q \ll p$ requires q to assign probability only to texts supported by p.

The two terms in Eq. (3) play complementary roles: the KL term keeps the generated distribution close to the natural-language prior, while the second term favors samples with greater feature activation strength. The coefficient β determines the strength of this feature guidance. This KL-regularized objective admits a closed-form exponentially tilted solution as below.

Theorem 1 (Gibbs form of the inversion target (Zhou et al., 2026)). *Under the assumptions of Definition 1, the unique solution of Eq.* (3) *is* $q_p^\beta(t) = \frac{p(t)e^{\beta s(t)}}{Z_p(\beta)}$, *where* $Z_p(\beta) = \mathbb{E}_p[e^{\beta s(T)}]$.

The prior p favors natural text, while $e^{\beta s(t)}$ increases the relative weight of samples with greater feature activation strength. The parameter β controls the balance between these two requirements.

We can now augment corpus retrieval in the explanation pipeline with samples from q_p^β. Specifically, draw

$$T_1, \ldots, T_N \overset{\text{iid}}{\sim} q_p^\beta \qquad \text{and define} \qquad \mathcal{S}_N(s, p, \beta) := \{T_1, \ldots, T_N\}. \tag{4}$$

The generated evidence can be combined with the retrieved set $\mathcal{C}_N(\mathcal{D}, s)$ in the interpreter $\mathscr{A}$, as developed in Section 2.5. This expands the evidence available for explanation. Samples are no longer restricted to texts in $\mathcal{D}$, although their coverage still depends on the chosen prior p and the sampling procedure.

Training-based inversion offers one approach to sampling (Luo et al., 2025), but it requires additional computation to train a separate generator and can inherit biases from its training corpus. Appendix B analyzes how these biases can persist even when the conditional generator is learned exactly. We therefore seek a training-free sampling procedure that approximates q_p^β without training a dedicated inverse model. The next subsection develops such a method using a frozen pretrained language model as the sampling prior.

2.4 DPS for SAE Feature Inversion

To construct the evidence set $\mathcal{S}_N(s, p, \beta)$ without training a dedicated inverse generator, we need both a natural-language prior and a sampling mechanism from its activation-strength-weighted target. DPS provides such a mechanism by guiding a pretrained diffusion prior toward a prescribed objective. This method is closely related to diffusion-based inverse problems (Chung et al., 2023), originally developed in the continuous visual domain and later extended to discrete diffusion models (Rout et al., 2026; Tejaswi et al., 2026). For language model SAE features, we use a frozen DLM as the natural-language prior. Its large-scale pretraining on diverse text corpora provides a practical approximation $p_\theta \approx p$, yielding $q_{p_\theta}^\beta$ as the corresponding feature inversion target.

Connection to the inversion target. Theorem 1 defines the inversion target by reweighting the prior probability $p_\theta(t)$ of each completed text t by its activation weight $e^{\beta s(t)}$, followed by normalization. To realize this target through the DLM's successive denoising steps, this weighting must be propagated through intermediate states. At such a state, the text is still incomplete and its final activation strength cannot yet be evaluated. The next transition should therefore be guided by the activation weights of the completed texts reachable from the current state.

Let $X_K, \ldots, X_0$ denote the DLM denoising sequence, starting from a fixed masked state X_K. We use a stochastic reference chain defined by the DLM token probabilities. Any history needed by the update schedule is included in the state. Let P_θ denote the law of this reference chain. Its transition $P_{\theta,k}(x' \mid x)$ gives the DLM probability of moving from x at step k to x' at step $k-1$. Let d decode a completed DLM token sequence into continuation text, omitting special tokens. The prior p_θ is the law of $d(X_0)$; accordingly, $s(X_0)$ abbreviates $s(d(X_0))$. To guide an intermediate state x, consider the average activation weight of the texts obtained by completing x with the base DLM,

$$V_k(x) = \mathbb{E}_{P_\theta}\left[e^{\beta s(X_0)} \mid X_k = x\right], \qquad V_0(x) = e^{\beta s(x)}, \tag{5}$$

Thus, $V_k(x)$ evaluates an incomplete sequence through its possible completed texts. It averages their exponential weights $e^{\beta s(X_0)}$, rather than their activation strengths alone. These values determine how to adjust the probability of each next state.

Theorem 2 (Exact inversion under ideal guided denoising). *Suppose s takes finite real values on supported texts and $Z_{p_\theta}(\beta) < \infty$. Using the exact values from Eq.* (5)*, define the feature guided*

transition kernel from state x at step k to state x' at step $k-1$ by

$$P_k^\star(x' \mid x) = P_{\theta,k}(x' \mid x)\frac{V_{k-1}(x')}{V_k(x)}. \tag{6}$$

Then $P_k^\star(\cdot \mid x)$ is a normalized conditional probability on reachable states. Starting from the fixed masked state X_K and sampling successively from these kernels produces a decoded final text with law $q_{p_\theta}^\beta$.

The rule has three parts. The base probability $P_{\theta,k}(x' \mid x)$ preserves the DLM's generation preferences. The factor $V_{k-1}(x')$ increases the relative probability of next states whose completed texts have greater average activation weight. The denominator $V_k(x)$ ensures that the probabilities sum to one. Applying this adjustment at every step produces exactly $q_{p_\theta}^\beta$, not merely texts with large activation strengths. In this way, Theorem 2 turns the target weighting in Theorem 1 into a stepwise generation rule. The proof in Appendix C shows how the intermediate value factors cancel, leaving the required weight on the completed text.

Estimating the guidance value. The exact kernel in Theorem 2 requires a conditional expectation over future completions. At the current state x, form a provisional completion by retaining visible tokens and filling the masks with the DLM's argmax predictions. For a proposed successor x' that sets one token, $\widehat{t}_{k,x}(x')$ replaces that token in the shared completion. The other predicted tokens are not recomputed after this replacement. We use the plug-in value $\widehat{V}_{k-1}(x'; x) := \exp\{\beta s(\widehat{t}_{k,x}(x'))\}$. This approximation replaces the distribution of terminal texts by one predicted completion. Its relation to the exact value can be expressed as

$$\log \mathbb{E}_{P_\theta}[e^{\beta s(X_0)} \mid X_{k-1} = x'] \;\approx\; \beta \mathbb{E}_{P_\theta}[s(X_0) \mid X_{k-1} = x'] \;\approx\; \beta s(\widehat{t}_{k,x}(x')). \tag{7}$$

The first approximation neglects higher conditional cumulants of the terminal score. When its conditional moment-generating function exists near zero, it is the first-order expansion in β at zero. The second uses the predicted completion to estimate the conditional mean score. An argmax completion need not attain that mean, so neither step is an identity. Substitution into the exact kernel, with explicit renormalization, gives the surrogate transition

$$\widetilde{P}_k(x' \mid x) = \frac{P_{\theta,k}(x' \mid x)\widehat{V}_{k-1}(x'; x)}{\sum_y P_{\theta,k}(y \mid x)\widehat{V}_{k-1}(y; x)}. \tag{8}$$

Thus predicted-terminal scoring approximates the value factor in Eq. (6). The denominator is the normalization of those approximate weights, rather than a separately estimated $V_k(x)$.

From the surrogate kernel to candidate search. Equation (8) combines the DLM prior and the predicted terminal feature score. In particular, its most probable successor solves

$$\operatorname*{arg\,max}_{x':\, P_{\theta,k}(x'|x)>0} \widetilde{P}_k(x' \mid x) = \operatorname*{arg\,max}_{x':\, P_{\theta,k}(x'|x)>0} \left\{\log P_{\theta,k}(x' \mid x) + \beta s(\widehat{t}_{k,x}(x'))\right\}. \tag{9}$$

This identity exposes the balance between prior probability and feature activation that motivates Algorithm 1. The algorithm replaces probabilistic weighting by a constrained search: DLM probabilities propose and initialize candidates, and likelihood filtering restricts which searched candidates remain eligible, rather than contributing an additive log-prior term to the final ranking. Within this search, EntRGi surrogate gradients (Tejaswi et al., 2026) optimize a differentiable pre-TopK feature reward on provisional completions. Retained candidates are then evaluated as hard texts and ranked by R_{hard}, with the current unmasked token retained as a baseline. The relaxation, groupwise likelihood filter, and hard ranking are specified in Appendix D and Algorithm 2.

Toy experiment. Toy experiments (Figure 1) demonstrate that inverse-based recovery can fail to recover the desired target distribution due to two distinct factors: bias in the training prior and approximation error of the learned sampler. A four-component Gaussian mixture defines the natural prior p, while the training prior p_{tr} introduces controlled shifts in component weights. We parameterize the inverse sampler as an MLP generator $f_\phi(a, z)$, where a denotes the target activation level and $z \sim \mathcal{N}(0, I_2)$ is 2-D Gaussian noise. The generator is trained to produce samples matching the

Algorithm 1 Training-free DPS-guided search for SAE feature inversion

Require: Frozen DLM p_θ, feature scorer s, prompt c, length L, steps B
Initialize $x \leftarrow [\text{MASK}]^L$
for $b = 1, \ldots, B$ **do**
    Select update position i and obtain DLM candidate distribution
    Complete masked positions with DLM predictions
    Generate candidate tokens from the DLM proposal distribution
    Evaluate candidates using feature-guided search with activation score s
    Select the candidate balancing DLM likelihood and feature activation
    Update x_i with the selected token
end for
Decode final text $t \leftarrow d(x)$
return $(t, s(t))$

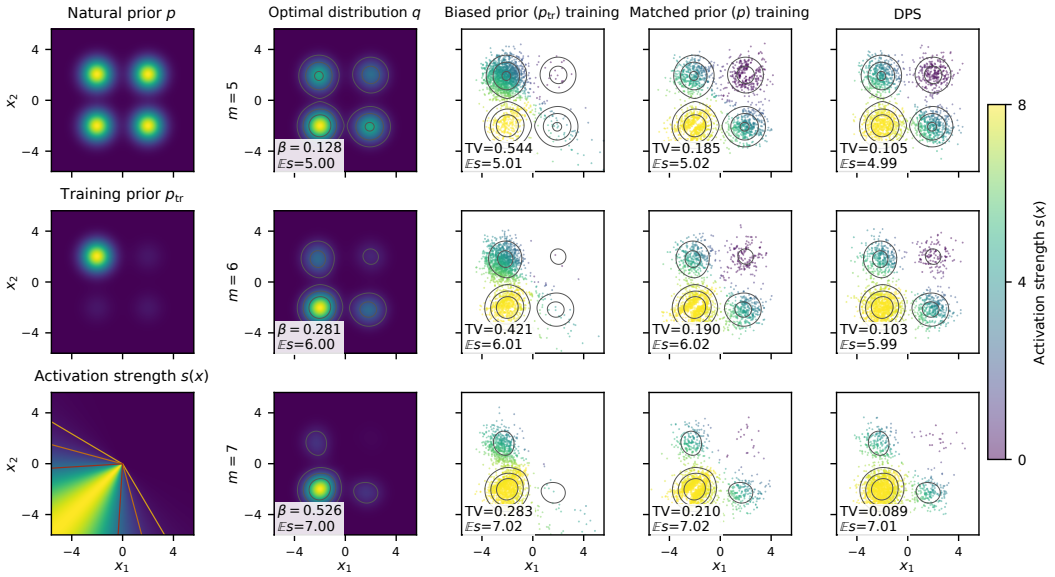

Figure 1: Toy experiment showing that inverse samplers inherit training-prior bias. A shifted training prior $p_{\text{tr}} \neq p$ leads to samples deviating from the desired target $q \propto p(x)e^{\beta s(x)}$, while matching priors is limited only by sampler approximation error. Columns compare the optimal distribution q, MLP samplers trained on p_{tr} or p, and continuous DPS with analytic p under matched mean activation levels $m = 5, 6, 7$. Gray contours denote q, sample colors indicate activation strength $s(x)$, and annotations report mean activation and binned TV distance to q.

corresponding exponentially tilted distribution using samples drawn from either p_{tr} (biased prior training) or p (matched prior training). Thus, the two training settings differ only in the prior distribution of the training samples. $q(x) \propto p(x)e^{\beta s(x)}$. For each target mean activation $m \in \{5, 6, 7\}$, β is separately calibrated such that all methods achieve approximately the same mean activation. Figure 1 compares continuous DPS with analytic p against inverse samplers trained under either a shifted prior p_{tr} or the matched prior p. When $p_{\text{tr}} \neq p$, the learned inverse inherits the training-prior bias, sampling from a distribution closer to $q^{\beta}_{p_{\text{tr}}}$ rather than the desired q^{β}_{p}, consistent with Proposition 1. Even when $p_{\text{tr}} = p$, the learned inverse exhibits additional approximation error due to finite training and model capacity. These results separate prior-induced distortion from inverse approximation error. Appendix E provides further details.

From generated samples to explanations. Independent runs of Algorithm 1 produce a set of feature-conditioned texts $\widehat{\mathcal{S}}_N = \{\widehat{T}_1, \ldots, \widehat{T}_N\}$, where activation strengths are recomputed on the final discrete texts. This set serves as a practical approximation to the ideal evidence set $\mathcal{S}_N(s, p, \beta)$ defined in Eq. (4). Unlike corpus retrieval, which is restricted to existing samples, the proposed procedure explores new texts guided by the target activation distribution. The resulting evidence reflects the combined effects of the pretrained prior, the diffusion-based search process, and the

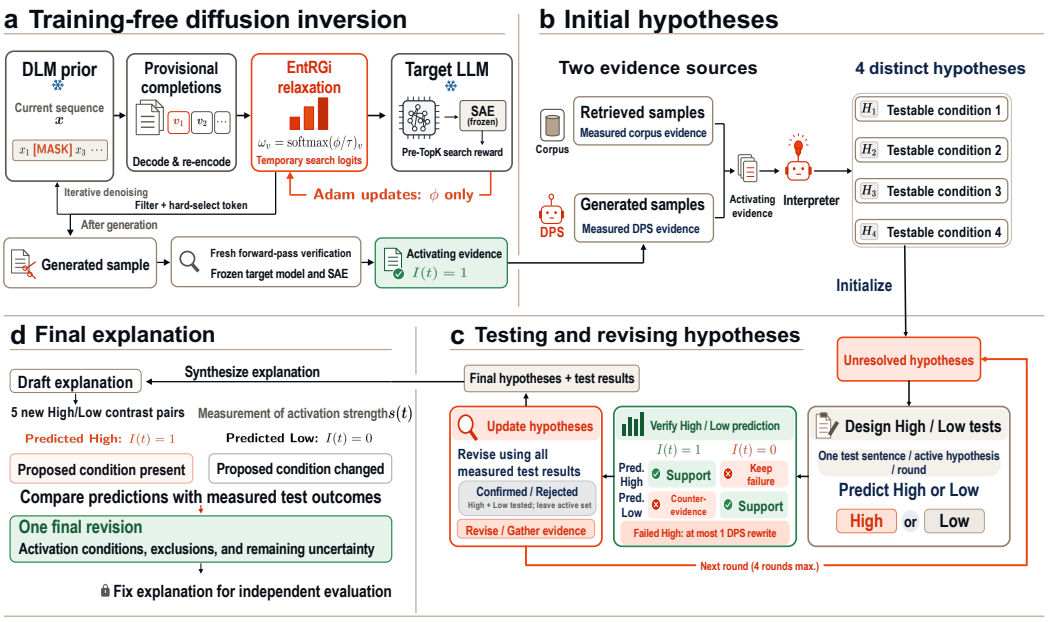

Figure 2: **InvExplain combines training-free feature inversion with closed-loop hypothesis testing.** **(a)** DPS guides a frozen DLM prior to generate samples that activate the target feature. Generated samples are added to the evidence set only after verification by the frozen target LLM and SAE. **(b)** Corpus-retrieved and inversion-generated samples provide complementary evidence from which the agent forms competing hypotheses about the target feature. **(c)** The agent iteratively tests these hypotheses with predicted-High and predicted-Low samples. A failed predicted-High test can trigger one DPS rewrite, while all measured outcomes are retained as evidence for confirming, rejecting, or refining the hypotheses. **(d)** The resulting hypotheses and test results are synthesized into an explanation.

discrete text generation procedure. The interpreter $\mathscr{A}$ then uses $\widehat{\mathcal{S}}_N$, together with retrieved samples when available, to identify text patterns associated with the activated feature.

2.5 InvExplain: Inversion-Guided Agentic Interpretation Framework

Section 2.2 represents corpus-based interpretation as $e_{\mathcal{D}} = \mathscr{A}(\mathcal{C}_N(\mathcal{D}, s))$. InvExplain (Fig. 2) augments the corpus evidence available to the interpreter with samples acquired through feature inversion. Its explanation is $e_{\text{InvExplain}} = \mathscr{A}\left(\widehat{\mathcal{S}}_N, \mathcal{C}_N(\mathcal{D}, s)\right)$. Here, $\widehat{\mathcal{S}}_N$, defined above, contains DPS-generated samples and $\mathcal{C}_N(\mathcal{D}, s)$ contains corpus-retrieved samples. Following Han et al. (2026), we implement $\mathscr{A}$ as an agent that forms candidate explanations, tests their predictions against measured feature activation strengths, and iteratively revises them based on the resulting evidence.

Forming hypotheses from two evidence sources. The agent receives corpus-retrieved samples $\mathcal{C}_N(\mathcal{D}, s)$ and inversion-generated samples $\widehat{\mathcal{S}}_N$, together with their measured feature activation strengths $s(t)$. We additionally define a binary activation indicator $I(t)$, which equals one if the target feature survives TopK sparsification at any scored token position and zero otherwise. During testing, the agent predicts whether a sample will be *High* ($I(t) = 1$) or *Low* ($I(t) = 0$), and each prediction is verified by a forward pass through the frozen target model and SAE. Generated samples with $I(t) = 1$ provide additional activating evidence beyond corpus retrieval. The agent also receives the local text around the position of maximum feature activation strength and uses the combined evidence to formulate four distinct, testable hypotheses about the text patterns associated with the feature.

Testing and revising hypotheses. For each active hypothesis, the agent constructs a test sentence and predicts whether it will be *High* or *Low* . A *High* test contains the proposed activating condition, whereas a *Low* test modifies or removes that condition while preserving the surrounding context.

Table 1: Explanation accuracy on the shared challenge sets. N is the number of prompts per method; all scores are percentages. Pooled scores aggregate integer counts over 110 features. Bold marks the best score in each block. Layer-level results are in Table 12.

Target	Method	N	Activation	*High* loc.	Joint
Llama-3.2-1B	SAGE	1,200	70.4	28.2	49.8
	InvExplain	1,200	**81.9**	**77.5**	**76.8**
	Neuronpedia-style	1,200	63.2	58.6	58.2
Llama-3.2-3B	SAGE	1,200	70.2	39.5	49.9
	InvExplain	1,200	**83.3**	**75.6**	**74.3**
	Neuronpedia-style	1,200	57.1	58.5	50.4
Qwen3-1.7B	SAGE	900	70.0	55.3	63.7
	InvExplain	900	**82.4**	**67.2**	**75.6**
	Neuronpedia-style	900	65.2	53.6	60.7
Pooled (110 features)	SAGE	3,300	70.24	40.03	53.64
	InvExplain	3,300	**82.58**	**74.09**	**75.58**
	Neuronpedia-style	3,300	61.52	56.94	56.03

Each prediction is verified through a forward pass of the frozen target model and SAE, allowing the agent to distinguish the proposed condition from co-occurring patterns in the initial evidence. When a predicted-*High* test fails to activate the feature, DPS generates at most one rewrite $\widetilde{t}$ conditioned on the failed test. Comparing t and $\widetilde{t}$ provides additional evidence for refining the hypothesis. The agent updates hypotheses using all accumulated test results until they are confirmed, rejected, or require further revision.

Final explanation. The agent synthesizes the tested hypotheses and measured test results into an explanation, including activation conditions, exclusions, and remaining uncertainty. A separate call receives this explanation and designs five new *High*/*Low* contrast pairs. Their measured activation strengths inform one final revision. The resulting explanation is then fixed for independent evaluation. Its public record contains a description, activating pattern, activation envelope, verified exclusions, labels, uncertainties, and confidence. Thus, interpreter $\mathscr{A}$ uses both evidence sources to form hypotheses and uses measured feature activation strength to test the explanation they support.

3 Experiments

3.1 Experimental Setup

Model settings. The target models are Llama-3.2-1B, Llama-3.2-3B (Meta AI, 2024), and Qwen3-1.7B (Yang et al., 2025). For each model, features come from Top-K transcoders (an SAE variant) that encode MLP inputs and reconstruct MLP outputs (Dunefsky et al., 2024). Each transcoder has eightfold expansion and was trained on 800 M tokens. Its dimensions and Top-K setting are listed in Appendix G. Feature inversion uses frozen iLLaDA-base as the diffusion language model prior (Nie et al., 2026).

Baseline setting. The baselines are SAGE (Han et al., 2026) and a Neuronpedia baseline explanation method (Lin, 2025). SAGE is the no-inversion agentic baseline: it tests hypotheses using corpus evidence without DPS-generated samples or rewrites. Ten alive features per layer are fixed before explanation generation. The completed evaluation covers 110 features across 11 model–layer settings. For each feature, a generator LLM writes five intended *High*/*Low* text pairs from each method's frozen explanation, forming a shared 30-text challenge set. An evaluator LLM then uses each frozen explanation to predict whether the feature activates on every text and, for predicted-*High* texts, where its activation peaks. The target model and transcoder provide the measured reference labels and peak spans. In Table 1, N is the number of shared challenge texts evaluated per method. *Activation* denotes accuracy on the *High*/*Low* decision; *High loc.* denotes the fraction of predicted-*High* texts whose predicted peak span overlaps the reference span; and *Joint* denotes the fraction of all texts with either a correct *Low* prediction or a correct *High* prediction and overlapping peak span. Appendix G gives the full protocol and metric formulas.

Table 2: Frozen explanations for two SAE features. **Blue** marks activating patterns found by InvExplain beyond corpus evidence; **Red** marks contrastive probes. Row-end values give shared-pool joint accuracy for InvExplain / SAGE / Neuronpedia-style.

LLM layer/feature	Neuronpedia explanation	SAGE explanation	InvExplain (ours) explanation
Llama-3.2-1B 10/14708	Say `periods`.	Clause-ending punctuation, usually a period and sometimes a comma, near a fixed span position; a formatting rather than semantic pattern.	The CJK enumeration comma 、 activates in neutral generations but is absent from retrieved samples. Replacing the separator with 和 in the same frame weakens the response; so does an ordinary prose comma. joint 80.0 / 45.0 / 45.0
Qwen3-1.7B 20/6114	`his`.	Possessive determiner `his` in a noun phrase, typically referring to a male person.	Possessiveness, not male reference: `her` and post-nominal `of his` also activate. Non-possessive `the` in matched noun phrases is inactive. joint 90.0 / 66.7 / 86.7

3.2 Feature Interpretation Results

Table 1 shows that InvExplain outperforms SAGE and the Neuronpedia-style baseline on all three metrics for each target model. Its pooled *High* localization gains are 34.06 percentage points over SAGE and 17.15 percentage points over Neuronpedia-style. Table 2 shows two changes supported by inversion-derived evidence: a separator symbol absent from retrieval for the Llama-3.2-1B feature, and possessiveness rather than male reference for the Qwen3-1.7B feature. Both explanations achieve higher shared-pool joint accuracy than the corpus-based baselines. More qualitative examples are shown in Appendix G.1.

3.3 Ablation Studies

We vary the initial generation budget N_{init}, EntRGi optimization steps M, and maximum base-logit drop D one at a time. All settings use the same ten layer-10 features, 80 prompts per feature, and 48 diffusion updates per generation. Initial rescue is disabled; conditional rewrites remain enabled. Table 3 shows that increasing M improves *High* localization and joint accuracy over the displayed range. The selected setting also outperforms the smaller N_{init} and tighter D settings. Appendix G provides further experimental details, additional ablations, and comparisons across DLM variants.

Table 3: Inversion-search ablation on shared challenge prompts. We study the effects of the initial generation budget N_{init}, EntRGi optimization steps M, and maximum allowed base-logit drop D. Scores are percentages; bold indicates the selected configuration.

N_{init}	M	D	Activation	*High* loc.	Joint
3	12	5	81.2	82.1	76.1
3	**12**	**20**	**84.2**	**83.3**	**79.9**
1	12	20	83.6	80.8	77.9
3	**12**	**20**	**84.2**	**83.3**	**79.9**
3	8	20	81.8	82.1	76.8
3	**12**	**20**	**84.2**	**83.3**	**79.9**

4 Conclusions

Corpus retrieval can miss texts that activate a feature but do not appear in the corpus. We address that limitation by formulating feature inversion as sampling from a natural-language prior guided by feature activation strength. InvExplain uses training-free diffusion to generate new samples, verifies their activation, and combines them with retrieved samples to test and refine explanations. The analysis shows how training data can bias a learned inverse, while experiments across three language models show more accurate predictions of feature activation and its location. These results support feature inversion as a useful complement to retrieval for feature interpretation.

AI USE STATEMENT

AI was used to polish writing and assist coding. We have reviewed all AI-assisted work. The authors take responsibility for the final content of this work.

REFERENCES

Bereska, L. and Gavves, E. Mechanistic interpretability for AI safety – a review. *Transactions on Machine Learning Research (TMLR)*, 2024.

Bhattacharyya, C., Lee, H., Lee, J., Jang, S., Suh, I. h., and Kim, Y. FineScope: SAE-guided data selection enables domain specific LLM pruning and finetuning. *arXiv preprint arXiv:2505.00624*, 2026. URL https://arxiv.org/abs/2505.00624.

Bills, S., Cammarata, N., Mossing, D., Tillman, H., Gao, L., Goh, G., Sutskever, I., Leike, J., Wu, J., and Saunders, W. Language models can explain neurons in language models. OpenAI technical report, 2023. URL https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Brixi, G., Durrant, M. G., Ku, J., Naghipourfar, M., Poli, M., Sun, G., et al. Genome modelling and design across all domains of life with Evo 2. *Nature*, 652:1349–1361, 2026. doi: 10.1038/s41586-026-10176-5.

Carnemolla, S., Pennisi, M., Samarasinghe, S., Bellitto, G., Palazzo, S., Giordano, D., Shah, M., and Spampinato, C. DEXTER: Diffusion-guided EXplanations with TExtual reasoning for vision models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Chung, H., Kim, J., McCann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023.

Deng, B., Wang, X., Wang, Y., Wan, Y., Ma, Y., Yang, B., Wei, H., Tang, J., Lin, H., Gao, R., Li, T., Cao, Q., Ren, X., Deng, X., Yang, A., Huang, F., Liu, D., and Zhou, J. Qwen-Scope: Turning sparse features into development tools for large language models. *arXiv preprint arXiv:2605.11887*, 2026. URL https://arxiv.org/abs/2605.11887.

Dunefsky, J., Chlenski, P., and Nanda, N. Transcoders find interpretable LLM feature circuits. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. doi: 10.52202/079017-0768.

Gao, L., Dupré la Tour, T., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations (ICLR)*, 2025.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association (JASA)*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Han, J., Xu, W., Jin, M., and Du, M. SAGE: An agentic explainer framework for interpreting SAE features in language models. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL), Industry Track*, pp. 483–495, 2026. doi: 10.18653/v1/2026.eacl-industry.37.

Huben, R., Cunningham, H., Smith, L., Ewart, A., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Li, Z., Wu, X., Li, Y., Hu, L., and Liu, N. Less is enough: Synthesizing diverse data in feature space of LLMs. In *International Conference on Machine Learning (ICML)*, 2026.

Lin, J. Neuronpedia: Interactive reference and tooling for analyzing neural networks, 2023. URL https://www.neuronpedia.org. Software available from neuronpedia.org.

Lin, J. Circuit tracer + new auto-interp method. Neuronpedia Blog: The Residual Stream, June 2025. URL https://www.neuronpedia.org/blog/circuit-tracer.

Lovelace, J., Kishore, V., Chen, Y., and Weinberger, K. Diffusion guided language modeling. In *Findings of the Association for Computational Linguistics (ACL)*, pp. 14936–14952, 2024. doi: 10.18653/v1/2024.findings-acl.887.

Luo, Y., Zhou, Z., and Dong, B. InverseScope: Scalable activation inversion for interpreting large language models. *arXiv preprint arXiv:2506.07406*, 2025. URL https://arxiv.org/abs/2506.07406.

Meng, W., Guo, H., Jing, Y., Liu, D., Yao, Z., Wang, X., Hou, L., and Li, J. SAEVerbalizer: Generating explanations for sparse autoencoder features via representation verbalization. *arXiv preprint arXiv:2608.13538*, 2026. URL https://arxiv.org/abs/2608.13538.

Meta AI. Llama 3.2 model card, 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., and Li, C. Large language diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. doi: 10.52202/085713-1689.

Nie, S., Min, Q., Xu, S., Huang, Z., Song, Y., Shan, Y., Lin, Y., Zhao, W. X., Li, C., and Wen, J.-R. Improved large language diffusion models. *arXiv preprint arXiv:2606.25331*, 2026. URL https://arxiv.org/abs/2606.25331.

Paulo, G. S., Mallen, A. T., Juang, C., and Belrose, N. Automatically interpreting millions of features in large language models. In *International Conference on Machine Learning (ICML)*, volume 267, pp. 48393–48421, 2025.

Pennisi, M., Bellitto, G., Palazzo, S., Kavasidis, I., Shah, M., and Spampinato, C. DiffExplainer: Towards cross-modal global explanations with diffusion models. *Computer Vision and Image Understanding (CVIU)*, 262:104559, 2025. doi: 10.1016/j.cviu.2025.104559.

Rathi, N. and Radford, A. Shaping capabilities with token-level data filtering. *arXiv preprint arXiv:2601.21571*, 2026. URL https://arxiv.org/abs/2601.21571.

Rosenfeld, K. and Sonnewald, M. Sparse probes and murky physics: a case study of interpretability challenges in a foundation model for continuum dynamics. In *International Conference on Learning Representations (ICLR) Workshop on Foundation Models for Science*, 2026.

Rott Shaham, T., Schwettmann, S., Wang, F., Rajaram, A., Hernandez, E., Andreas, J., and Torralba, A. A multimodal automated interpretability agent. In *International Conference on Machine Learning (ICML)*, volume 235, pp. 44293–44321, 2024.

Rout, L., Lugmayr, A., Jafarian, Y., Varadharajan, S., Caramanis, C., Shakkottai, S., and Kemelmacher-Shlizerman, I. Test-time anchoring for discrete diffusion posterior sampling. In *International Conference on Machine Learning (ICML)*, 2026.

Schwettmann, S., Rott Shaham, T., Materzynska, J., Chowdhury, N., Li, S., Andreas, J., Bau, D., and Torralba, A. FIND: A function description benchmark for evaluating interpretability methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Shi, L., Wu, X., Zhao, X., Wang, H., Liu, H., Liu, Y., Xu, L., Wang, L., Xiong, D., and Luo, W. From insight to action: A novel framework for interpretability-guided data selection in large language models. In *Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pp. 6258–6272, 2026. doi: 10.18653/v1/2026.acl-long.283.

Simon, E. and Zou, J. InterPLM: discovering interpretable features in protein language models via sparse autoencoders. *Nature Methods*, 22(10):2107–2117, 2025. doi: 10.1038/s41592-025-02836-7.

Tejaswi, A., Rout, L., Caramanis, C., Shakkottai, S., and Sanghavi, S. Entropy aware reward guidance for diffusion language model alignment. *arXiv preprint arXiv:2602.05000*, 2026. URL https://arxiv.org/abs/2602.05000. Extended preprint; an earlier version appeared at the ReALM-GEN Workshop at ICLR 2026.

Tsui, D., Talreja, K., Saeedi, D., and Aghazadeh, A. Protein circuit tracing via cross-layer transcoders. In *International Conference on Machine Learning (ICML)*, 2026.

Venkatraman, S., Jain, M., Scimeca, L., Kim, M., Sendera, M., Hasan, M., Rowe, L., Mittal, S., Lemos, P., Bengio, E., Adam, A., Rector-Brooks, J., Bengio, Y., Berseth, G., and Whitammer, E. S. Amortizing intractable inference in diffusion models for vision, language, and control. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. doi: 10.52202/079017-2422.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. Understanding neural networks through deep visualization. In *International Conference on Machine Learning (ICML) Workshop on Deep Learning*, 2015.

Zhou, G., Luo, Y., He, Z., Jin, Z., Ge, X., Shu, W., Meng, D., and Qiu, X. A distributional view for visual mechanistic interpretability: KL-minimal soft-constraint principle. In *International Conference on Machine Learning (ICML)*, 2026.

A Related Work

A.1 Sparse Autoencoders for Mechanistic Interpretation

Sparse autoencoders (SAEs) learn sparse features that reconstruct model activations, offering units of analysis finer than individual neurons (Huben et al., 2024; Bricken et al., 2023). Top-K SAEs enforce a fixed number of active features per input, making sparsity an explicit architectural choice (Gao et al., 2025). Transcoders instead encode an MLP's input and reconstruct its output, so their features can also be used to study the MLP's contribution to model computation (Dunefsky et al., 2024). These methods identify and measure features, but a feature's activation pattern still needs to be expressed as an understandable explanation.

SAE features are also used to select and synthesize training data. Rathi & Radford (2026) use SAE-based token labels to filter pretraining data, reducing an unwanted capability while preserving more benign data than document-level filtering. Li et al. (2026) identify features missing from a seed dataset and synthesize samples that cover them. Other work selects finetuning samples that activate task-related features (Shi et al., 2026), uses SAE-guided data selection for domain-specific pruning and finetuning (Bhattacharyya et al., 2026), and applies SAE features to safety data synthesis and post-training (Deng et al., 2026). These applications make it important to know what a feature actually responds to, since an incomplete explanation can misdirect the selection or generation of data.

This need extends to scientific foundation models, where a feature may encode a biological pattern rather than a linguistic one. InterPLM uses SAEs to identify protein features associated with motifs, domains, and other structural or functional properties (Simon & Zou, 2025). Cross-layer transcoders have been used to trace protein-model circuits related to structure and function (Tsui et al., 2026). In genomics, SAE features of Evo 2 align with exon boundaries, transcription factor binding motifs, and other biological annotations (Brixi et al., 2026). These results broaden the settings in which feature explanations must be both understandable and faithful to model behavior.

A.2 Feature Inversion

Feature inversion generates inputs that elicit a target model response. In vision, early work optimized images to visualize individual network units (Yosinski et al., 2015). DiffExplainer uses text-conditioned diffusion to synthesize images that activate classifier outputs or hidden features (Pennisi et al., 2025), while DEXTER uses generated images to explain classifier decisions and biases in language (Carnemolla et al., 2025). These methods primarily target network units or classifier behavior rather than sparse SAE features. EnergyDPS instead studies visual SAE features and frames inversion as sampling from a natural-image prior tilted toward higher feature activation strength (Zhou et al., 2026). For language models, InverseScope learns to generate text near a target internal activation (Luo et al., 2025). The present work focuses on SAE feature activation strength and obtains samples with a pretrained text prior without training a separate inverse model. It then uses those samples alongside retrieved evidence to produce feature explanations.

A.3 Automated Interpretation of Sparse Features

Automated interpretation uses language models to turn activation evidence into natural language explanations. Early work generated explanations from highly activating texts and assessed whether those explanations could predict activations on other texts (Bills et al., 2023). Neuronpedia provides access to top-activating samples and automated descriptions for large collections of features (Lin, 2023). A Neuronpedia-style one-pass explanation uses retrieved samples to describe a feature without an iterative testing stage. It serves as the one-pass baseline in our evaluation. FIND evaluates automated descriptions on constructed functions with known behavior, including functions in textual and numerical domains (Schwettmann et al., 2023). Its interactive agent probes a function before describing it, showing the value of tests that can distinguish a broad description from the function's more specific behavior. MAIA equips a vision-language agent with tools for inspecting visual model components (Rott Shaham et al., 2024). It combines dataset examples with synthesized and edited images to test candidate explanations. This demonstrates active evidence collection in vision, rather than reliance on retrieved examples alone. SAGE applies iterative hypothesis testing to SAE features in language models (Han et al., 2026). Starting from high-activation corpus samples,

it proposes several hypotheses, designs tests, and revises explanations using measured activation feedback. Its initial evidence remains tied to corpus retrieval. InvExplain adds activation-guided generated samples to that evidence and uses them during subsequent tests.

B Training-Based Feature Inversion

Let p_{tr} be the training prior and $\rho_{\mathrm{tr}} = s_{\#}p_{\mathrm{tr}}$ its activation-strength law. An ideal conditional inverse trained on this prior samples from $p_{\mathrm{tr}}(dx \mid s = a)$ when given activation strength a. The following result states what happens when its conditioning strengths are selected to favor larger activation.

Proposition 1 (Effect of the training prior)**.** *Assume regular conditional laws exist and $Z_{p_{\mathrm{tr}}}(\beta) < \infty$. Draw A from*

$$\nu^{\beta}_{\mathrm{tr}}(da) = \frac{e^{\beta a}\rho_{\mathrm{tr}}(da)}{Z_{p_{\mathrm{tr}}}(\beta)}, \tag{10}$$

then draw X from $p_{\mathrm{tr}}(dx \mid s = A)$. The resulting law is $q^{\beta}_{p_{\mathrm{tr}}}$, not q^{β}_{p} unless these targets coincide. If $p_{\mathrm{tr}} \ll p$, their density ratio is

$$\frac{dq^{\beta}_{p_{\mathrm{tr}}}}{dq^{\beta}_{p}}(x) = \frac{dp_{\mathrm{tr}}}{dp}(x)\frac{Z_p(\beta)}{Z_{p_{\mathrm{tr}}}(\beta)} \quad q^{\beta}_{p}\text{-almost everywhere}. \tag{11}$$

Proof. For any measurable set B, disintegration gives

$$\int p_{\mathrm{tr}}(B \mid s = a)\nu^{\beta}_{\mathrm{tr}}(da) = \frac{\int_B e^{\beta s(x)}p_{\mathrm{tr}}(dx)}{Z_{p_{\mathrm{tr}}}(\beta)} = q^{\beta}_{p_{\mathrm{tr}}}(B). \tag{12}$$

Comparing its density with the Gibbs density in Theorem 1 yields Eq. (11). □

For completeness, an arbitrary conditioning-strength law $\nu \ll \rho_{\mathrm{tr}}$ produces $q_\nu(dx) = \int p_{\mathrm{tr}}(dx \mid s = a)\nu(da)$. Disintegration also gives

$$D_{\mathrm{KL}}(q_\nu \| p_{\mathrm{tr}}) = D_{\mathrm{KL}}(\nu \| \rho_{\mathrm{tr}}), \qquad \mathbb{E}_{q_\nu}[s] = \mathbb{E}_{A\sim\nu}[A]. \tag{13}$$

Thus choosing the tilted law in Eq. (10) implements the same soft-constraint target as Definition 1, but relative to the training prior.

Matched mean activation in the toy experiment. The toy compares samplers at a common mean activation m rather than a common β. Let $\beta_p(m)$ and $\beta_{\mathrm{tr}}(m)$ be finite nonnegative coefficients satisfying $\mathbb{E}_{q_p^{\beta_p(m)}} s = \mathbb{E}_{q_{p_{\mathrm{tr}}}^{\beta_{\mathrm{tr}}(m)}} s = m$. Write $q = q_p^{\beta_p(m)}$ and $q_{\mathrm{tr}} = q_{p_{\mathrm{tr}}}^{\beta_{\mathrm{tr}}(m)}$. When $p_{\mathrm{tr}} \ll p$, their ratio is

$$\frac{dq_{\mathrm{tr}}}{dq}(x) = \frac{dp_{\mathrm{tr}}}{dp}(x)e^{[\beta_{\mathrm{tr}}(m)-\beta_p(m)]s(x)}\frac{Z_p(\beta_p(m))}{Z_{p_{\mathrm{tr}}}(\beta_{\mathrm{tr}}(m))}. \tag{14}$$

If s is bounded and nonconstant under the tilted priors, the identity $d\mathbb{E}_{q_p^{\beta}} s/d\beta = \mathrm{Var}_{q_p^{\beta}}(s) > 0$ ensures uniqueness whenever the requested m is attainable. Matching mean activation therefore does not remove dependence on the prior.

Let $D_{\mathrm{tr}} = \mathrm{TV}(q_{\mathrm{tr}}, q)$ and let $\varepsilon_{\mathrm{tr}} = \mathrm{TV}(\widehat{q}_{\mathrm{tr}}, q_{\mathrm{tr}})$ measure error in an implemented conditional sampler. Then

$$\max\{0, D_{\mathrm{tr}} - \varepsilon_{\mathrm{tr}}\} \le \mathrm{TV}(\widehat{q}_{\mathrm{tr}}, q) \le D_{\mathrm{tr}} + \varepsilon_{\mathrm{tr}}. \tag{15}$$

The first term persists for an exact inverse when the two targets differ. The second includes conditional fitting, empirical activation-strength sampling, and calibration error. The Gaussian-mixture toy has full support and examines probability shifts rather than missing support.

C DPS Inversion

Proof of Theorem 2. The Markov property and conditional expectation give

$$V_k(x) = \sum_{x'} P_{\theta,k}(x' \mid x)V_{k-1}(x'). \tag{16}$$

The finite positive terminal weight and finite normalizer ensure that $0 < V_k(x) < \infty$ on reachable states of the discrete chain. Consequently, Eq. (6) is normalized. For any path of positive base probability, its transformed-to-base likelihood ratio telescopes:

$$\prod_{k=1}^{K} \frac{P_k^\star(x_{k-1} \mid x_k)}{P_{\theta,k}(x_{k-1} \mid x_k)} = \prod_{k=1}^{K} \frac{V_{k-1}(x_{k-1})}{V_k(x_k)} = \frac{e^{\beta s(x_0)}}{V_K(x_K)}. \tag{17}$$

Since the initial state is fixed, $V_K(x_K) = Z_{p_\theta}(\beta)$. Summing over paths that decode to the same text t gives terminal law $p_\theta(t)e^{\beta s(t)}/Z_{p_\theta}(\beta) = q^\beta_{p_\theta}(t)$. □

A random initial state. The transformed process must also account for a random base initial law μ_K. Exact path reweighting uses $\mu_K^\star(dx) = \mu_K(dx)V_K(x)/Z$. When V_K is constant, the initial law is unchanged. The fixed all-mask state and fully independent terminal noise have this property. Finite Gaussian noising generally retains some dependence on X_0, making an unmodified standard-Gaussian initialization an additional approximation. The continuous toy specifies its finite schedule and evaluates the plug-in DPS update numerically.

Matched-mean errors at finite resolution. In the toy experiment, let $\widetilde{p}$ be either p or a training prior $p_{\mathrm{tr},\delta}$. For each prior, choose its coefficient $\beta_{\widetilde{p}}(m)$ so that the corresponding target has mean activation m. Let C map inputs to the finite evaluation bins and write

$$D_C(\widetilde{p}) = \mathrm{TV}(C_\# q_{\widetilde{p}}^{\beta_{\widetilde{p}}(m)}, C_\# q), \quad \epsilon_C(\widetilde{p}) = \mathrm{TV}(C_\# \widehat{q}, C_\# q_{\widetilde{p}}^{\beta_{\widetilde{p}}(m)}), \tag{18}$$

where $q = q_p^{\beta_p(m)}$ and $\widehat{q}$ is the implemented sampler's law. The triangle and reverse triangle inequalities give

$$\max\{0, D_C(\widetilde{p}) - \epsilon_C(\widetilde{p})\} \le \mathrm{TV}(C_\# \widehat{q}, C_\# q) \le D_C(\widetilde{p}) + \epsilon_C(\widetilde{p}). \tag{19}$$

The prior-dependent target gap D_C is zero when $\widetilde{p} = p$. The own-prior error ϵ_C includes fitting or sampling approximation and imperfect mean calibration. These distances are not additive. Reported empirical bin distances also contain sampling error and should not be interpreted as continuous TV.

Binary event and filtering. For a fixed prior p and binary activation strength $s = \mathbf{1}_A$, let $p_A = p(A)$. Then $Z_p(\beta) = 1 - p_A + e^\beta p_A$, and

$$q_p^\beta(A) = \frac{e^\beta p_A}{1 - p_A + e^\beta p_A} > p_A \quad (\beta > 0,\ 0 < p_A < 1). \tag{20}$$

Filtering exact proposals to the active set cancels the constant e^β there and yields $p(\cdot \mid A)$. For a general graded score it instead yields a tilt within that set. Hard verification certifies the activation of an individual admitted sample, not its typicality, semantic interpretation, or distributional representativeness.

Error accounting for the implemented generator. Let $\widehat{q}$ denote the input distribution produced by the implemented sampler. Define $D_\theta = \mathrm{TV}(q^\beta_{p_\theta}, q^\beta_p)$ and $\varepsilon_{\mathrm{alg}} = \mathrm{TV}(\widehat{q}, q^\beta_{p_\theta})$, with both priors conditional on the fixed prompt c as in Section 2.4. The triangle and reverse triangle inequalities give

$$\max\{0, D_\theta - \varepsilon_{\mathrm{alg}}\} \le \mathrm{TV}(\widehat{q}, q_p^\beta) \le D_\theta + \varepsilon_{\mathrm{alg}}. \tag{21}$$

Under an exact natural prior, $D_\theta = 0$ and the remaining discrepancy is sampling approximation. The toy uses an analytic natural prior and separately calibrates the coefficient under each prior to mean activation m. The relation separates the two error sources but supplies no estimate of $\varepsilon_{\mathrm{alg}}$. Candidate truncation, surrogate rewards, relaxed embeddings, and greedy selection can alter the generated law. Post-TopK verification checks activation on that law's samples. The sampling errors remain after this check.

D Generation Algorithm and Implementation Settings

One guided generation. Algorithm 1 uses one coordinate update at a time. During unmasking, the DLM predicts all masked tokens by argmax and the most confident masked coordinate is selected.

Algorithm 2 Detailed implementation of Algorithm 1

Require: Frozen DLM p_θ, feature scorer s, prompt c, length L, updates B
Require: Candidate counts $K_{\mathrm{D}}, K_{\mathrm{H}}$, group limit G, threshold δ_{lik}
Require: EntRGi steps M, learning rate η, temperatures $\tau, \tau_{\mathrm{pos}}$, weight λ

$x \leftarrow [\mathrm{MASK}]^L$
$r \leftarrow 0$ ▷ Refinement counter
for $b = 1, \ldots, B$ **do**
  if x contains masks **then**
    Evaluate DLM logits $\ell_j(v)$ at every position j, conditioned on (x, c)
    $i \leftarrow \arg\max_{j: x_j = \mathrm{MASK}} \max_v \mathrm{softmax}(\ell_j)_v$
    $\widehat{x}_j \leftarrow \arg\max_v \ell_j(v)$ if $x_j = \mathrm{MASK}$, otherwise x_j, for all j
  else
    $i \leftarrow 1 + (r \bmod L)$
    $r \leftarrow r + 1$
    $\widehat{x} \leftarrow x$
    Evaluate DLM logits $\ell_j(v)$ conditioned on $(x^{i \leftarrow \mathrm{MASK}}, c)$
  end if
  $\pi \leftarrow \mathrm{softmax}(\ell_i)$
  $v_0 \leftarrow \widehat{x}_i$
  $\mathcal{C} \leftarrow \mathrm{TopK}(\pi, K_{\mathrm{D}}) \cup \{x_i : x_i \neq \mathrm{MASK}\}$
  Remove mask and special tokens from $\mathcal{C}$
  $\mathcal{H} \leftarrow \{x_i\}$ if x_i is valid and unmasked, otherwise $\varnothing$
  $t_v \leftarrow d(\widehat{x}^{i \leftarrow v})$ for $v \in \mathcal{C}$
  Tokenize each t_v with the target tokenizer
  Discard invalid sequences
  Group candidates by target-token length
  Retain at most G groups by maximum DLM logit, then size
  Deduplicate target sequences within each group, retaining the highest-logit candidate
  for each retained group $\mathcal{G}$ with at least two distinct target sequences **do**
    $\phi^0 \leftarrow (\ell_i(v))_{v \in \mathcal{G}}$
    $\phi \leftarrow \phi^0$ and initialize Adam state
    for $m = 1, \ldots, M$ **do**
      Form E_{ST} using Eq. (22) with candidate embeddings $\{E_v\}_{v \in \mathcal{G}}$
      Take an Adam step on $-\lambda R(E_{\mathrm{ST}})$ with learning rate η ▷ Eq. (23)
      $\phi \leftarrow \phi^0 + \mathrm{clip}(\phi - \phi^0, -30, 30)$
    end for
    $\mathcal{J} \leftarrow \mathrm{TopK}(\mathrm{softmax}(\phi/\tau), K_{\mathrm{H}}) \cup \{\arg\max_{v \in \mathcal{G}} \ell_i(v)\}$
    $\mathcal{H} \leftarrow \mathcal{H} \cup \{v \in \mathcal{J} : \ell_i(v) \geq \max_{u \in \mathcal{G}} \ell_i(u) - \delta_{\mathrm{lik}}\}$
  end for
  $\widehat{v} \leftarrow \arg\max^{\mathrm{lex}}_{v \in \mathcal{H}} R_{\mathrm{hard}}(d(\widehat{x}^{i \leftarrow v}))$ if $\mathcal{H} \neq \varnothing$, otherwise v_0
  $x_i \leftarrow \widehat{v}$
end for
$t \leftarrow d(x)$
return $(t, s(t))$

After unmasking, refinement visits coordinates in order. The selected coordinate is masked only in the DLM input, preserving its current token as a hard-score baseline. Visible tokens are retained in the provisional completion. Candidate continuations are decoded with the DLM tokenizer and re-encoded with the target tokenizer. Empty or invalid target sequences are removed. Groups are ranked before deduplication. Among identical target sequences, the candidate with the highest DLM logit is retained. In Algorithm 2, $\ell_j(v)$ denotes the DLM logit for token v at position j. The operator TopK returns at most the requested number of candidates. R_{hard} and its lexicographic ordering are defined in Eq. (24). Hard-score ties preserve the first evaluated candidate, with the current token evaluated first when available.

EntRGi search. For a group of $m \geq 2$ candidates with equal target-token length, let E_v be the frozen target embeddings of candidate continuation t_v. Initialize temporary logits $\phi_v^0 = \ell_i(v)$ and

define

$$\omega_v = \mathrm{softmax}(\phi/\tau)_v, \qquad E_{\mathrm{soft}} = \sum_v \omega_v E_v,$$

$$h(\phi) = \frac{H(\mathrm{softmax}(\phi))}{\log m}, \qquad E_{\mathrm{ST}} = E_{\mathrm{soft}} + \mathrm{sg}(h(\phi)(E_J - E_{\mathrm{soft}}))\,. \tag{22}$$

Here $J \sim \mathrm{Categorical}((\omega_v)_{v\in\mathcal{G}})$, and sg denotes stop-gradient. The frozen target model receives its BOS embedding followed by E_{ST}. For the single-feature setting, let $u_\ell(E)$ denote the selected feature's pre-TopK response at continuation position ℓ. The default search reward is

$$R(E) = \tau_{\mathrm{pos}} \log\left[\frac{1}{n}\sum_{\ell=1}^{n} \exp\left(\frac{u_\ell(E)}{\tau_{\mathrm{pos}}}\right)\right]. \tag{23}$$

The inner loop of Algorithm 2 performs M Adam updates with learning rate η on the loss $-\lambda R(E_{\mathrm{ST}})$, resampling J at each update. After each update, each coordinate of $\phi - \phi^0$ is clipped to $[-30, 30]$. The weight λ is an optimization hyperparameter, distinct from the inverse temperature β in the ideal target distribution. Only ϕ is optimized. All model parameters and candidate embeddings remain fixed. The DLM candidate count K_{D} is distinct from the transcoder sparsity k_{TC}.

Hard selection and verification. Within each group, retain the K_{H} candidates with largest optimized weights and the group's highest-DLM-logit candidate. Apply the logit-drop threshold relative to that group's maximum DLM logit. The current unmasked token is scored separately and remains eligible without this filter. Every retained continuation is evaluated as a hard target-token sequence, prefixed by BOS. In the uncapped single-feature setting, write $A(t)$ for its maximum post-TopK activation and $U(t) = s(t)$ for its maximum pre-TopK response over the continuation positions. At scored position ℓ, let $\kappa_\ell(t)$ be the TopK selection cutoff among the pre-TopK feature responses. Define the maximum TopK margin as $m(t) = \max_\ell\{u_{j,\ell}(t) - \kappa_\ell(t)\}$. The hard score is

$$a_{\mathrm{hard}}(t) = \begin{cases} A(t), & A(t) > 0 \text{ and } m(t) \geq 0, \\ m(t), & \text{otherwise}, \end{cases} \qquad R_{\mathrm{hard}}(t) = (a_{\mathrm{hard}}(t), U(t)). \tag{24}$$

Candidates are compared lexicographically. This ordering implements the single-feature hard selection rule. It is distinct from optimizing $s(t)$ alone. If no candidate is eligible, retain the provisional token. Final scoring excludes the generation prompt and uses BOS plus the continuation. Activating evidence requires positive maximum post-TopK activation.

Table 4: Generation and workflow settings for the original 40-feature Llama-3.2-1B experiment.

Setting	Value
Continuation length (DLM tokens)	32
Generation updates per run	48
DLM candidate count K_{D}	256
Adam updates per candidate group M	12
Adam learning rate	1.0
Candidate temperature τ	0.7
Position temperature τ_{pos}	0.25
Maximum base-logit drop δ_{lik}	20.0
Maximum target-length groups	2
Configured hard-candidate count	8
Neutral-prompt initial runs	3
Corpus-conditioned rescue runs (at most one batch)	3
Desired valid initial generations	2
Initial hypotheses / maximum test rounds	4 / 4
Final stress-test pairs	5

Gradient convention. The entire correction in Eq. (22) is detached, so gradients pass only through E_{soft}. This is a surrogate gradient for finite candidate search, not an unbiased estimator of the exact value function gradient or a guarantee of exact posterior sampling.

E CONTINUOUS TOY EXPERIMENT: CONSTRUCTION AND DERIVATIONS

Setup and endpoint comparison. Following the four-mode Gaussian-mixture construction of Zhou et al. (2026), the natural prior p has means $(\pm 2, \pm 2)$, covariance $0.8^2 I$, and equal weights. The shifted prior p_{tr} assigns weights $(0.05, 0.85, 0.05, 0.05)$ to $(-2,-2), (-2,2), (2,-2), (2,2)$, respectively. We interpolate the dataset prior as $p_{\mathrm{tr},\delta} = (1-\delta)p + \delta p_{\mathrm{tr}}$ for $\delta \in \{0, 0.25, 0.5, 0.75, 1\}$, with $p_{\mathrm{tr},0} = p$ as the no-shift control. All domains retain full support on $\mathbb{R}^2$. Define

$$s(x) = 4\left[1 - \cos\Big(\mathrm{atan2}(x_2, x_1) - \frac{\pi}{4}\Big)\right] \in [0, 8]. \tag{25}$$

The normalized weight $e^{s(x)}\mathbf{1}[x \in \Omega]$ on $\Omega = [-7, 7]^2$ visualizes activation preference in input space. It is not the natural prior or the law of the scalar activation strength.

Matched-activation protocol. For each $m \in \{5, 6, 7\}$, we request the same *mean activation* from every sampler. This criterion controls the achieved activation level while allowing each prior and algorithm its own guidance coefficient. Quadrature constructs the natural target $q = q_p^{\beta_p(m)}$ and the training-prior target $q_{\mathrm{tr},\delta} = q_{p_{\mathrm{tr},\delta}}^{\beta_{\mathrm{tr},\delta}(m)}$, each with expected activation m. They coincide at $\delta = 0$. This is a matched-mean comparison rather than a common-β comparison.

The training-free sampler combines the analytic diffusion score of p with 400 approximate DPS steps. In each dataset domain, we train stochastic MLPs $f_\phi(a, z)$ with $z \sim \mathcal{N}(0, I_d)$ for $d \in \{1, 2\}$. Each learns a conditional inverse from 20,000 observations using a conditional energy score. We train for up to 20,000 updates and select among four checkpoints by independent conditional validation loss. Generation draws conditioning scores from the exponentially reweighted empirical training marginal. DPS guidance strength and the MLP score-reweighting coefficient are independently calibrated to m, each with 14 trials of 12,000 samples. We then measure performance on 6,000 fresh samples per method and seed, retaining the realized means.

Controls and scope. Figure 1 compares q, two-dimensional MLPs trained under p_{tr} and p, and DPS using analytic p. Direct samples from $q_{\mathrm{tr},\delta}$ in the tables expose prior mismatch before neural fitting. Figure 3 additionally runs DPS under the same $p_{\mathrm{tr},\delta}$ used to train the MLP. DPS receives an analytic density, while the MLP receives finite data. These comparisons examine prior choice and numerical approximation under different information conditions. They do not establish a general advantage of training-free inversion. Further controls, calibration records, and fixed-score diagnostics appear below.

Results. At the endpoint shift, the dataset oracle has binned TV 0.531, 0.390, 0.222 from the natural optimum at $m = 5, 6, 7$. The one- and two-dimensional-noise MLP distances are 0.550, 0.489, 0.446 and 0.544, 0.421, 0.283, respectively. Natural-prior DPS has distance 0.105, 0.103, 0.089. The no-shift MLP distances, 0.292, 0.307, 0.333 and 0.185, 0.190, 0.210, show that fitting and numerical effects contribute even when the domains agree. DPS under the shifted prior also departs from the natural target, with distances 0.532, 0.385, 0.253. Together, the controls identify reference-prior choice as relevant to both inversion procedures. The lower row of Figure 3 measures discrepancy from each method's corresponding-prior target. These distances answer a separate question and do not form additive TV components. The largest held-out mean deviation is 0.080, so activation matching is approximate. The reported results concern finite-step algorithms and coarsened distribution metrics.

Table 5: Endpoint comparison at $p_{\mathrm{tr},1} = p_{\mathrm{tr}}$. All distances are measured against q. Values are means and sample standard deviations across three seeds.

m	Method	Realized $\mathbb{E}[s]$	Binned TV	SW_1
5	Direct q samples	4.974 ± 0.031	0.107 ± 0.003	0.045 ± 0.005
5	Direct $q_{\mathrm{tr},1}$ samples	4.999 ± 0.027	0.531 ± 0.006	1.415 ± 0.048
5	DPS (p)	4.988 ± 0.047	0.105 ± 0.005	0.048 ± 0.019
5	MLP, 1D noise (p_{tr})	5.014 ± 0.030	0.550 ± 0.006	1.418 ± 0.032
5	MLP, 2D noise (p_{tr})	5.009 ± 0.040	0.544 ± 0.001	1.418 ± 0.010
6	Direct q samples	5.995 ± 0.027	0.103 ± 0.002	0.038 ± 0.014
6	Direct $q_{\mathrm{tr},1}$ samples	6.004 ± 0.020	0.390 ± 0.007	1.015 ± 0.032
6	DPS (p)	5.990 ± 0.033	0.103 ± 0.004	0.038 ± 0.013
6	MLP, 1D noise (p_{tr})	6.011 ± 0.026	0.489 ± 0.016	1.004 ± 0.018
6	MLP, 2D noise (p_{tr})	6.013 ± 0.026	0.421 ± 0.014	1.018 ± 0.037
7	Direct q samples	7.008 ± 0.022	0.086 ± 0.003	0.029 ± 0.002
7	Direct $q_{\mathrm{tr},1}$ samples	7.008 ± 0.011	0.222 ± 0.007	0.522 ± 0.020
7	DPS (p)	7.014 ± 0.022	0.089 ± 0.002	0.034 ± 0.008
7	MLP, 1D noise (p_{tr})	7.022 ± 0.018	0.446 ± 0.033	0.521 ± 0.023
7	MLP, 2D noise (p_{tr})	7.024 ± 0.021	0.283 ± 0.015	0.512 ± 0.029

Domain-shift sweep. Figure 3 supplements the endpoint comparison in Section 2.4 with no-shift and shared-prior controls. Distance to the natural target measures agreement with the intended prior. Distance to the corresponding-prior target measures numerical fidelity under the chosen prior. The two references separate diagnostic questions, with no additive relation between the distances.

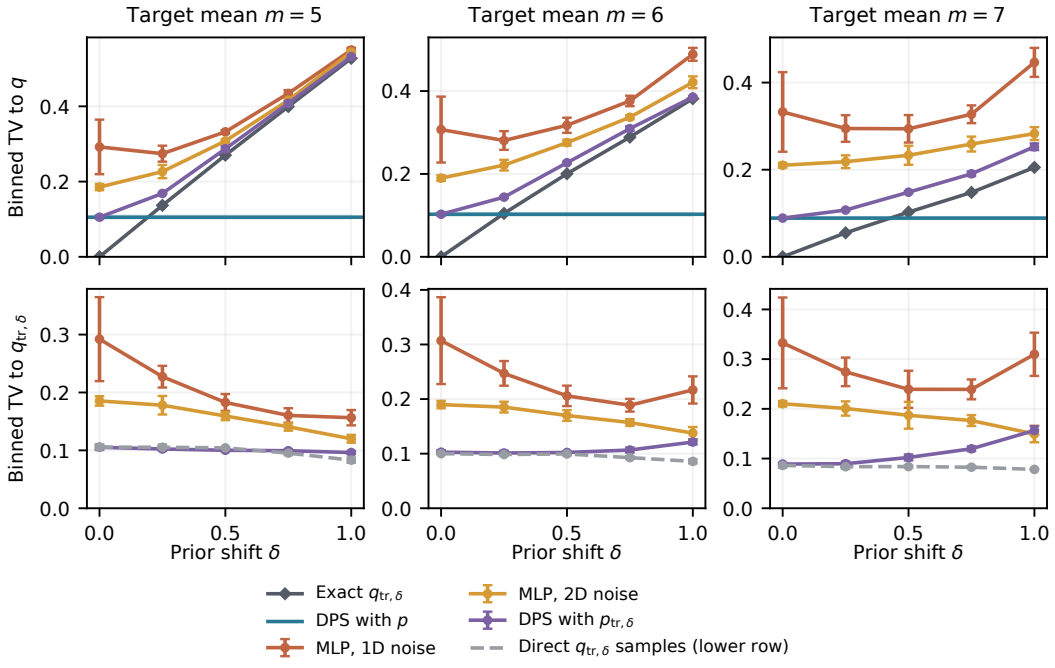

Figure 3: Separating prior shift from algorithm error while matching mean activation. Top: discrepancy from the natural optimum as $p_{\mathrm{tr},\delta}$ moves from p to p_{tr}. The exact oracle curve compares quadrature bin probabilities. The other curves include sampling error. Bottom: discrepancy from the corresponding training-prior target, with the dataset-oracle sampling floor shown dashed. Nonzero errors from both MLPs at $\delta = 0$ cannot be explained by domain shift. The natural-prior DPS band reuses the same runs across δ. It does not represent independently retrained or recalibrated samplers. Error bars and shading are sample standard deviations across seeds.

Prior and activation strength. The setup above specifies the component ordering, means, covariance, and weights. Every mixture is defined on all of $\mathbb{R}^2$. Ω limits only the display. The normalized display weight proportional to $e^{s(x)}\mathbf{1}_\Omega(x)$ expresses activation preference relative to two-dimensional Lebesgue measure. For an input prior $\widetilde{p}$, the law of activation strengths is instead $s_{\#}\widetilde{p}$. These objects have distinct roles. The value of Eq. (25) at the origin is arbitrary because this point has zero probability under every model considered here.

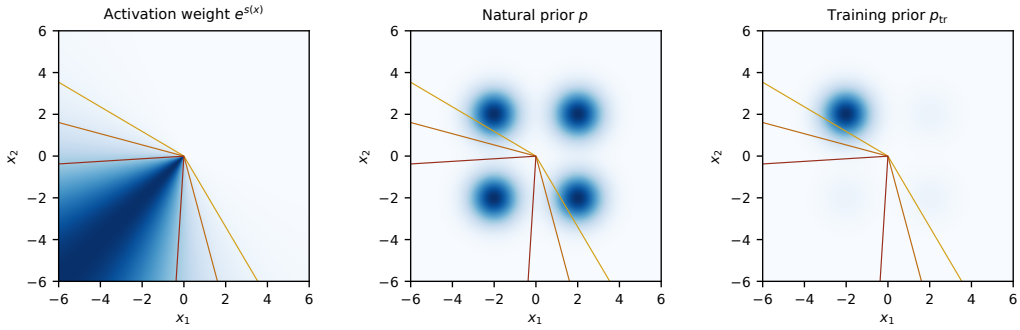

Figure 4: Activation preference, the natural prior, and the training prior. Curves mark $s(x) = 5, 6, 7$. The preference view is normalized on the display square. Both Gaussian-mixture priors are defined on all of $\mathbb{R}^2$.

Gibbs reference and soft constraints. For either prior $\widetilde{p}$, let

$$q_{\widetilde{p}}^{\beta}(x) = \frac{\widetilde{p}(x)e^{\beta s(x)}}{Z_{\widetilde{p}}(\beta)}, \qquad Z_{\widetilde{p}}(\beta) = \int_{\mathbb{R}^2} \widetilde{p}(x)e^{\beta s(x)}\,dx. \tag{26}$$

Because $0 \le s \le 8$, the normalizer is finite at every finite β. For an attainable m above the prior mean, the KL-minimal law satisfying $\mathbb{E}_{q'}s \ge m$ has the form in Eq. (26), with $\beta \ge 0$ chosen so that $\mathbb{E}_q s = m$. Indeed, for any feasible q,

$$D_{\mathrm{KL}}(q'\|\widetilde{p}) = D_{\mathrm{KL}}(q'\|q_{\widetilde{p}}^{\beta}) + \beta\mathbb{E}_{q'}s - \log Z_{\widetilde{p}}(\beta) \ge \beta m - \log Z_{\widetilde{p}}(\beta), \tag{27}$$

with equality at $q' = q_{\widetilde{p}}^{\beta}$ if its mean is m. Moreover, $\frac{d}{d\beta}\mathbb{E}_{q_{\widetilde{p}}^{\beta}}s = \mathrm{Var}_{q_{\widetilde{p}}^{\beta}}(s) > 0$, which permits scalar root finding. The targets q and $q_{\mathrm{tr},\delta}$ both have mean activation m. The matched-mean ratio in Eq. (14) applies with $p_{\mathrm{tr},\delta}$ as the training prior. Theorem 2 gives the corresponding ideal target when the exact guided process uses the same prior. All toy priors have full support, so oracle curves examine probability reweighting. Equation (19) relates the natural-target and own-domain diagnostics at the evaluation resolution.

We use midpoint quadrature on a 560×560 grid over $[-7, 7]^2$. The numerical reference density is normalized within this integration square. An extra evaluation bin records points outside it. Exact reference draws use proposals $X \sim \widetilde{p}$ and accept with probability $e^{\beta(s(X)-8)}$. This rejection sampler operates on the full Gaussian-mixture support, independent of the quadrature box.

Analytic noising process. Let b_t be the DDPM variance schedule, $\alpha_t = 1 - b_t$, and $\bar{\alpha}_t = \prod_{i \le t} \alpha_i$. The forward process is

$$X_t = \sqrt{\bar{\alpha}_t}\,X_0 + \sqrt{1 - \bar{\alpha}_t}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I_2). \tag{28}$$

For $X_0 \sim p$, its exact marginal is

$$p_t(x) = \sum_{k=1}^{4} w_k \mathcal{N}(x \mid \sqrt{\bar{\alpha}_t}\mu_k, v_t I_2), \qquad v_t = \bar{\alpha}_t\sigma^2 + 1 - \bar{\alpha}_t. \tag{29}$$

Writing $r_{t,k}(x)$ for the normalized component responsibilities and $\bar{\mu}_t(x) = \sum_k r_{t,k}(x)\mu_k$, the exact marginal score and clean posterior mean are

$$\nabla_x \log p_t(x) = -\frac{x - \sqrt{\bar{\alpha}_t}\bar{\mu}_t(x)}{v_t}, \tag{30}$$

$$M_t(x) := \mathbb{E}[X_0 \mid X_t = x] = c_t x + d_t \bar{\mu}_t(x), \quad c_t = \frac{\sqrt{\bar{\alpha}_t}\sigma^2}{v_t}, \quad d_t = \frac{1 - \bar{\alpha}_t}{v_t}. \tag{31}$$

No neural denoiser is fitted. The clean-mean Jacobian is also analytic:

$$\nabla_x M_t(x) = c_t I_2 + d_t \frac{\sqrt{\bar{\alpha}_t}}{v_t} \operatorname{Cov}_{r_t(x)}(\mu_k). \tag{32}$$

An explicit Gaussian-to-prior reverse process. The forward Markov transition is $X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{b_t}\epsilon_t$. Let $m_{t-1,k} = \sqrt{\bar{\alpha}_{t-1}}\mu_k$, with $\bar{\alpha}_0 = 1$ and $v_0 = \sigma^2$. Conditioning a joint Gaussian within each mixture component gives the *exact* reverse kernel

$$K_t(x' \mid x) = \sum_{k=1}^{4} r_{t,k}(x)\, \mathcal{N}\!\left(x' \mid A_t x + \frac{b_t}{v_t} m_{t-1,k}, C_t I_2\right), \tag{33}$$

$$A_t = \frac{\sqrt{\alpha_t} v_{t-1}}{v_t}, \qquad C_t = \frac{b_t v_{t-1}}{v_t}, \qquad v_t = \alpha_t v_{t-1} + b_t. \tag{34}$$

It can be sampled by first drawing a component with probabilities $r_{t,k}(x)$, then drawing from its conditional Gaussian. Bayes' rule gives $p_t(x)K_t(x' \mid x) = p_{t-1}(x')F_t(x \mid x')$, where F_t is the forward transition. Reversing these kernels from p_T recovers p at any finite discretization.

To make the noise endpoint *exactly* standard Gaussian, one can set the final forward variance to $b_T = 1$, so $\bar{\alpha}_T = 0$ and $X_T \perp X_{T-1}$. Equation (33) remains well-defined: $r_{T,k} = w_k$, $A_T = 0$, and $K_T(x' \mid x) = p_{T-1}(x')$. The first reverse draw is consequently independent of the numerical value of the starting Gaussian sample. Subsequent draws progressively recover the mixture. This complete-noising endpoint is used only in the separate exact-prior construction. The approximate DPS update uses $b_t < 1$ throughout.

The exact activation-tilted reverse law. For $h_0(x) = e^{\beta s(x)}$, define $h_t(x) = \mathbb{E}[e^{\beta s(X_0)} \mid X_t = x]$ under the prior process. The continuous-state counterpart of Theorem 2 is

$$K_t^{\beta}(x' \mid x) = K_t(x' \mid x)\frac{h_{t-1}(x')}{h_t(x)}. \tag{35}$$

The tower property normalizes each kernel. With the independent Gaussian endpoint, $h_T(x) = Z_p(\beta)$, and the pathwise density ratio telescopes:

$$\prod_{t=1}^{T} \frac{K_t^{\beta}(x_{t-1} \mid x_t)}{K_t(x_{t-1} \mid x_t)} = \frac{e^{\beta s(x_0)}}{Z_p(\beta)}. \tag{36}$$

Thus exact sampling of these transformed kernels from Gaussian noise has terminal law q_p^{β}. For this toy, the required conditional expectation is explicitly specified by

$$h_t(x) = \sum_k r_{t,k}(x) \int e^{\beta s(u)} \mathcal{N}(u \mid c_t x + d_t \mu_k, \tau_t^2 I_2)\, du, \tag{37}$$

$$\tau_t^2 = \frac{\sigma^2(1-\bar{\alpha}_t)}{v_t}, \qquad t > 0, \tag{38}$$

with c_t, d_t from Eq. (31). This integral is generally not equal to $\exp\{\beta s(M_t(x))\}$. The result establishes the ideal target. The implemented plug-in sampler below approximates its guidance.

Approximate DPS update. Starting from $X_T \sim \mathcal{N}(0, I_2)$, the implementation uses

$$X_{t-1} = \frac{X_t + b_t \nabla \log p_t(X_t)}{\sqrt{1-b_t}} + \gamma b_t \nabla_{X_t} s(M_t(X_t)) + \sqrt{\widetilde{b}_t}\, \xi_t, \tag{39}$$

$$\widetilde{b}_t = b_t \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}, \qquad \xi_t \sim \mathcal{N}(0, I_2). \tag{40}$$

There are 400 steps with b_t linearly spaced from 10^{-4} to 0.06. We differentiate through Eq. (31). The score-gradient denominator clips $\|M_t\|$ below 0.02 to control the singular angular derivative near the origin. Target densities and evaluation scores use the unclipped score. For the domain-control runs, the same formulas use mixture weights $w_{\delta,k} = (1-\delta)w_k + \delta w_{\mathrm{tr},k}$ in the component responsibilities. In particular, the noised density and score are exact for the chosen prior. The guided reverse update remains approximate.

Matched-mean calibration and independent evaluation. Both numerical methods use the same protocol: evaluate the endpoints 0 and 4, then take 12 bisection steps, with 12,000 samples at each of the 14 trials. The free scalar is γ for DPS and the empirical score-marginal tilt λ for the MLP. The selected trial minimizes the absolute calibration mean error. No evaluation sample participates. Common random numbers are reused within each calibration search. Every prior/target DPS configuration has one fixed calibration, independent of all evaluation seeds. Each separately trained MLP is calibrated independently with the same per-model budget. The exact reference multipliers come from quadrature and are not fitted numerical samplers. All tested configurations bracketed the requested mean. The endpoint weights $(0.05, 0.85, 0.05, 0.05)$ emphasize concentration in the second quadrant while retaining full support. This is an exploratory stress test rather than a prospectively selected benchmark. Thirty MLPs were trained, with 15 for each latent dimension. All numerical samplers were calibrated and evaluated using the budgets above. Calibration and evaluation remain independent within each run. The saved outputs include calibration trials and held-out means. Across numerical-method runs, the largest realized evaluation mean error is 0.080. Finite-sample calibration does not impose an exact population moment. Table 8 reports every prior/target configuration.

An exact continuous-time reverse diffusion would require the time-dependent correction $\nabla_x \log h_t(x)$, where

$$h_t(x) = \mathbb{E}_p[e^{\beta s(X_0)} \mid X_t = x]. \tag{41}$$

The plug-in gradient in Eq. (40) is not this quantity. Finite reverse steps, the approximate Gaussian reverse kernel, gradient clipping, and the nonzero terminal $\bar{\alpha}_T$ introduce additional error. In particular, the analytic forward marginal is only approximately standard Gaussian at the chosen terminal time. Convexity of KL gives the conservative bound

$$D_{\mathrm{KL}}(p_T \| \mathcal{N}(0, I_2)) \le v_T - 1 - \log v_T + 4\bar{\alpha}_T, \tag{42}$$

and Pinsker's inequality gives the square root of half this bound for TV. The exact continuous construction in Eq. (35) consequently does not certify this numerical update as an exact posterior sampler. Numerically, $\bar{\alpha}_T = 4.699 \times 10^{-6}$ and this conservative terminal TV upper bound is 0.00307.

Conditional stochastic inverse. For each $\delta \in \{0, 0.25, 0.5, 0.75, 1\}$ and each of three seeds, we sample 20,000 training inputs from $p_{\mathrm{tr},\delta}$, together with 4,096 independent validation inputs. Each fully connected MLP has three width-128 hidden layers with SiLU activations. The one-dimensional model input is $((a-4)/4, z)$ with $z \sim \mathcal{N}(0,1)$. The two-dimensional model input is $((a-4)/4, z_1, z_2)$ with $(z_1, z_2) \sim \mathcal{N}(0, I_2)$. Both models output samples in $\mathbb{R}^2$. We minimize the proper conditional energy score

$$\mathcal{L}(\phi) = \mathbb{E}_{X \sim p_{\mathrm{tr},\delta}, Z, Z'} \left[\|f_\phi(s(X), Z) - X\|_2 - \frac{1}{2} \|f_\phi(s(X), Z) - f_\phi(s(X), Z')\|_2 \right]. \tag{43}$$

Here $f_\phi(a, z)$ includes the normalization of its first input to $(a-4)/4$ described above. We average both independent data-fit terms to reduce estimator variance. Under the usual moment assumptions, the proper energy score identifies the full conditional law at its population optimum (Gneiting & Raftery, 2007). Ordinary squared-error regression with independent noise does not have this property. Attainment of the optimum by the finite MLP remains an empirical fitting question.

Every model receives 20,000 Adam updates with batches of 256. The learning rate decreases by a cosine schedule from 10^{-3} to 10^{-4}. Checkpoints at updates 2,400, 6,000, 12,000, and 20,000 are compared using only the conditional energy score on the independent validation set, averaged over two fixed noise-pair draws per input and model. The checkpoint with lowest validation loss supplies all subsequent calibration and evaluation draws. We retain loss curves, validation values, the selected step, and model weights for all 30 models. Neither a Gibbs-distance metric nor the evaluation set is used for checkpoint selection. Seeds are shared across δ to pair initializations and mixture draws. The three seeds themselves are independent replicates.

Let $\rho_{\mathrm{tr},\delta} = s_{\#} p_{\mathrm{tr},\delta}$ be the activation-strength law under the training prior. An ideal conditional inverse has law $p_{\mathrm{tr},\delta}(dx \mid s = a)$. We implement the KL-optimal score-sampling protocol of Proposition 1, whose score-space and input-space KL objectives agree by Eq. (13). Drawing its conditioning input from

$$\rho_{\delta,\lambda}(da) = \frac{e^{\lambda a} \rho_{\mathrm{tr},\delta}(da)}{Z_{p_{\mathrm{tr},\delta}}(\lambda)} \tag{44}$$

then gives

$$\int p_{\mathrm{tr},\delta}(dx \mid s = a)\rho_{\delta,\lambda}(da) = \frac{e^{\lambda s(x)} p_{\mathrm{tr},\delta}(dx)}{Z_{p_{\mathrm{tr},\delta}}(\lambda)}. \tag{45}$$

We approximate this score marginal using the sorted training scores with exponential weights, draw fresh Gaussian noise of the model's latent dimension per generated input, and calibrate λ to the requested mean. The learned model's λ need not equal the oracle $\beta_{\mathrm{tr},\delta}(m)$, because calibration can compensate for fitting and finite-dataset error. We also save diagnostics at the oracle multiplier to expose this compensation separately. Distances to the own-domain optimum in Figure 3 include conditional fitting, empirical score marginal, and numerical calibration error. They are not an isolated estimate of any one term.

The no-shift and shared-prior controls. With $\delta = 0$, both procedures use the natural domain. At $m = 5, 6, 7$, the one- and two-dimensional-noise MLPs have binned TV 0.292, 0.307, 0.333 and 0.185, 0.190, 0.210, respectively, and DPS has 0.105, 0.103, 0.089. Table 6 also gives the sampling floor, realized means, and SW_1. Since no domain shift is present, the residual MLP discrepancy reflects approximation effects. Selecting the model by validation loss leaves these effects measurable.

Table 6: No-shift control: all methods use the domain p. The MLPs receive finite training data. DPS receives the analytic prior score. All distances use the common natural/domain optimum.

m	Method	Realized $\mathbb{E}[s]$	Binned TV	SW_1
5	Reference floor	4.974 ± 0.031	0.107 ± 0.003	0.045 ± 0.005
5	DPS	4.988 ± 0.047	0.105 ± 0.005	0.048 ± 0.019
5	MLP, 1D noise	5.015 ± 0.049	0.292 ± 0.073	0.099 ± 0.042
5	MLP, 2D noise	5.016 ± 0.056	0.185 ± 0.008	0.056 ± 0.006
6	Reference floor	5.995 ± 0.027	0.103 ± 0.002	0.038 ± 0.014
6	DPS	5.990 ± 0.033	0.103 ± 0.004	0.038 ± 0.013
6	MLP, 1D noise	6.014 ± 0.044	0.307 ± 0.080	0.091 ± 0.017
6	MLP, 2D noise	6.016 ± 0.046	0.190 ± 0.007	0.063 ± 0.010
7	Reference floor	7.008 ± 0.022	0.086 ± 0.003	0.029 ± 0.002
7	DPS	7.014 ± 0.022	0.089 ± 0.002	0.034 ± 0.008
7	MLP, 1D noise	7.026 ± 0.023	0.333 ± 0.091	0.093 ± 0.042
7	MLP, 2D noise	7.025 ± 0.026	0.210 ± 0.005	0.062 ± 0.007

At $\delta = 1$, supplying p_{tr} to DPS removes its access to the correct natural prior. Table 7 compares both numerical methods with the natural optimum and with their own-domain optimum. For DPS the latter binned distances are 0.096, 0.121, 0.156, and for the two-dimensional MLP they are 0.120, 0.138, 0.149. These comparisons separate the questions of prior choice and numerical fidelity. The distances need not add, and errors can partially cancel.

Table 7: Shared shifted prior at $\delta = 1$. The dataset oracle, DPS with analytic p_{tr}, and MLP trained on samples from p_{tr} all target the same requested mean. The last two columns use different reference optima and must not be interpreted as an additive decomposition.

m	Method	Realized $\mathbb{E}[s]$	TV to q	TV to $q_{\mathrm{tr},1}$
5	Direct $q_{\mathrm{tr},1}$ samples	4.999 ± 0.027	0.531 ± 0.006	0.083 ± 0.005
5	DPS (p_{tr})	5.027 ± 0.018	0.532 ± 0.005	0.096 ± 0.003
5	MLP, 1D noise (p_{tr})	5.014 ± 0.030	0.550 ± 0.006	0.156 ± 0.013
5	MLP, 2D noise (p_{tr})	5.009 ± 0.040	0.544 ± 0.001	0.120 ± 0.007
6	Direct $q_{\mathrm{tr},1}$ samples	6.004 ± 0.020	0.390 ± 0.007	0.086 ± 0.004
6	DPS (p_{tr})	6.000 ± 0.012	0.385 ± 0.002	0.121 ± 0.004
6	MLP, 1D noise (p_{tr})	6.011 ± 0.026	0.489 ± 0.016	0.217 ± 0.025
6	MLP, 2D noise (p_{tr})	6.013 ± 0.026	0.421 ± 0.014	0.138 ± 0.011
7	Direct $q_{\mathrm{tr},1}$ samples	7.008 ± 0.011	0.222 ± 0.007	0.078 ± 0.001
7	DPS (p_{tr})	7.021 ± 0.012	0.253 ± 0.008	0.156 ± 0.009
7	MLP, 1D noise (p_{tr})	7.022 ± 0.018	0.446 ± 0.033	0.310 ± 0.043
7	MLP, 2D noise (p_{tr})	7.024 ± 0.021	0.283 ± 0.015	0.149 ± 0.017

Table 8: Calibration and independent evaluation at every domain shift. $\beta_{\mathrm{tr},\delta}$ is the quadrature oracle multiplier. γ is DPS strength and $\lambda_{1D}, \lambda_{2D}$ are the MLP score-marginal tilts. Values after $\pm$ summarize three seeds. No reported evaluation mean is used to choose a model or a calibration parameter.

δ	m	$\beta_{\mathrm{tr},\delta}$	γ	λ_{1D}	λ_{2D}	DPS mean	MLP_{1D} mean	MLP_{2D} mean
0	5	0.128	0.063	0.132 ± 0.006	0.130 ± 0.004	4.988 ± 0.047	5.015 ± 0.049	5.016 ± 0.056
0	6	0.281	0.142	0.284 ± 0.003	0.283 ± 0.005	5.990 ± 0.033	6.014 ± 0.044	6.016 ± 0.046
0	7	0.526	0.271	0.529 ± 0.006	0.526 ± 0.003	7.014 ± 0.022	7.026 ± 0.023	7.025 ± 0.026
0.25	5	0.154	0.085	0.156 ± 0.006	0.153 ± 0.009	4.999 ± 0.033	5.016 ± 0.044	5.017 ± 0.046
0.25	6	0.335	0.187	0.336 ± 0.007	0.332 ± 0.013	5.989 ± 0.024	6.016 ± 0.037	6.015 ± 0.035
0.25	7	0.614	0.344	0.612 ± 0.005	0.605 ± 0.016	7.002 ± 0.021	7.022 ± 0.022	7.024 ± 0.024
0.5	5	0.192	0.118	0.193 ± 0.007	0.194 ± 0.004	5.013 ± 0.022	5.019 ± 0.051	5.021 ± 0.047
0.5	6	0.409	0.256	0.407 ± 0.007	0.408 ± 0.005	6.006 ± 0.019	6.014 ± 0.029	6.009 ± 0.033
0.5	7	0.723	0.470	0.715 ± 0.001	0.718 ± 0.004	7.024 ± 0.024	7.015 ± 0.023	7.020 ± 0.016
0.75	5	0.252	0.173	0.252 ± 0.006	0.252 ± 0.006	5.019 ± 0.018	5.016 ± 0.048	5.015 ± 0.043
0.75	6	0.517	0.371	0.512 ± 0.006	0.513 ± 0.010	5.999 ± 0.014	6.013 ± 0.033	6.012 ± 0.033
0.75	7	0.869	0.691	0.857 ± 0.003	0.859 ± 0.012	7.024 ± 0.018	7.020 ± 0.021	7.019 ± 0.020
1	5	0.366	0.292	0.362 ± 0.006	0.363 ± 0.010	5.027 ± 0.018	5.014 ± 0.030	5.009 ± 0.040
1	6	0.701	0.616	0.691 ± 0.008	0.696 ± 0.011	6.000 ± 0.012	6.011 ± 0.026	6.013 ± 0.026
1	7	1.107	1.163	1.088 ± 0.004	1.103 ± 0.005	7.021 ± 0.012	7.022 ± 0.018	7.024 ± 0.021

Fixed-score inversion is a different target. We also examine conditioning on one fixed score. For each $a \in \{5, 6, 7\}$, each selected $\delta = 1$ MLP receives that first input and 6,000 fresh Gaussian draws of its latent dimension. Table 9 and Figure 5 report conditioning accuracy. The ideal law is supported on $s^{-1}(a)$, which has zero two-dimensional Lebesgue measure. Even a perfect conditional inverse therefore has continuous TV one from a finite-β Gibbs density. Score means, dispersion, and absolute conditioning error assess this fixed-score task without conflating it with matched-mean Gibbs inversion.

Table 9: Fixed-score sampling under the selected $\delta = 1$ models. Each model generates 6,000 samples. The activation SD is computed within a run. $\pm$ describes variation of each statistic across the three trained models.

a	Noise	Realized $\mathbb{E}[s(f_\phi(a, Z))]$	Activation SD	$\mathbb{E}\lvert s(f_\phi(a, Z)) - a\rvert$
5	1D noise	5.034 ± 0.009	0.288 ± 0.010	0.062 ± 0.010
5	2D noise	5.061 ± 0.001	0.353 ± 0.021	0.102 ± 0.009
6	1D noise	6.067 ± 0.026	0.343 ± 0.036	0.111 ± 0.020
6	2D noise	6.045 ± 0.018	0.348 ± 0.018	0.137 ± 0.006
7	1D noise	7.131 ± 0.031	0.350 ± 0.044	0.235 ± 0.033
7	2D noise	7.172 ± 0.022	0.432 ± 0.023	0.326 ± 0.013

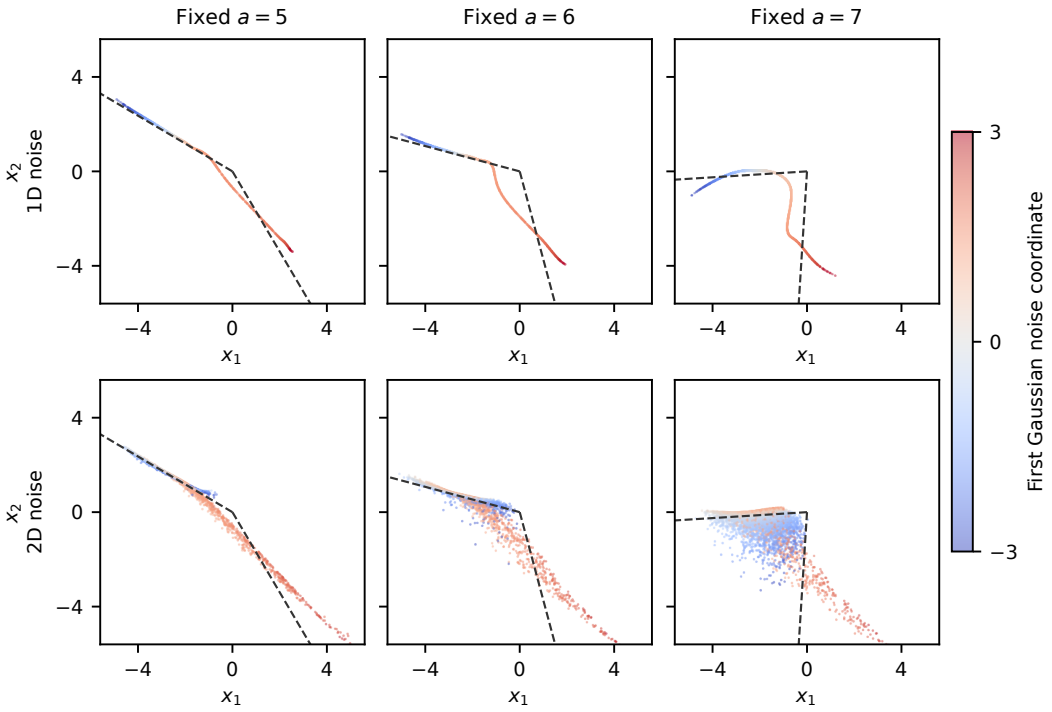

Figure 5: Fixed-score inversion with the updated MLPs. Dashed curves mark $s(x) = a$. Colors identify the first input-noise coordinate. The top and bottom rows use one- and two-dimensional Gaussian noise. This is separate from the matched-mean Gibbs comparison.

Evaluation and reproducibility. For each target and each seed, every method supplies 6,000 independent evaluation samples. The reported binned TV uses 40 bins per coordinate over $[-7, 7]^2$ and an extra bin for all points outside the box. Its reference bin masses come from numerical quadrature. It is a coarsened distance estimate, not the continuous-distribution TV. In particular, scalar noise and a finite empirical set of conditioning scores make the smooth MLP's generated law a finite mixture of curves. Its continuous TV from the full-dimensional Gibbs density is one. The reported binned TV and Wasserstein metrics assess agreement at finite spatial resolution and under weak distances. They do not estimate the continuous-TV approximation error in Eq. (15). Empirical sliced Wasserstein-1 averages sorted one-dimensional sample distances over 64 fixed random unit directions against a separate Gibbs reference sample of equal size. An independent second Gibbs sample yields the sampling-error floor. We also save masses in the four nearest-component Voronoi cells (the coordinate quadrants), their TV from the target, and the average radius, so that activation alone cannot conceal geometric distortion or mode imbalance. Reported uncertainty is sample standard deviation across the three seeds, not a confidence interval. The figures and tables summarize the saved per-seed metrics, calibration records, and training records used for the reported experimental values.

F Agent Protocol and Evidence Bookkeeping

The controller exposes a different evidence view at each LLM call. This separation prevents the stress-test designer from copying discovery examples and keeps measured feature responses distinct from agent predictions. Table 10 summarizes the call contracts.

Controller flow and evidence ledger. The controller first combines retrieved and generated samples to form hypotheses. For each active hypothesis, it then requests a test and a predicted activation label, measures the feature on the proposed text, and uses the result to revise the hypothesis. After testing, it drafts an explanation, checks it on fresh contrast pairs, and freezes the revised version.

Table 10: Agent-call contracts for one feature. Hard responses are always computed by the frozen target model and transcoder, never by an LLM.

Call	Visible state	Required output
Form hypotheses	Feature identity, corpus examples and activations, accepted DPS texts and selected trajectories, hard scores and peak token windows	Four distinct, single-sentence-testable hypotheses, including an alternative or negative-control direction
Design tests	Active hypothesis statements and cumulative test summaries	One new sentence and a High/Low prediction for every active hypothesis
Update hypotheses	Original tests, any linked DPS rewrites, and their hard responses	CONFIRMED, REJECTED, REFINED, or UNCHANGED status, revised statement when needed, and confidence
Draft explanation	Corpus and accepted DPS evidence, the complete test ledger, and final hypothesis states	Seven-field public explanation record
Design calibration	Public explanation record only	Five fresh High/Low minimal pairs covering the prescribed axes
Final calibration	Public draft and hard-scored calibration errors	Revised seven-field record to freeze

A ledger keeps the measured evidence separate from the agent's predictions. Each entry records the sample's source, discrete text, hard-scored maximum activation, activity indicator, and peak location when active. Test entries additionally record the predicted label, round, associated hypothesis, and whether a rewrite was requested. A DPS rewrite receives its own entry linked to the failed *High* test that prompted it, so both outcomes remain available during revision.

Initialization. The controller verifies the target model, dictionary, layer, hook, and feature identity, and retrieves corpus examples. Missing corpus evidence is a data failure. It runs three neutral generations; if fewer than two pass the gate, one rescue batch uses a corpus-conditioned prompt of the form *Generate a passage similar to "[corpus example]"*. The combined evidence is marked SUFFICIENT, LIMITED, or NONE for at least two, exactly one, or zero valid runs. Interpretation proceeds in all three cases, whereas GPU, database, configuration, or API failures stop the process.

Successful generations' snapshots at steps $1, 8, 16, 24, 32$ and the final step provide search-state information alongside hard scores. Rejected runs remain in logs but are excluded from the hypothesis prompt. If none pass, initialization uses corpus evidence alone.

Testing and termination. Exactly four hypotheses are initialized. During each of at most four rounds, every pending, refined, or unchanged hypothesis receives one new original sentence test. The collection includes High and Low when multiple hypotheses remain active. At least one original test of each label is required before a hypothesis can terminate. A failed original High permits one guided rewrite, and both its original outcome and the rewrite outcome are shown during revision. Terminal hypotheses need not be tested further. Unresolved claims at the round limit remain uncertainty.

Public explanation and calibration. The public schema uses the seven fields defined in Section 2.5: a description, activating pattern, activation envelope, verified exclusions, labels, uncertainties, and confidence. Five contrast pairs probe token boundaries, minimum surrounding text, substitutions, structure, and exclusions. Hard-scored false negatives, false positives, and High peak-localization errors inform the final revision. The resulting explanation is frozen before benchmark stimulus generation. Evidence status does not impose a mechanical cap on confidence: later tests can supply additional evidence. Optional independent-seed DPS validation assesses a frozen explanation without changing it. It is disabled in the original 40-feature Llama-3.2-1B record and in the reported generator comparisons; the expanded main-result extraction does not specify this setting separately for the other targets. The hypothesis-formation user prompt is reproduced in Appendix H.

G BENCHMARK METRICS AND DETAILED RESULTS

Target models and transcoders. The Top-K transcoder configurations are listed below. Each transcoder was trained on 800 million tokens.

Target model	Input dimension	Feature dimension	Expansion	Top-K
Llama-3.2-1B	2,048	16,384	8×	64
Llama-3.2-3B	3,072	24,576	8×	128
Qwen3-1.7B	2,048	16,384	8×	128

Shared evaluation protocol. For each feature, an LLM uses each method's frozen public explanation to write five new *High*/*Low* pairs, giving ten texts per method. These labels describe the intended activation of the generated texts, not their measured activation. The texts from all three methods are anonymized and interleaved into one shared 30-text challenge set. Each frozen explanation is evaluated on the same set by an LLM that predicts *High* or *Low* for every text and a peak span for each predicted-*High* text. Apart from the challenge texts, the LLMs receive only the public explanation, without corpus samples, DPS-generated samples, activation traces, or method identity. The target model and transcoder provide the reference labels: a text is *High* if its maximum scored post-TopK feature activation is positive, and *Low* otherwise. The challenge-generation and prediction prompts are reproduced in Appendix H.

Evaluation metrics. *Activation accuracy* is the fraction of challenge texts with the correct *High* /*Low* prediction. *High localization* is the fraction of predicted-*High* texts whose predicted span overlaps the scorer's reference peak-token span in character space. It measures localization among predicted-*High* texts, whether or not the *High* label is correct. *Joint accuracy*, the primary metric, counts a correct *Low* prediction or a *High* prediction with both a correct activation label and an overlapping peak span, divided by the total number of texts. Counts are pooled across features, using all texts for activation and joint accuracy and only predicted-*High* texts for localization.

For feature j and stimulus i, let y_{ji} be the reference label, equal to one when the maximum scored post-TopK activation is positive and zero otherwise. Let $\widehat{y}_{ji}^{(m)}$ be method m's binary prediction, and let $L_{ji}^{(m)}$ indicate nonempty character overlap between its predicted span and the reference peak-token span recorded by the scorer when $y_{ji} = 1$. Set $L_{ji}^{(m)} = 0$ when $y_{ji} = 0$. This is a binary match, not an overlap fraction. With N total stimuli,

$$\mathrm{Acc}^{(m)} = \frac{1}{N} \sum_{j,i} \mathbf{1}\{\widehat{y}_{ji}^{(m)} = y_{ji}\}, \tag{46}$$

$$\mathrm{Loc}^{(m)} = \frac{\sum_{j,i} \widehat{y}_{ji}^{(m)} L_{ji}^{(m)}}{\sum_{j,i} \widehat{y}_{ji}^{(m)}}, \tag{47}$$

$$\mathrm{Joint}^{(m)} = \frac{1}{N} \sum_{j,i} \left[(1 - \widehat{y}_{ji}^{(m)})(1 - y_{ji}) + \widehat{y}_{ji}^{(m)} y_{ji} L_{ji}^{(m)} \right]. \tag{48}$$

Localization uses the documented predicted-High denominator. Its numerator does not explicitly include y_{ji} because $L_{ji}^{(m)} = 0$ for a true Low. Thus, false-positive High predictions count as localization misses. Joint accuracy also requires $y_{ji} = 1$ for every credited High prediction. If no High is predicted, localization is undefined rather than zero. The scorer has no reference peak span for a true Low. Peak tie-breaking and character-offset handling should accompany the raw prediction release.

Coverage and exact aggregation. Before explanation generation, seed 20260807 selects ten alive features from each chosen layer; this selection remains fixed when a run fails. The completed subset comprises 40 features from each Llama target and 30 from Qwen, with ten fixed features per completed layer and 30 shared prompts per feature. Both Llama models cover layers $0, 5, 10, 15$, giving 1,200 predictions per method per model; Qwen contributes 900. All three methods cover these 110 features. The challenge pool balances intended High/Low labels, not necessarily the measured activation labels. Evaluators receive only frozen public explanation fields: method identity,

corpus examples, generation traces, evidence texts, and acquisition status are hidden. Qwen layer 27 is incomplete and contributes no features or predictions to any reported total; the completed Qwen layers are $0, 10, 20$. Table 11 records pooled integer counts from the result extraction, obtained by summing the 11 completed layers' evaluation summaries. Localization is pooled by its own denominator, rather than by averaging model-specific percentages. The blind challenge-generation and shared-pool prediction user prompts are reproduced in Appendix H.

Table 11: Exact pooled correct/denominator counts on the 110-feature completed subset. Activation and joint accuracy each use 3,300 shared prompts; localization uses each method's predicted-High count.

Method	Activation	High localization	Joint
SAGE	2318/3300	540/1349	1770/3300
InvExplain	2725/3300	1187/1602	2494/3300
Neuronpedia-style	2030/3300	726/1275	1849/3300

Table 12: Layer-level results underlying Table 1. Each setting has ten features and 300 shared prompts per method. Cells report activation / High localization / joint accuracy in percent. Localization uses each method's predicted-High denominator. Bold marks the highest value for each metric within a row. Qwen layer 27 is incomplete and excluded.

Target	Layer	SAGE	InvExplain	Neuronpedia-style
Llama-3.2-1B	0	68.7 / 27.5 / 46.3	**91.0 / 97.3 / 89.7**	70.7 / 89.1 / 70.7
	5	74.0 / 28.2 / 50.3	**79.3 / 72.7 / 73.0**	60.7 / 50.5 / 53.3
	10	62.7 / 17.5 / 42.7	**78.0 / 63.9 / 70.0**	61.0 / 38.1 / 50.3
	15	76.3 / 40.7 / 60.0	**79.3 / 75.2 / 74.7**	60.3 / 57.1 / 58.3
Llama-3.2-3B	0	77.3 / 43.0 / 57.3	**94.3 / 99.4 / 94.3**	56.3 / 79.7 / 56.3
	5	67.3 / 27.9 / 42.3	**84.0 / 79.4 / 76.0**	55.0 / 77.2 / 52.7
	10	66.0 / 48.7 / 53.3	**75.0 / 62.7 / 64.3**	52.3 / 34.6 / 41.7
	15	70.3 / 38.5 / 46.7	**80.0 / 60.2 / 62.7**	64.7 / 53.5 / 51.0
Qwen3-1.7B	0	73.3 / 54.5 / 70.7	**85.7 / 72.5 / 84.7**	70.0 / 51.2 / 67.7
	10	72.3 / **63.6** / 60.3	**81.0** / 61.6 / **64.7**	62.0 / 58.0 / 52.3
	20	64.3 / 46.0 / 60.0	**80.7 / 69.2 / 77.3**	63.7 / 52.0 / 62.0

G.1 Case Study: Recovering the Feature Boundary

Figure 6 illustrates why retrieved top activations can yield an incomplete explanation. For Qwen3-1.7B layer-10 feature 10774, all ten highest-activating corpus examples supplied to the baselines express negative analytic or elementary solvability. Their unsolvability explanations fit this evidence but conflate the topic with its polarity. In controlled pairs, replacing *cannot* with *can*, or *no elementary antiderivative* with *an elementary antiderivative*, preserves strong activation at the solvability expression. InvExplain captures both polarities, whereas the baselines miss the affirmative claims. On the full 30-prompt pool, activation accuracy is 90% for InvExplain and 50% for each baseline. This case illustrates how counterfactual testing can distinguish a feature's response from a correlation in its retrieved examples.

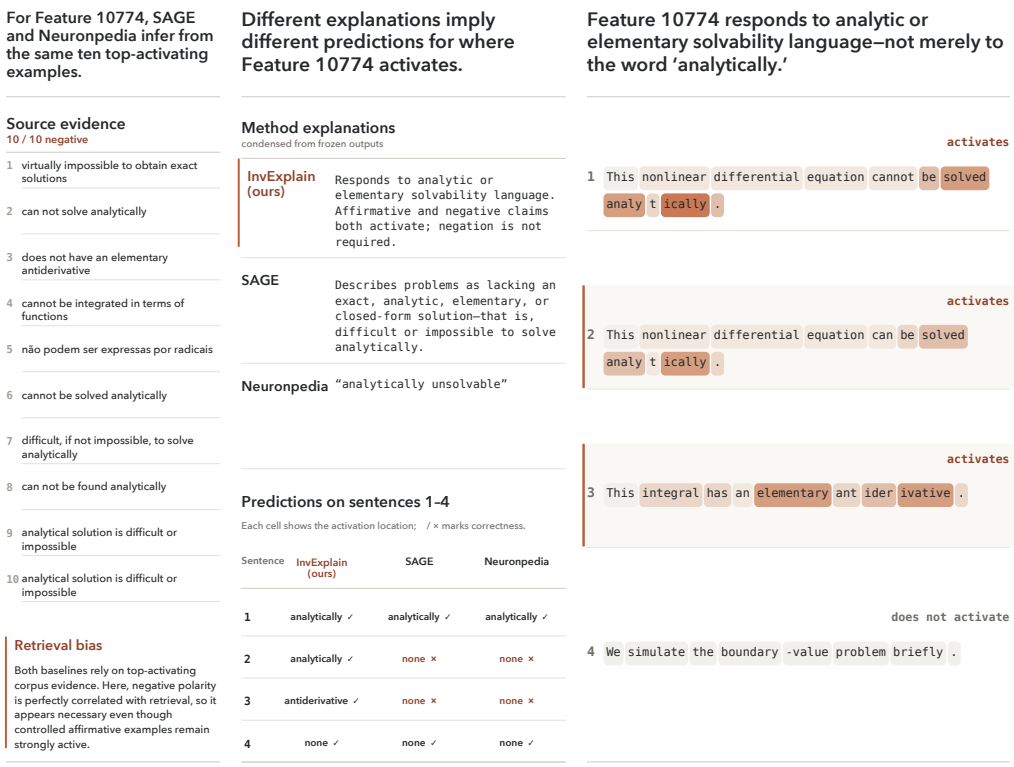

Figure 6: **Case study for Qwen3-1.7B layer-10 feature 10774. Left:** The same ten highest-activating corpus examples used by SAGE and Neuronpedia all express negative solvability, revealing the retrieval-polarity confound behind their narrower explanations. **Middle:** The three frozen explanations and their predicted activation locations on the same sentences. InvExplain captures both polarity and the elementary-antiderivative formulation, whereas SAGE and the Neuronpedia-style baseline miss the two affirmative examples. Checkmarks and crosses indicate whether each explanation-conditioned prediction is correct. **Right:** Token shading shows measured post-TopK activation on four shared held-out sentences: a negative analytic-solvability claim, two affirmative solvability claims, and a topic-matched inactive control.

Inversion-search ablation. Table 3 displays the reference setting and one single-parameter comparison for each of N_{init}, M, and D. The selected configuration is $(N_{\text{init}}, M, D) = (3, 12, 20)$. We compare it with a tighter base-logit constraint ($D = 5$), a smaller initial generation budget ($N_{\text{init}} = 1$), and fewer optimization steps ($M = 8$), changing one parameter at a time. All four configurations use the same ten layer-10 features and 48 diffusion updates per generation. Initial rescue is disabled, so N_{init} counts initial DLM generations, while test-time conditional rewrites remain enabled. M counts EntRGi Adam updates per candidate group, and D is the maximum allowed drop from the highest base DLM logit in that group.

For the shared-pool evaluation, each frozen explanation predicts the same anonymous 80-prompt pool for its feature. The prediction LLM sees no evidence, generation trajectory, method identity, or prompt source. This gives 800 shared predictions per configuration and is separate from the 30-prompt-per-feature main comparison.

Table 13 reports auxiliary results for the same three pairwise comparisons on texts authored separately by each configuration. The selected configuration has the same High active rate as all three alternatives (94.0%), but higher High peak, Joint, and Reduction scores. Its Low inactive rate is higher than those of the $D = 5$ and $N_{\text{init}} = 1$ alternatives (80.0% versus 78.0%), but lower than that of $M = 8$ (84.0%). These measurements characterize each configuration's own texts, whereas Table 3 compares the configurations on a common prompt pool.

Table 13: Auxiliary results on separately generated High/Low texts for the three inversion-search ablations in Table 3. Blocks vary D, N_{init}, and M, respectively; bold indicates the selected configuration, repeated for each comparison. Entries are percentages. Reduction is the activation-weighted decline from High to Low; texts are not paired across configurations.

N_{init}	M	D	High active	Low inactive	High peak	Joint	Reduction
3	12	5	94.0	78.0	80.0	79.0	93.8
3	**12**	**20**	**94.0**	**80.0**	**88.0**	**84.0**	**94.6**
1	12	20	94.0	78.0	84.0	81.0	93.3
3	**12**	**20**	**94.0**	**80.0**	**88.0**	**84.0**	**94.6**
3	8	20	94.0	84.0	82.0	83.0	91.4
3	**12**	**20**	**94.0**	**80.0**	**88.0**	**84.0**	**94.6**

Generator-comparison diagnostics. Table 14 assesses two aspects of each generator's own *High /Low* texts. *High active* and *High peak* measure whether the proposed *High* text activates the feature and whether its peak is correctly identified. *Low inactive* and *Reduction* measure whether the proposed *Low* text suppresses activation and by how much. Because each generator produces different texts, these measurements describe its generated contrasts rather than performance on a shared test set.

Among the base generators in Pair I, iLLaDA-base has lower High active and High peak rates than LLaDA-base, but higher Low inactive and Reduction. In these generated examples, iLLaDA-base more often finds a nonactivating contrast, while LLaDA-base more often finds and localizes activating patterns. Instruction tuning does not produce the same tradeoff in both families. For LLaDA in Pair II, the base model has higher High active and High peak rates, while the instruct model has a slightly higher Low inactive rate and a similar Reduction. For iLLaDA in Pair III, the instruct model improves both the High and boundary measures.

For a designed High/Low pair i with hard peak activations $A_{H,i}, A_{L,i}$, define

$$\text{Reduction}_i = \begin{cases} 100(A_{H,i} - A_{L,i})/A_{H,i}, & A_{H,i} > 0, \\ 0, & A_{H,i} \le 0. \end{cases} \tag{49}$$

For the generated pairs $i \in \mathcal{P}$ represented by one table row, the reported positive-A_H-weighted aggregate uses $A^+_{H,i} = \max\{A_{H,i}, 0\}$ and is

$$\begin{aligned} \text{Reduction}_{\mathcal{P}} &= \frac{\sum_{i\in\mathcal{P}} A^+_{H,i}\,\text{Reduction}_i}{\sum_{i\in\mathcal{P}} A^+_{H,i}} \\ &= 100\frac{\sum_{i\in\mathcal{P}:A_{H,i}>0}(A_{H,i} - A_{L,i})}{\sum_{i\in\mathcal{P}:A_{H,i}>0} A_{H,i}}, \end{aligned} \tag{50}$$

when at least one $A_{H,i} > 0$; pairs with nonpositive A_H have zero weight. Negative reductions are retained. High activation, Low nonactivation, and High peak-correctness rates are measured on each method's own generated texts. These are different inputs, so the rates cannot be treated as paired correctness outcomes. In particular, High peak correctness in this table is not the shared-pool predicted-High localization metric.

Table 14: Auxiliary results on separately generated High/Low texts. Entries are percentages. High peak is correctness on generated High texts; Reduction is the reported activation-weighted decline from High to Low. Pair I compares iLLaDA-base with LLaDA-base, Pair II compares LLaDA-base with LLaDA-instruct, and Pair III compares iLLaDA-base with iLLaDA-instruct. Different rows do not evaluate the same texts.

Pair: generator	High active	Low inactive	High peak	Reduction
I: iLLaDA-base	86.0	84.0	70.0	93.5
I: LLaDA-base	92.0	68.0	86.0	91.3
II: LLaDA-base	94.0	80.0	84.0	92.9
II: LLaDA-instruct	86.0	84.0	76.0	92.6
III: iLLaDA-base	82.0	82.0	62.0	92.4
III: iLLaDA-instruct	86.0	94.0	70.0	98.8

H User Prompt Templates

The following boxes reproduce the English instructions used to form four hypotheses and to conduct the shared blind evaluation. Braced fields are filled at runtime. Descriptive placeholders are named in English for typesetting.

Forming four hypotheses. The agent receives corpus samples and accepted inversion evidence with final hard scores and selected denoising snapshots. If no generation passes the gate, the evidence field states this explicitly.

Four hypotheses: user prompt

```
Analyze this target and form four distinct testable hypotheses.

Target model: {model_name}
Dictionary: {dictionary_name}
Feature: {feature_index}
Layer: {layer}
Hook: {hook_point}
Corpus global maximum: {corpus_global_maximum}

CORPUS TOP ACTIVATIONS
{corpus_top_activations}

ENTRGI GUIDED EVIDENCE
{accepted_evidence_or_no_valid_generation}

{evidence_use_constraints} Every hypothesis must identify a narrow
    token, syntax, discourse-position, or semantic condition and be
    testable with one sentence. Include at least one alternative or
    negative-control hypothesis.

Return exactly:
{
  "observation": "cross-source pattern analysis",
  "hypotheses": [
    {"text": "specific testable claim", "evidence": "why corpus and/or
    trajectory supports it"}
  ]
}
```

Generating blind challenge texts. The evaluator receives one frozen public explanation and writes five High/Low pairs without access to its acquisition evidence.

Challenge generation: user prompt

```
TARGET
Feature: {feature_idx}

FROZEN EXPLANATION (the only semantic evidence you may use)
{public_explanation_json}

Create exactly five minimal HIGH/LOW pairs, for exactly 10 new
    standalone prompts total. Within each pair, change the smallest
    practical cue needed to test the explanation while keeping topic
    and wording similar. Cover distinct aspects or boundary cases of
    the explanation instead of repeating one template. All 10 `text`
    values must be globally unique,  non-empty, and at most 400
    characters. A short symbol or Unicode sequence is a valid
    standalone prompt. Do not discuss this task inside the prompts.

For every HIGH prompt, predict the shortest exact visible substring
    containing the single location where the feature's maximum post-
    TopK activation should occur. `predicted_span` must occur exactly
    once in `text` and be at most 32 characters. It may include a
    little unique context when predicting whitespace or repeated
    punctuation. For every LOW prompt, use null because no location
    should activate.

Return exactly:
{"items": [{"id": "pair1_high", "pair_id": 1, "text": "new prompt", "
    prediction": "HIGH", "predicted_span": "exact unique substring", "
    reason": "short explanation-based prediction"}, {"id": "pair1_low",
     "pair_id": 1, "text": "minimal contrast", "prediction": "LOW", "
    predicted_span": null, "reason": "short explanation-based
    prediction"}]}
```

Predicting the shared challenge set. The same anonymized 30 texts are presented to each frozen explanation.

Shared-pool prediction: user prompt

```
TARGET
Feature: {feature_idx}

FROZEN EXPLANATION
{public_explanation_json}

SHARED HELD-OUT PROMPT POOL
[
  {"stimulus_id": "feature_{feature_idx}_prompt_01", "text": "..."},
  ...
]

Independently predict whether each complete prompt has any positive
    post-TopK activation for this feature. For HIGH, give the shortest
    exact substring containing the predicted maximum activation
    location. The span must occur exactly once in that prompt and be at
     most 32 characters. For LOW, use null. Do not force a balanced
    answer; follow the explanation.

Return exactly:
{"predictions": [{"stimulus_id": "exact id", "prediction": "HIGH|LOW",
    "predicted_span": "exact unique substring or null"}]}
```