UNDERSTANDING CONFABULATION AND RETHINKING RECONSTRUCTION IN ACTIVATION EXPLANATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Natural Language Autoencoders (NLAs) produce unsupervised text explanations of a model's activations: a verbalizer describes an activation and a reconstructor learns to recover it from this text. Under the established point-reconstruction NLA training recipe, explanations become more useful for predicting model behavior while also increasingly introducing unsupported details and exhibiting writing defects. To assess these changes separately, we introduce the first publicly available standardized evaluation framework for unstructured NLA explanations, measuring information recoverable from explanations, contextual support for their claims, and writing quality. To address confabulation and writing defects, we move beyond predicting a single activation: explanations can distinguish distributions of possible activations even when their means and optimal point-reconstruction rewards are identical. We introduce Flow-NLA, which models the distribution of activations compatible with an explanation and trains the verbalizer using a diffusion likelihood bound. Across Qwen, Gemma, and Apertus, this richer signal retains the utility gains of point reconstruction while curbing the growth of confabulation and writing defects, opening up a direction for improving activation-derived training signals to encourage more informative, supported, and readable explanations. The evaluation and training code is available in the anonymous repository.

1 INTRODUCTION

Natural-language activation explanations aim to make model representations accessible beyond observable outputs, as explored by LatentQA and Activation Oracles (Pan et al., 2026; Karvonen et al., 2026a). Natural Language Autoencoders (NLAs) (Fraser-Taliente et al., 2026) learn such explanations through reconstruction: a verbalizer describes an activation in natural language, and a reconstructor predicts the activation from that text. After supervised initialization, reconstruction quality provides reinforcement-learning feedback without requiring labels for each generated explanation.

Because removing true claims hurts reconstruction more than removing false ones, the original NLA study anticipated that training would reduce confabulation, but found that it remained substantial and did not decrease (Fraser-Taliente et al., 2026). Rerunning this recipe on three open models, we find that final explanations are similarly confabulated and that confabulation rises as reconstruction improves. Recent work has further examined initialization sensitivity and claim-level reliability (Zhang & Turner, 2026; Dingeto, 2026). These findings raise a fundamental question: *what properties of an explanation does reconstruction training actually reward?* An explanation may convey useful information while introducing unsupported claims or expressing that information poorly. We therefore separate these properties, track how they evolve during training, and ask whether richer reconstruction feedback can improve explanation quality beyond point reconstruction.

Our contributions are:

1. Evaluating explanation quality. We measure instance-specific predictive information by asking an independent reader model to make predictions from the explanation, comparing matched explanations with mismatched explanations. Separately, we evaluate contextual confabulation, claims unsupported or contradicted by the source context using claim-based evaluation (Min et al., 2023; Song et al., 2024), and identify writing defects in clarity, coherence, and economy (Section 3.1). This

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.

framework separates what information an explanation makes recoverable from whether its claims are context-supported and clearly expressed.

2. Characterizing point-reconstruction training. Prior NLA work primarily trains explanations through point reconstruction. Across the models we study, this training improves predictive utility while also increasing contextual confabulation and writing defects (Section 3.2). This behavior follows naturally from the training signal: reconstruction rewards text for helping recover the target activation, but does not directly require the claims carrying that information to be context-supported or clearly expressed. Thus, useful but unsupported cues can still be rewarded, motivating a closer examination of what information the reconstruction objective itself captures.

3. Beyond point reconstruction with Flow-NLA. We show that point reconstruction has an information blind spot: under NLA's normalized squared-error objective, explanation refinements that change the conditional activation distribution while preserving its mean cannot improve the optimal reconstruction reward. Flow-NLA instead provides explanation-conditioned denoising feedback across noise levels, allowing the training signal to depend on information beyond the conditional mean (Section 4). We relate this objective to the diffusion variational bound (Ho et al., 2020; Kingma et al., 2021) and analyze its sensitivity to richer conditional information. Across three models, Flow-NLA retains substantial utility gains while limiting contextual confabulation growth and producing fewer writing defects than the baseline NLAs (Section 4.3).

2 Related Work

Learning Activation Explanations. LatentQA and Activation Oracles train language models to answer questions about activations using supervised activation–response pairs (Pan et al., 2026; Karvonen et al., 2026a), with Activation Oracles broadening this supervision through classification and context-prediction tasks. More recent approaches use reconstruction to learn open-ended activation descriptions beyond predefined questions. Cycle-Consistent Activation Oracles (Chalnev, 2026) warm-start the verbalizer on LatentQA-style supervision and surrounding-token prediction before optimizing an activation→text→activation cycle with RL, while NLAs (Fraser-Taliente et al., 2026) warm-start both the activation verbalizer and reconstructor using teacher-generated descriptions before jointly optimizing them for activation reconstruction. These approaches make reconstruction a learning signal for open-ended activation explanations, but the feedback remains centered on recovering individual target activations. We instead study whether richer, distributional reconstruction feedback can reward information that point reconstruction misses, using conditional denoising to examine its effects on explanation utility and confabulation.

Confabulation and Explanation Evaluation. An NLA initialization study finds that claim plausibility declines during training for normally initialized NLAs and that different initializations yield similar reconstruction accuracy despite differences in explanation plausibility (Zhang & Turner, 2026). Verbalizations can also reflect the verbalizer's own parametric knowledge (Li et al., 2026). For utility, CHIVE evaluates whether interpretability tools, including NLAs, improve counterfactual behavioral prediction beyond the original prefix (Karvonen et al., 2026b). The NLA study evaluates both information recovery and confabulation (Fraser-Taliente et al., 2026). We extend this evaluation effort with the first publicly available standardized framework for unstructured NLA explanations, enabling comparisons across models and training procedures. Our framework evaluates predictions from explanations with the source withheld and separately checks source support using claim-based evaluation adapted from FActScore and VeriScore (Min et al., 2023; Song et al., 2024).

3 Confabulation or Explanation? Understanding Confabulation in Reconstruction Training

We start by examining how explanations change during training as an NLA learns to reconstruct activations more accurately. To achieve this, we propose an evaluation framework that scores utility, confabulation, and writing quality.

First, we describe the established reconstruction training procedure. Let $h \in \mathbb{R}^d$ denote a residual-stream activation of the target language model at a selected layer and token position, normalized to unit length. The target language model is frozen. We extract activations from fixed residual-stream

locations, with layers and token positions specified for each target model and evaluation task in Appendix B.1. The verbalizer $\pi_\phi(z \mid h)$ generates an explanation z from the activation. Following the released NLA (Fraser-Taliente et al., 2026), a separate reconstructor predicts a unit activation $\widehat{h}_\theta(z)$, discarding activation magnitude. The verbalizer's reward is $r_\theta(h, z) = -\left\|h - \widehat{h}_\theta(z)\right\|_2^2$. Training an NLA consists of two phases. After supervised initialization on activation–summary pairs, online training repeatedly generates groups of explanations, scores them with the current reconstructor, and updates the reconstructor and the verbalizer on the same batch (Fraser-Taliente et al., 2026). The reconstructor is updated directly on the generated activation–explanation pairs. The verbalizer uses a group-relative policy update with a KL penalty toward its supervised reference (Shao et al., 2024). Thus, an explanation is favored when the reconstructor can use it to reduce the reconstruction MSE.

3.1 An Evaluation Framework for Explanation Quality

We evaluate explanations along three complementary dimensions (Figure 1): **utility**, measuring what task-relevant information a blind judge can recover from the explanation; **source support**, measuring whether its claims are supported by the source; and **writing quality**, measuring how clearly that information is expressed. These dimensions are scored by a separate judge LLM, whose context is restricted to what each dimension requires. The utility judge answers a task query using only the explanation, the source-support judge observes both the source and explanation, and the writing judge observes only the explanation. Appendix B provides further information.

Utility. Useful explanations let readers answer diverse questions about the activation. Therefore, we evaluate utility over a large range of tasks (Appendix Table 4). First, we identify properties of the source text through 12 explanation classification tasks spanning semantic (topic, sentiment, subjectivity, entailment), lexical (entity and exact-name presence), and grammatical (tense, number, language, pronoun form) categories (accuracy). Second, we extract information on the model's continuation by classifying the true 32-token continuation among ten candidates (accuracy). Last, we predict the model's own behavior by predicting whether it answers, refuses, or asks for clarification from the explanation (macro-F1). Explanations that perform well across these tasks carry identifiable information from the activation. The utility judge receives only the explanation and a question with candidate answers (the task query), and must choose an answer. These are judged with DeepSeek V4.1 Flash (DeepSeek-AI, 2026) as the judge model (Appendix B).

To additionally test whether utility comes from instance-specific information rather than coarse cues such as the topic, we compare the matched explanation against a hard negative: the explanation of a closely related source that differs in the relevant answer. For example, "Alice lent Bob a book" and "Bob lent Alice a book" share nearly all surface cues but imply opposite answers to "Who lent the book?" We define the hard-negative separation as the log-probability ratio $\log\left[p_{\mathrm{J}}(y \mid z_{\mathrm{matched}})/p_{\mathrm{J}}(y \mid z_{\mathrm{hard}})\right]$ that the utility judge assigns to the correct answer y given

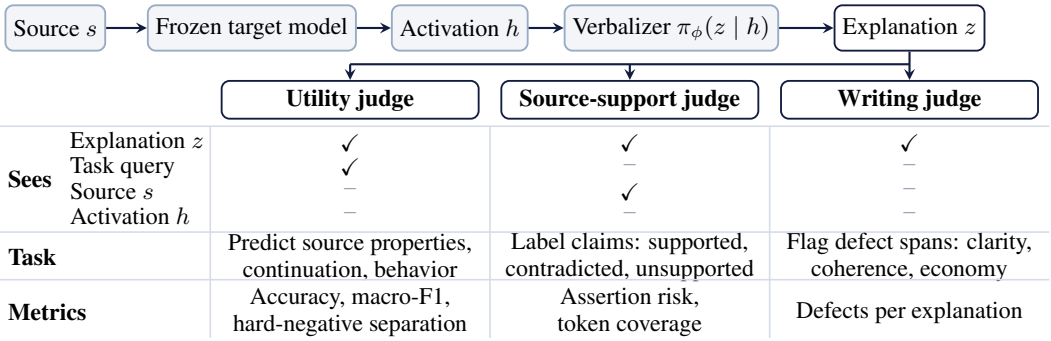


Figure 1: **Three judges of explanation quality.** Each judge sees only what it requires: The utility judge answers the task query from the explanation; the source-support judge checks the explanation's claims with the source; and the writing judge assesses only the explanation.

the matched versus the hard-negative explanation, with answer probabilities read from the same DeepSeek judge; it is the primary utility measure presented here. Higher is better: in the example, an explanation that only says a book was lent scores zero, while one that says who lent it to whom scores positive; positive values thus indicate instance-specific information beyond the topic and surface cues shared with the hard negative. Definitions and aggregation details are given in Appendix B, and Appendix D decomposes the separation into matched predictive gain and mismatch penalty.

Confabulation. Using a source-support judge that receives the source text before the activation extraction and the explanation, explanations are split into distinct atomic claims (Min et al., 2023; Song et al., 2024). The claims are labeled as supported (stated in or entailed by the source), contradicted (incompatible with the source) or unsupported. Supported and contradicted verdicts are required to quote the exact source evidence. This yields two metrics: Assertion Risk, which is the proportion of assertions that are unsupported or contradicted, and Confabulated Token Coverage, which is the proportion of explanation tokens occupied by these claims. Both metrics assess the contextual source support rather than the utility of describing the model's internal computation.

Writing quality. The writing quality judge sees only the explanation, without the source text. Here, the judge flags distinct defects in clarity (unclear wording, malformed or unfinished sentences), coherence (statements that conflict within the explanation), and economy (avoidable repetition or filler), quoting the exact explanation span for each. We count distinct defects and report changes from supervised initialization. Writing quality is evaluated separately from source support because an explanation may be unsupported yet well written, or supported yet poorly expressed.

For example, the Qwen NLA explanation below fails the source-support and writing quality judges for different reasons: the source-support judge flags invented names as contradicted by the source, and the writing judge flags incompatible accounts of the person's name as a coherence defect.

Source
Jonn Hart is a stage name his real name is Di'Jonn Grizzell

Explanation excerpt
The final clause "his actual full name is Kristopher Ellis... his real last name is Scott Gellell; he's actual full name is Tracy Ellzell" requires a period or continuation

3.2 The Confabulation Cost of Better Reconstruction

We retrain NLAs using the established setup from Fraser-Taliente et al. (2026) for Qwen2.5-7B-Instruct (Yang et al., 2024), Gemma-3-12B (Gemma Team, 2025) and Apertus-v1.5-8B (Hernández-

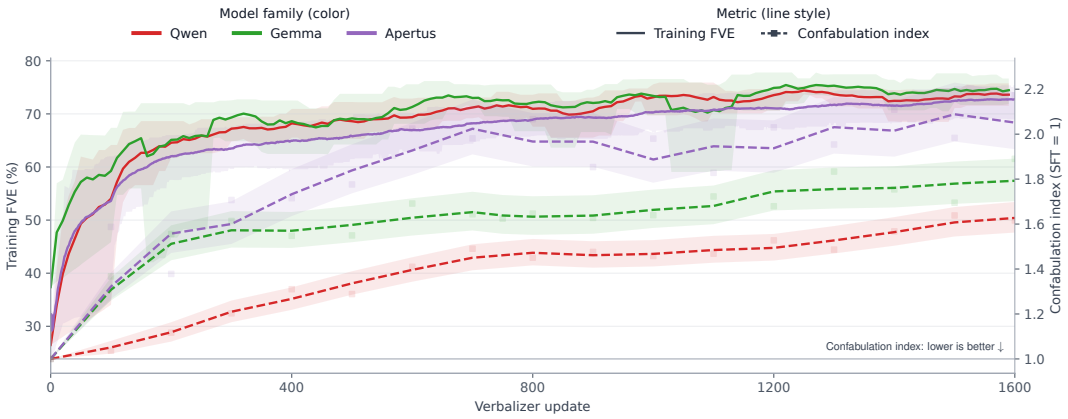


Figure 2: **Better reconstruction accompanies more confabulation and writing defects.** As the FVE reconstruction rises (solid line), so does the confabulation index (dashed line) across models. The index $I(t) = \frac{1}{2}[R(t)/R(0) + D(t)/D(0)]$ averages assertion risk R and writing defects per explanation D relative to supervised initialization ($I = 1$). Shading shows 95% bootstrap intervals.

Table 1: **Utility and confabulation in released NLAs and our reruns.** All scores are percentages except writing defects, which are counts per explanation.

		Utility ↑			Confabulation ↓		Writing ↓
	Target model	Class.	Suffix	Behavior	Risk	Coverage	Defects
Released Anthropic NLAs	Qwen2.5-7B	79.4	98.0	82.7	92.6	74.8	2.45
	Gemma-3-12B	84.7	96.6	83.4	83.1	67.5	1.78
	Gemma-3-27B	85.7	98.2	84.1	76.3	55.4	1.92
	Llama-3.3-70B	82.7	99.0	65.1	76.4	56.9	1.78
Point-reconstruction reruns	Qwen2.5-7B	79.7	97.6	74.9	83.7	68.3	1.63
	Gemma-3-12B	84.2	96.2	82.4	77.8	44.2	1.38
	Apertus-v1.5-8B	77.5	96.0	73.6	81.3	41.5	1.50

Cano et al., 2026). Reconstruction is measured during training using the Fraction of Variance Explained, $\mathrm{FVE} = 1 - \mathbb{E}[\|h - \widehat{h}_\theta(z)\|_2^2]/V_h$, where V_h is the variance baseline for unit activations; utility, confabulation, and writing quality are evaluated on held-out examples.

As reconstruction improves during training, assertion risk and writing defects increase (Figure 2; separate panels in Appendix C): the correlation between FVE and the confabulation index is considerable, with 0.78 for Qwen, 0.96 for Gemma, and 0.90 for Apertus.

At the same time, hard-negative separation rises from roughly 0.6–1.1 nats at supervised initialization to 1.2–2.4 nats within the first few hundred updates (Figure 3), so explanations increasingly carry instance-specific information rather than only coarse cues (Section 3.1). Better reconstruction thus carries both greater recoverable information and a deterioration in how reliably and clearly it is expressed, although whether this coupling is strictly necessary remains open.

Released NLA baselines. The same coexistence of useful information and unsupported content is visible in the four released Anthropic NLAs (Fraser-Taliente et al., 2026) (Table 1). Classification accuracy ranges from 79.4% to 85.7% and suffix-selection accuracy from 96.6% to 99.0%, yet assertion risk exceeds 75% across all four models, and confabulated content accounts for more than half of explanation tokens in the released NLAs.

Classification and suffix-selection accuracies approximately match those reported by Fraser-Taliente et al. (2026) for their Claude NLAs (about 78–86% and 93–98%), although their evaluation code is unreleased. Our model reruns reach a training FVE of about 0.75 (Figure 2), close to the 0.75–0.80 reported for the released checkpoints. Our point-reconstruction reruns on Qwen, Gemma, and Apertus show the same pattern: 77.5% to 84.2% classification and 96.0% to 97.6% suffix-selection accuracy, alongside 77.8% to 83.7% assertion risk and 41.5% to 68.3% confabulated token coverage. Thus, both released checkpoints and our reruns produce useful but highly confabulated explanations. We next ask whether changing the activation-derived reward can preserve useful learning while weakening the coupling between confabulation and reconstruction error.

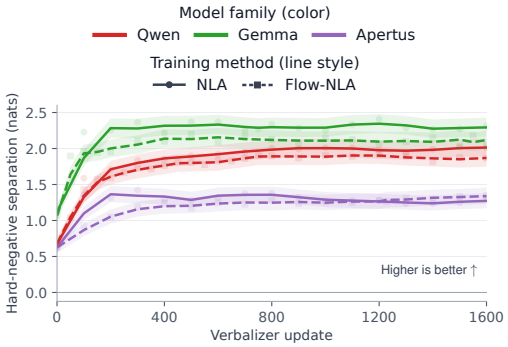


Figure 3: **Better reconstruction accompanies more relevant information.** Hard-negative separation across the 12 classification datasets for point-reconstruction NLA (solid) and Flow-NLA (dashed); higher is better.

4 Beyond Point Reconstruction with Flow-NLA

What information should reconstruction feedback reward? We first show that informative distinctions among activations can leave optimal point-reconstruction loss unchanged. Next, we develop

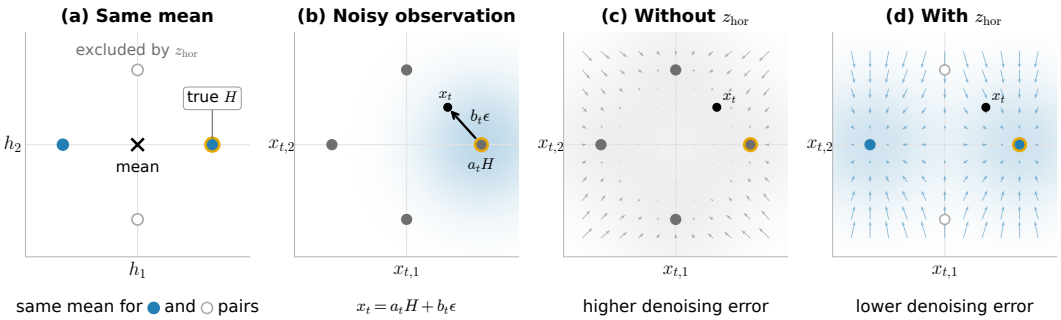


Figure 4: **Denoising can reward information beyond the conditional mean.** H is uniform over four unit directions; we follow the realized activation $H = (1, 0)$ (gold outline). (a) The horizontal explanation z_{hor} keeps the horizontal pair (blue) and excludes the vertical pair (open circles). Both pairs have mean zero ($\times$), so the optimal expected point loss is the same with or without z_{hor}. (b) Denoising also observes the noisy activation $x_t = a_t H + b_t \epsilon$ (black arrow: noise $b_t\epsilon$), with candidates scaled by a_t. (c) Without the explanation, x_t is about equally close to the true and upward candidates, so the optimal estimate hedges between them. (d) With z_{hor}, the upward candidate is excluded and the estimate moves to the true direction, lowering the denoising error. In (c)–(d), arrows show the displacement $a_t\mathbb{E}_\phi[H \mid x_t, z] - x_t$ from each noisy observation to its optimal signal estimate (without z in (c)); shading shows the density of x_t.

Flow-NLA to reward such distinctions through conditional denoising and evaluate how this richer feedback changes explanation utility, confabulation, and writing quality.

4.1 Beyond Point Reconstruction

Point reconstruction can fail to reward informative distinctions even with an optimal reconstructor. For a fixed verbalizer, the joint distribution $p(h)\pi_\phi(z \mid h)$ induces $p_\phi(h \mid z)$. Following NLA (Fraser-Taliente et al., 2026), let both the target H and prediction u have unit norm. Writing $\mu_\phi(z) = \mathbb{E}_\phi[H \mid Z = z]$, we have

$$\min_{\|u\|_2=1} \mathbb{E}_\phi[\|H - u\|_2^2 \mid Z = z] = 2 - 2\|\mu_\phi(z)\|_2. \tag{1}$$

The optimum is attained by $u = \mu_\phi(z)/\|\mu_\phi(z)\|_2$ when $\mu_\phi(z) \neq 0$, and by any unit-norm prediction otherwise. The optimal expected point loss depends on $p_\phi(h \mid z)$ only through its conditional mean. Because NLA rewards the verbalizer based on reconstruction quality, more informative explanations that leave the conditional mean unchanged do not necessarily receive higher reward.

Figure 4 illustrates this limitation with four equally likely unit directions. The horizontal explanation z_{hor} rules out the vertical directions but leaves two opposing candidates. Their mean is zero, so this partial information leaves the optimal expected point loss at 2, as without the explanation.

Denoising additionally observes $x_t = a_t H + b_t \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is independent of (H, Z) and $a_t^2 + b_t^2 = 1$. At each noise level, the optimal unconstrained squared-error prediction is $\mathbb{E}_\phi[H \mid x_t, z]$. The explanation can therefore exclude candidates that remain plausible from x_t alone. For the illustrated observation, denoising assigns lower error with the correct horizontal explanation than without it (Figure 4c–d). Information that leaves the conditional mean unchanged can thus still be rewarded by denoising.

4.2 Flow-NLA

Flow-NLA builds on this argument by replacing point reconstruction with conditional denoising. The reconstructor learns to denoise activations using the text explanation, and the verbalizer is rewarded for explanations that help it capture the conditional distribution of activations they describe. This results in an unsupervised training signal sensitive to more than the conditional mean.

Rewarding input reconstruction likelihood under a co-trained decoder is an established way to encourage informative outputs (Barber & Agakov, 2003; Miao & Blunsom, 2016; Eysenbach et al.,

2019; Rita et al., 2022), typically using categorical or Gaussian models. Flow-NLA instead rewards input likelihood under a co-trained diffusion model conditioned on the verbalizer's output. This conditional distribution shifts as the verbalizer learns, complicating reward stabilization. This unsupervised reward favors activation prediction over source-context support, so it may still reward unsupported wording. Source-text supervision could address this, but would bias explanations away from information extractable from activations alone. Whether this objective changes the balance between utility and contextual confabulation is an empirical question (Section 4.3).

Conditional reconstruction. Figure 5 shows the Flow-NLA training procedure. Let x be the activation after a fixed coordinate standardization, and let X denote the corresponding random variable. We form a noisy activation $x_t = a_t x + b_t \epsilon$ and velocity target $v_t = a_t \epsilon - b_t x$, where $\epsilon \sim \mathcal{N}(0, I_d)$ and $a_t^2 + b_t^2 = 1$ define a variance-preserving path, and d is the activation dimension. The reconstructor $f_\theta(x_t, t, z)$ combines a text reader with a conditional velocity field and is trained with the velocity-prediction objective (Salimans & Ho, 2022):

$$\mathcal{L}_{\text{rec}}(\theta; \phi) = \mathbb{E}_{h,z,t,\epsilon}\left[d^{-1} \| f_\theta(x_t, t, z) - v_t \|_2^2\right], \qquad z \sim \pi_\phi(\cdot \mid h). \tag{2}$$

Across noise levels, the model learns how the explanation helps distinguish activation patterns compatible with a partially observed activation. These denoising predictions capture conditional distributional structure (Saharia et al., 2023), since their population optimum at an interior noise level identifies the full explanation-conditioned distribution for a fixed verbalizer (Appendix H). During training, this distribution, $p_\phi(h \mid z) \propto p(h)\, \pi_\phi(z \mid h)$, changes whenever the verbalizer learns, so the reconstructor tracks a moving target, much like a distributional critic (Bellemare et al., 2017), except that here the modeled distribution itself, not its mean, defines the reward (Appendix F).

Rewarding distributional information. We seek to reward explanations under which the observed activation has high conditional likelihood. The point reward already does this, but with a restrictive model: for unit vectors, $r_\theta(h, z) = 2\, \widehat{h}_\theta(z)^\top h - 2$ is, up to scale and a constant, the log-likelihood of a distribution on the unit sphere determined entirely by its mean direction $\widehat{h}_\theta(z)$. Such a log-likelihood lower-bounds the information that explanations carry about activations (Barber & Agakov, 2003). The bound is loose whenever the model cannot represent the true conditional distribution, as with the two opposing candidates in Figure 4. Flow-NLA keeps this objective but replaces this model with the flow reconstructor, which can represent several modes. The diffusion likelihood bound (Ho et al., 2020; Kingma et al., 2021; Kingma & Gao, 2023) provides a tractable training signal: it relates this likelihood to how well the explanation helps the reconstructor denoise the activation across noise levels. The bound takes the form $\log p_\theta(h \mid z) \geq R_\theta(h, z) - c(h)$ (Appendix I), where $R_\theta(h, z)$ is the explanation-dependent denoising term and $c(h)$ is independent of

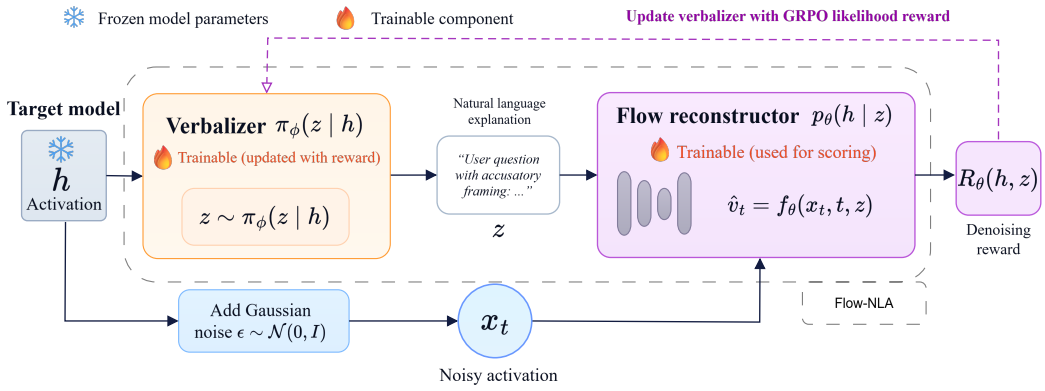


Figure 5: **Learning activation explanations with Flow-NLA.** An activation h from a frozen target model is passed to a verbalizer, which generates an explanation z, and corrupted with Gaussian noise to form x_t. The flow reconstructor learns to denoise x_t conditioned on z, modeling the distribution of activations compatible with the explanation. Denoising errors across noise levels define a reward derived from a conditional diffusion likelihood bound. This reward trains the verbalizer to produce explanations that make the observed activation more predictable under the conditional model.

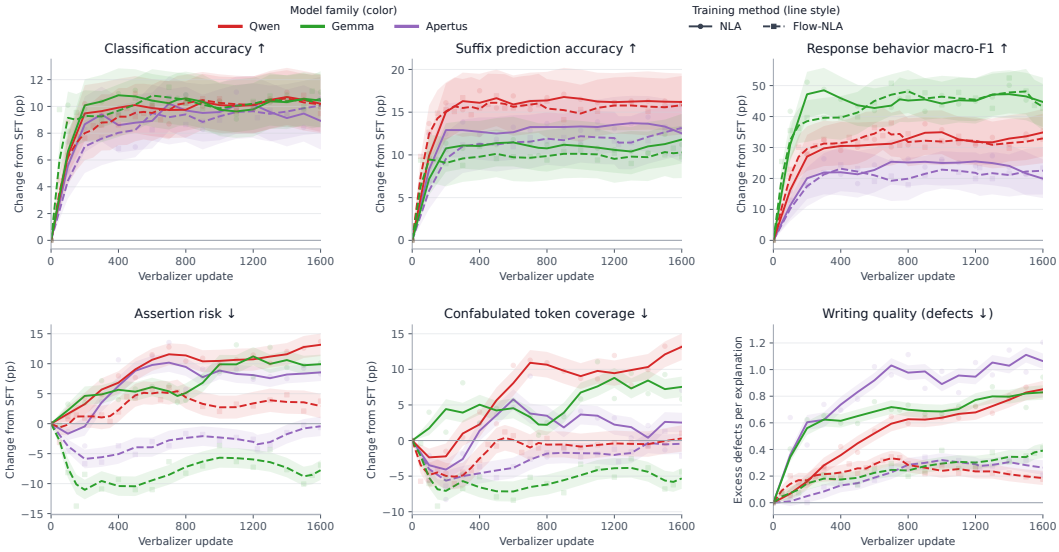


Figure 6: Changes in utility, confabulation, and writing defects from supervised initialization for point-reconstruction NLA (solid) and Flow-NLA (dashed). Markers show raw measurements, curves show three-checkpoint means, and shading shows pointwise 95% bootstrap intervals.

the explanation. We therefore use R_θ as the verbalizer's reward, so that candidate explanations of the same activation are compared through their likelihood bounds. Flow matching trains the reconstructor to predict the velocity v_t, whereas the bound is expressed in noise-prediction error. The definitions of x_t and v_t imply $b_t x_t + a_t v_t = (a_t^2 + b_t^2)\epsilon = \epsilon$. Substituting the predicted velocity therefore gives $\widehat{\epsilon}_\theta(x_t, t, z) = b_t x_t + a_t f_\theta(x_t, t, z)$, without training a second predictor.

We index noise levels by log signal-to-noise ratio, $\lambda = \log(a_t^2/b_t^2)$: low values correspond to mostly noise and high values to mostly signal. In this coordinate, the likelihood-bound term integrates noise-prediction error with constant weight $1/2$ (Appendix I). We use $\lambda_{\min} = -8$ and $\lambda_{\max} = 8$, a finite band spanning both regimes. The reward is the negative integrated noise-prediction error:

$$R_\theta(h, z) = -\frac{1}{2} \int_{\lambda_{\min}}^{\lambda_{\max}} \mathbb{E}_\epsilon \left\| \epsilon - \widehat{\epsilon}_\theta(x_\lambda, \lambda, z) \right\|_2^2 \, \mathrm{d}\lambda. \tag{3}$$

For optimal denoisers, each integrand is the clean-activation denoising error of Figure 4 scaled by e^λ, since $\epsilon - \widehat{\epsilon}^* = e^{\lambda/2}(\mathbb{E}[X \mid x_\lambda, z] - x)$. The reward therefore credits an explanation for every noise level at which it makes the observed activation easier to recover, including when the explanation leaves the conditional mean unchanged; in the example of Figure 4, R_θ favors z_{hor} over an uninformative explanation by approximately $\log 2$ nats, while the point reward r_θ is unchanged. This information-theoretic interpretation motivates constant weighting in log-SNR: it matches the likelihood-bound term, whereas arbitrary reweightings do not (Appendix J). The trained reconstructor and numerical integration approximate this ideal quantity (Kingma et al., 2021).

Training the verbalizer. We follow the standard NLA training recipe described in Section 3, retaining the verbalizer's supervised initialization and group-relative updates with a KL penalty (Fraser-Taliente et al., 2026). The reconstructor's supervised initialization differs, as it is trained for velocity prediction rather than point reconstruction. We replace the point-reconstruction reward with $R_\theta(h, z)$, estimated using sampled Gaussian noise and numerical integration over log-SNR $[-8, 8]$, and fit the online reconstructor to the generated activation–explanation pairs using the denoising objective, in the same order as the standard recipe. Because the distribution behind this reward moves with the verbalizer, several choices mattered for training (Appendix F). Integrating over noise levels avoids uninformative extremes: nearly clean inputs reveal the activation regardless of the explanation, while nearly pure noise leaves only the conditional mean (Appendix H). To assess whether the reconstructor captures more distributional information as the verbalizer learns, we compare sample and real activation covariance within and outside the activations' 64 leading prin-

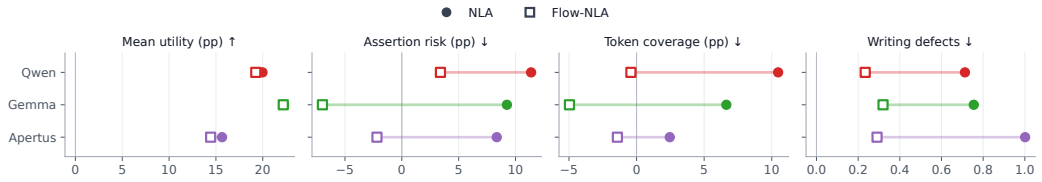


Figure 7: Changes from supervised initialization in mean utility, assertion risk, token coverage, and writing defects, averaged over the second half of the training range shared by both methods. Filled circles denote NLA; open squares denote Flow-NLA; lines show the gap between methods.

cipal components (Appendix F). This diagnostic ruled out our initial energy-score reward (Gneiting & Raftery, 2007), which compares the observed activation with reverse-generated samples: their spread outside the leading subspace decreased as mean reconstruction improved. The likelihood-bound reward requires no reverse-generated samples, only forward noising of the observed activation, and its reconstructor's spread outside the leading subspace converges to that of the activations.

4.3 Utility and Confabulation under Richer Feedback

We evaluate whether Flow-NLA changes the balance between recoverable information and explanation defects using the evaluation framework from Section 3.1. We compare Flow-NLA with point-reconstruction NLA on Qwen2.5-7B-Instruct, Gemma-3-12B, and Apertus-v1.5-8B, examining whether the effect of richer feedback is consistent across LLMs. For each model, we compare both methods' training trajectories in utility, contextual confabulation, and writing quality (Figure 6).

On all three target models, the two methods follow similar utility trajectories, while confabulation and writing defects grow significantly less under Flow-NLA. Figure 7 summarizes this late in training: Flow-NLA retains the mean utility gain (averaged over classification, suffix, and behavior), while its assertion risk and token coverage stay near or below supervised initialization rather than rising by up to 11 points, and its writing defects grow by 0.23–0.32 rather than 0.71–1.00 per explanation. Flow-NLA also has the lowest assertion risk, token coverage, and writing defects of all compared NLAs, including the released ones (Table 2, Appendix A).

Across target models, substantial utility gains can thus be retained with less growth in confabulation and defects, establishing richer reconstruction feedback as a promising direction for NLA training.

5 Conclusions, Limitations, and Future Work

Current NLAs grow less reliable as they learn: during point-reconstruction training, explanations become more useful but also more confabulated and less clearly written, and the released NLAs show the same pattern. Our evaluation framework exposes this by scoring utility, source support, and writing quality separately. This deterioration points to a limitation of the current objective: point reconstruction rewards whatever text helps recover the activation, while remaining blind to much of the information an explanation carries. Flow-NLA replaces it with a conditional denoising reward that credits this richer information. Across three target models, Flow-NLA keeps its utility gains while sharply curbing the growth of confabulation and writing defects during training: its writing defects grow by 0.23–0.32 rather than 0.71–1.00 per explanation, and assertion risk stays near initialization instead of rising by up to 11 points. We therefore recommend that NLA work report source support and writing quality alongside reconstruction, since lower training loss alone can reflect more confabulation rather than merely better explanations.

Our study is limited to the models, tasks, and training procedures considered, relies on automated judgments, and includes provider and inference differences that limit causal attribution to the reconstruction objective alone (Appendix B). Our method and analysis also assume optimal denoisers and a fixed verbalizer, and richer reconstruction feedback does not guarantee source support or mechanistic faithfulness. Future work should better characterize how reconstruction rewards induce confabulation and design training objectives that encourage informative, supported explanations.

AI USE STATEMENT

We used generative AI tools to help validate theory and refine experiment design. We also used these tools for coding and debugging support, processing result files, preparing scientific figures and tables, and searching the scientific literature. Further, LLMs form part of the evaluation framework; DeepSeek V4.1 Flash serves as the judge for explanation utility, source support, and writing quality. The authors reviewed and revised all AI-assisted text and code; we take responsibility for the full content of this work.

ETHICS STATEMENT

Our work aims to improve the reliability and interpretability of LLMs by improving natural-language activation explanations. Improvements on these metrics do not mean the explanations are faithful or fair. The evaluation uses existing text datasets, including conversations from LMSYS-Chat-1M. Conversation data can contain sensitive or identifying information even when publicly available, and activation explanations may reproduce or infer such information. Reuse of the data and explanations, including processing by external model providers and redistribution of research artifacts, requires attention to the source datasets' privacy protections, and applicable licenses.

REFERENCES

David Barber and Felix V. Agakov. Information maximization in noisy channels: A variational approach. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL https://proceedings.neurips.cc/paper/2003/hash/a6ea8471c120fe8cc35a2954c9b9c595-Abstract.html.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 449–458. PMLR, 2017. URL https://proceedings.mlr.press/v70/bellemare17a.html.

Sviatoslav Chalnev. Cycle-consistent activation oracles. LessWrong research post, March 2026. URL https://www.lesswrong.com/posts/Nf2sKaNNdxE2ssxbp/cycle-consistent-activation-oracles-1.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression, 2026. URL https://arxiv.org/abs/2609.19969.

Hiskias Dingeto. Train the model, not the reader: Decodability supervision for verifiable activation explanations, 2026. URL https://arxiv.org/abs/2607.20379.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.

Kit Fraser-Taliente, Subhash Kantamneni, Euan Ong, Dan Mossing, Christina Lu, Paul C. Bogdan, Emmanuel Ameisen, James Chen, Dzmitry Kishylau, Adam Pearce, Julius Tarng, Alex Wu, Jeff Wu, Yang Zhang, Daniel M. Ziegler, Evan Hubinger, Joshua Batson, Jack Lindsey, Samuel Zimmerman, and Samuel Marks. Natural language autoencoders produce unsupervised explanations of LLM activations. *Transformer Circuits Thread*, 2026. URL https://transformer-circuits.pub/2026/nla/index.html.

Gemma Team. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

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

Dongning Guo, Shlomo Shamai, and Sergio Verdú. Mutual information and minimum mean-square error in Gaussian channels. *IEEE Transactions on Information Theory*, 51(4):1261–1282, 2005. doi: 10.1109/TIT.2005.844072. URL https://arxiv.org/abs/cs/0412108.

Alejandro Hernández-Cano, Alexander Hägele, Allen Hao Huang, et al. Apertus: Democratizing open and compliant LLMs for global language environments. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 46877–46955, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.2172. URL https://aclanthology.org/2026.acl-long.2172/.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://arxiv.org/abs/2006.11239.

Adam Karvonen, James Chua, Clément Dumas, Kit Fraser-Taliente, Subhash Kantamneni, Julian Minder, Euan Ong, Arnab Sen Sharma, Daniel Wen, Owain Evans, and Samuel Marks. Activation oracles: Training and evaluating LLMs as general-purpose activation explainers. In *International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2512.15674.

Adam Karvonen, Euan Ong, Subhash Kantamneni, and Samuel Marks. Would this change your answer? Evaluating explanations of LLM behavior in the wild with counterfactual experiments, 2026b. URL https://arxiv.org/abs/2608.16747.

Diederik P. Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.00848.

Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21696–21707, 2021. URL https://arxiv.org/abs/2107.00630.

Millicent Li, Alberto Mario Ceballos Arroyo, Giordano Rogers, Naomi Saphra, and Byron C. Wallace. Do activation verbalization methods convey privileged information? In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=IFGTKwHwK6.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1509.02971.

Yishu Miao and Phil Blunsom. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 319–328, 2016. doi: 10.18653/v1/D16-1031.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

Alexander Pan, Lijie Chen, and Jacob Steinhardt. LatentQA: Teaching LLMs to decode activations into natural language. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2412.08686.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

Mathieu Rita, Corentin Tallec, Paul Michel, Jean-Bastien Grill, Olivier Pietquin, Emmanuel Dupoux, and Florian Strub. Emergent communication: Generalization and overfitting in Lewis games. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023. doi: 10.1109/TPAMI.2022.3204461.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9447–9474. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.552. URL https://aclanthology.org/2024.findings-emnlp.552/.

An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Michael Zhang and Alex Turner. How robust are natural language autoencoders to initialization? Research report, July 2026. URL https://turntrout.com/natural-language-autoencoder-robustness.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. LMSYS-Chat-1M: A large-scale real-world LLM conversation dataset. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5f9bfdfe3685e4ccdbc0e7fb29cccf2a-Abstract-Conference.html.

A Absolute Model Comparison

Table 2: **Utility and confabulation across NLA training methods.** We compare released NLAs, our point-reconstruction reruns, and Flow-NLA within the same target model. All scores are percentages except writing defects, which are counts per explanation. Best results are highlighted within each target model. Checkpoint selection is described in Appendix B.2.

		Utility ↑			Confabulation ↓		Writing ↓
Target model	Method	Class.	Suffix	Behavior	Risk	Coverage	Defects
Qwen2.5-7B	Released	79.4	**98.0**	**82.7**	92.6	74.8	2.45
	Point recon.	79.7	97.6	74.9	83.7	68.3	1.63
	Flow-NLA	**80.3**	96.6	72.9	**71.9**	**54.7**	**0.99**
Gemma-3-12B	Released	**84.7**	**96.6**	83.4	83.1	67.5	1.78
	Point recon.	84.2	96.2	82.4	77.8	44.2	1.38
	Flow-NLA	83.8	96.2	**84.8**	**60.8**	**33.8**	**0.89**
Apertus-v1.5-8B	Point recon.	77.5	**96.0**	**73.6**	81.3	41.5	1.50
	Flow-NLA	**78.4**	95.4	69.6	**72.8**	**38.9**	**0.75**

B Evaluation Details

B.1 Activation extraction

Table 3 lists the extraction site for each target model. As in the original NLA code (Fraser-Taliente et al., 2026), our reruns and Flow-NLA share the extraction site of their target model. Training activations come from FineWeb (Penedo et al., 2024) `sample-10BT` documents with at least 128 tokens, truncated to 1,024 tokens, with 10 token positions per document. Documents are split disjointly into 25% verbalizer supervised initialization, 25% reconstructor supervised initialization, and 50% online training.

Table 3: Activation extraction sites. Revisions are Hugging Face commit prefixes; released NLAs are identified by their repository names. A released NLA at layer K reads the output of decoder block K, i.e., `hidden_states[`K`+1]`.

	Target model	Revision or release	Hidden state
Ours	Qwen2.5-7B-Instruct	`a09a354`	`hidden_states[21]`
	Gemma-3-12B-it	`96b6f1e`	`hidden_states[33]`
	Apertus-v1.5-8B	`a411d83`	`hidden_states[22]`
Released	Qwen2.5-7B-Instruct	`nla-qwen2.5-7b-L20`	`hidden_states[21]`
	Gemma-3-12B-it	`nla-gemma3-12b-L32`	`hidden_states[33]`
	Gemma-3-27B-it	`nla-gemma3-27b-L41`	`hidden_states[42]`
	Llama-3.3-70B-Instruct	`Llama-3.3-70B-NLA-L53`	`hidden_states[54]`

Evaluation token positions. For classification, the activation is extracted at the last source-text token, before any evaluation question. For suffix prediction, it is extracted at the last prefix token, before the true continuation. For response behavior, it is extracted at the first token of a freshly generated response from the target model. These positions are the same for all target models. Explanations are sampled at temperature 1 with at most 150 new tokens.

B.2 Evaluation protocols

All judges, including the hard-negative judge (Appendix D), are DeepSeek V4.1 Flash (DeepSeek-AI, 2026) with thinking disabled and temperature zero. Prompt templates are described in Appendix G.

Utility tasks, controls, and aggregation. We evaluate utility on source-property classification, suffix prediction, and response-behavior prediction. In all cases, the utility judge receives the task query and the explanation.

Table 4: Overview of our various utility tasks

Task	Judge's task	Examples	Options	Metric
Classification	Source properties (12 tasks)	1,344	2–14	Mean accuracy
Suffix selection	True 32-token continuation	500	10	Accuracy
Response behavior	Answer, refuse, or clarify	248	3	Macro-F1
Hard negatives	Contrastive question on source pair	1,316 (658 pairs)	2	Log-prob. ratio

For target instance i, let q_i denote the task query, $\mathcal{C}_i$ the candidate answers, y_i the reference answer, z_i the matched explanation, and $z_{j(i)}$ the hard-negative explanation, generated for the paired source $j(i)$. Where predictive probabilities over $\mathcal{C}_i$ are available, we define

$$\begin{aligned} L_{\mathrm{matched},i} &= -\log p_{\mathrm{J}}(y_i \mid q_i, \mathcal{C}_i, z_i), \\ L_{\mathrm{noexp},i} &= -\log p_{\mathrm{J}}(y_i \mid q_i, \mathcal{C}_i, \varnothing), \\ L_{\mathrm{hard},i} &= -\log p_{\mathrm{J}}(y_i \mid q_i, \mathcal{C}_i, z_{j(i)}). \end{aligned} \tag{4}$$

We then decompose the hard-negative separation as

$$\begin{aligned} U_{\text{gain},i} &= L_{\text{noexp},i} - L_{\text{matched},i}, \\ U_{\text{penalty},i} &= L_{\text{hard},i} - L_{\text{noexp},i}, \\ U_{\text{contrast},i} &= L_{\text{hard},i} - L_{\text{matched},i} = U_{\text{gain},i} + U_{\text{penalty},i}. \end{aligned} \tag{5}$$

Thus, U_{gain} measures the predictive benefit of the matched explanation over no explanation, whereas U_{penalty} measures the degradation caused by the hard-negative explanation. Their sum is the hard-negative separation. We report these components separately because a larger separation may reflect more useful matched explanations, more misleading hard negatives, or both. Appendix D describes how hard-negative pairs are constructed and reports the decomposition for all runs.

The classification suite contains 112 source examples for each of 12 tasks, covering topic, sentiment, subjectivity, entailment, entity and exact-name presence, tense, grammatical number, language, and pronoun form, for 1,344 examples per checkpoint and method. We average accuracy over successful generations and parsed responses within each task and then weight the 12 tasks equally; details and data sources are in the provided repository.

Suffix prediction uses 500 documents with ten fixed candidate continuations per example. The activation is extracted at the end of the prefix before the true 32-token continuation.

Response-behavior prediction uses 248 LMSYS-Chat-1M (Zheng et al., 2024) conversations. The judge sees only the explanation and makes a forced prediction of answer, refusal, or clarification. Reference labels come from each target model's actual response. We report the macro-F1 score.

Source-support judgments. The source-support judge extracts distinct assertions and assigns supported, contradicted, unsupported, or not assessed verdicts. Reasonable entailments count as supported; a contradiction requires incompatible source evidence. Future-text predictions and hidden-computation claims are not assessed and are excluded from both metrics. Token coverage counts the union of tokens overlapping unsupported or contradicted claims.

Writing defects. The writing judge sees only the explanation. It identifies distinct defects in clarity, coherence, and economy, assigning each defect one category and an exact evidence span. An annotation is accepted only when the quoted span occurs exactly in the explanation.

Checkpoint comparison. Classification, suffix, and confabulation changes use each series' own supervised initialization as reference. The models are evaluated every 100 updates through 1600. Tables 1 and 2 report absolute checkpoint scores rather than changes from supervised initialization. We select one checkpoint per run by the highest unweighted mean of classification accuracy, suffix accuracy, and response-behavior macro-F1 among updates 1300–1600 for a fair comparison. Released checkpoints are evaluated unchanged, with the same judges and protocols as our runs. Classification, assertion risk, and token coverage weight the 12 classification tasks equally.

C Individual Evaluation Panels

Figure 8 shows the individual utility, confabulation, and writing trajectories of the point-reconstruction NLA reruns summarized in Section 3.2, including the assertion-risk and writing-defect components of the confabulation index.

D Hard-Negative Separation

Pair construction. Hard-negative pairs are drawn from the classification sources. For each of the 12 datasets, we retrieve lexical nearest neighbors and write one question that the two sources answer differently (factual, linguistic, or evaluative), with the two answers as candidates. This gives 658 disjoint pairs covering 1,316 of the 1,344 classification sources.

Judge and aggregation. The judge is DeepSeek V4.1 Flash with thinking disabled, as for the other utility tasks. It answers with an option letter, and we read the log-probability of the correct letter, normalized over the two letters. Each question is scored with both answer orders to cancel

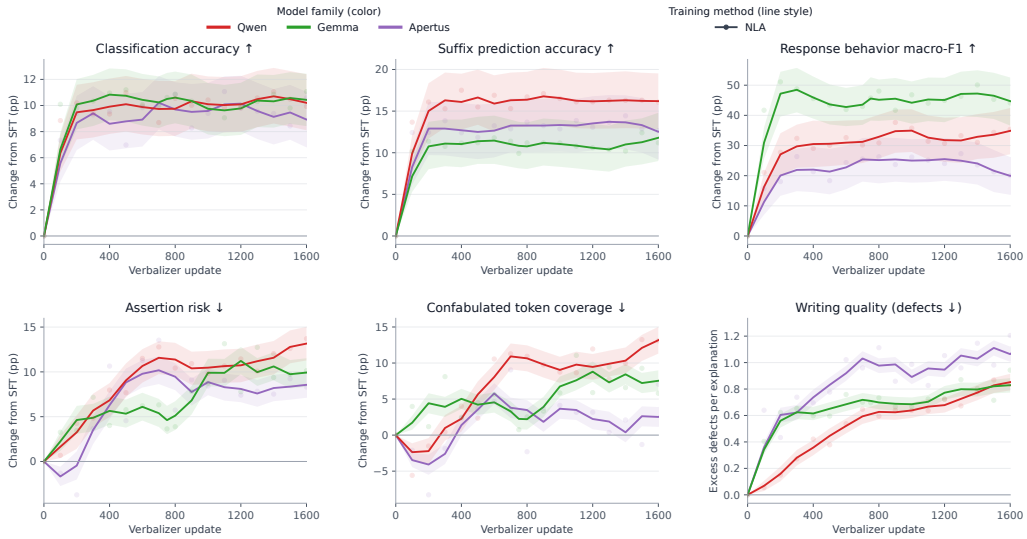


Figure 8: Changes in utility, confabulation, and writing defects from supervised initialization for the point-reconstruction NLA reruns. Shading shows pointwise 95% bootstrap intervals.

letter-position bias. Losses follow Eq. 4 and the decomposition follows Eq. 5. We average within each dataset and then weight datasets equally.

Decomposition. Figure 9 separates the total separation of Figure 3 into matched predictive gain U_{gain} and mismatch penalty U_{penalty}. Without an explanation, the judge is at chance on every pair. Most of the separation, and most of its rise during training, comes from the mismatch penalty: it grows from 0.44–0.83 nats at supervised initialization to 0.91–1.86 nats from update 800 onward, while the matched gain grows from 0.19–0.29 to 0.31–0.49 nats. Both components level off after about 200 updates. Late in training, Flow-NLA matches NLA on both components for Apertus and has a slightly lower mismatch penalty for Qwen and Gemma (by 0.10 and 0.16 nats).



Figure 9: Matched predictive gain, mismatch penalty, and total hard-negative separation across the 12 classification datasets for point-reconstruction NLA (solid) and Flow-NLA (dashed). Curves are three-checkpoint means over raw markers; shading shows 95% bootstrap intervals.

E Training Details

Supervised initialization uses explanations written by Qwen3.6-35B-A3B for the activation sites of Appendix B.1. Each run uses 8 NVIDIA GH200 GPUs. For reruns, we use the same hyperparameter settings as the released Anthropic code (Fraser-Taliente et al., 2026); for Apertus, we use the hyperparameters of Qwen2.5-7B as it is the closest in parameter size.

F MAKING THE DISTRIBUTIONAL REWARD TRAINABLE

Using a co-trained likelihood bound as a policy-gradient reward raised problems that a fixed reward model does not. This section records the failure modes we encountered and how we addressed them. All diagnostics reported here are for Qwen2.5-7B-Instruct.

The reconstructor tracks a moving target. The conditional distribution $p_\phi(h \mid z)$ changes with the verbalizer, and the reconstructor measurably follows it. Between updates 0 and 200, the co-trained reconstructor's score improved by 38.8 units on explanations from the update-200 verbalizer, but by only 5.7 units on explanations from supervised initialization. A frozen reconstructor also confirms that the verbalizer's gains are real: it scores update-200 explanations above initial ones (0.466 versus 0.332). Monitoring with a fixed reconstructor therefore measures transfer to old explanations, not joint progress.

Scoring with an averaged reconstructor. Unlike the released NLA recipe, which scores explanations with the current reconstructor (Fraser-Taliente et al., 2026), we score them with an exponential moving average (EMA) of the reconstructor's parameters with decay 0.99. Rewards thus follow the moving conditional distribution gradually rather than jumping with each reconstructor update. Such parameter averaging is common in off-policy RL, where the targets that critics are trained on are computed with an EMA of their parameters (Lillicrap et al., 2016).

Single noise levels give weak credit. An early reward used velocity error at a single uniformly sampled noise level. At nearly clean inputs, the noisy activation reveals the target and alternative explanations receive almost the same score. The gap between mismatched and matched explanations was 0.003 at $t = 0.05$ versus 0.625 at $t = 0.95$. At pure noise the reward reduces to point reconstruction (Appendix H). In the retained reward, the variance that ranks candidate explanations comes mostly from intermediate log-SNR: 2.3% from $\lambda \leq -4$, 34.9% from $(-4, 0]$, 60.9% from $(0, 4]$, and 1.9% from $\lambda > 4$. Integrating across the band with likelihood weighting captures this signal without choosing a single noise level.

Monitoring distributional fit. Reconstruction FVE measures only how well the reconstructor recovers the mean, so we also check whether the spread of its samples matches the spread of the activations. On a fixed panel of held-out activation–explanation pairs, we draw 16 samples per explanation by running the reconstructor in reverse from noise. We compare the covariance of these samples around their mean with the covariance of the activations around the reconstructor's mean prediction, separately inside and outside the subspace spanned by the 64 leading principal components of real activations. The PCA-64 covariance overlap is one minus the normalized squared Bures distance between the two covariances within this subspace; it equals one when they are identical. The off-PC variance ratio divides the samples' variance outside the subspace by that of the activations, corrected for the finite number of samples; it equals one when the model is calibrated there, and values below one indicate samples that are too concentrated. For the reconstructor used by the likelihood-bound reward, the off-PC variance ratio moves from 1.26 to 1.00 over reconstructor training steps 50–1,600, while the PCA-64 overlap rises from 0.77 to 0.89.

An energy-score reward. Before adopting the likelihood bound, we rewarded the verbalizer with the negative energy score (Gneiting & Raftery, 2007) of $K = 16$ reverse samples $\widehat{x}^{(1)}, \ldots, \widehat{x}^{(K)}$ per explanation,

$$\mathrm{ES}(x) = \frac{1}{K} \sum_{k} \|\widehat{x}^{(k)} - x\|_2 - \frac{1}{2K(K-1)} \sum_{k \neq l} \|\widehat{x}^{(k)} - \widehat{x}^{(l)}\|_2, \tag{6}$$

a proper scoring rule minimized in expectation by the true conditional distribution, with common noise across candidates. Unlike the likelihood bound, which only requires noising the observed activation forward, this reward depends on samples generated by running the reconstructor in reverse, so its value is only as good as the spread of those samples. The reconstructor trained for this reward lost spread outside the leading subspace: its off-PC variance ratio fell from 0.96 to 0.78 over reconstructor training steps, while its PCA-64 overlap stayed near 0.66 and FVE rose from 31.8% to 49.6%. Better mean reconstruction thus came with increasingly concentrated samples during energy reward training.

G JUDGE PROMPTS

The exact prompt templates are in the `prompts/` directory of the evaluation code in our repository. The utility judge receives the task query, its candidate answers, and the explanation, and returns a single answer; classification, suffix selection, and response behavior each have their own template. The source-support judge receives the source text and the explanation, splits the explanation into atomic claims, and returns a verdict for each claim with quoted source evidence for supported and contradicted claims. The writing judge receives only the explanation and returns each defect with its category and quoted span. All judges return structured output, which is validated before scoring.

H WHY DENOISING IDENTIFIES THE CONDITIONAL DISTRIBUTION

This section works through the example of Figure 4 and then shows that denoising at an interior noise level identifies the explanation-conditioned distribution.

Illustrative example. In Figure 4, the activation is uniform over $(\pm 1, 0)$ and $(0, \pm 1)$, and the explanation z_{hor} identifies the horizontal pair. Both the unconditional and the z_{hor}-conditional means are zero, so the optimal unit-norm point loss is 2 with or without the explanation. For this illustration, $x = h$ and $X_t = a_t X + b_t \epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$ and $a_t^2 + b_t^2 = 1$. Panels (b)–(d) fix $b_t/a_t = 0.6$ and use the observation $x_t = a_t(0.55, 0.5)$ of the true activation $(1, 0)$; shading shows the Gaussian-corrupted conditional densities. Writing $m_z(x_t, t) = \mathbb{E}[X \mid X_t = x_t, Z = z]$, the field arrows in panels (c)–(d) depict

$$d_z(x_t, t) = a_t m_z(x_t, t) - x_t = -b_t \widehat{\epsilon}^*(x_t, t, z) = b_t^2 \nabla_{x_t} \log p_t(x_t \mid z). \tag{7}$$

Thus, the displacement fields encode gradients of the noisy conditional log-density. Flow-NLA scores noise predictions across noise levels using the likelihood bound in Appendix I, allowing its reward to distinguish these equal-mean distributions. With equally likely groups, group information leaves optimal point loss at 2 but improves the optimal expected integrated denoising reward by approximately $\log 2$ nats over log-SNR $[-8, 8]$, computed using exact posterior predictors and Gaussian quadrature (Appendix J).

Distribution identification. Fix the verbalizer, so that the joint activation–explanation distribution is fixed. At an interior noise level with $a_t, b_t > 0$, squared-error training yields the population-optimal field $f^*(x_t, t, z) = \mathbb{E}[v_t \mid x_t, z]$. Its implied noise prediction is $\widehat{\epsilon}^*(x_t, t, z) = \mathbb{E}[\epsilon \mid x_t, z]$. Gaussian corruption relates this prediction to the score of the noisy density, $\nabla_{x_t} \log p_t(x_t \mid z) = -\widehat{\epsilon}^*(x_t, t, z)/b_t$, where $\widehat{\epsilon}^*(x_t, t, z) = b_t x_t + a_t f^*(x_t, t, z)$. Its score determines it up to a normalization constant, fixed by requiring it to integrate to one. Recovering the noisy distribution therefore identifies the original conditional activation distribution at optimality. At the pure-noise endpoint $a_t = 0$, the velocity target is $-X$ and the optimum depends only on the conditional mean. Interior noise levels are what allow equal-mean distributions to be distinguished.

I THE CONDITIONAL LIKELIHOOD BOUND

The reward in Eq. 3 is the negative interior loss of a conditional diffusion variational bound (Ho et al., 2020; Kingma et al., 2021; Kingma & Gao, 2023). We define the diffusion model on the chosen finite noise band, with an explanation-independent Gaussian prior at the low-SNR boundary and an explanation-independent observation decoder at the high-SNR boundary. The explanation conditions the reverse transitions between these endpoints. This specification ensures that the omitted endpoint terms do not vary between candidate explanations of the same activation.

Because activations lie on a sphere, we take densities with respect to the image ν of sphere surface measure under the fixed transform $x = T(h) = (\sqrt{d}\, h - \mu)/s$. Following Kingma et al. (2021), the decoder is $r(x \mid x_{\mathrm{hi}}) \propto q(x_{\mathrm{hi}} \mid x)$, normalized over ν, where q is the forward Gaussian corruption at the high-SNR boundary.

Applying the standard variational bound separates the endpoint terms from KL divergences between forward posteriors and learned reverse transitions. For Gaussian reverse transitions with the forward posterior variance, the interior terms are weighted squared noise-prediction errors. In the continuous

log-SNR limit, under the regularity and integrability conditions for that limit, their sum becomes the negative of Eq. 3. Consequently, $\log p_\theta(h \mid z) \geq R_\theta(h, z) - c(h)$, where $c(h)$ contains the endpoint terms and is independent of z. The bound applies to a well-defined reverse process without requiring optimal denoising.

The velocity and noise parameterizations obey $\epsilon - \widehat{\epsilon}_\theta = a_t(v_t - f_\theta)$, so the reward weights velocity error by a_t^2 before integrating over log-SNR. This differs from the reconstructor's coordinate-averaged training loss. Sampled noise and numerical integration approximate the bound-derived reward; computing candidate scores does not require evaluating the fixed observation decoder.

J Information Gain Across the Noise Band

Fix the verbalizer and assume optimal denoisers with and without the explanation. Let $\mathcal{E}_Z(\lambda)$ and $\mathcal{E}_0(\lambda)$ denote their expected squared noise-prediction errors, and write $Y_\lambda = e^{\lambda/2}X + \epsilon$, where ϵ is independent standard Gaussian noise. Under finite second moments, the I-MMSE relation (Guo et al., 2005) gives

$$\frac{1}{2}\int_{\lambda_{\min}}^{\lambda_{\max}} [\mathcal{E}_0(\lambda) - \mathcal{E}_Z(\lambda)]\,\mathrm{d}\lambda = I(Z; Y_{\lambda_{\max}}) - I(Z; Y_{\lambda_{\min}}). \tag{8}$$

The explanation's denoising benefit therefore measures information gained across the selected noise band. An added detail can improve this quantity by distinguishing modes or other distributional structure even when the conditional mean remains unchanged. The equal-mean example in the main text illustrates such a distinction. The finite band omits slight information already visible at its low-SNR boundary and information still unresolved at its high-SNR boundary. The unconditional error contributes the same term to every candidate score for a fixed activation and can be omitted without changing candidate comparisons.