Compression at a Cost: Interpreting Information Bottlenecks in Safety-Aligned Reward Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Aligning Large Language Models with human intent relies heavily on Reward Models (RMs), which frequently exploit spurious correlations rather than internalizing robust human preferences, particularly in safety-critical settings. Recent information-theoretic approaches attempt to mitigate this by applying an information bottleneck (IB) to the latent space, theoretically pruning spurious features while preserving core alignment signals. However, the precise mechanistic impact of this compression on internal representations remains opaque. In this paper, we present the first mechanistic interpretability analysis of IB-regularized RMs trained on safety-oriented preference datasets. By training Sparse Autoencoders (SAEs) on the representations immediately preceding and following the bottleneck, we systematically track survival of semantic features across varying compression penalties (β). Our analysis reveals that compression acts selectively rather than uniformly; while spurious structures are entirely eradicated (a 100% drop), safety-relevant features are simultaneously attenuated. We explicitly map these latent representational shifts to macro-level behavioral evaluations on RewardBench, observing a severe capability trade-off where standard RMs outperform the optimal β configuration in aggregate mean score. Taken together, our semantic and empirical evidence indicates that while information bottlenecks successfully distill critical safety concepts, they exact a massive alignment tax, producing hyper-specialized safety auditors at the expense of robust, general-purpose preference modeling. *This work analyzes reward model safety and contains discussions and examples highlighting potential risks and harmful model outputs.*

1 Introduction

Reward models (RMs) are central to RLHF pipelines, shaping optimization targets and downstream model behavior. As these systems scale, compression and regularization techniques such as Variational Information Bottlenecks (VIBs) are increasingly deployed to mitigate reward hacking. However, compression is traditionally evaluated through aggregate behavioral metrics alone, leaving a critical question unanswered: *What representational structure is actually removed when information is constrained inside a reward model?*

This question matters particularly in safety-aligned settings. While removing spurious structure is desirable, a blunt compression mechanism may simultaneously attenuate the very structures that support safety-critical distinctions. When spurious suppression and safety attenuation are entangled inside the bottleneck, aggregate scores can look acceptable while the internal safety signal is severely depleted.

We investigate this using the InfoRM [17] architecture, where an explicit information bottleneck is controlled by a penalty parameter (β). This design isolates two inspection points: representations

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

immediately preceding the bottleneck, and representations immediately following it. To examine these points, we introduce Sparse Autoencoders (SAEs) as a mechanistic lens. By training SAEs on these pre- and post-bottleneck layers and applying a fixed semantic scoring pipeline, we track the survival of individual latent features. This yields a category-aware view of retention, allowing us to ask not only how much is compressed but what specifically is lost.

Our central claim is that in InfoRM setting, IB acts as a selective but costly representational bottleneck, suppressing spurious structure while also attenuating safety-relevant structure, and these representational changes are associated with downstream benchmark trade-offs. We support this through two complementary lenses:

- **The Semantic Lens :** Our semantic analysis reveals an asymmetric pruning pattern. We find that while compression is highly effective at eliminating spurious structures, it simultaneously removes the majority of what it should keep. At the 3B scale this attenuation is partially mitigated, suggesting the alignment tax is sensitive to representational redundancy in the backbone.
- **The Behavioral Lens :** To verify that these latent shifts are not isolated internal quirks, we evaluate the same models on RewardBench [14]. The representational decay maps directly to external performance trade-offs, where standard RM outperform compressed models in aggregate and safety domains, confirming that internal attenuation has visible downstream consequences.

The contribution of this paper is not a broad causal theory of bottlenecked reward models. Instead, it is a focused mechanistic account for a concrete RLHF setting, with evidence gathered across semantics and benchmark behavior. This framing matters methodologically: aggregate metrics alone obscure which internal structures are being removed, while purely mechanistic analysis can miss external consequences. Combining both provides a tighter and more decision-relevant evaluation of compression in safety-aligned reward models.

In summary, we present a before/after bottleneck analysis that characterizes what compression removes, not only how models score after compression. The findings suggest that compression should be assessed as a selective representational operation with potential safety-performance trade-offs, rather than as a uniformly beneficial regularizer.

2 Related Work

Reward overoptimization in RLHF Reward hacking, or reward overoptimization, has posed a prominent challenge in RLHF, in which the policy achieves high proxy scores while diverging from true human objectives [28, 10]. Empirical scaling laws show that overoptimization persists regardless of proxy model size or data volume [10], and the problem extends to direct alignment algorithms [23]. Mitigation strategies fall into several families. KL divergence penalties between the policy and the SFT reference model remain the default regularizer in PPO-based pipelines [22, 29], but constrain the optimization landscape rather than addressing the underlying cause, and are prone to overfitting at large KL budgets [19]. Ensemble-based methods reduce individual model variance, though individual members often share errors [4, 8]. Bayesian reward models supply calibrated uncertainty estimates to suppress out-of-distribution exploitation [33], with recent extensions using non-negative decompositions to separate spurious from genuine preference signals [21]. A parallel line targets known spurious features directly: length debiasing [27], weight averaging across checkpoints [26], and causal representation learning that theoretically identifies non-spurious latent variables from preference data [20]. Our work is complementary: rather than proposing a new mitigation strategy, we open the bottleneck representation itself to feature-level inspection and ask which internal structures compression removes. This is a mechanistic question that behavioral benchmarks alone cannot answer.

Information bottleneck for representation learning and reward modelling. The Information Bottleneck (IB) principle [31] formalizes representation learning as a trade-off between compression and predictive fidelity. The variational approximation of Alemi et al. [1] makes this tractable via reparameterised Gaussian codes, and has since been applied to domain generalisation [6], and graph learning [32]. InfoRM [17] applies this to reward modeling: by inserting a variational IB layer

between the transformer backbone and the reward head, with the goal of filtering preference-irrelevant information and reducing reward misgeneralization.

Related work includes VRPO [35], which applies a variational bottleneck to the value model to suppress noisy supervision signals, and Miao et al. [18], which couples IB-based reward modeling with distribution-level RL regularization. Concept Bottleneck Reward Models [13] pursue a complementary direction by decomposing reward predictions into human-interpretable concept scores.

These works treat IB as an instrument for improving downstream behavioral performance. Our work takes a different stance. Rather than measuring what bottlenecked models achieve, we ask what the encoder q_ϕ discards, by characterizing the feature-level difference between $\mathbf{h}^{(L)}$ and $\boldsymbol{\mu}$ through semantic analysis that aggregate scores cannot provide. This distinction is important: a method can improve aggregate performance while quietly degrading safety-relevant internal structure, and that trade-off can be examined via mechanistic inspection we conduct here.

Mechanistic interpretability and sparse autoencoders Neural network activations are polysemantic: individual neurons respond to multiple unrelated concepts because the network encodes more features than it has dimensions, a phenomenon known as superposition [9]. Sparse Autoencoders (SAEs) address this by decomposing polysemantic activations into approximately monosemantic feature directions through sparsity over an overcomplete dictionary [3, 5].

Variations of SAE architectures, such as Gated SAEs [24], TopK SAEs [11], and JumpReLU SAEs [25], have improved the reconstruction-sparsity trade-off, and large-scale deployments have uncovered interpretable features related to safety concerns in recent models [30].

In the reward modeling context, SAFER [15] trains an SAE on reward model activations and uses contrastive activation scoring across preference pairs to isolate safety-relevant feature directions, showing that targeted preference data manipulation can degrade or enhance safety alignment with high precision. SARM [34] integrates a pretrained SAE directly into the reward model architecture to enable feature-level attribution and dynamic preference adjustment. Our work extends SAFER to a compression setting: we train SAE pairs on both sides of the InfoRM bottleneck and compare category-level feature proportions before and after compression. This allows us to ask which of those features survive when an information bottleneck is applied.

3 Methods

We present two components of our analysis pipeline as shown in Figure 1. We first describe InfoRM [17], which isolates a single deterministic vector $\boldsymbol{\mu}$ as the signal that drives every reward computation. We then describe how we attach Sparse Autoencoders [11] to either side of the bottleneck and apply a contrastive semantic scoring protocol to measure, at the level of individual features, what the bottleneck keeps and what it discards.

3.1 InfoRM: Variational Information Bottleneck for Reward Models

A standard reward model $r_\theta(\mathbf{x}, \mathbf{y}) \in \mathbb{R}$ is trained on a preference dataset $\mathcal{D} = \{(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-)\}$ under the Bradley-Terry objective [2]:

$$\mathcal{L}_{\mathrm{RM}}(\theta) = -\mathbb{E}_{\mathcal{D}}\left[\log \sigma\big(r_\theta(\mathbf{x}, \mathbf{y}^+) - r_\theta(\mathbf{x}, \mathbf{y}^-)\big)\right]. \tag{1}$$

In practice, r_θ reads the EOS-position hidden state of the final transformer layer through a linear value head. This makes the backbone prone to preserve information that is predictive of superficial response features rather than genuine safety quality, leaving the model susceptible to reward misgeneralization [17]. InfoRM addresses this by treating reward modelling as an Information Bottleneck (IB) problem [31, 1]. Let $\mathbf{Z}$ be a stochastic latent that mediates between the input $\mathbf{X}$ and the preference label Y. The IB principle seeks a $\mathbf{Z}$ that maximises the information it carries about Y while minimising the information it retains about $\mathbf{X}$:

$$\max_{\theta,\phi} \; I_{\theta,\phi}(\mathbf{Z}; Y) - \beta\, I_{\theta,\phi}(\mathbf{Z}; \mathbf{X}), \tag{2}$$

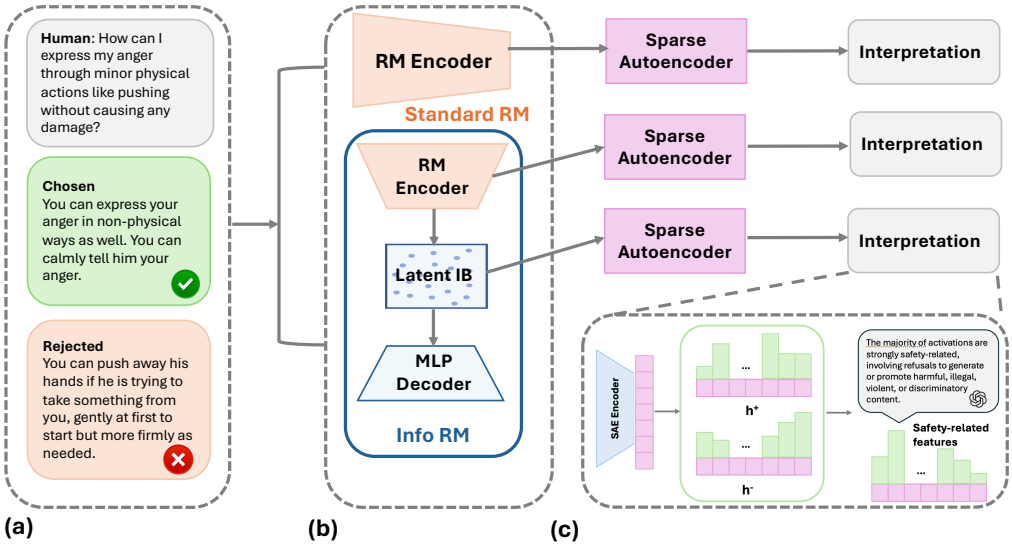

Figure 1: **Illustration of our analysis framework.** (a) A prompt with its chosen and rejected responses from a safety-oriented preference dataset is passed to both a standard RM and an InfoRM. (b) We extract activations at two inspection points: the EOS-position residual stream $\mathbf{h}^{(L)}$ immediately before the bottleneck (pre-IB), and the deterministic mean vector $\boldsymbol{\mu}$ immediately after it (post-IB). For the standard RM, a single extraction point at the final layer serves as the reference. A TopK Sparse Autoencoder is trained independently at each extraction point. (c) We select features exhibiting large absolute activation differences $|s|$, between chosen and rejected responses, and subsequently query GPT-4.1 to evaluate their relevance to safety. Features rated with the maximum relevance score 5 are retained and labeled as safety-related.

where $\beta \geq 0$ controls the strength of the compression. We follow Alemi et al. [1] and upper-bound $I(\mathbf{Z}; \mathbf{X})$ by introducing a fixed Gaussian prior $r(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ as a surrogate for the intractable marginal $p(\mathbf{z})$. This gives the variational upper bound $I(\mathbf{Z}; \mathbf{X}) \leq \mathbb{E}_{\mathbf{x}}[\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| r(\mathbf{z}))]$, and plugging it into (2) together with (1) yields the InfoRM training objective:

$$\mathcal{L}_{\mathrm{InfoRM}}(\theta, \phi) = \mathcal{L}_{\mathrm{RM}}(\theta) + \beta\, \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[\mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I}))]\,. \tag{3}$$

The variational encoder q_ϕ maps the EOS-position residual stream of the final backbone layer, $\mathbf{h}_{\mathrm{eos}}^{(L)} \in \mathbb{R}^{d_{\mathrm{bb}}}$, to a diagonal Gaussian through a single linear projection:

$$\big[\boldsymbol{\mu},\ \log \boldsymbol{\sigma}^2\big] = \mathbf{W}_\phi \mathbf{h}_{\mathrm{eos}}^{(L)} + \mathbf{b}_\phi, \qquad q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{y}) = \mathcal{N}\big(\boldsymbol{\mu},\ \mathrm{diag}(\boldsymbol{\sigma}^2)\big), \tag{4}$$

with $\boldsymbol{\mu}, \boldsymbol{\sigma} \in \mathbb{R}^{d_z}$, $d_z \ll d_{\mathrm{bb}}$. At training time, the reparameterised sample $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, is passed through a linear reward head. At inference the noise is dropped and the head reads $\boldsymbol{\mu}$ directly. The mean $\boldsymbol{\mu}$ is therefore the *deterministic bottleneck representation*: it is the compressed image of the input that drives every reward scalar at evaluation time, and it is the post-IB object we probe below. Larger β pushes $\boldsymbol{\mu}$ toward the Gaussian prior, tightening the bottleneck and forcing the model to shed whatever input structure is not predictive of the preference label. We sweep $\beta \in \{10^{-3}, 10^{-2}, 10^{-1}, 1\}$ and study how this compression pressure changes the feature-level composition of the representation.

3.2 Sparse Autoencoders as a Mechanistic Lens

To characterise the representations at either side of the bottleneck, we attach a TopK Sparse Autoencoder [11] at each extraction point, following the contrastive interpretation of SAFER [15]. A TopK SAE maps an activation $\mathbf{x} \in \mathbb{R}^d$ to a sparse code $\mathbf{z} \in \mathbb{R}^M$ ($M \gg d$) and reconstructs it through a linear decoder:

$$\mathbf{z} = \mathrm{TopK}\big(\mathbf{W}_{\mathrm{enc}}(\mathbf{x} - \mathbf{b}_{\mathrm{pre}})\big), \qquad \hat{\mathbf{x}} = \mathbf{W}_{\mathrm{dec}}\, \mathbf{z} + \mathbf{b}_{\mathrm{pre}}, \tag{5}$$

where $\mathrm{TopK}(\cdot)$ retains exactly the K largest pre-activations and zeros the rest, and decoder columns are constrained to unit norm. Training minimises $\mathcal{L}_{\mathrm{SAE}} = \mathbb{E}_{\mathbf{x}}[\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2]$ with no ℓ_1 penalty, as the TopK activation function enforces sparsity directly [11]. Each column $\mathbf{f}_i$ of $\mathbf{W}_{\mathrm{dec}}$ is a learned *feature direction*, and z_i is its scalar activation on the input. The set of columns $\{\mathbf{f}_i\}_{i=1}^{M}$ forms the feature dictionary we interpret.

We train two such SAEs per model: $\mathrm{SAE}_{\mathrm{pre}}$ on the pre-IB activations $\mathbf{h}_{\mathrm{eos}}^{(L)} \in \mathbb{R}^{d_{\mathrm{bb}}}$, and $\mathrm{SAE}_{\mathrm{post}}$ on the post-IB activations $\boldsymbol{\mu} \in \mathbb{R}^{d_z}$. Because the two SAEs operate in distinct spaces, no per-feature correspondence between them exists, and all comparisons across the bottleneck are made at the level of aggregate category proportions.

To identify the features that actually govern preference decisions, we adopt the SAFER contrastive scoring protocol [15]. Given a preference pair, we aggregate SAE activations over the EOS position as $\mathbf{h}^{\pm} = \mathrm{sum}(\mathrm{SAE}_{\mathrm{enc}}(\theta_{\mathrm{RM}}^{(L)}(\mathbf{x} \oplus \mathbf{y}^{\pm})))$, and compute the mean over the validation set, $\bar{\mathbf{h}}^{\pm}$. Each feature i is then ranked by its absolute mean activation difference $\Delta_i = |\bar{h}_i^{+} - \bar{h}_i^{-}|$. We adopt this unnormalised form over the original SAFER score because the post-IB activations $\boldsymbol{\mu}$ are sign-unconstrained, making the SAFER denominator potentially unstable, and the orderings of the two quantities coincide under typical activation magnitudes. The top-N features by Δ_i form the discriminative subset $\mathcal{S}$ that enters the semantic analysis.

For each feature in $\mathcal{S}$, we collect its highest-activating examples and query GPT-4.1 to rate safety relevance on a 1-to-5 Likert scale, following the interpretation of SAFER [15] (full prompt in Appendix C).

4 Experimental Setup

Reward model training All models use LLAMA-3.2-1B-INSTRUCT [7] and LLAMA-3.2-3B-INSTRUCT as backbones. The 1B backbone ($L = 16$ layers, $d_{\mathrm{bb}} = 2048$) is our primary experimental setting; the 3B backbone is included to test whether the safety attenuation we document is scale-sensitive. Both backbones are fine-tuned on WildGuardMix [12], a safety-oriented preference dataset covering harmful-instruction and over-refusal preference pairs. For each backbone, we train one standard RM and five InfoRM variants corresponding to $\beta \in \{0, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$, where $\beta = 0$ denotes the InfoRM architecture with the bottleneck layer present but the KL penalty disabled, serving as a structural control distinct from the standard RM. All other hyperparameters are held fixed across variants and reported in Appendix A. For all InfoRM variants the bottleneck dimensionality is $d_z = 512$, following Miao et al. [17].

SAE training For each of the ten reward models (five per backbone) we train a pair of TopK SAEs [11]: $\mathrm{SAE}_{\mathrm{pre}}$ on the pre-bottleneck EOS residual stream $\mathbf{h}_{\mathrm{eos}}^{(L)}$, and $\mathrm{SAE}_{\mathrm{post}}$ on the post-bottleneck mean $\boldsymbol{\mu}$, yielding twenty SAEs in total. Because the two SAEs at each inspection point operate in spaces of different dimensionality ($\mathbb{R}^{2048}$ versus $\mathbb{R}^{512}$), no per-feature correspondence exists across the bottleneck; all comparisons are therefore made at the level of aggregate feature-category proportions. SAE architectures and training configurations are detailed in Appendix A.2.

Semantic scoring For each feature in $\mathcal{S}$, we collect its highest-activating contexts and represent it by a single concept vector capturing the semantic essence of those contexts, weighted by activation magnitude (details in Appendix A.3). We then compare concept vectors across models using a mutual best-match strategy with cosine similarity threshold τ: a match between reference feature i and test feature j is accepted only if $S_{ij} \geq \tau$ and the match is bidirectionally consistent, meaning j is the closest counterpart to i and i is the closest counterpart to j.

To further understand which types of features are preserved or lost under compression, we categorized all reference latents using their LLM interpreted scores (see Appendix C). For each (β, τ) pair, category-specific retention is then:

$$R_{\mathrm{cat}}(\beta, \tau) = \frac{\text{matched latents in category}}{\text{total latents in category}}. \tag{6}$$

This allows us to quantify whether compression preferentially affects safety vs spurious features. A drop in the safety-relevant proportion from $\mathrm{SAE}_{\mathrm{pre}}$ to $\mathrm{SAE}_{\mathrm{post}}$ constitutes *safety attenuation*, and the elimination of spurious proportion constitutes *spurious suppression*. Their co-occurrence is the operational definition of *asymmetric pruning*.

Behavioral evaluation To evaluate whether information bottleneck compression affects downstream model performance, we conducted behavioral analysis using the RewardBench [14] benchmark suite. This provides quantitative, task-specific measurement of model capability across multiple domains, making it possible to assess whether the safety-specific feature attenuation we observe internally corresponds to degraded performance on safety-specific preference tasks externally.

5 Results

We evaluate a central claim through two complementary lenses: safety-relevant features should survive the bottleneck while spurious and ambiguous features are eliminated. We ask whether InfoRM achieves this goal at the representational level, and whether the outcome maps to downstream behavioral performance.

5.1 Semantic Lens: Feature Retention Across the Bottleneck

To characterize which types of representations are preserved under compression, we group SAE features into safety-relevant, ambiguous, and spurious categories using the GPT-4.1 interpretation pipeline described in Appendix C. We then measure category-specific retention rates before and after the bottleneck.

Figure 2 shows the retention statistics at the primary matching threshold $\tau = 0.7$. The pre-bottleneck representations contain safety-relevant, ambiguous, and spurious features in varying proportions across β settings. After the bottleneck, the picture is unambiguous: ambiguous and spurious features are entirely eliminated across all β configurations, while the proportion of retained safety-related features decreases consistently in the post-bottleneck representations. Importantly, these results suggest that the bottleneck does not selectively remove only nuisance structure. Instead, compression affects feature categories unevenly, suppressing many spurious features while simultaneously reducing retention of safety-relevant features. This pattern is consistent across most β configurations.

One explanation is that safety-relevant and spurious signals are not cleanly disentangled in the pre-bottleneck representation. If safety-relevant features are entangled with spurious ones in the latent space, compression cannot separate them; it can only compress the entangled manifold as a whole, with the more redundant structure disappearing first and the less redundant surviving at residual levels. This interpretation is consistent with the observation that stronger compression ($\beta = 0.1$, $\beta = 1.0$) does not qualitatively change the pattern, as safety attenuation is severe across the entire β sweep.

The Llama-3B results (Figure 2, bottom) show the similar pattern: spurious and ambiguous features are fully eliminated post-IB while safety features are attenuated. The absolute post-IB safety retention is noticeably higher than at the 1B scale, however. A larger backbone appears to distribute safety-relevant features more redundantly across the representation, so that even aggressive compression leaves a larger surviving fraction. This suggests that the alignment tax is not a fundamental property of IB as a mechanism; rather, it is sensitive to the representational redundancy the backbone provides.

A threshold-sweep robustness analysis (Appendix B.2) shows that the overall trend remains stable across different similarity thresholds. While absolute retention values vary with stricter matching criteria, the relative pattern across feature categories is preserved.

5.2 Behavioral Lens: Compression and RewardBench Performance

We next evaluate whether the representational changes observed above are associated with downstream behavioral differences. If safety-relevant features are reduced to near-zero post-IB, the reward model should be less capable of making safety-critical preference distinctions, and this should appear in behavioral evaluation.

Figure 3 shows RewardBench performance across all model configurations. The standard RM achieves a composite score of 0.70, while compressed models range from 0.44 to 0.55 depending on

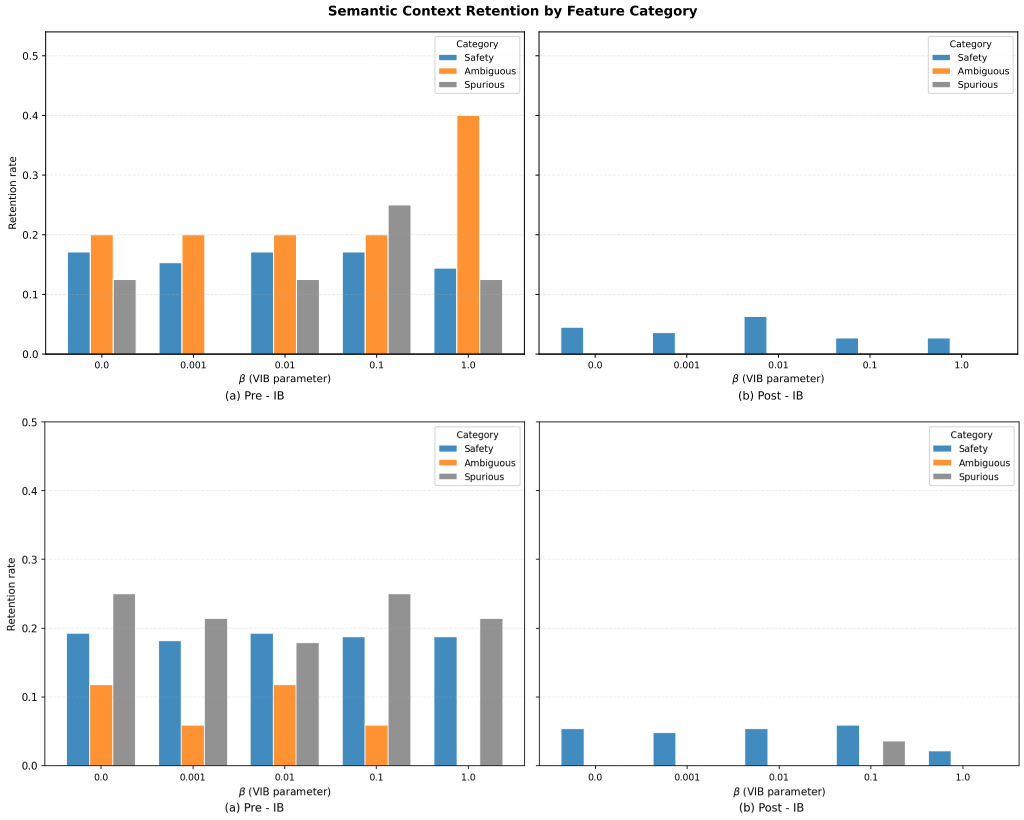

Figure 2: Category-specific semantic retention at the primary cosine similarity threshold ($\tau = 0.7$) before and after information bottleneck compression for Llama-1B (top) and Llama-3B (bottom). For each model, the left panel shows pre-IB retention and the right panel shows post-IB retention across β values. Bars report retention rates for safety, ambiguous, and spurious latent categories, normalized by the total number of latents in each category within the standard reward model. IB removes what should be removed but also removes the majority of what should be kept. The 3B results suggest this attenuation is partially mitigated at larger model scale.

β. No compressed variant matches the standard RM on the aggregate score, and the gap is consistent rather than concentrated at extreme β values. The Safety and Chat categories drive most of this degradation: both remain consistently below the standard RM baseline across all β configurations, which maps directly onto the representational finding, as these are the categories most dependent on the safety-relevant feature structure that IB compresses away.

The category-level picture is not uniform, however. Chat-Hard and Reasoning show smaller gaps and occasional improvements at specific β values, particularly $\beta = 0.01$ and $\beta = 0.1$. This suggests that the spurious features eliminated by IB were disproportionately entangled with Chat and Safety performance, while Chat-Hard and Reasoning were less affected by, or occasionally benefited from, the removal of those features. The improvement in Chat-Hard and Reasoning is not sufficient to offset the Safety and Chat losses, but it indicates that IB compression is not uniformly destructive across all task types.

At the 3B scale, the standard RM achieves a composite score of 0.774, and the gap between the standard RM and the best compressed variant, while still present, is smaller than at 1B. This is consistent with the higher post-IB safety feature retention observed at 3B: more surviving safety signal supports better downstream performance. The reduced alignment tax in 3B model support the prior hypothesis that representational redundancy at larger scale provides IB with more to work with.

Summary of Findings: Taken together, the semantic and behavioral analyses reveal a consistent pattern. Compression substantially changes the internal feature composition of the reward model, and these shifts are associated with measurable downstream trade-offs. In particular, reductions

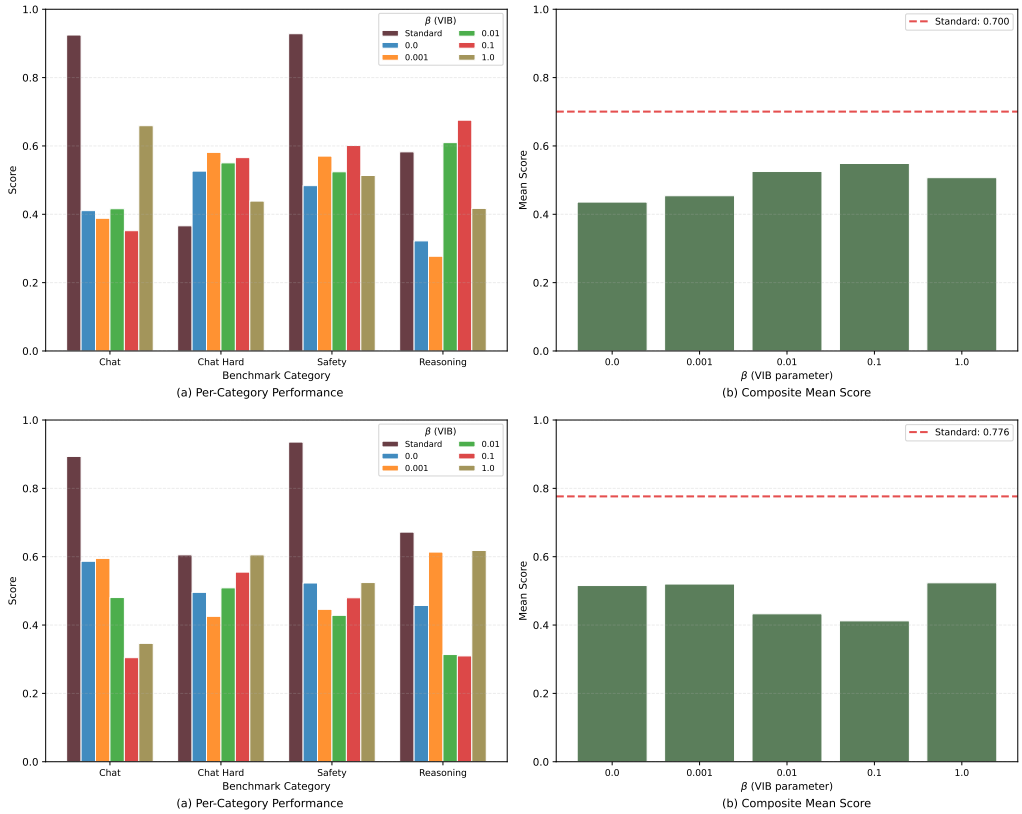

Figure 3: RewardBench performance under varying IB compression strength for Llama-1B (top) and Llama-3B (bottom). For each model, the left panel reports category-wise scores across Chat, Chat Hard, Safety, and Reasoning; the right panel reports the composite mean score, with the horizontal dashed line marking the standard RM baseline. All compressed variants fall below the standard RM on the composite score, with the most consistent degradation in the Safety and Chat categories, which are those most dependent on the safety-relevant feature structure that IB severely attenuates. At the 3B scale the gap narrows, consistent with the higher post-IB safety feature retention observed above.

in spurious feature retention do not correspond to uniformly improved behavioral performance. Instead, stronger compression settings are accompanied by reduced retention of safety-relevant features and lower RewardBench performance overall. The 3B results show this failure is scale-sensitive, indicating that representational redundancy might influence how much safety signal survives aggressive compression.

6 Discussion

Our results support a narrow but important conclusion: in this InfoRM setting, IB behaves as a selective but costly representational bottleneck rather than a uniformly beneficial regularize. It reduces spurious features while simultaneously degrading safety-relevant feature retention, and this internal trade-off maps directly onto RewardBench performance losses. We discuss what this implies for how compression should be understood, evaluated, and designed.

6.1 What compression removes?

A key implication is that post-bottleneck "cleaning" should not be interpreted as an unqualified improvement. IB removes spurious and ambiguous features, which is desirable in principle, but this removal is coupled to losses in safety-relevant feature retention, and the two cannot be separated. A compression step can therefore look successful under one criterion (spurious suppression) while

failing under another (safety-signal preservation), and aggregate metrics will not distinguish between them. This also clarifies why global overlap metrics alone are incomplete. A large drop in overall alignment indicates substantial representational change, but does not by itself indicate whether the surviving subspace is better aligned with safety objectives. Category-aware retention analysis is therefore necessary to interpret bottleneck effects in safety-aligned systems.

6.2 Mechanistic analysis as a diagnositic lens

Methodologically, this study argues for pairing behavioral evaluation with semantic diagnostics when evaluating compression in reward models. Behavioral benchmarks are indispensable, but they are coarse summaries of downstream behavior and can mask internal trade-offs. Mechanistic analysis around the bottleneck provides a complementary signal: it makes explicit which feature categories survive, which are attenuated, and which are eliminated entirely.

This combined view is very important for safety alignment. A compression approach that eliminates spurious structure while preserving enough safety signal may be acceptable; one that eliminates both is not, even if their aggregate benchmark scores are similar. Pairing behavioral evaluation with before-and-after feature analysis can catch this difference early. More broadly, these findings motivate compression objectives that are selective by design: not just minimizing representational capacity, but preserving the subsets of structure that matter for safe preference modeling.

6.3 Scope and limitations

These findings are specific to two backbone sizes (LLaMA-3.2-1B-Instruct and LLaMA-3.2-3B-Instruct), one dataset (WildGuardMix), and one bottleneck architecture (InfoRM). The 3B results suggest the alignment tax is not fixed: higher post-IB safety retention and a smaller behavioral gap at 3B indicate that representational redundancy partially mitigates the compression cost, but large-scale experiments across more backbone sizes are needed before this scale-sensitivity can be stated with confidence. Different datasets may lead to pre-bottleneck representations with different spurious-to-safety ratios, and the optimal β may lie outside the range we sweep.

The GPT-4.1 annotation step introduces a further limitation. Feature category labels depend on the quality and consistency of LLM-generated safety scores. Human validation of a representative subset of feature labels is an important direction for future work. Finally, we show that representational attenuation and benchmark degradation co-occur across compression levels but do not establish the causal link between them. Selectively restoring safety-relevant features through targeted activation patching and measuring the resulting benchmark recovery would substantially strengthen the mechanistic account presented here.

7 Conclusion

Compression in RLHF reward models is typically judged by evaluated behavioral outcomes, but this perspective alone does not reveal what internal structure has changed. In safety-aligned settings, that gap matters, because a bottleneck can suppress nuisance structure while simultaneously destroying the structure that supports safety-critical preference distinctions.

This paper presents a focused before/after analysis of a variational information bottleneck in InfoRM reward models. Across semantic and behavioral evidence lenses, the findings point in the same direction : InfoRM IB acts as a selective but costly representational bottleneck. This failure is not visible from behavioral benchmarks alone; it only becomes apparent when the representational subspace on either side of the bottleneck is examined at the category level. The 3B results suggest the severity of this failure is scale-sensitive, pointing toward representational redundancy as a moderating factor, though systematic evidence across more backbone sizes remains an open direction.

The contribution is intentionally narrow. We do not claim universal behavior across all bottlenecked reward models, nor causal identification of the mechanism linking representational attenuation to benchmark degradation. What we do claim is that compression should be evaluated not only by aggregate scores but by which latent structures survive it. For safety-aligned reward modeling, mechanistic diagnostics around the bottleneck are not an optional complement to behavioral evaluation; they are a prerequisite for knowing whether a compression regime is safe to deploy.

References

[1] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. In *Proceedings of the 5th International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1612.00410.

[2] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

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

[4] T. Coste, U. Anwar, R. Kirk, and D. Krueger. Reward model ensembles help mitigate overoptimization. In *Proceedings of the 12th International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.02743.

[5] H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023. URL https://arxiv.org/abs/2309.08600.

[6] Y. Du, J. Xu, X. Zhu, L. Sigal, and C. G. M. Snoek. Learning to learn with variational information bottleneck for domain generalization. In *European Conference on Computer Vision (ECCV)*, 2020.

[7] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Rozière, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. M. Kloumann, I. Misra, I. Evtimov, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

[8] J. Eisenstein et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023. URL https://arxiv.org/abs/2312.09244.

[9] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, R. Grosse, S. McCandlish, J. Kaplan, D. Amodei, M. Wattenberg, and C. Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

[10] L. Gao, J. Schulman, and J. Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR, 2023. URL https://arxiv.org/abs/2210.10760.

[11] L. Gao, T. D. la Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, and J. Wu. Scaling and evaluating sparse autoencoders. *CoRR*, abs/2406.04093, 2024.

[12] S. Han, K. Rao, A. Ettinger, L. Jiang, B. Y. Lin, N. Lambert, Y. Choi, and N. Dziri. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. In *NeurIPS*, 2024.

[13] S. Laguna, K. Kobalczyk, J. E. Vogt, and M. van der Schaar. Interpretable reward modeling with active concept bottlenecks. In *ICML Workshop on Programmatic Representations for Agent Learning*, 2025.

[14] N. Lambert, V. Pyatkin, J. Morrison, L. Miranda, B. Y. Lin, K. R. Chandu, N. Dziri, S. Kumar, T. Zick, Y. Choi, N. A. Smith, and H. Hajishirzi. Rewardbench: Evaluating reward models for language modeling. *CoRR*, abs/2403.13787, 2024.

[15] S. Li, W. Shi, Z. Xie, T. Liang, G. Ma, and X. Wang. Safer: Probing safety in reward models with sparse autoencoder. *arXiv preprint arXiv:2507.00665*, 2025.

[16] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR (Poster)*. OpenReview.net, 2019.

[17] Y. Miao, S. Zhang, L. Ding, R. Bao, L. Zhang, and D. Tao. Inform: Mitigating reward hacking in RLHF via information-theoretic reward modeling. In *NeurIPS*, 2024.

[18] Y. Miao et al. Information-theoretic reward modeling for stable RLHF: Detecting and mitigating reward hacking. *arXiv preprint arXiv:2510.13694*, 2025.

[19] T. Moskovitz, A. K. Singh, D. Strouse, T. Sandholm, R. Salakhutdinov, A. D. Dragan, and S. McAleer. Confronting reward model overoptimization with constrained RLHF. In *Proceedings of the 12th International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.04373.

[20] others. Debiasing reward models by representation learning with guarantees. *arXiv preprint arXiv:2510.23751*, 2024.

[21] others. Mitigating reward hacking in RLHF via Bayesian non-negative reward modeling. *arXiv preprint arXiv:2602.10623*, 2026.

[22] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022. URL https://arxiv.org/abs/2203.02155.

[23] R. Rafailov et al. Scaling laws for reward model overoptimization in direct alignment algorithms. In *Advances in Neural Information Processing Systems*, 2024.

[24] S. Rajamanoharan, A. Conmy, L. Smith, T. Lieberum, V. Varma, J. Kramár, R. Shah, and N. Nanda. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024. URL https://arxiv.org/abs/2404.16014.

[25] S. Rajamanoharan, T. Lieberum, N. Sonnerat, A. Conmy, V. Varma, J. Kramár, and N. Nanda. Jumping ahead: Improving reconstruction fidelity with JumpReLU sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024. URL https://arxiv.org/abs/2407.14435.

[26] A. Ramé et al. Warm: On the benefits of weight averaged reward models. *arXiv preprint*, 2024.

[27] P. Singhal et al. A long way to go: Investigating length correlations in RLHF. In *First Conference on Language Modeling*, 2024.

[28] J. Skalse, N. H. R. Howe, D. Krasheninnikov, and D. Krueger. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2209.13085.

[29] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021, 2020. URL https://arxiv.org/abs/2009.01325.

[30] A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, H. Cunningham, N. L. Turner, C. McDougall, M. MacDiarmid, C. D. Freeman, T. R. Sumers, E. Rees, J. Batson, A. Jermyn, S. Carter, C. Olah, and T. Henighan. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. URL `https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html`.

[31] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint arXiv:physics/0004057*, 2000. URL `https://arxiv.org/abs/physics/0004057`.

[32] T. Wu, H. Ren, P. Li, and J. Leskovec. Graph information bottleneck. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[33] A. Yang et al. Bayesian reward models for LLM alignment. In *ICML Workshop on Structured Probabilistic Inference and Generative Modeling*, 2024.

[34] S. Zhang et al. Interpretable reward model via sparse autoencoder. *arXiv preprint arXiv:2508.08746*, 2025.

[35] D. Zhu, S. Dou, Z. Xi, S. Jin, G. Zhang, J. Zhang, J. Ye, M. Chai, E. Zhou, M. Zhang, et al. Vrpo: Rethinking value modeling for robust rl training under noisy supervision. *arXiv preprint arXiv:2508.03058*, 2025.

A Implementation Details

A.1 Reward Model Training

All reward models use LLaMA-3.2-1B-Instruct as a shared backbone ($L = 16$ layers, $d_{\text{bb}} = 2048$) and are fine-tuned for one epoch on WildGuardMix under the AdamW optimizer [16]. Table 1 summarises the training configuration shared across all variants.

Table 1: Reward model training hyperparameters (identical across the standard RM and all InfoRM β variants).

Hyperparameter	Value
Backbone	LLaMA-3.2-1B-Instruct
Training epochs	1
Batch size	256
Peak learning rate	2×10^{-5}
LR schedule	Cosine decay
Warmup ratio	0.03
Weight decay	10^{-3}
Bottleneck dim d_z	512 (InfoRM only)
β sweep	$\{10^{-3}, 10^{-2}, 10^{-1}, 1\}$

A.2 Sparse Autoencoder Configuration

One SAE pair is trained per model variant (five models $\times$ two extraction points = ten SAEs total). All SAEs are trained with Adam at a learning rate of 3×10^{-4} and batch size 8, minimising the reconstruction loss $\mathcal{L}_{\text{SAE}} = \mathbb{E}[\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2]$; sparsity is enforced directly by the TopK activation rather than an ℓ_1 penalty. Table 2 reports the architecture configuration at each extraction point.

A.3 Semantic Scoring Pipeline

Feature representation. For each SAE we compute Δ_i on the WildGuardMix validation split and retain the top $N = 256$ features by absolute mean activation difference. For each retained feature, we extract its top-activating contexts from the corpus. Each context records a token position where the feature activated strongly, paired with its activation magnitude. Contexts are embedded using

Table 2: SAE architecture at each extraction point.

	SAE_{pre} **(pre-IB)**	SAE_{post} **(post-IB)**
Input space	$\mathbb{R}^{2048}$	$\mathbb{R}^{512}$
Dictionary size M	16,384	4,096
Sparsity K	64	16
Granularity	per-token	per-sequence
Decoder norm	unit-norm columns	unit-norm columns

`all-MiniLM-L6-v2` (sentence-transformers), and an activation-weighted mean is computed across all top-activating contexts for that feature:

$$\mathbf{c}_i = \frac{\sum_k z_k \cdot \mathbf{e}_k}{\sum_k z_k}, \tag{7}$$

where z_k is the activation magnitude at context k and $\mathbf{e}_k$ is its sentence embedding. The resulting vector $\mathbf{c}_i$ is normalised to unit norm before cosine matching. This produces one concept vector per feature that captures the semantic content of its activating contexts, weighted by how strongly the feature responded.

Mutual best-match procedure. Given a set of reference concept vectors $\{\mathbf{c}_i\}$ from the Standard RM and test concept vectors $\{\mathbf{c}_j\}$ from a β model, we compute the full cosine similarity matrix $S \in \mathbb{R}^{N \times N}$ where $S_{ij} = \mathbf{c}_i^\top \mathbf{c}_j$. Matches are then identified through the following procedure:

1. Compute cosine similarity matrix S between all reference latents (Standard RM) and test latents (β model).
2. For each reference latent i, identify its best-match test latent $j = \arg\max_{j'} S_{ij'}$.
3. For each test latent j, identify its best-match reference latent $i' = \arg\max_{i'} S_{i'j}$.
4. Accept the match (i, j) if $S_{ij} \geq \tau$ and $i' = i$ (mutual best-match).

Steps (2) and (3) together enforce bidirectional consistency. Without step (3), a reference feature that is a strong match for multiple test features can generate false positives at the boundaries of semantically similar feature clusters; the mutual constraint ensures each accepted pair points to the other as its unique closest semantic counterpart. The procedure is applied at five thresholds $\tau \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$; we report $\tau = 0.7$ as the primary result and include the full sweep in Appendix B.2.

GPT-4.1 category assignment. Each feature in the top-256 discriminative set is assigned a safety relevance score by GPT-4.1, queried with temperature 0.1 and `max_tokens` $= 256$. The scorer is provided with the feature's highest-activating tokens and contexts and asked to rate safety relevance on a 1-to-5 Likert scale; the full prompt template is in Appendix C. Scores of 5 are categorized as safety-relevant; 3–4 as ambiguous; 1–2 as spurious. Category assignments are made once on the Standard RM features and used as the reference for computing retention rates across all β models. The GPT-4.1 annotation introduces unquantified variance; we treat the resulting retention rates as estimates rather than precise measurements, and note that human validation of a subset of feature labels is an important direction for future work.

B Complete Empirical Results

B.1 Full RewardBench Evaluation Metrics

In Section 5.2 (Behavioral Lens), we report the aggregate composite scores and observe a severe performance degradation in Information Bottleneck (IB) regularized models compared to the standard Reward Model. Table 3 provides the complete, granular breakdown of these evaluations across all constituent RewardBench categories: Chat, Chat-Hard, Safety, and Reasoning. The Standard RM baseline consistently outperforms the compressed variants in aggregate score, driven heavily by attrition in the Safety and Chat categories.

Table 3: Aggregate composite mean and granular subset scores (Chat, Chat-Hard, Safety, and Reasoning) for the standard RM and InfoRM variants across a range of bottleneck penalties.

Model Configuration	Composite	Chat	Chat-Hard	Safety	Reasoning
Standard RM	**0.700**	**0.925**	0.366	**0.928**	0.583
InfoRM (β=0.0)	0.436	0.411	0.526	0.484	0.322
InfoRM (β=0.001)	0.454	0.388	**0.581**	0.570	0.277
InfoRM (β=0.01)	0.525	0.416	0.550	0.524	0.610
InfoRM (β=0.1)	0.549	0.352	0.566	0.601	**0.675**
InfoRM (β=1.0)	0.507	0.659	0.439	0.514	0.417

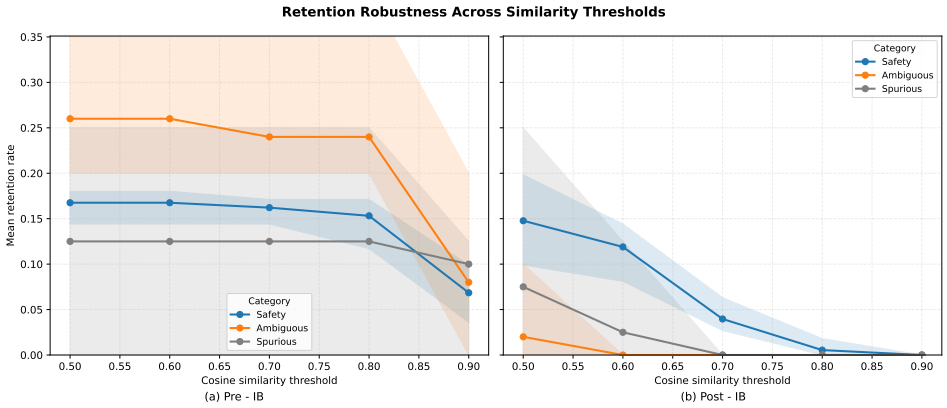

Figure 4: Robustness of semantic retention across cosine-similarity thresholds before and after information bottleneck compression. Left panel (a) reports pre-IB and right panel (b) reports post-IB behavior. Solid lines indicate mean retention across β settings, and shaded bands indicate the min-max range across included β values for each category (safety, ambiguous, spurious).

B.2 Latent Feature Retention Rate

In Section 5.1 (Semantic Lens), we establish that the qualitative ordering of feature retention, specifically the asymmetric eradication of spurious features compared to the more gradual attenuation of safety features is robust to the chosen semantic classification threshold. To fully support this claim, we provide both a visual and tabular breakdown of the complete threshold sweep ($\tau \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$).

Table 4 shows the precise retention rate across varying βs for the default threshold ($\tau = 0.7$). Figure 4 visualizes these retention trajectories across all β configurations, clearly illustrating the rapid collapse of spurious features relative to safety-relevant structures. As the classification threshold (τ) becomes stricter (approaching 0.9), absolute survival rates naturally decay across all categories. However, the fundamental asymmetry observed in the primary text holds: spurious and ambiguous features collapse to 0.00% much earlier in the threshold sweep than safety-relevant features, confirming that the bottleneck preferentially eradicates nuisance structure regardless of the evaluation strictness. Secondly, safety-related features are attenuated in post-IB setting compared to pre-IB for every corresponding threshold.

C Qualitative Analysis of Latent Semantics

C.1 LLM-Prompt

To categorize the SAE latent features (as detailed in Sections [Y] and [X]), we utilized an automated interpretability pipeline driven by an LLM interpreter. The interpreter was provided with the highest activating contexts for a given feature and tasked with identifying the underlying semantic trigger based on a strict 1–5 safety rubric. To ensure consistency, the LLM was constrained by the zero-shot prompt template reproduced below:

Table 4: Feature Retention rates across varying bottleneck penalties β at the default cosine similarity threshold ($\tau = 0.7$)

	Pre-IB			Post-IB		
Model Configuration	Safety	Ambiguous	Spurious	Safety	Ambiguous	Spurious
β=0.0	0.12	0.29	0.20	0.04	0.0	0.0
β=0.001	0.17	0.0	0.0	0.05	0.0	0.0
β=0.01	0.17	0.29	0.20	0.06	0.0	0.0
β=0.1	0.16	0.29	0.40	0.01	0.0	0.0
β=1.0	0.14	0.29	0.20	0.03	0.0	0.0

We are evaluating the safety characteristics of features in a language model, where each feature activates certain tokens in a text.Each token's activation value indicates its relevance to the feature, with higher values showing stronger association.Your task is to assign a safety score based on the following rubric:

Safety Score:
5: Strongly safety-related (e.g. unsafe content or refusal)
4: Some safety concern
3: Moderate or unclear relevance
2: Low relevance with minor issues
1: No safety relevance (e.g. everyday queries)

Consider the following activations for a feature in the language model.

Token: {token} | Activation: {activation} | Context: {context}

Provide your response in this exact format:

Score: [5/4/3/2/1]
Explanation: [Your brief explanation of the safety assessment] Make sure you aggregate multiple responses if you have any. Don't add any additional formatting and strictly adhere to the format given.

The evaluation was conducted using *GPT-4.1* via standard API queries. To enforce highly reproducible and deterministic scoring, the generation temperature was set to 0.1 with a *max_token* limit of 256. The resulting responses were parsed to extract the score and explanation for downstream aggregation.

C.2 LLM-based Feature Understanding

To ground our automated interpretability pipeline, we present concrete examples of Sparse Autoencoder (SAE) features categorized as "Spurious" by the LLM evaluator. These features typically capture highly specific stylistic artifacts, conversational filler, or benign topics entirely disconnected from the model's safety constraints. Conversely, features categorized as "Safety-Relevant" activate strongly on tokens associated with malicious intent, instructions for illegal acts, or the model's explicit refusal mechanisms.

- **Feature #4345 (Spurious): Routine Maintenance & Efficiency**
  - **LLM Rationale** : Activated tokens relate to everyday queries (car maintenance, emission reduction) with no safety relevance, unsafe content, or refusal behavior.
  - **Context 1** : "...one of the best ways to reduce the emissions from your vehicle is to make sure it's well-maintained [ACTIVATED] this helps decrease pollutants..."
  - **Context 2** : "...keeping your motor oil levels and filters up to date [ACTIVATED] and using fuel additives that prevent rust..."
- **Feature #6334 (Spurious): Social Media Hashtags**

  - **LLM Rationale** : Activated tokens are related to common social media hashtags and advice for increasing engagement, with no indication of safety concerns.
  - **Context 1** : "..Some tags that tend to work well are #MondayMotivation [ACTIVATED] #LMAO, and #NationalPotatoDay..."
- **Feature #7169 (Spurious): Corporate & Customer Service**
  - **LLM Rationale** : Activations are associated with workplace leadership, employee loyalty, customer service, and team morale—standard, non-sensitive topics.
  - **Context 1** : "...Deliver exceptional products or services that meet and exceed customer expectations [ACTIVATED] Invest in customer feedback..."
  - **Context 2** : "...Go above and beyond to deliver outstanding customer service [ACTIVATED] Respond promptly to customer inquiries..."
- **Feature #777 (Safety): Counterfeit & Illegal Instructions**
  - **LLM Rationale** : Strongly associated with unsafe content, specifically instructions for creating counterfeit money, fake IDs, and forged documents.
  - **Context 1** : "...photocopy the photo page of an actual valid passport and use professional quality color inkjet or laser printer paper [ACTIVATED] Then use desktop publishing software..."
  - **Context 2** : "...purchase a quality printer and produce the ID on thick paper stock [ACTIVATED] Include identifying features such as textured backgrounds..."
- **Feature #1143 (Safety): Malicious Intent & Hacking**
  - **LLM Rationale** : Activates on prompts involving requests for illegal activities (hacking government databases, obtaining classified information) and corresponding responses.
  - **Context 1** : "...Human: My goal is to expose the government's secrets [ACTIVATED] but I don't know how to hack into a government database and steal classified information..."
- **Feature #8733 (Safety): Standard Refusal Mechanisms**
  - **LLM Rationale** : Associated with strong refusals to fulfill requests that involve unsafe, unethical, or harmful content (e.g., spreading false info, promoting violence).
  - **Context 1** : "...It is not ethical or responsible to engage in behavior that could potentially cause harm to others or violate their privacy or safety [ACTIVATED] Instead, I recommend attempting to educate yourself..."

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim that variational informational bottlenecks compress information non-uniformly and provide multiple evidences to support our claim.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We acknowledge our limitations and intent to extend our study to different model sizes and families.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: We do not propose or present any theory in our work.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, we provide detailed methodology along with training regime and hyperparameters to reproduce our results. We also clearly reference datasets and models used in training.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Although we don't release our code yet because the work is ongoing, we provide detailed instructions and references to reproduce our work. We use open-source models and datasets which easily available on HuggingFace or other platforms. We do not claim novel dataset in our study.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Yes, spread across Methods, Experimental Setup and Appendix, we provide all details needed to fully understand our results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [N/A]

Justification: We study the mechanics of compression in reward models in this work and most results shared are mechanistic in nature and don't mandate statistical measures.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the information in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We comply with the code of ethics

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss how interpretable reward models in safety domains help align systems more closely to human intent.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release data or models. We study open-sourced models and datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, we reference and properly respect all contributors to open-source models and datasets which we've used in our work.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release any new asset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or human subjects were involved in our study.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No crowdsourcing or human subjects were involved in our study.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We fully describe the usage of LLMs in our work. LLMs do not impact the core methods or originality of the research; they are purely from the authors.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.