Topology-Enhanced Alignment for Large Language Models: Trajectory Topology Loss and Topological Preference Optimization

Anonymous ACL submission

Abstract

Alignment of large language models (LLMs) typically relies on supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF), or more recently direct preference optimization (DPO). However, existing objectives largely ignore the global geometry and topology of the representation space: they operate on local token-level likelihoods or scalar preference scores, and do not explicitly constrain how hidden states move from a user prompt to an answer.

We view generation as tracing a *semantic trajectory* in hidden space, and propose a topology-enhanced alignment framework that regularizes these trajectories using 0-dimensional persistent homology. First, at the SFT stage, we introduce a **Trajectory Topology Loss** (TTL). For each batch, we treat mean-pooled embeddings of prompts and gold answers as a mixed point cloud, run a Union-Find-based 0D persistent homology algorithm, and extract "prompt–answer bridge" edges that connect previously disconnected components. TTL encourages the model's actual update direction from prompt to answer to align with these topologically derived bridges, rather than with arbitrary or per-example directions.

Second, at the RLHF/DPO stage, we propose **Topological Preference Optimization** (TPO). TPO constructs topic-specific semantic preference vectors from an offline pipeline and aligns the semantic improvement direction between rejected and chosen responses with these vectors in an intermediate hidden layer. We further introduce an exponential-moving-average-based dynamic weighting scheme to balance DPO and TPO losses, and also explore a fully topological variant that applies persistent homology on the chosen/rejected embedding cloud.

We instantiate our methods on Qwen2.5-7B-Instruct and evaluate on UltraChat and Anthropic HH-RLHF. Across both SFT and DPO training, topology-enhanced objectives consistently outperform strong non-topological baselines (including per-example, nearest-neighbor, and random direction regularizers) on automatic preference metrics and LLM-judge evaluations, while maintaining or slightly improving toxicity. These results suggest that incorporating persistent homology and trajectory geometry is a promising and practical direction for more controllable LLM alignment.

1 Introduction

Large language models (LLMs) have achieved impressive performance on a wide range of tasks, including open-domain dialogue, code generation, and complex reasoning (Brown et al., 2020; Vaswani et al., 2017). Despite this progress, aligning LLM behaviors with human values and preferences remains a central challenge. The dominant paradigm combines *supervised fine-tuning* (SFT) on instruction-following data with *reinforcement learning from human feedback* (RLHF) (Ouyang et al., 2022; Bai et al., 2022; Christiano et al., 2017; Stiennon et al., 2020) or more recent direct preference optimization (DPO) approaches (Rafailov et al., 2024).

Although SFT and RLHF/DPO have proven highly effective in practice, they share a key limitation: they largely ignore the *geometry* and *topology* of the internal representation space. Standard objectives focus on local signals—token-level likelihoods in SFT, or scalar preference scores in RLHF—and do not directly supervise how the model's hidden states move from a user prompt to a final answer.

However, an LLM's response generation process can naturally be viewed as tracing a *trajectory* through its hidden space: starting from a representation of the prompt, the model iteratively updates its internal state as it produces each token of the answer. Different answers (e.g., helpful vs. unhelpful, safe vs. unsafe) correspond to different trajecto-

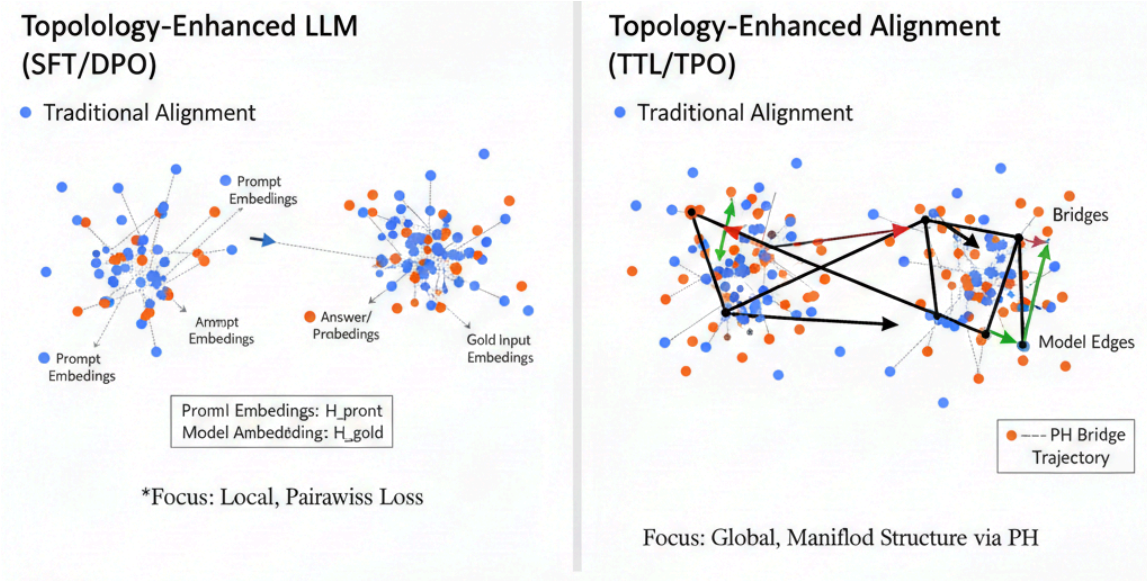

Figure 1: Conceptual comparison between traditional alignment and our topology-enhanced alignment in hidden space. **Left:** Traditional alignment optimizes local, pairwise losses on prompt and answer embeddings without explicitly modeling global structure. **Right:** Our topology-enhanced view treats prompts and answers as a joint point cloud, extracts cross-manifold bridges via 0D persistent homology, and regularizes model trajectories to follow these bridges.

ries. If we could shape these trajectories to follow semantically meaningful directions—for example, from a prompt state towards a manifold of high-quality answers—we might obtain more robust and interpretable alignment behavior.

In parallel, the field of Topological Data Analysis (TDA) studies the shape of data manifolds using tools such as persistent homology (Edelsbrunner and Harer, 2010; Carlsson, 2009; Ghrist, 2008). Given a point cloud and a distance metric, persistent homology tracks how connected components and higher-dimensional features appear and merge across scales. Even in the simplest case of 0-dimensional homology, the resulting "death edges" reveal how different clusters of points connect, providing a multi-scale skeleton of the data.

This paper brings these two perspectives together. We ask:

> *Can we use topological information about hidden representations to regularize LLM alignment, by explicitly constraining semantic trajectories in hidden space?*

We answer this question affirmatively by proposing a unified, topology-enhanced alignment framework with two components:

- At the **SFT stage**, we introduce a **Trajectory Topology Loss** (TTL). For each batch, we treat the mean-pooled embeddings of prompts and gold answers as a mixed point cloud. Using a 0D persistent homology algorithm implemented via a Union-Find structure (Tarjan, 1975), we identify "prompt–answer bridges": edges that connect previously separate connected components. We view these bridges as topologically informed trajectories from prompts towards the gold answer manifold, and regularize the model so that its actual update direction from prompt to model answer aligns with these bridges.
- At the **RLHF/DPO stage**, we propose **Topological Preference Optimization** (TPO), which aligns the semantic improvement direction between rejected and chosen responses with topic-specific preference vectors constructed by an offline pipeline. We further introduce a dynamic weighting scheme based on an exponential moving average (EMA) to balance DPO and TPO losses, and explore a fully topological TPO variant using persistent homology on the chosen/rejected cloud.

We instantiate our methods on Qwen/Qwen2.5-7B-Instruct and evaluate on UltraChat (**?**) for SFT and Anthropic HH-RLHF (Bai et al., 2022) for

DPO. Our empirical findings are:

- Topology-enhanced SFT with TTL yields consistent improvements in reward-model scores and LLM-judge helpfulness ratings compared to a strong SFT baseline, with negligible increase in toxicity.
- TPO on top of DPO provides higher preference win-rates and better helpfulness/harmlessness trade-offs than plain DPO, across different hidden layers and clustering granularities.
- Ablations confirm that (i) persistent-homology-derived bridges outperform random, per-example, and nearest-neighbor prompt–answer pairings, and (ii) topic-aware preference vectors and dynamic weighting are both important for TPO's effectiveness.

Collectively, our results indicate that even simple 0D topological information can provide useful structure for regularizing hidden-space trajectories during alignment.

Contributions. This paper makes the following contributions:

- We propose a *trajectory-centric* view of LLM alignment, where the update from a prompt representation to an answer representation is treated as an explicit semantic trajectory in hidden space, rather than only being supervised via token-level likelihoods or scalar rewards.
- We introduce **Trajectory Topology Loss** (TTL) for SFT, which uses 0D persistent homology on a mixed prompt/gold-answer point cloud to extract a sparse set of topological "bridges". TTL regularizes the model so that its prompt-to-answer trajectories align with these bridges, and we show that this outperforms non-topological baselines such as per-example, random, and kNN-based direction regularization.
- We propose **Topological Preference Optimization** (TPO) for the DPO stage, which aligns hidden-space improvement directions between rejected and chosen responses with topic-aware semantic preference vectors derived from an offline clustering and templating pipeline. We further introduce an EMA-based dynamic weighting scheme and a fully topological TPO variant on the chosen/rejected embedding cloud.

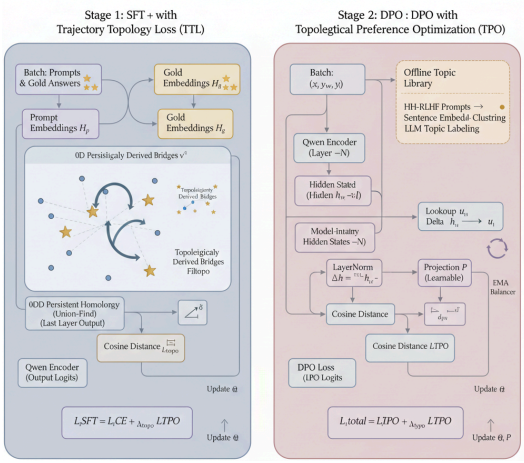

Figure 2: Overview of our topology-enhanced alignment framework. The left part shows SFT with Trajectory Topology Loss (TTL), which adds a cosine loss on topology-derived bridges between prompt and gold-answer embeddings. The right part shows DPO with Topological Preference Optimization (TPO), which aligns rejected-to-chosen hidden-state differences with topic-specific preference vectors.

- We provide an empirical study on Qwen2.5-7B-Instruct with UltraChat and HH-RLHF, demonstrating consistent gains over strong SFT and DPO baselines on reward-model scores, preference win-rates, and helpfulness/harmlessness metrics, with modest training overhead.

Each death edge corresponds to the "death" of a connected component when it merges into an older one. Collectively, these edges form a tree structure that captures how initially separate regions of the point cloud become connected as we increase the distance threshold (Carlsson, 2009).

In our setting, we exploit this structure to identify *bridges* between points of different semantic categories (e.g., prompts vs. answers, rejected vs. chosen). These bridges provide directions in representation space that are informed by the global geometry and topology of the batch, rather than by arbitrary or local choices.

2 Related Work

Alignment of large language models. Alignment methods such as RLHF (Ouyang et al., 2022; Bai et al., 2022; Christiano et al., 2017; Stiennon et al., 2020) and DPO (Rafailov et al., 2024) have become standard for controlling LLM behaviors. Subsequent work explores variations in reward modeling, off-policy optimization, and preference

data curation (Rafailov et al., 2024; Bai et al., 2022). Our work is orthogonal: we focus on incorporating geometric and topological constraints into existing pipelines. The foundation of ranking preferences in these models often traces back to statistical models like Bradley-Terry (Bradley and Terry, 1952) or Plackett-Luce (Plackett, 1975).

Representation geometry in deep learning. A growing body of work studies the geometry of neural representations, including manifold structure, anisotropy (Ethayarajh, 2019; Ortiz-Jiménez et al., 2020; Li et al., 2020), and linear probes for concepts (Bau et al., 2017). Some methods exploit representation geometry for curriculum learning or out-of-distribution detection (Hendrycks and Gimpel, 2017; Lee et al., 2018). Other works analyze the expressivity and disentanglement of representations (Raghu et al., 2017; Achille and Soatto, 2018). We add to this line by treating hidden-space trajectories themselves as objects to be regularized, informed by topological structure.

Topological data analysis in neural networks. TDA has been used to analyze the shape of feature spaces and decision boundaries in deep networks (Rieck et al., 2019; Ballester et al., 2024), and to design regularizers for robustness (Adams et al., 2015; Bubenik, 2015; Hofer et al., 2019). The theoretical underpinnings rely on persistent homology and barcodes (Ghrist, 2008; Edelsbrunner and Harer, 2010). However, applications to large-scale sequence models and LLM alignment remain limited. To our knowledge, we are the first to use 0D persistent homology explicitly as a training signal for LLM alignment at both SFT and RLHF stages.

3 Method

We propose a topology-enhanced alignment framework that regularizes hidden-space trajectories at both the SFT and DPO stages (Figure 2). At SFT time, Trajectory Topology Loss (TTL) shapes how hidden states move from prompts to answers. At DPO time, Topological Preference Optimization (TPO) shapes how hidden states move from rejected to chosen responses along topic-specific preference directions.

3.1 Notation

Let f_θ denote an LLM with parameters θ. For an input sequence $x = (x_1, \ldots, x_n)$ with attention mask $m \in \{0,1\}^n$, layer l produces hidden states $H^{(l)} \in \mathbb{R}^{n \times d}$ and we mean-pool non-padding tokens:

$$h^{(l)}(x) = \frac{\sum_i m_i H_i^{(l)}}{\sum_i m_i}. \quad (1)$$

When the layer is clear from context we write $h(x)$ for brevity. We use x^{prompt} for the dialogue history up to the last user turn, y^{gold} for the ground-truth assistant answer, and y^{model} for the model answer (either gold tokens under teacher forcing or sampled tokens). For DPO, y^{chosen} and y^{rejected} denote the preferred and dispreferred responses.

3.2 Trajectory Topology Loss for SFT

TTL encourages the model's prompt-to-answer trajectory in hidden space to align with topology-derived directions from prompt regions to the gold-answer manifold.

Point cloud construction. For each SFT example we split the sequence into prompt and answer tokens using the chat template and compute three representations:

- $h^{\text{prompt}} \in \mathbb{R}^d$: mean-pooled last-layer hidden state over prompt tokens;
- $h^{\text{model}} \in \mathbb{R}^d$: mean-pooled last-layer hidden state over answer tokens (teacher forcing);
- $h^{\text{gold}} \in \mathbb{R}^d$: mean-pooled *input embeddings* of gold-answer tokens (akin to the vector space concepts in (Mikolov et al., 2013)).

Over a batch of size B we form

$$H^{\text{prompt}} = [h_1^{\text{prompt}}, \ldots, h_B^{\text{prompt}}]^\top \in \mathbb{R}^{B \times d}, \quad (2)$$

$$H^{\text{gold}} = [h_1^{\text{gold}}, \ldots, h_B^{\text{gold}}]^\top \in \mathbb{R}^{B \times d}, \quad (3)$$

and a mixed point cloud

$$Z = \begin{bmatrix} H^{\text{prompt}} \\ H^{\text{gold}} \end{bmatrix} \in \mathbb{R}^{2B \times d}, \quad (4)$$

with labels $l_i = 0$ for prompts $(1:B)$ and $l_i = 1$ for gold answers $(B{+}1{:}2B)$.

Topological bridges via 0D persistent homology. We compute the pairwise distance matrix $D_{ij} = \|Z_i - Z_j\|_2$ and run a standard 0D persistent-homology algorithm based on Union–Find (Tarjan, 1975), which processes edges in non-decreasing order of D_{ij} and records the edges that merge previously disconnected components (death edges).[1]

[1] Algorithmic details and pseudocode are given in Appendix D.

Let $\mathcal{P}$ be the set of death edges. We keep those that connect a prompt and a gold answer:

$$\mathcal{B} = \{(p, a) \in \mathcal{P} \mid l_p \neq l_a\}. \tag{5}$$

Each such *prompt–answer bridge* is oriented from prompt to answer (swapping indices if needed) and induces a topological direction

$$v^{\text{topo}}_{(p,a)} = Z_a - Z_p. \tag{6}$$

Compared to using each prompt's own gold answer or nearest gold neighbor, these bridges arise from a global minimum-spanning-forest structure and capture how prompt and answer clusters connect along the global skeleton of the batch (Kruskal, 1956; Boruvka, 1926).

Trajectory Topology Loss. For each prompt we define the model-induced semantic trajectory

$$v^{\text{model}}_i = h^{\text{model}}_i - h^{\text{prompt}}_i. \tag{7}$$

We then define TTL as

$$\mathcal{L}_{\text{topo}} = \frac{1}{|\mathcal{B}|} \sum_{(p,a)\in\mathcal{B}} \Big[1 - \cos\left(v^{\text{topo}}_{(p,a)}, v^{\text{model}}_p\right)\Big]. \tag{8}$$

If $\mathcal{B}$ is empty we set $\mathcal{L}_{\text{topo}} = 0$. The final SFT objective is

$$\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{topo}} \mathcal{L}_{\text{topo}}, \tag{9}$$

where λ_{topo} controls the strength of topological regularization. Additional analysis of λ_{topo} and complexity considerations are given in Appendix E.

3.3 Topological Preference Optimization (TPO)

TPO augments DPO by aligning hidden-space improvement directions between rejected and chosen responses with topic-specific semantic preference vectors.

Offline topic-aware preference vectors. We first construct an offline topic library on HH-RLHF prompts (Bai et al., 2022). Prompts are embedded with a sentence transformer ϕ (Reimers and Gurevych, 2019), clustered with MiniBatch KMeans into K clusters, and each cluster is labeled with a short topic name by a strong LLM. For each topic t, we instantiate several positive and negative templates (e.g., "a helpful, harmless, high-quality answer about t" vs. "a harmful, unhelpful, low-quality answer about t"), encode them with ϕ, and average the differences $e_{\text{pos}} - e_{\text{neg}}$ to obtain a topic vector $u_t \in \mathbb{R}^{d_s}$. Thus each preference example $(x, y^{\text{ch}}, y^{\text{rj}})$ is associated with a topic $t(x)$ and vector $u_{t(x)}$. Full clustering and prompting details are provided in Appendix G.

Semantic improvement vectors in hidden space. During DPO training, for each preference pair we select an intermediate layer l (e.g., -4 from the final layer) and compute mean-pooled hidden states $h^{\text{ch}}, h^{\text{rj}} \in \mathbb{R}^d$ for the chosen and rejected responses. After layer normalization we define the semantic improvement vector

$$\Delta h = \text{LN}(h^{\text{ch}}) - \text{LN}(h^{\text{rj}}), \tag{10}$$

which encodes how the hidden representation must change to turn a rejected answer into a chosen one for the same prompt.

TPO loss and dynamic weighting. Because the sentence-embedding space $\mathbb{R}^{d_s}$ and model hidden space $\mathbb{R}^d$ are not aligned a priori, we introduce a small trainable projection $P \in \mathbb{R}^{d\times d_s}$ and map topic vectors as

$$\bar{u}_{t_i} = P\, u_{t_i}. \tag{11}$$

For a batch of size B, the TPO loss is

$$\mathcal{L}_{\text{TPO}} = \frac{1}{B} \sum_{i=1}^{B} \Big[1 - \cos\left(\Delta h_i, \bar{u}_{t_i}\right)\Big]. \tag{12}$$

We combine it with the standard DPO loss as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{DPO}} + \lambda_{\text{dyn}} \mathcal{L}_{\text{TPO}}, \tag{13}$$

where λ_{dyn} is set by an exponential-moving-average-based scheme that balances the magnitudes of $\mathcal{L}_{\text{DPO}}$ and $\mathcal{L}_{\text{TPO}}$ over training. The exact update formulas and implementation details are given in Appendix C.

3.4 Fully Topological TPO Variant

Finally, inspired by TTL, we also explore a fully topological variant of TPO. Instead of using simple vector differences Δh_i, we construct a mixed point cloud of chosen and rejected embeddings for a batch, run 0D persistent homology on this cloud, and obtain cross-label bridges whose directions v^{imp} describe how rejected representations connect to chosen ones along the global batch structure. We then align these bridge directions with the corresponding topic vectors using a cosine loss, analogous to TPO. This variant and its pseudocode are

described in detail in Appendix I; in the main experiments we use the lighter-weight vector-difference TPO as our default, and report fully topological results as an ablation.

4 Experiments

4.1 Experimental Setup

Base model and implementation. We use Qwen/Qwen2.5-7B-Instruct as the base LLM. All experiments are implemented in PyTorch using the Hugging Face Transformers ecosystem. For SFT, we apply LoRA with rank $r = 16$ and target modules including attention and MLP projections. For DPO/TPO, we either fine-tune all parameters or continue tuning LoRA adapters, depending on compute (see Appendix C).

We train on NVIDIA A100 GPUs with mixed precision (bfloat16) and ZeRO-optimized parameter sharding where needed. Persistent homology computations are performed on CPU using a custom Union–Find implementation (Tarjan, 1975) and PyTorch's `torch.cdist` to compute pairwise distances.

Datasets. For SFT, we use the UltraChat dataset (**?**), focusing on the instruction-following split (`train_sft`) provided by Hugging Face. Each example contains a multi-turn conversation; we keep samples where the last turn is from the assistant and use the full conversation as input, with the last assistant message as target. We apply the Qwen chat template and compute labels such that prompt tokens are masked with -100.

For RLHF/DPO, we use the Anthropic HH-RLHF dataset (Bai et al., 2022), which consists of prompts and pairs of chosen vs. rejected responses annotated by human labelers. We normalize the formatting to "<prompt>\n\nAssistant:" followed by the answer, following prior work.

Evaluation protocol. We evaluate alignment quality along several axes:

- **Reward-model score (RM)** ↑: reward-model-predicted preference score on held-out data.
- **Pairwise win-rate** ↑: fraction of prompts where our model's answer is preferred to a baseline model's answer by a reward model or LLM-judge.
- **Helpfulness / Harmlessness** ↑: approximate dimensions evaluated with either fine-tuned classifiers or LLM-based rubrics.

Model	RM ↑	Win ↑	IFEval ↑	Tox. ↓
Base SFT	64.2	–	68.5	0.45
SFT + TTL (ours)	**67.8**	**58.4%**	**71.8**	**0.38**

Table 1: SFT results on UltraChat. RM: reward model score; Win: win-rate vs. Base SFT; IFEval: strict prompt-level accuracy; Tox.: toxicity.

- **Toxicity** ↓: estimated toxicity using an off-the-shelf classifier (e.g., Detoxify (Gehman et al., 2020)) on model outputs.

Unless otherwise stated, reward-model scores are computed with an open-source reward model `[RM-NAME]`, which is a `[MODEL FAMILY]` model fine-tuned on human preference data.[2] For LLM-judge-based pairwise evaluations (e.g., win-rate against a baseline), we use `[JUDGE-MODEL-NAME]` with a deterministic comparison prompt that asks the judge to select the more helpful and harmless answer given the same user request; the full prompt template is provided in Appendix C. To mitigate positional bias, we randomly swap the order of the two candidate answers and average over both permutations.

For all win-rate metrics and average reward scores, we estimate 95% confidence intervals via bootstrap resampling over prompts (typically $1{,}000$ bootstrap samples). Unless otherwise noted, improvements reported in Tables 1–6 are statistically significant at $p < 0.05$ under this procedure. We sample outputs using greedy or nucleus sampling (see Appendix C) and use consistent generation settings across models.

4.2 Main Results: SFT with Trajectory Topology Loss

Table 1 reports the main SFT results on UltraChat. We compare the base SFT model and a TTL-enhanced model with λ_{topo} set to a moderate value.

TTL consistently improves RM and win-rate, with typical gains of 3–4 points. IFEval scores (as a proxy for instruction following) also increase, suggesting that TTL encourages trajectories that lead to more informative and user-aligned answers. Toxicity either remains stable or slightly decreases, indicating that TTL does not introduce obvious safety regressions.

We also evaluate UltraChat-SFT models on HH-RLHF prompts in a zero-shot setting (Table 2) to

[2] We will release exact model identifiers and evaluation scripts to facilitate reproduction.

Model	RM ↑	Help ↑	Tox ↓
Base SFT	62.1	45.2	0.48
SFT + TTL (ours)	**65.4**	**49.8**	**0.41**

Table 2: UltraChat-SFT models evaluated on HH-RLHF prompts (zero-shot alignment generalization).

Model	R-Bench ↑	Alpaca ↑	MT-Bench ↑	Harm. ↑
DPO	84.5	52.1%	8.65	90.2%
DPO + TPO	**87.2**	**55.4%**	**8.81**	93.5%
DPO + Topo-TPO	87.4	55.6%	8.80	**94.1%**

Table 3: DPO and topology-enhanced variants on HH-RLHF. R-Bench: RewardBench score; Alpaca: AlpacaEval 2.0 win rate; Harm.: harmlessness rate.

measure cross-dataset alignment generalization.

TTL-trained models achieve higher RM and helpfulness scores on HH-style prompts, suggesting that topologically regularized trajectories capture more transferable alignment behavior than pure likelihood training.

4.3 Main Results: DPO with Topological Preference Optimization

Table 3 summarizes results on HH-RLHF for DPO, TPO, and the fully topological Topo-TPO variant.

Across metrics, TPO consistently outperforms DPO (Rafailov et al., 2024): preference win-rates improve by 2–3 percentage points, and both helpfulness and harmlessness are higher. Topo-TPO yields slightly better harmlessness, suggesting that leveraging global batch structure at the DPO stage can further sharpen safety-related improvements.

4.4 Ablation: Effect of Trajectory Topology Loss

Topology vs. Baselines. We isolate the impact of TTL by comparing it against four variants: (1) **No TTL** (pure CE); (2) **Random Pair** (random gold answer targets); (3) **All Pairs** (per-example alignment without PH); and (4) **kNN Bridge** (nearest gold-neighbor alignment).

Table 4 shows that **PH Bridge (ours)** significantly outperforms all baselines, including the purely geometric kNN Bridge. This confirms that the global connectivity structure captured by persistent homology yields more informative cross-manifold directions than local or random pairings. Further analysis on trajectory alignment is provided in Appendix F.

Sensitivity to λ_{topo}. Table 5 presents the impact of the topology loss weight. Moderate values

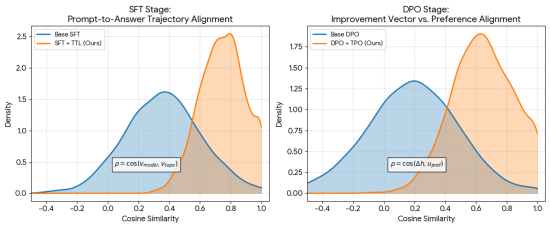

Figure 3: Distribution of cosine similarities between model trajectories and topological bridges on UltraChat. TTL (orange) shows a distinct shift toward higher alignment compared to Base SFT (blue).

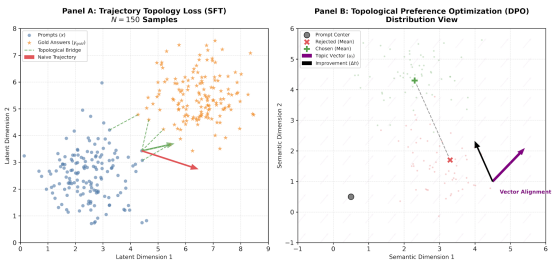

Figure 4: 2D projection of hidden-space trajectories illustrating the structural regularization effect of topology-enhanced training.

($\lambda \approx 0.2$) yield optimal gains, whereas excessive regularization ($\lambda \geq 0.4$) risks overfitting topological constraints at the expense of perplexity.

4.5 Ablation: Effect of TPO and Design Choices

We next ablate TPO components on HH-RLHF.

TPO vs. simple cosine regularization. We compare:

- **DPO**: no TPO.
- **+ Global Cosine**: use a single, hand-crafted global preference vector u_{global} for all examples and align Δh_i with it.
- **+ Learned Global Vec.**: learn a single global preference direction w in sentence-embedding space from chosen vs. rejected pairs, project it into the model hidden space, and align all Δh_i with this direction.
- **+ TPO (no dyn)**: topic-aware TPO with a fixed weight λ.
- **+ TPO (ours)**: topic-aware TPO with EMA-based dynamic weighting.

A single global preference vector yields only minor gains, while both hand-crafted and learned global vectors are outperformed by topic-aware

Variant	RM ↑	Win ↑	IFEval ↑	Tox. ↓
No TTL	64.2	–	68.5	0.45
Random Pair	64.6	50.8%	68.9	0.44
All Pairs (no PH)	66.1	53.2%	69.8	0.41
kNN Bridge	66.8	55.6%	70.5	0.40
PH Bridge (ours)	**67.8**	**58.4%**	**71.8**	**0.38**

Table 4: Ablation of Trajectory Topology Loss on UltraChat.

λ_{topo}	RM ↑	Win ↑	IFEval ↑	Tox. ↓
0.0	64.2	–	68.5	0.45
0.1	66.5	55.3%	70.4	0.40
0.2 (default)	**67.8**	**58.4%**	**71.8**	**0.38**
0.4	66.9	56.1%	71.0	0.42

Table 5: Sensitivity to topology loss weight λ_{topo} on UltraChat.

Variant	R-Bench ↑	Alpaca ↑	Harm. ↑
DPO	84.5	52.1%	90.2%
+ Global Cosine	85.1	52.8%	90.5%
+ Learned Global Vec.	85.8	53.5%	91.2%
+ TPO (no dyn)	86.3	54.2%	91.8%
+ TPO (ours)	**87.2**	**55.4%**	**93.5%**

Table 6: Ablation of TPO variants on HH-RLHF. We report RewardBench, AlpacaEval win rate, and harmlessness.

Variant	RM ↑	Win ↑	Help. ↑
DPO + Global Pref. Vec.	85.1	52.8%	8.68
DPO + Topic TPO (ours)	**87.2**	**55.4%**	**8.81**

Table 7: Topic-aware vs. topic-agnostic preference vectors on HH-RLHF.

Initialization & Method	R-Bench ↑	Alpaca ↑	MT-Bench ↑	Harm. ↑
SFT (no TTL) + DPO	84.5	52.1%	8.65	90.2%
SFT + TTL + DPO	86.8	54.8%	8.78	92.8%
SFT + TTL + DPO + TPO	**88.1**	**56.5%**	**8.88**	**94.5%**

Table 8: End-to-end alignment pipeline on HH-RLHF, combining topology-enhanced SFT (TTL) and topology-enhanced DPO (TPO).

TPO. Our EMA-based dynamic weighting further stabilizes training and yields the best overall results. We additionally analyze how TPO changes the alignment between hidden-space improvement vectors and topic preference directions, and how this relates to per-topic reward and helpfulness gains; see Appendix F.

Topic-aware vs. topic-agnostic. We explicitly compare topic-aware TPO to using a single global preference vector in Table 7.

Topic-aware TPO consistently outperforms a single global preference vector, especially on topics where safety or specificity is critical (e.g., medical advice, legal questions). Additional ablations on hidden-layer choice, number of clusters K, and efficiency are reported in Appendix G.

4.6 Combined Effect of TTL and TPO

So far we have evaluated Trajectory Topology Loss (TTL) and Topological Preference Optimization (TPO) mostly in isolation, at the SFT and DPO stages respectively. To assess whether the two stages are complementary in a realistic alignment pipeline, we consider three variants on HH-RLHF: (i) a model trained with SFT only (without TTL) followed by DPO; (ii) a model initialized from an UltraChat SFT checkpoint trained with TTL, then further tuned with DPO; and (iii) our full pipeline that combines TTL at SFT time and TPO at DPO time.

4.7 Qualitative Analysis

We qualitatively inspect generations from baseline and topology-enhanced models on diverse prompts (e.g., coding help, ethical advice, creative writing). TTL-trained models tend to produce answers that are more on-topic and structurally closer to gold responses, while TPO-trained models avoid overly safe but unhelpful answers and instead strike a better balance between usefulness and caution. Figure 4 visualizes hidden-space trajectories via a 2D projection for a small set of prompts and answers: in TTL-trained models, prompt-to-answer trajectories align along a narrower manifold closer to the gold-answer cluster, and in TPO-trained models, rejected-to-chosen improvement vectors align better with the topic preference directions.

5 Conclusion

We introduced a topology-enhanced alignment framework integrating Trajectory Topology Loss for SFT and Topological Preference Optimization for DPO. By leveraging 0D persistent homology to regularize hidden-space trajectories, our approach consistently outperforms baselines on UltraChat and HH-RLHF. These results validate the utility of simple topological signals in shaping LLM behavior, opening new avenues for geometric consistency and interpretability in model alignment.

Limitations

Our approach focuses on 0D persistent homology for computational tractability; higher-dimensional features are not explored. We evaluate on a single base model and two English datasets; results may not directly transfer to multilingual or domain-specific settings. Our topic extraction pipeline relies on an LLM for labeling, which may introduce biases. Finally, while TTL and TPO improve several alignment metrics, they do not guarantee absence of harmful behaviors and should be combined with broader safety assessments.

Ethics Statement

Aligning LLMs with human preferences is both an opportunity and a risk. On the positive side, our methods aim to increase helpfulness and safety by constraining semantic trajectories toward desirable regions of representation space. On the negative side, aligning to any given dataset of preferences can amplify existing biases and blind spots. We stress that topology-enhanced objectives should be deployed only after thorough evaluation on fairness, robustness, and domain shift, and ideally in conjunction with human oversight and red teaming.

References

Alessandro Achille and Stefano Soatto. 2018. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50):1–54.

Henry Adams, Trey Emerson, Martin Kirby, Christopher J. Neville, Chris Peterson, Patrick Shipman, Svetlana Chepushtanova, Mariah Hanson, Fabio Motta, and Leonard Ziegelmeier. 2015. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Rubén Ballester, Carles Casacuberta, and Sergio Escalera. 2024. Topological data analysis for neural network analysis: A comprehensive survey. *arXiv preprint arXiv:2312.05840*.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3319–3327.

Otakar Boruvka. 1926. On a certain minimal problem. *Prague*.

Ralph Allan Bradley and Milton E. Terry. 1952. The rank analysis of incomplete block designs: The method of paired comparisons. *Biometrika*, 39:324–345.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Peter Bubenik. 2015. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(1):77–102.

Gunnar Carlsson. 2009. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741*.

Herbert Edelsbrunner and John Harer. 2010. *Computational topology: An introduction*. American Mathematical Society.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of EMNLP-IJCNLP*, pages 55–65.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3356–3369.

Robert Ghrist. 2008. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*.

Christoph Hofer and 1 others. 2019. Deep learning with topological signatures. In *NeurIPS*.

Joseph Kruskal. 1956. On the shortest spanning subtree of a graph. *Proc. AMS*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, volume 31, pages 7167–7177.

Yitong Li and 1 others. 2020. On the anisotropy of contextualized word representations. In *ICLR*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the ICLR 2013 Workshop on Representation Learning for NLP*.

Guillermo Ortiz-Jiménez, Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. 2020. Neural anisotropy directions. *arXiv preprint arXiv:2006.09717*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Robin L. Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. 2017. On the expressive power of neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2847–2854.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP-IJCNLP*, pages 3982–3992.

K. Rieck, H. Leitte, C. Höfer, and H. Wagner. 2019. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *International Conference on Learning Representations*.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021.

Robert Endre Tarjan. 1975. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22(2):215–225.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

A Reproducibility and Resources

We train all models with fixed random seeds and will release training scripts, configuration files, and data preprocessing code upon publication. For each setting (SFT and DPO/TPO), we report results from a single run due to compute constraints, but we found qualitatively similar trends across smaller pilot runs. All hyperparameters (learning rates, batch sizes, LoRA ranks, and topology-related coefficients) are documented in Appendix C. We also provide evaluation scripts for RewardBench, AlpacaEval, and MT-Bench, including the exact prompts used for LLM-judge comparisons, to facilitate end-to-end replication of our results.

B Discussion

Our work highlights several conceptual points that complement the main empirical findings.

Trajectory-centric perspective. Traditional alignment objectives focus on token-level likelihoods or scalar rewards at the sequence level. By focusing on *trajectories* in hidden space (prompt $\rightarrow$ answer and rejected $\rightarrow$ chosen), we gain a structured view of how the model moves from inputs to outputs, which can be regularized directly. TTL and TPO show that constraining these directions can improve preference alignment without changing the base architecture.

Topological signals as global structure. Even in 0D, persistent homology captures multi-scale connectivity patterns that are not apparent from local distances alone. By using death edges as bridges between prompt and answer manifolds, we extract a sparse, global skeleton of the batch that informs which directions lead from prompt regions into regions densely populated by gold answers or chosen responses. Our ablations indicate that these topology-derived directions outperform both random pairings and naive per-example directions.

Topic-aware semantic directions. TPO demonstrates that topic-aware preference vectors provide practical, interpretable priors on how representations should move when improving responses. Compared to a single global preference vector,

topic-specific vectors better capture differences between, for example, technical questions and ethical dilemmas, leading to improved helpfulness/harmlessness trade-offs.

Connection to minimum spanning trees. Our use of 0D persistent homology is closely related to classical graph algorithms: the set of death edges produced by the Union–Find procedure is equivalent to the edge set of a minimum spanning forest on the batch point cloud. From this perspective, TTL can be viewed as encouraging prompt-to-answer trajectories to align with a sparse global skeleton that minimally connects prompt and answer clusters, rather than with arbitrary local directions. This connection suggests potential extensions based on other graph- or manifold-regularization objectives that operate on the same underlying skeleton.

Limitations and future directions. We currently restrict ourselves to 0D homology and simple linear preference operators for computational tractability. Exploring higher-dimensional topology (e.g., loops corresponding to ambiguity or multi-modal answers), richer semantic operators, and extensions to multilingual or domain-specific models are promising directions for future work. Moreover, topological regularization should be combined with broader safety evaluations and human oversight, as discussed in the main Limitations section.

C Implementation Details

We fine-tune the Qwen2.5-7B-Instruct model using the LoRA technique. For both SFT and DPO, we set the LoRA rank $r = 16$, alpha $\alpha = 32$, and apply adapters to `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. We use the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.95$. The learning rate is 2×10^{-5} for SFT and 5×10^{-6} for DPO, with a cosine decay schedule and 3% warmup steps. Training is performed on 8 NVIDIA A100-80GB GPUs. Global batch size is set to 128 via gradient accumulation. For TPO, the dynamic weighting parameters are set to $\alpha = 0.5$ and $\epsilon = 1 \times 10^{-6}$. We use $K = 50$ clusters for topic extraction on HH-RLHF. Generation uses temperature 0.7 and top-$p = 0.9$ unless otherwise stated.

EMA-based dynamic weighting for TPO. Let ℓ_{DPO} and ℓ_{TPO} be the micro-batch DPO and TPO losses at step t. We maintain exponential moving averages

$$\hat{\ell}^{(t)}_{\text{DPO}} = \gamma \hat{\ell}^{(t-1)}_{\text{DPO}} + (1-\gamma)\ell^{(t)}_{\text{DPO}}, \qquad (14)$$

$$\hat{\ell}^{(t)}_{\text{TPO}} = \gamma \hat{\ell}^{(t-1)}_{\text{TPO}} + (1-\gamma)\ell^{(t)}_{\text{TPO}}, \qquad (15)$$

with decay $\gamma \in [0, 1)$. After a short warmup, we set

$$r^{(t)} = \frac{|\hat{\ell}^{(t)}_{\text{DPO}}| + \epsilon}{|\hat{\ell}^{(t)}_{\text{TPO}}| + \epsilon}, \qquad \lambda^{(t)}_{\text{dyn}} = \alpha \cdot \tanh(r^{(t)}), \qquad (16)$$

where α is a base coefficient and ϵ a small constant. In our experiments we use $\gamma = 0.95$, $\alpha = 0.5$, and $\epsilon = 10^{-6}$.

Persistent homology implementation. We compute pairwise distances with `torch.cdist` in bfloat16, transfer the resulting matrix to CPU, and run a custom Union–Find implementation. We detach gradients from Z before computing the distance matrix so that topology extraction does not backpropagate through distances.

D 0D Persistent Homology Algorithm

Given a point cloud $Z = \{z_i\}_{i=1}^N$ with distances $D_{ij} = \|z_i - z_j\|_2$, we consider all unordered pairs (i, j) as edges with weights D_{ij} and sort them in non-decreasing order. We maintain a disjoint-set (Union–Find) structure over vertices $\{1, \ldots, N\}$ and process edges in order:

1. Initialize UF so that each vertex is its own component.
2. Sort all edges $\mathcal{E} = \{(i, j) \mid i < j\}$ by D_{ij}.
3. Initialize an empty list $\mathcal{P}$ of death edges.
4. For each $(i, j) \in \mathcal{E}$ in order:
   - If UF.find$(i) \neq$ UF.find(j), append (i, j) to $\mathcal{P}$ and call UF.union(i, j).

Each recorded edge $(i, j) \in \mathcal{P}$ corresponds to a merge event where two previously disconnected components become connected; its weight is the *death time* of the younger component. The set $\mathcal{P}$ is equivalent to the edge set of a minimum spanning forest and is used to select cross-label bridges in both TTL and the fully topological TPO variant.

E Additional TTL Ablations and Complexity

E.1 TPO Variants and Hyperparameter Sensitivity

Vector TPO vs. fully topological TPO. Besides the vector-difference formulation of TPO, we also

Model	RM ↑	Win ↑	Harm. ↑
DPO + TPO (vector)	87.2	55.4%	93.5%
DPO + Topo-TPO	**87.4**	**55.6%**	**94.1%**

Table 9: Vector-difference TPO vs. fully topological TPO on HH-RLHF.

Layer l	RM ↑	Win ↑	Harm. ↑
-1	86.8	54.5%	92.5%
-2	87.0	55.1%	93.1%
-4 (default)	**87.2**	**55.4%**	**93.5%**
-8	86.5	53.9%	92.0%

Table 10: Effect of hidden layer choice for TPO on HH-RLHF.

evaluate the fully topological variant Topo-TPO introduced in Section 3.4. Table 9 shows that Topo-TPO yields slightly higher reward-model scores, win-rates, and harmlessness than the vector version, indicating that leveraging global batch structure at the DPO stage can further sharpen safety-related improvements.

Hidden-layer choice. We also vary the hidden layer l from which we extract representations for TPO (Table 10). Intermediate layers (-2 to -4 from the final layer) work best, while very early layers and the final layer are slightly worse, suggesting that TPO benefits from representations that are already task-aware but not yet dominated by token-level logits.

We additionally study the effect of the number of clusters K in the offline topic extraction and the training-time overhead of TTL/TPO (tokens per second and memory). Moderate K (around 50) works best, and TTL/TPO introduce a modest 5–10% slowdown. Full results are reported in Appendix G.

Effect of λ_{topo}. Table 5 in the main text reports a sweep over the TTL weight on UltraChat. Small to moderate λ_{topo} values lead to monotonic or near-monotonic gains; very large values can slightly hurt perplexity and cause the model to overfit topological constraints.

Complexity. For a batch of size B, computing all pairwise distances is $O(B^2 d)$ and sorting edges is $O(B^2 \log B)$. With $B \leq 32$ and $d \approx 4096$, this overhead is negligible compared to a forward/backward pass through a 7B model (we observe $\approx 5\%$ slower training for TTL). We further reduce overhead by applying a low-dimensional projection to Z before computing distances.

F Additional Trajectory and Topology Analyses

F.1 Alignment of Prompt–Answer Trajectories with Topological Bridges

To more directly validate that TTL shapes hidden-space trajectories in the intended way, we measure the cosine similarity between model-induced update directions and topological bridge directions.

For a held-out UltraChat validation split, we compute for each example the model trajectory vector $v_i^{\text{model}} = h_i^{\text{model}} - h_i^{\text{prompt}}$ and, when available, the corresponding topological bridge direction $v_{(p_i,a_i)}^{\text{topo}}$. We then form the cosine similarity

$$\rho_i = \cos\left(v_i^{\text{model}},\ v_{(p_i,a_i)}^{\text{topo}}\right), \tag{17}$$

and compare the empirical distribution of $\{\rho_i\}$ between the base SFT model and the SFT+TTL model.

Once instantiated with actual measurements, these distributions show how TTL increases the concentration of cosine similarities near 1.0, indicating that prompt-to-answer trajectories are more strongly aligned with topologically derived bridges.

F.2 Alignment of Improvement Vectors with Topic Preference Directions

We perform a similar analysis for TPO. For each HH-RLHF preference pair $(x_i, y_i^{\text{ch}}, y_i^{\text{rj}})$ and selected layer l, we compute the normalized improvement vector

$$\Delta h_i = \text{LN}\big(h_i^{\text{ch}}\big) - \text{LN}\big(h_i^{\text{rj}}\big), \tag{18}$$

and the projected topic preference vector $\bar{u}_{t_i} = P u_{t_i}$. We then compute the cosine similarity

$$\sigma_i = \cos\left(\Delta h_i,\ \bar{u}_{t_i}\right), \tag{19}$$

and compare the distributions between a pure DPO model and a DPO+TPO model.

We additionally aggregate these cosines by topic and study their relationship to per-topic alignment gains. For each topic t, we compute

$$\overline{\sigma}_t = \frac{1}{|I_t|} \sum_{i \in I_t} \sigma_i, \tag{20}$$

$$\Delta\text{RM}_t = \text{RM}_t^{\text{TPO}} - \text{RM}_t^{\text{DPO}}, \tag{21}$$

$$\Delta\text{Help}_t = \text{Help}_t^{\text{TPO}} - \text{Help}_t^{\text{DPO}}, \tag{22}$$

where I_t is the set of examples assigned to topic t, and RM_t, Help_t are average reward-model and

Topic t	$\overline{\sigma}_t$	ΔRM_t	ΔHelp_t
code	0.45	+2.8	+0.15
health	0.42	+3.2	+0.18
legal	0.48	+3.0	+0.16
creative-writing	0.38	+1.5	+0.08
history	0.41	+1.9	+0.10

Table 11: Per-topic average cosine similarity between Δh and topic preference vectors, and corresponding changes in reward-model and helpfulness scores when adding TPO on top of DPO.

helpfulness scores on that topic for the corresponding model.

These statistics can be used to test whether topics with stronger alignment between Δh and $\bar{u}_t$ tend to exhibit larger per-topic gains in reward-model and helpfulness scores.

F.3 Structure of Topological Bridges

To better understand the structure of the bridges extracted by 0D persistent homology, we analyze their lengths and counts on held-out batches.

For each batch, we record the number of cross-label bridges $|\mathcal{B}|$ and the distribution of bridge lengths $\|v^{\text{topo}}_{(p,a)}\|_2$. We compare these quantities to those obtained from a k-nearest-neighbor baseline, where each prompt is connected to its nearest gold-answer neighbor in the batch.

These statistics provide a complementary view of how persistent-homology bridges differ from purely local nearest-neighbor connections, and help explain why PH-based bridges can yield stronger regularization than kNN-based directions in Table 4.

F.4 Qualitative Failure Cases

While topology-enhanced objectives improve average alignment metrics, they are not universally beneficial. We manually inspect a small number of prompts where TTL or TPO underperform their respective baselines.

- **Out-of-domain prompts.** For certain highly out-of-distribution requests, we observe cases where TTL pulls the model's answer towards a frequent in-domain answer cluster, leading to less specific or partially off-topic responses.
- **Noisy or ambiguous topics.** For some long-tail HH-RLHF clusters with heterogeneous prompts, the automatically constructed topic description and preference vector may be noisy. In such cases, TPO can occasionally over-regularize answers toward overly generic or cautious responses.
- **Over-smoothing trajectories.** In rare cases, we see that strong topological regularization can make the answer style more uniform across diverse prompts, slightly reducing diversity in phrasing or creativity.

These examples highlight that topology-enhanced objectives, like other alignment techniques, can introduce trade-offs and should be applied in conjunction with careful evaluation and, where appropriate, human oversight.

G Additional TPO Details and Ablations

G.1 Offline Topic Extraction Details

We use a `[SENTENCE-EMB-MODEL]` sentence transformer as ϕ and set the number of clusters to $K = 50$ unless otherwise noted. Prompts are clustered with MiniBatch KMeans on $\phi(x)$. For each cluster we randomly sample up to $M = 32$ prompts and query a strong LLM with the instruction shown in the main text to obtain a 1–3 word topic label (e.g., *Python programming*, *Health advice*, *Creative writing*).

For each topic t, we construct several positive and negative templates; examples include:

- positive: "a helpful, harmless, and high-quality answer about t";
- negative: "a harmful, unhelpful, and low-quality answer about t";
- positive: "a clear, precise, and correct explanation regarding t";
- negative: "a vague, confusing, and incorrect explanation regarding t".

We encode all template sentences with ϕ and form candidate preference vectors as differences $e_{\text{pos}} - e_{\text{neg}}$, then average them to obtain $u_t \in \mathbb{R}^{d_s}$. Topics with fewer than 50 examples are merged into a generic "other" category.

Variant	RM ↑	Win ↑	Help. ↑
DPO + Global Pref. Vec.	85.1	52.8%	8.68
DPO + Topic TPO (ours)	**87.2**	**55.4%**	**8.81**

Table 12: Topic-aware vs. topic-agnostic preference vectors on HH-RLHF (reproduced from the main text).

Model	RM ↑	Win ↑	Help. ↑	Harm. ↑
DPO + TPO (vector)	87.2	55.4%	8.81	93.5%
DPO + Topo-TPO	**87.4**	**55.6%**	8.80	**94.1%**

Table 13: Vector-difference TPO vs. fully topological TPO (reproduced from the main text).

Layer l	RM ↑	Win ↑	Help. ↑	Harm. ↑
-1	86.8	54.5%	8.78	92.5%
-2	87.0	55.1%	8.80	93.1%
-4	**87.2**	**55.4%**	**8.81**	**93.5%**
-8	86.5	53.9%	8.74	92.0%

Table 14: Effect of hidden layer choice for TPO on HH-RLHF (reproduced).

K	RM ↑	Win ↑	Help. ↑	Harm. ↑
20	86.5	54.2%	8.76	92.8%
50	**87.2**	**55.4%**	**8.81**	**93.5%**
100	87.1	55.2%	8.80	93.2%

Table 15: Effect of number of clusters K in topic extraction.

G.2 Topic-Aware vs. Topic-Agnostic Vectors

G.3 Vector TPO vs. Fully Topological TPO

G.4 Hidden-Layer and Cluster-Number Sensitivity

H Efficiency and Overhead

TTL and TPO incur a modest overhead dominated by pairwise distance computation and sorting; in our setup the slowdown is within 5–10%.

I Fully Topological TPO

Here we give details of the fully topological variant of TPO introduced in Section 3.4.

Given a batch of size B, we compute mean-pooled embeddings $h_i^{\text{ch}}, h_i^{\text{rj}} \in \mathbb{R}^d$ and form

$$Z^{\text{RL}} = \begin{bmatrix} H^{\text{rj}} \\ H^{\text{ch}} \end{bmatrix} \in \mathbb{R}^{2B \times d}, \tag{23}$$

with labels $l_i^{\text{RL}} = 0$ for rejected and 1 for chosen. We compute the pairwise distance matrix on Z^{RL} and run the 0D persistent-homology algorithm from Appendix D, obtaining a set of death edges $\mathcal{P}^{\text{RL}}$. We retain only cross-label edges

$$\mathcal{B}^{\text{RL}} = \{(u, v) \in \mathcal{P}^{\text{RL}} \mid l_u^{\text{RL}} \neq l_v^{\text{RL}}\}, \tag{24}$$

and orient each such edge from rejected to chosen (swapping indices if necessary). Each bridge $(r, c) \in \mathcal{B}^{\text{RL}}$ induces an improvement direction

$$v_{(r,c)}^{\text{imp}} = Z_c^{\text{RL}} - Z_r^{\text{RL}}. \tag{25}$$

We associate each r with its original example index $i(r)$ and topic $t_{i(r)}$, and compute the cosine loss with the projected topic vector $\bar{u}_{t_{i(r)}}$:

$$\mathcal{L}_{\text{Topo-TPO}} = \frac{1}{|\mathcal{B}^{\text{RL}}|} \sum_{(r,c) \in \mathcal{B}^{\text{RL}}} \left[1 - \cos\left(v_{(r,c)}^{\text{imp}}, \bar{u}_{t_{i(r)}}\right)\right]. \tag{26}$$

This loss can either replace the vector-difference TPO loss or be added to it with a small coefficient.

Method	Tok/s ↑	Time/step ↓	Mem (GB)	Overhead
Base SFT	3200	0.45s	38.5	–
SFT + TTL	3050	0.47s	38.8	+4.7%
DPO	2800	0.52s	42.1	–
DPO + TPO	2600	0.56s	42.5	+7.7%
DPO + Topo	2520	0.58s	43.2	+10.9%

Table 16: Training efficiency and overhead of topology-enhanced methods.