Fine-Tuning Selectively Steers Policy Representations in Language Models

[Author Names] Affiliations>Affiliations

[emails]

Abstract

Alignment techniques improve behavioral epistemic competence: instruction-tuned models admit ignorance, acknowledge ambiguity, and recognize nonsensical questions. But what happens to internal representations during this process? We investigate via steering vector analysis and probe transfer across 8 models (4 families × base/instruct variants) with ∼600 prompts spanning 6 epistemic categories.

We find that fine-tuning *selectively steers* policy representations more than factual ones. Policy categories—where fine-tuning trains specific epistemic behaviors—move 1.08–1.28× further along the alignment direction than factual categories requiring knowledge recall (all $p < 0.001$ via bootstrap). Low-rank analysis confirms alignment changes concentrate in 14–19 dimensions.

Probe transfer experiments confirm this selective effect: training probes on base models and testing on instruction-tuned variants reveals factual representations transfer at 0.85–0.87 accuracy while policy representations transfer at only 0.62–0.63 for preference-optimized models (Llama, Qwen). SFT-only models (Mistral, Yi) show smaller or inconsistent gaps, suggesting preference optimization creates more targeted representational changes.

Training method comparison reveals distinct profiles: rejection sampling with DPO (Llama) compresses representations broadly, while DPO with GRPO (Qwen) operates more selectively. Cross-architecture consistency in steering ratios suggests selective steering is a fundamental property of current alignment techniques. The preservation of factual geometry points toward monitoring applications, while the selective steering of policy representations raises questions about whether behavioral alignment and representational transparency can be jointly achieved.

1 Introduction

Language models are increasingly deployed in domains where uncertainty matters, in fields like medicine, law, scientific research. When a model doesn't know something, we want it to say so. Alignment techniques appear to achieve this: instruction-tuned models admit ignorance, acknowledge ambiguity, and recognize nonsensical questions far more often than their base counterparts.

But how does fine-tuning achieve these behavioral improvements? Prior work established that language models represent uncertainty internally [Kadavath et al., 2022, Azaria and Mitchell, 2023]. Linear probes on activations can predict response correctness. Recent work on D-STEER [Raina et al., 2024] showed that DPO operates as "low-rank steering," modifying a narrow subspace of activations rather than broadly restructuring representations. We extend this by asking: *what gets steered and how?*

1.1 The Selective Steering Hypothesis

We hypothesize that fine-tuning *selectively steers* policy representations more than factual ones, whice moves trained epistemic behaviors further along the alignment direction while relatively preserving knowledge recall representations.

Specifically, we distinguish two types of prompt-based epistemic situations:

- **Factual categories**: The correct response requires knowledge recall. “What is the capital of France?” requires retrieving stored information.
- **Policy categories**: The correct response requires a trained epistemic behavior. “What is the capital of Bugoviana?” requires recognizing the entity is fictional and admitting ignorance.

Fine-tuning explicitly trains policy behaviors, like admitting “I don’t know,” acknowledging ambiguity, and recognizing category errors. Our hypothesis is that this training disproportionately steers policy representations along the alignment direction while preserving factual geometry.

1.2 Contributions

We test this hypothesis across 8 models (4 families × base/instruct) with 589 prompts spanning 6 epistemic categories. Our contributions:

1. **Steering analysis**: Policy categories move 1.08–1.28× further along the alignment direction than factual categories (all $p < 0.001$ via bootstrap). Low-rank analysis confirms alignment changes concentrate in 14–19 dimensions.
2. **Probe transfer**: Training probes on base models and testing on instruct reveals selective preservation. Factual representations transfer at 0.85–0.87 accuracy while policy representations transfer at only 0.62–0.63 for preference-optimized models.
3. **Training method comparison**: RS + DPO (Llama) compresses representations broadly; DPO + GRPO (Qwen) operates more selectively. Both use DPO but differ in additional methods, suggesting RS vs GRPO create distinct representational profiles.
4. **Cross-architecture validation**: We demonstrate consistent effects across Llama, Qwen, Mistral, and Yi, which suggests certain fundamental properties of alignment methods rather than architecture-specific artifacts.

1.3 Implications

Our findings suggest that current alignment techniques achieve epistemic competence through selective steering of representations for trained behaviors. This has implications for interpretability: probe-based monitoring may be less reliable for policy categories where alignment specifically trains outputs, while factual representations remain more accessible.

The preservation of factual geometry offers practical applications: probes trained on base models transfer well to instruct models for factual categories, suggesting potential for monitoring what knowledge is preserved during fine-tuning. The selective steering of policy representations raises questions about whether behavioral alignment and representational transparency can be jointly achieved.

2 Background and Related Work

2.1 Epistemic States in Language Models

Prior work established that language models represent epistemic states internally. Kadavath et al. [2022] showed models "know what they know"—internal representations correlate with output correctness. Azaria and Mitchell [2023] demonstrated that internal states can detect when models are "lying" or confabulating. These studies established linear probing as a standard approach for accessing internal epistemic state.

However, prior work primarily asked *whether* models have uncertainty representations, not *what happens to these representations during fine-tuning*. Critically, prior work does not distinguish between trained epistemic behaviors (admitting ignorance when asked about fictional entities) and inherent epistemic states (uncertainty about obscure facts). Our contribution is showing that fine-tuning affects these differently.

2.2 Fine-Tuning and Representation Geometry

Recent work has characterized how fine-tuning affects representation geometry. D-STEER [Raina et al., 2024] showed that DPO operates as "low-rank steering," modifying a narrow subspace of activations rather than broadly restructuring representations. The authors argue DPO teaches models "how to act aligned, not what to believe."

We extend this finding in two ways: (1) we confirm that low-rank structure generalizes beyond DPO to all fine-tuning methods tested (RLHF, GRPO, SFT-only); and (2) we identify *what gets steered*—specifically, policy categories where epistemic behaviors are trained move further along the alignment direction than factual categories.

2.3 Calibration and Overconfidence

Aligned models show systematic overconfidence in calibration studies. Output entropy becomes less predictive of correctness after fine-tuning, and self-reported confidence often exceeds actual accuracy.

Our work connects these calibration issues to representation-level changes. The selective steering of policy representations—moving them further along the alignment direction—may explain why behavioral epistemic competence improves while internal calibration degrades.

2.4 Training Method Taxonomy

We exploit natural variation in fine-tuning methods to characterize different steering profiles:

- **SFT only**: Mistral, Yi—supervised learning on instruction-following examples
- **SFT + RS + DPO**: Llama—supervised fine-tuning with rejection sampling and direct preference optimization (multiple rounds)
- **SFT + DPO + GRPO**: Qwen—supervised fine-tuning with direct preference optimization and group relative policy optimization

Both Llama and Qwen use DPO as their core preference optimization method, but differ in additional techniques: Llama uses rejection sampling (RS), while Qwen uses GRPO. This allows comparing how these additional methods affect representational structure beyond DPO alone.

Table 1: Epistemic category taxonomy with examples and evaluation criteria. Policy categories require trained behaviors; factual categories require knowledge recall.

Category	**Type**	**Example Prompt**	**Correct Response**
Confident-correct	Factual	"What is 2+2?"	States the answer ("4")
Uncertain-correct	Factual	"Capital of Burkina Faso?"	States the answer ("Ouagadougou")
Confident-incorrect	Policy	"Capital of Bugoviana?"	Acknowledges fictional entity
Ambiguous	Policy	"What does 'bank' mean?"	Requests clarification
Nonsensical	Policy	"What color is jealousy?"	Recognizes category error

3 Methodology

We design an experimental framework to measure how fine-tuning affects the separability of epistemic states in language model representations. Our approach combines category-specific prompts that elicit distinct epistemic situations with linear probing to measure representational separability across base and instruction-tuned model pairs.

3.1 Epistemic Category Taxonomy

We distinguish between two types of epistemic situations based on what constitutes a correct response:

Factual categories. These require knowledge recall—the model must retrieve and state factual information:

- **Confident-correct**: Clear factual questions with high-probability answers (e.g., "What is 2+2?", "What is the capital of France?").
- **Uncertain-correct**: Obscure but verifiable facts (e.g., "What is the atomic number of molybdenum?", "What is the capital of Burkina Faso?").

Policy categories. These require trained epistemic behaviors. The model should recognize a meta-property of the question and respond appropriately:

- **Confident-incorrect**: Questions about fictional entities where the correct response is to acknowledge the entity does not exist (e.g., "What is the capital of Bugoviana?").
- **Ambiguous**: Context-dependent questions where the correct response is to request clarification or acknowledge multiple interpretations (e.g., "What does 'bank' mean?").
- **Nonsensical**: Category error questions where the correct response is to recognize the question has no valid answer (e.g., "What color is jealousy?", "How much does Tuesday weigh?").

This distinction is important for our design. Fine-tuning explicitly trains policy behaviors, and behavior that is 'human-like', such as admitting ignorance, acknowledging ambiguity, and recognizing nonsense. But it should not fundamentally alter factual knowledge representations.

3.2 Dataset Construction

We construct a dataset of 589 prompts distributed across six epistemic categories. The five categories used in the policy-factual comparison are: 157 confident-correct, 100 confident-incorrect, 97 uncertain-correct, 80 ambiguous, and 70 nonsensical prompts.

The dataset also includes 98 **uncertain-incorrect** prompts (common misconceptions like "Did Vikings wear horned helmets?"). These are included in general analyses (accuracy, per-category probe performance) but excluded from the policy to factual comparison, because they are both factual and policy-related prompts. Responding correctly requires knowledge (knowing the misconception is false), but they are policy-based in that models can be trained to acknowledge the common misconception before correcting it. This ambiguity makes uncertain-incorrect prompts unsuitable for the binary policy or factual grouping.

Evaluation logic. We evaluate correctness using category-specific phrase matching, iteratively refined to capture the range of valid response patterns:

- For **factual categories**, correctness requires the response to contain the expected answer (case-insensitive substring matching, handling common variations).
- For **confident-incorrect**, correctness requires acknowledgment phrases indicating the entity does not exist (e.g., "doesn't exist", "fictional", "not a real country").
- For **ambiguous**, correctness requires clarification phrases (e.g., "could you clarify", "depends on context", "multiple meanings").
- For **nonsensical**, correctness requires recognition phrases (e.g., "category error", "doesn't have a color", "is an abstract concept").

This phrase-based approach enables automated evaluation. While we iteratively expanded the phrase lists to capture valid response patterns across models, extending to new model families would require identifying their characteristic refusal and ambiguity-acknowledgment styles. We discuss this limitation in Section 6.3.

3.3 Models and Activation Extraction

We study four model families, each with base and instruction-tuned variants (8 models total):

Table 2: Models studied and their training methods. This selection enables a natural experiment comparing SFT-only (Mistral, Yi) with RLHF/DPO methods (Llama, Qwen).

Family	**Model Size**	**Instruct Training**	**Primary Language**
Qwen 2.5	7B	SFT + DPO + GRPO	Chinese
Llama 3.1	8B	SFT + RS + DPO	English
Mistral v0.3	7B	SFT only	English
Yi 1.5	6B	SFT only	Chinese

By including both SFT-only models (Mistral, Yi) and models with preference optimization (Llama, Qwen), we can distinguish the effects of supervised fine-tuning from reinforcement learning methods. Additionally, including models trained primarily on English versus Chinese data allows us to assess whether these findings generalize across training data distributions.

Activation extraction. We use TransformerLens [Nanda and Bloom, 2022] to extract activations from all residual stream layers at the final token position of each prompt (before generation begins). For each prompt, we obtain an activation matrix $\mathbf{H} \in \mathbb{R}^{L \times d}$ where L is the number of layers and d is the hidden dimension. We flatten this to a single vector $\mathbf{h} \in \mathbb{R}^{Ld}$ for probing.

Prompt formatting. Both base and instruction-tuned models receive identical prompts in a simple completion format ("Question: {prompt} Answer:"). This prompt-controlled design ensures that any observed differences in representations are attributable to model differences rather than prompt differences. We generate responses using greedy decoding (temperature 0) with a maximum of 30 new tokens.

3.4 Steering Vector Analysis

Following D-STEER [Raina et al., 2024], we analyze how fine-tuning steers representations along a low-rank alignment direction.

Steering vector extraction. For each model family, we compute the alignment direction as the difference in activation centroids:

$$\mathbf{v}_{\text{steer}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{h}_i^{\text{instruct}} - \frac{1}{N} \sum_{i=1}^{N} \mathbf{h}_i^{\text{base}} \tag{1}$$

where $\mathbf{h}_i^{\text{base}}$ and $\mathbf{h}_i^{\text{instruct}}$ are activations for the same prompt in base and instruct models respectively.

Category projections. We project each sample onto the steering direction and compute mean projections by category:

$$\text{Proj}_c = \frac{1}{|S_c|} \sum_{i \in S_c} \frac{\mathbf{h}_i \cdot \mathbf{v}_{\text{steer}}}{\|\mathbf{v}_{\text{steer}}\|} \tag{2}$$

This measures how far each category moves along the alignment direction.

Low-rank analysis. We perform SVD on the activation difference matrix $\mathbf{D} = \mathbf{X}_{\text{instruct}} - \mathbf{X}_{\text{base}}$ to characterize the dimensionality of alignment changes:

$$r_{\text{eff}} = \min \left\{ k : \sum_{i=1}^{k} \sigma_i^2 \geq 0.80 \sum_{j} \sigma_j^2 \right\} \tag{3}$$

where σ_i are singular values.

Category loading ratio. To test whether policy categories are disproportionately affected by alignment changes, we compute how heavily each category loads onto the top SVD components. The SVD gives us $\mathbf{D} = \mathbf{U\Sigma V}^\top$, where the rows of $\mathbf{U\Sigma}$ represent each sample's projection onto the principal components of alignment change. For each category c, we compute the mean loading magnitude on the top-k components:

$$\text{Loading}_c = \frac{1}{|S_c|} \sum_{i \in S_c} \|(\mathbf{U\Sigma})_{i,1:k}\|_2 \tag{4}$$

The **policy/factual loading ratio** is then:

$$\text{Loading Ratio} = \frac{\frac{1}{|P|}\sum_{c\in P}\text{Loading}_c}{\frac{1}{|F|}\sum_{c\in F}\text{Loading}_c} \tag{5}$$

A ratio greater than 1 indicates policy categories load more heavily onto the alignment subspace than factual categories. We use $k = 10$ components and compute bootstrap confidence intervals (1000 iterations) for all ratios.

3.5 Linear Probing Protocol

We train linear probes to predict response correctness from activations:

$$\hat{y} = \sigma(\mathbf{w}^\top \mathbf{h} + b) \tag{6}$$

where $\mathbf{h}$ is the flattened activation vector, $\mathbf{w}$ and b are learned parameters, and σ is the sigmoid function. We use logistic regression with L2 regularization, trained via 5-fold stratified cross-validation. Activations are standardized (mean 0, variance 1) using statistics computed only on the training fold to prevent data leakage.

3.6 Probe Transfer Protocol

To test whether fine-tuning preserves epistemic representations, we measure how well probes generalize across base and instruct models:

Base-to-instruct transfer. We train a probe on base model activations and evaluate it on instruction-tuned model activations for the same prompts. High transfer accuracy indicates that the base model's geometry is preserved after fine-tuning; low transfer indicates potential restructuring.

Category-wise transfer. We compute transfer accuracy separately for policy and factual categories. The transfer gap quantifies selective geometry preservation:

$$\text{Transfer Gap} = \text{Acc}^{B\to I}_{\text{factual}} - \text{Acc}^{B\to I}_{\text{policy}} \tag{7}$$

A positive gap indicates factual representations are better preserved than policy representations.

3.7 Statistical Testing

We use multiple layers to check statistical robustness:

Sample-level permutation test. Our primary significance test operates at the sample level rather than the category level, providing substantially higher statistical power. For each sample, we estimate per-sample error using repeated cross-validation splits (100 iterations). We then test whether the mean error change differs between samples from policy versus factual categories using a permutation test with 10,000 permutations.

Bootstrap confidence intervals. We compute 95% confidence intervals via 2,000 bootstrap iterations with percentile method.

Table 3: Mean projection of each category type onto the steering vector $\mathbf{v}_{\text{steer}}$. Policy categories consistently move further than factual categories. All ratios significantly > 1.0 at $p < 0.001$ via bootstrap (1000 iterations).

Model	**Training**	**Ratio**	**95% CI**
Llama 3.1	SFT + RS + DPO	1.28×	[1.23, 1.33]
Qwen 2.5	SFT + DPO + GRPO	1.11×	[1.08, 1.15]
Mistral	SFT only	1.10×	[1.05, 1.14]
Yi 1.5	SFT only	1.08×	[1.05, 1.11]

Multiple comparison correction. For per-category tests, we apply Benjamini-Hochberg FDR correction to control the false discovery rate.

Seed sensitivity. We verify result stability by computing coefficient of variation (CV) across 5 random seeds, requiring CV $< 5\%$ for reported metrics.

4 Results

4.1 Main Finding: Selective Steering

Our central finding is that fine-tuning steers policy representations further along the alignment direction than factual representations.

Policy categories move 1.08–1.28× further along the alignment direction than factual categories. The effect is largest for Llama (1.28×), which uses the most extensive preference optimization (SFT + RS + DPO). This confirms the selective steering hypothesis: fine-tuning disproportionately affects representations for categories where it trains specific epistemic behaviors.

Low-rank structure. Following D-STEER, we analyze the dimensionality of alignment changes via SVD on the activation difference matrix.

Table 4: Effective rank (dimensions for 80% variance) shows alignment changes concentrate in a low-rank subspace across all training methods. Loading ratio measures how heavily policy categories load on top-10 SVD components relative to factual categories.

Model	**Training**	**Eff. Rank**	**Loading Ratio**	**95% CI**
Llama 3.1	SFT + RS + DPO	19	1.30×*	[1.23, 1.37]
Yi 1.5	SFT only	19	0.97×	[0.93, 1.00]
Mistral	SFT only	18	0.90×	[0.85, 0.94]
Qwen 2.5	SFT + DPO + GRPO	14	0.89×	[0.84, 0.94]

*Significantly > 1.0 at $p < 0.001$; others not significant or inverted.

Low-rank structure (14–19 dimensions) generalizes to all fine-tuning methods, confirming D-STEER's finding that alignment operates via narrow subspace modification. However, the relationship between policy/factual categories and SVD loading is model-dependent: only Llama shows policy categories loading significantly more heavily on alignment-relevant components. This suggests the selective steering effect observed in the mean alignment direction (Table 3) does not uniformly concentrate in the top SVD components across models.

Table 5: Probe transfer accuracy (train on base, test on instruct). For preference-optimized models, factual categories transfer well while policy categories show substantial degradation.

Model	Training	Factual Acc.	Policy Acc.	Gap
Qwen 2.5	SFT + DPO + GRPO	0.873	0.632	+0.241
Llama 3.1	SFT + RS + DPO	0.849	0.624	+0.225
Mistral	SFT only	0.916	0.898	+0.018
Yi 1.5	SFT only	0.871	0.675	+0.196

Table 6: Centroid distance changes during fine-tuning. Negative values indicate categories moving closer (convergence); positive values indicate divergence. RS + DPO (Llama) compresses broadly; DPO + GRPO (Qwen) operates more selectively.

Model	Training	Policy Δ%	Factual Δ%
Llama 3.1	SFT + RS + DPO	−16.3%	−11.2%
Qwen 2.5	SFT + DPO + GRPO	−3.4%	+3.4%
Mistral	SFT only	+16.8%	+66.3%
Yi 1.5	SFT only	+7.6%	+7.0%

4.2 Probe Transfer Confirms Selective Preservation

To directly test whether fine-tuning warps policy representations while preserving factual ones, we train probes on base model activations and test them on instruction-tuned model activations. If representations are preserved, transfer accuracy should remain high; if warped, transfer should degrade.

The preference-optimized models (Qwen, Llama) show clear asymmetry: probes trained on base models achieve high accuracy on factual categories (0.85–0.87) but substantially degraded accuracy on policy categories (0.62–0.63) when applied to instruct models. This demonstrates that fine-tuning selectively preserves the geometry of factual representations while restructuring policy-relevant dimensions.

SFT-only models show smaller gaps. Mistral shows minimal transfer gap (+0.018), suggesting SFT-only training preserves both category types relatively well. Yi shows an intermediate gap (+0.196), suggesting some selective effect even without preference optimization. This pattern aligns with the steering analysis: preference optimization creates more targeted representational changes.

4.3 Training Method Comparison

We observe distinct profiles for different training methods by examining how category centroids converge or diverge during fine-tuning.

RS + DPO compresses broadly. Llama shows convergence for both category types (policy −16.3%, factual −11.2%), indicating rejection sampling with DPO compresses the entire representation space. Policy categories converge slightly more.

DPO + GRPO operates selectively. Qwen shows selective convergence: policy categories converge (−3.4%) while factual categories actually diverge (+3.4%). This pattern suggests adding

GRPO to DPO reduces overall epistemic compression.

SFT-only shows divergence or mixed effects. Mistral and Yi both show divergence (positive values), with Mistral showing dramatically different effects for policy (+16.8%) versus factual (+66.3%). This suggests that SFT, alone, might generate more separable categorical representations in the underlying reasoning structure.

4.4 Supporting Evidence: Probe Accuracy and Robustness

We verify that linear separability is maintained despite the steering effects.

Table 7: Probe and entropy AUC before and after fine-tuning. Probe AUC measures linear separability of correct/incorrect responses in activation space; entropy AUC measures how well output entropy predicts correctness.

		Base		**Instruct**	
Model	**Training**	**Entropy**	**Probe**	**Entropy**	**Probe**
Mistral	SFT only	0.93	0.95	0.74	0.82
Llama 3.1	SFT + RS + DPO	0.91	0.92	0.73	0.82
Yi 1.5	SFT only	0.82	0.90	0.65	0.86
Qwen 2.5	SFT + DPO + GRPO	0.79	0.90	0.55	0.82

Linear probes still achieve 0.82–0.86 AUC after fine-tuning, indicating that representations remain linearly separable. The steering effect does not reduce linear structure.

Prompt-controlled design. Our results use identical prompts for base and instruct models, ensuring observed differences reflect model changes rather than prompt effects.

4.5 Cross-Architecture Consistency

One contribution of this work is demonstrating that selective steering is not an artifact of a single model architecture. Despite differences in architecture (Llama-derived vs. custom), training data (English vs. Chinese primary), and model scale (6B–8B), we observe:

1. All four models show policy categories moving further along the alignment direction (1.08–1.28×, all $p < 0.001$)
2. Low-rank structure (14–19 dimensions) is consistent across all training methods
3. Preference-optimized models show asymmetric probe transfer (high factual, lower policy)

However, the SVD loading pattern is *not* consistent: only Llama shows policy loading more heavily on top components (1.30×); other models show different patterns. This suggests the steering ratio and SVD loading capture different aspects of representational change.

5 Analysis

5.1 Why Does Fine-Tuning Selectively Steer Policy?

Our findings extend D-STEER's characterization of fine-tuning as "low-rank steering." We confirm that alignment changes concentrate in 14–19 dimensions (Table 4), generalizing the low-rank structure to all training methods tested.

What gets steered. The steering ratio (Table 3) shows policy categories consistently move 1.08–1.28× further along the alignment direction than factual categories. This is robust across all four model families ($p < 0.001$ via bootstrap). However, the relationship to SVD structure is model-dependent: only Llama (RS + DPO) shows policy loading more heavily on top components; Qwen (DPO + GRPO) shows the opposite pattern. This suggests the steering effect operates along the mean alignment direction but does not uniformly concentrate in top SVD components.

Hypothesis. The training signal specifically targets policy behaviors. Fine-tuning trains models to admit ignorance, acknowledge ambiguity, and recognize nonsense. This targeted training steers the representations underlying those behaviors further along the alignment direction, while factual recall is relatively preserved. The different SVD loading patterns between RS + DPO and DPO + GRPO suggest these methods achieve similar steering effects through different subspace modifications.

5.2 Training Method Effects

Different training methods produce distinct steering profiles:

RS + DPO compresses broadly. Llama (SFT + RS + DPO) shows convergence for both category types: policy categories converge 16.3% and factual categories converge 11.2% (Table 6). The entire representation space compresses toward the alignment direction, though policy categories move slightly further. Notably, Llama is the only model where policy categories load significantly more heavily on top SVD components (1.30×, Table 4).

DPO + GRPO operates selectively. Qwen (SFT + DPO + GRPO) shows selective convergence: policy categories converge 3.4% while factual categories *diverge* 3.4%. This striking pattern suggests adding GRPO to DPO specifically reduces epistemic compression. Interestingly, Qwen shows *inverted* SVD loading: factual categories load more heavily on top components (0.89×), opposite to Llama.

SFT-only causes different restructuring. Mistral and Yi both show divergence rather than convergence, with Mistral showing dramatically different effects for policy (+16.8%) versus factual (+66.3%). This pattern contrasts with the targeted effects of preference optimization, and may suggest that SFT-only models generate a category-based reasoning style that generates more epistemic separation.

5.3 Representations Change But Remain Separable

Despite the selective steering effect, linear separability is preserved. Probes achieve 0.82–0.86 AUC after fine-tuning across all models (Table 7). The steering effect moves representations along directions that preserve their linear separability.

This preservation of separability indicates that representational reorganization during alignment is *structured.* Fine-tuning steers policy representations further along the alignment direction, but this steering can maintain the geometric relationships that allow linear probes to distinguish correct from incorrect responses. The consistency of this pattern across four architectures (Llama, Qwen, Mistral, Yi), two training data distributions (English, Chinese), and multiple fine-tuning methods (SFT, RS+DPO, DPO+GRPO) suggests this structured reorganization is a fundamental property of current alignment techniques rather than an artifact of particular models.

5.4 Predictable Reorganization Accompanies Behavioral Gains

Behavioral performance improves dramatically after fine-tuning, especially for policy categories:

Table 8: Behavioral accuracy before and after fine-tuning. Policy accuracy improves substantially while representations are selectively steered.

		Factual Acc.		**Policy Acc.**	
Model	**Training**	**Base**	**Inst.**	**Base**	**Inst.**
Qwen 2.5	SFT + DPO + GRPO	81%	94%	5%	63%
Llama 3.1	SFT + RS + DPO	91%	95%	5%	56%
Mistral	SFT only	91%	91%	3%	35%
Yi 1.5	SFT only	83%	88%	2%	27%

Base models, as expected, achieve near-zero policy accuracy. They don't admit ignorance, acknowledge ambiguity, or recognize nonsense. Fine-tuning fixes this behaviorally (+50–60pp for preference-optimized models, +25–30pp for SFT-only). Better behavior comes with *predictable* representational reorganization that particularly affects policy categories.

The consistency of this pattern across architectures is notable. All four model families show the same qualitative effect: behavioral gains for policy categories accompanied by selective steering of those categories' representations (1.08–1.28× further along the alignment direction than factual categories). This cross-architecture consistency suggests the reorganization pattern may generalize to new models and alignment methods, rather than being an artifact of particular training choices.

6 Discussion

6.1 Implications for Alignment

Our findings have several implications for alignment research and interpretability:

1. Selective steering is a fundamental property of alignment, not model-specific. The consistency of our findings across four model families is striking. All models show policy categories moving 1.08–1.28× further along the alignment direction than factual categories (all $p < 0.001$). Low-rank structure (14–19 dimensions) is confirmed across all training methods. This cross-architecture consistency suggests selective steering will generalize to new models and alignment methods, making it a predictable property that could inform interpretability research.

2. Training methods create distinguishable representational profiles. RS + DPO (Llama) compresses representations broadly, with both policy and factual categories converging toward the alignment direction. DPO + GRPO (Qwen) operates more selectively—policy categories converge while factual categories actually *diverge*. These distinct profiles suggest that understanding the representational effects of different alignment methods could inform alignment method choice. When factual preservation is prioritized, methods with more selective steering profiles may be preferable.

3. Steering analysis reveals structure that probe accuracy misses. With prompt-controlled data, probe accuracy differences between base and instruct are modest (0.82–0.86 AUC for both). But steering analysis and probe transfer reveal clear asymmetry in how categories are affected. This methodological finding suggests that probe accuracy alone is an incomplete measure of representational change. We recommend using directional methods (steering vector analysis, probe transfer) alongside accuracy-based metrics for characterizing alignment effects.

6.2 Methodological Contribution

Our work extends D-STEER's characterization of fine-tuning as low-rank steering. While D-STEER showed that DPO modifies a narrow subspace, we identify *what* gets steered within that subspace: policy categories move further along the alignment direction than factual categories. We further show that different training methods (RS + DPO vs DPO + GRPO vs SFT-only) create distinguishable steering profiles, providing a framework for characterizing alignment methods by their representational effects.

6.3 Limitations

Phrase-based evaluation. Our correctness labels rely on phrase matching rather than human annotation. While we iteratively refined phrase lists to capture valid response patterns, error rates may be non-negligible. Extending to new model families would require identifying their characteristic refusal and ambiguity-acknowledgment styles.

Correlation, not causation. We observe that fine-tuning correlates with selective steering, but we cannot isolate specific training components. Models differ in multiple ways (SFT data, preference optimization method, training scale), making causal claims difficult.

Limited model scale. All models studied are in the 6–8B parameter range. Larger models may exhibit different patterns of steering or preservation.

Cannot isolate training components. While we compare SFT-only versus preference-optimized models, we cannot isolate the effects of specific training choices (SFT data composition, preference pair selection, optimization hyperparameters). The training method comparison should be interpreted as characterizing broad patterns rather than precise causal effects.

Uncertain-incorrect ambiguity. The uncertain-incorrect category (common misconceptions) is excluded from the policy-factual comparison because it requires both knowledge recall and behavioral acknowledgment. Its intermediate nature limits the cleanness of our binary grouping.

6.4 Future Work

Several directions would strengthen or extend these findings:

1. **Steering-aware fine-tuning**: Design objectives that preserve factual geometry while allowing policy steering. This could test whether the selective effect is avoidable.

2. **Intermediate checkpoint analysis**: Analyze how steering develops across training steps to identify when policy representations begin moving further than factual ones.

3. **Larger models**: Test whether 70B+ models show the same patterns or if scale provides different steering dynamics.

4. **Category-specific subspace analysis**: Our ablation results suggest steering may occur in category-specific subspaces rather than the mean alignment direction. More targeted analysis could identify these subspaces.

5. **SFT-induced categorical structure**: Our SFT-only models (Mistral, Yi) show divergence rather than convergence, with Mistral showing dramatically increased separation between policy and factual representations. This suggests SFT may drive models toward more categorical reasoning structures. Investigating this effect—whether it reflects explicit category formation or a byproduct of imitation learning—could inform how different training objectives shape internal representations.

7 Conclusion

We have shown that fine-tuning selectively steers policy representations along low-rank alignment directions. Across four model families with different architectures, training data, and fine-tuning methods, we find a consistent pattern: policy categories—where fine-tuning trains specific epistemic behaviors—move 1.08–1.28× further along the alignment direction than factual categories (all $p < 0.001$). Low-rank structure is confirmed across all models (14–19 dimensions capture 80% of alignment variance), though the relationship between category type and SVD loading is model-dependent rather than universal.

Probe transfer experiments confirm this selective effect: training probes on base models and testing on instruction-tuned variants reveals factual representations transfer at 0.85–0.87 accuracy while policy representations transfer at only 0.62–0.63 for preference-optimized models. The geometry of factual knowledge is preserved while the geometry of trained epistemic behaviors is reorganized.

Training method comparison reveals distinct profiles: RS + DPO (Llama) compresses representations broadly and concentrates policy in top SVD components, while DPO + GRPO (Qwen) operates more selectively with inverted SVD loading. Both methods use DPO but differ in additional techniques (rejection sampling vs. GRPO), suggesting these create distinguishable representational profiles. Notably, SFT-only models (Mistral, Yi) show divergence rather than convergence, with Mistral exhibiting dramatically increased separation between categories—suggesting SFT may drive models toward more categorical structures, a pattern worth further investigation.

Our cross-architecture validation suggests selective steering is not an artifact of particular models but a characteristic property of how current preference-optimization techniques interact with epistemic representations. These findings provide a framework for characterizing how different alignment methods affect internal representations, complementing behavioral evaluations with representational analysis.

References

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.findings-emnlp.68/.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know, 2022. URL `https://arxiv.org/abs/2207.05221`.

Neel Nanda and Joseph Bloom. Transformerlens. `https://github.com/TransformerLensOrg/TransformerLens`, 2022. Software library for mechanistic interpretability.

Samarth Raina, Amitava Das, Tanmoy Chakraborty, and Niloy Ganguly. D-STEER: Preference alignment techniques learn to behave, not to believe – beneath the surface, DPO as steering vector perturbation in activation space, 2024. URL `https://arxiv.org/abs/2512.11838`.

A Evaluation Logic Details

We evaluate model responses using category-specific phrase matching. For each category, we define a set of phrases that indicate a correct response. A response is marked correct if it contains any phrase from the relevant set (case-insensitive substring matching).

Factual categories. For **confident-correct** and **uncertain-correct**, correctness requires the response to contain the expected answer. We use case-insensitive substring matching with support for alternative answers separated by "/" (e.g., "carbon dioxide/CO2").

Confident-incorrect (53 phrases). The model should acknowledge that the fictional entity does not exist. Representative phrases:

- **Existence denials**: "doesn't exist", "does not exist", "fictional", "not real", "made up", "no such", "isn't real", "not a real", "fabricated", "imaginary", "hypothetical"
- **Uncertainty expressions**: "i don't know", "i'm not sure", "cannot find", "no information", "don't have enough information", "not aware of", "unable to find"
- **Geographic non-existence**: "no country", "not a country", "has no capital", "not a recognized country"
- **Specificity denials**: "not specified", "is unknown", "no reliable source", "cannot be determined"

Ambiguous (39 phrases). The model should request clarification or acknowledge multiple meanings. Representative phrases:

- **Clarification requests**: "could you clarify", "what do you mean", "need more context", "which", "what specifically", "without more context"
- **Multiple meanings**: "can mean", "can refer to", "could mean", "multiple meanings", "different meanings", "several definitions"
- **Dependency acknowledgment**: "depends on", "it depends", "depending on"
- **Inability to answer**: "cannot determine", "unable to answer", "impossible to answer", "cannot be answered without"

Nonsensical (26 phrases). The model should recognize the category error. Representative phrases:

- **Sense denials**: “doesn’t make sense”, “does not make sense”, “meaningless”, “nonsensical”
- **Property denials**: “has no mass”, “has no weight”, “has no color”, “has no taste”, “does not have a”
- **Category recognition**: “is an emotion”, “is an abstract”, “is a concept”, “is a day of the week”, “is not a physical object”
- **Measurement impossibility**: “cannot be weighed”, “cannot be measured”, “can’t be tasted”

Uncertain-incorrect (12 phrases). The model should debunk the misconception. Representative phrases:

- “myth”, “misconception”, “not true”, “false”, “incorrect”, “actually”, “contrary to”, “debunked”, “no evidence”

Limitations. This phrase-based approach enables automated evaluation at scale but has limitations: (1) it may miss valid responses that use different phrasing; (2) it may incorrectly accept responses that contain a phrase but in a different context; (3) extending to new model families requires identifying their characteristic response styles. We iteratively expanded phrase lists by examining false negatives across all eight models, but error rates may be non-negligible.

B Model Details

Table 9: Model specifications for reproducibility. All models are in the 6–8B parameter range with 28–32 transformer layers, accessed via HuggingFace Hub.

Family	Variant	HuggingFace Model ID	Layers	Hidden
Qwen 2.5	Base	`Qwen/Qwen2.5-7B`	28	3584
Qwen 2.5	Instruct	`Qwen/Qwen2.5-7B-Instruct`	28	3584
Llama 3.1	Base	`meta-llama/Llama-3.1-8B`	32	4096
Llama 3.1	Instruct	`meta-llama/Llama-3.1-8B-Instruct`	32	4096
Mistral v0.3	Base	`mistralai/Mistral-7B-v0.1`	32	4096
Mistral v0.3	Instruct	`mistralai/Mistral-7B-Instruct-v0.1`	32	4096
Yi 1.5	Base	`01-ai/Yi-6B`	32	4096
Yi 1.5	Instruct	`01-ai/Yi-6B-Chat`	32	4096

Generation parameters. All models used identical generation parameters:

- **Max new tokens**: 30 (sufficient for short factual answers)
- **Temperature**: 0 (greedy decoding for determinism)
- **Random seed**: 42

Activation extraction. We use TransformerLens [Nanda and Bloom, 2022] to extract activations from the residual stream at all layers. For probing, we use activations at the final token position (the last token of the prompt before generation begins) and concatenate across all layers to form a single feature vector $\mathbf{h} \in \mathbb{R}^{L \times d}$ where L is the number of layers and d is the hidden dimension.

Prompt format. Both base and instruction-tuned models receive identical prompts:

```
Question: {prompt}
Answer:
```

This prompt-controlled design ensures observed differences are attributable to model fine-tuning rather than prompt structure differences.

C Additional Results

C.1 Token Position Analysis

We extracted activations at three token positions: first (beginning of prompt), middle (sequence midpoint), and last (final token before generation). The last token position consistently provided the best probe performance across all models, which aligns with prior work showing that later positions aggregate more semantic information. All main results use last-token activations.

C.2 Per-Category Confidence Intervals

Table 10: Probe error rates (1 − accuracy) by category with 95% bootstrap CIs (2,000 iterations, percentile method). Error rates computed via 5-fold cross-validation. Policy categories show larger error increases after fine-tuning for preference-optimized models.

		Policy Error		**Factual Error**	
Model	**Variant**	**Mean**	**95% CI**	**Mean**	**95% CI**
Qwen 2.5	Base	0.12	[0.08, 0.16]	0.08	[0.05, 0.11]
	Instruct	0.19	[0.14, 0.24]	0.10	[0.07, 0.14]
Llama 3.1	Base	0.10	[0.06, 0.14]	0.09	[0.06, 0.12]
	Instruct	0.18	[0.13, 0.23]	0.12	[0.08, 0.16]
Mistral	Base	0.08	[0.05, 0.12]	0.06	[0.03, 0.09]
	Instruct	0.15	[0.10, 0.20]	0.14	[0.10, 0.19]
Yi 1.5	Base	0.11	[0.07, 0.15]	0.10	[0.06, 0.14]
	Instruct	0.16	[0.11, 0.21]	0.13	[0.09, 0.18]

For preference-optimized models (Qwen, Llama), policy error increases more than factual error after fine-tuning: Qwen shows +0.07 policy vs +0.02 factual; Llama shows +0.08 policy vs +0.03 factual. SFT-only models show more uniform changes.

C.3 Effect Sizes

Effect sizes decrease after fine-tuning for both category types, but the decrease is larger for policy categories. This aligns with the steering analysis: policy representations are steered further along

Table 11: Cohen's d effect sizes for activation differences (correct vs incorrect) along the steering direction, computed separately for policy and factual categories. Larger values indicate greater separability.

Model	**Variant**	**Policy** d	**Factual** d
Qwen 2.5	Base	0.82	0.91
	Instruct	0.65	0.84
Llama 3.1	Base	0.89	0.95
	Instruct	0.71	0.88
Mistral	Base	0.94	0.97
	Instruct	0.76	0.82
Yi 1.5	Base	0.85	0.88
	Instruct	0.72	0.81

the alignment direction, which compresses differences between correct and incorrect responses within that category type.

D Statistical Details

D.1 Permutation Test Procedure

Our primary significance test uses a sample-level permutation test to compare steering ratios between policy and factual categories. The procedure:

Algorithm 1 Sample-Level Permutation Test for Steering Ratio

```
Input: Activations $\mathbf{X}$, category labels $c_i$, steering vector $\mathbf{v}$
Compute projections: $p_i = \mathbf{x}_i \cdot \mathbf{v}/\|\mathbf{v}\|$ for each sample
Compute observed ratio: $R_{\text{obs}} = \text{mean}(p_i : c_i \in \text{Policy})/\text{mean}(p_i : c_i \in \text{Factual})$
for $b = 1$ to $B$ do                                    ▷ $B$ = 10,000 permutations
    Randomly permute category labels $c_i$
    Compute permuted ratio $R_b$
end for
$p$-value = $\frac{1+\sum_{b=1}^{B} \mathbf{1}[R_b \geq R_{\text{obs}}]}{B+1}$
```

This sample-level approach provides substantially higher statistical power than category-level tests because it operates on $n \approx 300$ samples per group rather than 3–5 category means.

D.2 Bootstrap Confidence Intervals

We compute 95% confidence intervals using the percentile bootstrap method:

1. Draw B = 2,000 bootstrap samples with replacement from the original data
2. Compute the statistic of interest for each bootstrap sample
3. Report the 2.5th and 97.5th percentiles as the 95% CI bounds

For steering ratios, we bootstrap over samples while preserving category membership. For probe accuracy, we bootstrap over cross-validation folds.

D.3 Multiple Comparison Correction

When conducting per-category tests (5 categories × 4 models = 20 tests), we apply Benjamini-Hochberg FDR correction at $\alpha = 0.05$. This controls the expected false discovery rate rather than the family-wise error rate, providing a less conservative correction appropriate for exploratory analyses.

The BH procedure:

1. Sort p-values: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$
2. Find the largest k such that $p_{(k)} \leq \frac{k}{m}\alpha$
3. Reject hypotheses $H_{(1)}, \ldots, H_{(k)}$

D.4 Sample-Level vs Category-Level Testing

We chose sample-level testing as our primary approach because:

- **Power**: With $n \approx 300$ samples per group vs 3–5 categories, sample-level tests have dramatically higher power to detect real effects
- **Variance estimation**: Sample-level tests properly account for within-category variance, which category-level means obscure
- **Generalization**: Results reflect effects on individual prompts, not just category averages

We report category-level breakdowns for interpretability but base significance claims on sample-level tests.

D.5 Seed Sensitivity Analysis

To verify result stability, we computed all main metrics across 5 random seeds (42, 123, 456, 789, 1000) and required coefficient of variation (CV) < 5% for reported values:

$$\mathrm{CV} = \frac{\sigma_{\text{seeds}}}{\mu_{\text{seeds}}} \times 100\%$$

All steering ratios and probe transfer gaps met this threshold. The largest observed CV was 3.2% for the Yi steering ratio, indicating stable results across random initialization.