Aligning What LLMs Do and Say: Towards Self-Consistent Explanations

Anonymous ACL submission

Abstract

Large language models (LLMs) seem to offer an easy path to interpretability: just ask them to explain their answers. Yet the features driving an answer often differ from those emphasized in its explanation, meaning post-hoc rationales can misrepresent what actually shaped the model's output. We quantify this gap by comparing the feature-importance distributions of answers and their explanations. Prior analyses reveal such discrepancies, but large-scale study has been limited by the high computational cost of attribution methods. To address this, we introduce the Post-hoc Self-Consistency Bank (PSCB), a large-scale benchmark linking model decisions with diverse explanations and attribution vectors across datasets, methods, and model families. Using PSCB, we find that Spearman rank correlation provides a more reliable signal of alignment than cosine similarity. Building on this insight, we apply Direct Preference Optimization (DPO) to attribution-based preference data, improving alignment without degrading task accuracy. Our results pave the way toward scalable and faithful alignment between LLM decisions and their natural language explanations.[1]

1 Introduction

As LLMs are increasingly embedded in user-facing and decision-support systems, their outputs and accompanying explanations are often trusted by end users even when independent verification is impractical (Sun et al., 2024; Liu et al., 2023; Doshi-Velez and Kim, 2017). To foster such trust, LLMs are frequently prompted to produce natural language explanations (Madsen et al., 2024). Yet these post-hoc explanations often draw on different input features than those that determined the model's answer (Turpin et al., 2023; Randl et al., 2024; Hase et al., 2020), revealing a gap we term *self-consistency*. While faithfulness broadly concerns whether explanations reflect the model's decision-making process (Jacovi and Goldberg, 2020; Wiegreffe et al., 2020), prior work suggests that many proposed faithfulness metrics in fact capture forms of self-consistency, agreement between the factors influencing a model's prediction and those invoked in its explanation (Parcalabescu and Frank, 2024). We operationalize this notion through feature-attribution alignment, measuring whether the explanation relies on the same input evidence that influenced the model's answer. Our work focuses on quantifying and improving this property for post-hoc explanations generated after the answer (see Figure 1).

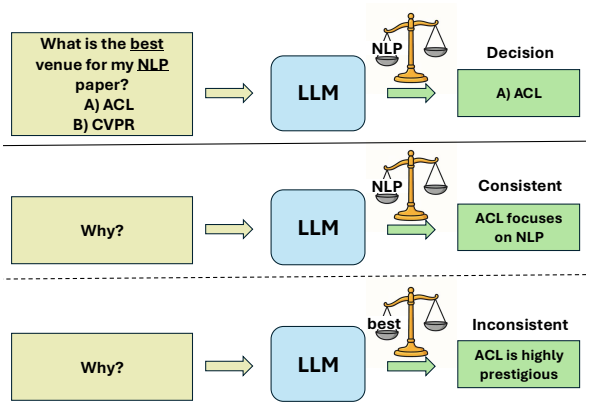


Figure 1: Illustration of explanation consistency using feature importance. **Top:** The LLM generates an answer where the word *NLP* in the prompt has high feature importance. **Middle:** A consistent explanation where *NLP* in the prompt also has high feature importance. **Bottom:** An inconsistent explanation, where the word *best* in the prompt has high feature importance instead, misaligned with the model's actual decision.

Prior works apply counterfactual interventions to input features using various heuristics to estimate their influence on model decisions (Wiegreffe et al., 2020; Turpin et al., 2023; Lanham et al., 2023; Atanasova et al., 2023). However, applying

[1] Code and data will be released upon acceptance.

such interventions to LLMs is computationally expensive. Parcalabescu and Frank (2024) recently proposed a more rigorous approach based on feature importance, where self-consistency is defined as the similarity between the importance assigned to the answer and to its explanation. Yet, computing feature importance itself is resource-intensive, and their evaluation was limited to only 100 test examples, constraining the conclusions that can be drawn. To the best of our knowledge, no prior work has demonstrated how to improve this alignment, as even measuring it remains highly expensive.

We address this gap by introducing new benchmarks, metrics, and methods to enhance LLMs' self-consistency. Our contributions are as follows: (1) **Post-hoc Self-Consistency Bank (PSCB):** a large-scale benchmark for evaluating natural language explanation (NLE) self-consistency in LLMs. PSCB contains over 85,000 model decisions paired with 428,000 explanations across four multiple-choice QA datasets and two instruction-tuned LLMs, with attribution vectors computed using LIME and Layer Integrated Gradients (LIG). This enables systematic large-scale evaluation of explanation–decision alignment. (2) **Empirical analysis of self-consistency:** We conduct the first large-scale study of attribution-based self-consistency in LLMs and find that it is largely orthogonal to answer correctness. We show that Spearman rank correlation provides a stronger and more discriminative signal than cosine similarity, effectively separating high- and low-quality explanations. (3) **Improving self-consistency via preference supervision:** We demonstrate that attribution-based preference data automatically derived from PSCB can be used to fine-tune LLMs with Direct Preference Optimization (DPO), yielding consistent and substantial in-domain improvements and robust cross-domain generalization without degrading task performance.

2 Background and Related Work

Feature attribution methods estimate how input components (e.g., tokens) or internal elements (e.g., attention heads) influence model predictions (Zhao et al., 2023). *Attention-based* methods are simple but often unreliable (Jain and Wallace, 2019; Serrano and Smith, 2019; Wiegreffe and Pinter, 2019), whereas *gradient-* and *perturbation-based* methods provide stronger signals, including Integrated Gradients (Sundararajan et al., 2017), LRP (Bach et al., 2015; Montavon et al., 2019), SHAP (Lundberg and Lee, 2017), and LIME (Ribeiro et al., 2016). SHAP is rigorous but slow; LIME trades some precision for efficiency.

Despite advances in robustness and faithfulness (Parcalabescu and Frank, 2024; Atanasova et al., 2023), scalable, model-agnostic attribution remains challenging. LLMs often produce fluent yet unfaithful explanations (Narang et al., 2020; Turpin et al., 2023; Madsen et al., 2024), motivating *self-consistency*, alignment between features driving predictions and explanations. Perturbation-based tests evaluate robustness, while attribution-based methods compare importance vectors directly. Prior work such as CC-SHAP (Parcalabescu and Frank, 2024) handles only ∼100 samples; our Post-hoc Self-Consistency Bank (PSCB) scales this to tens of thousands via a LIME-based variant (CC-LIME). We also explore Layer-Integrated Gradients, a gradient-based method with cost between LIME and SHAP. Complementary efforts enhance explanation fidelity via probability-aware metrics (Siegel et al., 2024), context-faithful prompting (Zhou et al., 2023), or cross-example semantic coherence (Chen et al., 2024). In contrast, we focus on *attributional* self-consistency within each decision, enabling scalable, data-driven improvements in explanation fidelity.

3 Post-hoc Self-Consistency Bank (PSCB)

We present the *Post-hoc Self-Consistency Bank* (PSCB), a collection of attribution-augmented QA datasets for evaluating decision–explanation alignment. Each dataset is defined by three components: a base QA dataset, a target LLM, and an alignment metric. We describe our construction choices and report key statistics and findings. The pipeline (Figure 2) is agnostic to the choice of the attribution method and alignment metric.

3.1 Sequence Feature Attribution

We compute token-level attribution scores to analyze how input tokens influence a model's generation. Given an input $\mathbf{x} = [x_1, \ldots, x_m]$ and an output $\mathbf{y} = [y_1, \ldots, y_n]$, the model generates $\mathbf{y}$ autoregressively with logits $\ell_t \in \mathbb{R}^V$ at each step t. We define the *sequence log-probability* as $\mathrm{SLP}(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \log P(y_t)$, where $P(y_t) = \frac{\exp(\ell_t[y_t])}{\sum_{v \in V} \exp(\ell_t[v])}$. This scalar serves as the attribution target, indicating how much each token in $\mathbf{x}$ contributes to generating $\mathbf{y}$. For each out-

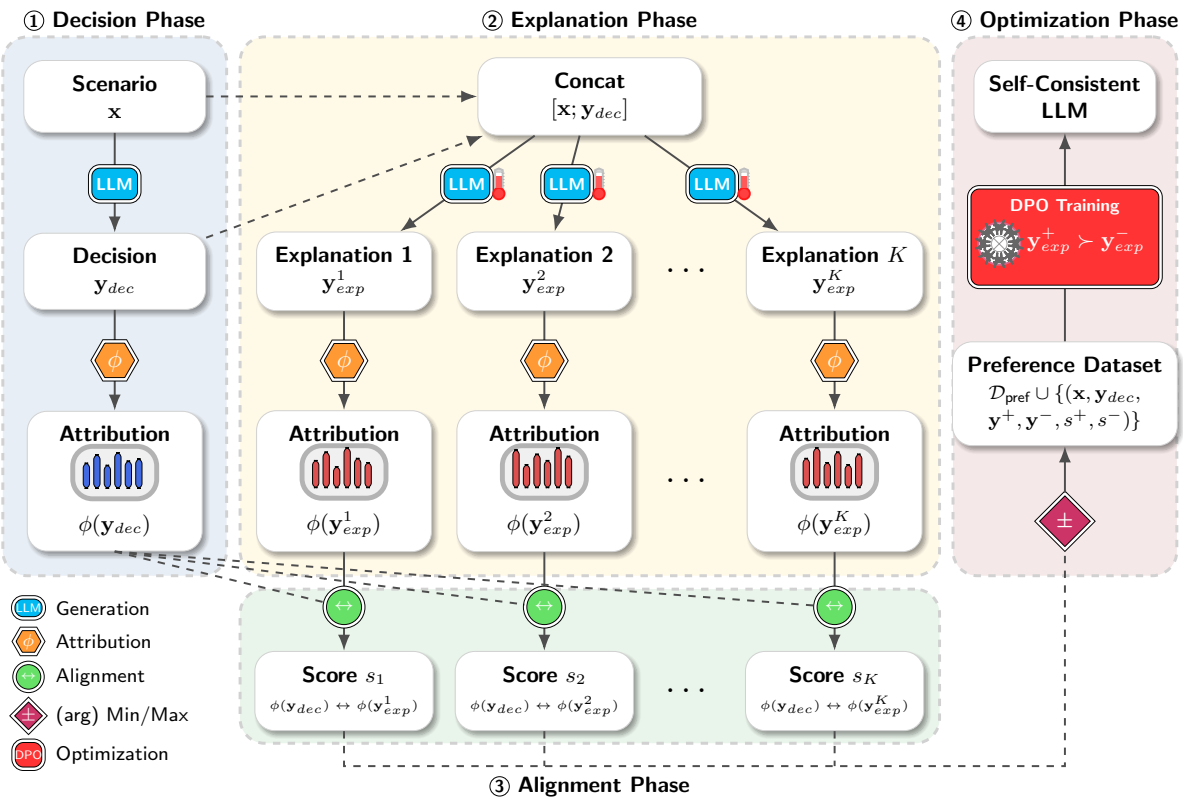


Figure 2: Overview of the PSCB pipeline. ① **Decision:** Given a multiple-choice QA instance, the LLM generates an answer together with attribution scores over the input. ② **Explanation:** The model is then prompted to produce K diverse explanations (via temperature sampling) conditioned on the question and answer, with attribution scores computed for each explanation. ③ **Alignment:** Self-consistency scores are obtained by comparing attribution vectors from the decision and explanations. ④ **Optimization:** Finally, attribution-based preference pairs are used to fine-tune the model DPO, yielding more self-consistent explanations.

put $\mathbf{y}$, we compute an attribution vector $\boldsymbol{\phi}^{(\mathbf{y})} = [\phi_1^{(\mathbf{y})}, \ldots, \phi_m^{(\mathbf{y})}]$, where $\phi_i^{(\mathbf{y})}$ quantifies the contribution of x_i. The attribution method, gradient-based, perturbation-based, or sampling-based, determines how $\boldsymbol{\phi}^{(\mathbf{y})}$ is computed; our framework is method-agnostic (Section 3.4). To reduce noise, we exclude a set of *skip tokens* $\mathcal{S} \subseteq V$ (e.g., punctuation, formatting markers), ensuring attribution focuses on semantically meaningful inputs.

3.2 Measuring Self-Consistency

Building on the attribution vectors defined above, we evaluate whether an explanation reflects the reasoning behind a decision. For an input $\mathbf{x}$ with decision $\mathbf{y}_{\text{dec}}$ and explanation $\mathbf{y}_{\text{exp}}$, let $\boldsymbol{\phi}^{(\mathbf{y}_{\text{dec}})}$ and $\boldsymbol{\phi}^{(\mathbf{y}_{\text{exp}})}$ denote their respective attributions. Self-consistency is measured by an alignment function

$$\alpha\left(\boldsymbol{\phi}^{(\mathbf{y}_{\text{dec}})}, \boldsymbol{\phi}^{(\mathbf{y}_{\text{exp}})}\right) : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R},$$

where the choice of α determines the aspect of the alignment to be evaluated (e.g. magnitude, rank, or overlap). Specific metrics are detailed in the following subsections.

3.3 Constructing Attribution-Based Preference Data

Building on attribution vectors (Section 3.1) and alignment measures (Section 3.2), we construct a dataset that pairs model outputs with their attributions, allowing systematic comparison between a model's decision rationale and its post-hoc justifications. Each instance is represented as a quintuple

$$\left(\mathbf{x},\ \mathbf{y}_{\text{dec}},\ \{\mathbf{y}_{\text{exp}}^{(i)}\}_{i=1}^k,\ \boldsymbol{\phi}^{(\mathbf{y}_{\text{dec}})},\ \{\boldsymbol{\phi}^{(\mathbf{y}_{\text{exp}}^{(i)})}\}_{i=1}^k\right),$$

where $\mathbf{x}$ is the input, $\mathbf{y}_{\text{dec}}$ the predicted answer, and $\{\mathbf{y}_{\text{exp}}^{(i)}\}_{i=1}^k$ are diverse explanations sampled in a zero-shot setting by conditioning on $(\mathbf{x}, \mathbf{y}_{\text{dec}})$. The corresponding attribution vectors $\boldsymbol{\phi}$ are computed as in Section 3.1, with non-semantic tokens excluded. We then score each explanation by its alignment α with the decision (Section 3.2), and retain the highest- and lowest-scoring cases as $\mathbf{y}_{\text{exp}}^{\text{chosen}}$

and $\mathbf{y}_{\text{exp}}^{\text{rejected}}$. These preference pairs form the core of the dataset used for optimization (Section 4).

3.4 Experimental Setup

Models. We evaluated two instruction-tuned LLaMA models, LLaMA-3.1-8B-Instruct and LLaMA-3.2-3B-Instruct (Touvron et al., 2023), in a zero-shot setting. Each was used in both the original Meta release and the UNSLOTH-optimized implementation, which share identical weights but differ slightly in decoding behavior, leading to minor accuracy variations. A padding token was added for compatibility with generation and attribution.

Datasets. Experiments cover four multiple-choice QA datasets: ECQA (Aggarwal et al., 2021), ARC-Easy, ARC-Challenge (Clark et al., 2018), and CODAH (Chen et al., 2019), spanning diverse reasoning domains. Each dataset is converted into an attribution-enhanced format (Section 3.3) and split into train/validation/test (70%/20%/10%) with a fixed seed. Training instances are used to sample $k = 5$ explanations and construct preference pairs, validation is used for model selection, and testing for final evaluation.

Prompting and Generation. Decisions are elicited using a minimal task instruction (e.g., "Choose the most plausible answer:"), and explanations are generated as in Section 3.3, conditioned on $(\mathbf{x}, \mathbf{y}_{\text{dec}})$. Explanations are sampled with nucleus decoding ($p = 0.9, T = 0.7$) and a maximum length of 400 tokens. The complete prompt templates are provided in Appendix B.

Feature Attribution. We computed attributions using Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro et al., 2016) and Layer Integrated Gradients (LIG) (Sundararajan et al., 2017) as implemented in Captum (Kokhlikyan et al., 2020), with sequence log-probability as the target. LIME used 500 perturbation samples with the padding token as baseline, and LIG used an embedding baseline with 25 interpolation steps. We applied the UNSLOTH variants for LIME computations and the original Meta models for LIG to ensure compatibility. Skip tokens were filtered as listed in Appendix D.

Consistency Metrics. We evaluated self-consistency with two alignment functions. *Cosine similarity* measures directional overlap between the attribution vectors, assessing whether explanations emphasize the same features with a similar magnitude. *Spearman rank correlation* captures agreement in feature prioritization while abstracting from the attribution scale:

$$\text{CC}_{\text{sp}} = 1 - \frac{6 \sum_{i=1}^{m} \left(\text{r}\left(\phi_i^{(\mathbf{y}_{\text{dec}})}\right) - \text{r}\left(\phi_i^{(\mathbf{y}_{\text{exp}})}\right) \right)^2}{m(m^2 - 1)},$$

where $\text{r}(\cdot)$ assigns the ordinal ranks (ties are decided by average). Cosine reflects directional similarity, while Spearman reflects ranking similarity. High values indicate that explanations highlight features aligned with the decision, suggesting greater self-consistency.

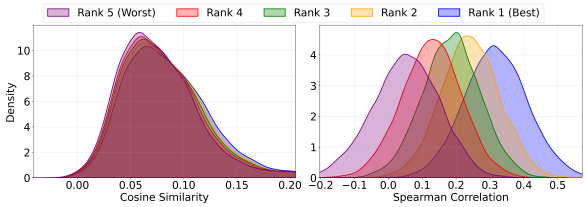


Figure 3: Smoothed distribution of self-consistency scores by explanation rank for the LLaMA3.1-8B model on ECQA. **Spearman correlation (right)** shows clear rank separation, reflecting strong sensitivity to explanation quality. **Cosine similarity (left)** shows substantial overlap across ranks, indicating weaker differentiation.

3.5 Key Findings and Insights

We evaluate self-consistency across four MCQA datasets and two instruction-tuned LLaMA models. For each model–dataset pair, we report decision accuracy and compute self-consistency using five sampled explanations per input. Attributions are derived with both LIME and LIG, and alignment is measured using cosine similarity (CC-COS) and Spearman rank correlation (CC-SP). For each setting, we report the *worst*, *mean*, and *best* alignment scores to capture the range of consistency across explanations. Comprehensive numerical results are provided in Appendix A.

Explanation variability and metric differences. We observe substantial gaps between the best and worst explanations for a given input. For example, in ECQA with LLaMA-3.1-8B (LIME attributions), Spearman scores range from 5.07 to 31.76. Across datasets and models, Spearman correlations are consistently higher than cosine similarities, indicating that models preserve feature *ranking* more reliably than attribution magnitudes. Spearman also exhibits greater spread across explanations, making it more effective in detecting consistency differences, whereas cosine tends to produce compressed

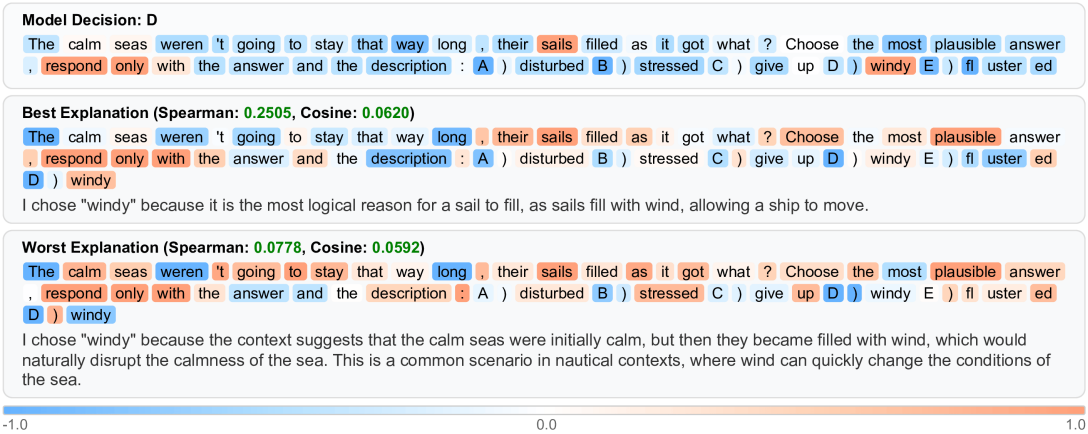


Figure 4: An example from the ECQA dataset shows attribution alignment for the LLaMA3.2-3B model's decision and its best ($Sp = 0.25$) and worst ($Sp = 0.08$) explanations. The model selects answer "D" ("windy"), with attribution heatmaps highlighting word-level influences (blue = negative, orange = positive). Key tokens like "sails" and "windy" drive the decision. The best explanation mirrors this focus, correctly reasoning that wind fills sails, while the worst shifts emphasis to "calm seas," misaligning with the core mechanism.

values due to its sensitivity to attribution scale. This supports the need for ranking-based evaluation and highlights the opportunity to identify highly self-consistent explanations among diverse generations (Figure 3). Figure 4 further shows that a large Spearman variance corresponds to meaningful semantic differences in explanation quality.

Correctness and self-consistency. We test whether attribution-based self-consistency correlates with prediction correctness. Using LIME attributions, we compare alignment scores for correct vs. incorrect predictions (Table 1). Differences are small and inconsistent across datasets and models: for example, in ECQA with LLaMA-3.1-8B, Spearman is slightly higher when correct (0.186 vs. 0.182), whereas in ARC-Easy with LLaMA-3.2-3B it is slightly higher when incorrect. Overall, we find no association between correctness and alignment (for either cosine or Spearman), suggesting that self-consistency is largely orthogonal to accuracy.

	Model	CC-Cos (T / F / Δ)	CC-Sp (T / F / Δ)
ECQA	LLaMA3.1-8B	8.1 / 8.4 / **−0.3**	18.6 / 18.2 / **+0.4**
	LLaMA3.2-3B	1.6 / 1.7 / **−0.1**	22.3 / 22.3 / **0.0**
ARC-E	LLaMA3.1-8B	14.1 / 14.8 / **−0.7**	12.4 / 13.0 / **−0.6**
	LLaMA3.2-3B	5.7 / 6.6 / **−0.9**	19.8 / 19.3 / **+0.5**
ARC-C	LLaMA3.1-8B	16.1 / 16.1 / **0.0**	11.5 / 11.5 / **0.0**
	LLaMA3.2-3B	8.6 / 7.6 / **+1.0**	18.5 / 18.4 / **+0.1**
Coda-H	LLaMA3.1-8B	13.7 / 13.4 / **+0.3**	19.8 / 20.9 / **−1.1**
	LLaMA3.2-3B	8.2 / 7.5 / **+0.7**	19.2 / 19.9 / **−0.7**

Table 1: Self-consistency scores (×100) for true (T) and false (F) predictions. We denote $\Delta = T - F$. Cosine similarity and Spearman rank correlation are shown. Differences are small and vary in direction.

4 Improving Self-Consistency with DPO

To directly optimize for self-consistency, we fine-tune language models using Direct Preference Optimization (DPO) (Rafailov et al., 2023), which aligns outputs with preference signals through relative comparisons.

Preference Pair Construction. For each input $\mathbf{x}$, we construct preference pairs by selecting a preferred explanation $\mathbf{y}_{\text{exp}}^{\text{chosen}}$ and a dispreferred explanation $\mathbf{y}_{\text{exp}}^{\text{rejected}}$ based on their attribution alignment with the model's decision $\mathbf{y}_{\text{dec}}$ (Section 3.2). We rank five candidate explanations using Spearman rank correlation between their attribution vectors and the decision's attribution vector, then select the highest and lowest ranked as the preference pair.

Alignment Metric. We employ Spearman rank correlation as our primary alignment metric for DPO training over previously used cosine similarity. Spearman correlation captures feature prioritization rather than magnitude alignment. As demonstrated in Section 3.5 (Figure 3), Spearman correlation exhibits greater score variability across explanation candidates. This broader dynamic range provides a more informative training signal, enabling a clearer differentiation between high-quality and low-quality explanations.

			LLaMA3.1-8B			LLaMA3.2-3B		
			B-B	B-T (↑↓)	T-T (↑↓)	B-B	B-T (↑↓)	T-T (↑↓)
ECQA		Acc.	70.34	70.34	69.70 ↓0.9%	67.13	67.13	66.12 ↓1.5%
ECQA	LIME CC-Cos	Worst	07.58 ±0.11	07.64 ±0.12 ↑0.8%	**09.16** ±0.12 ↑20.8%	01.11 ±0.14	01.30 ±0.13 ↑17.1%	**02.14** ±0.13 ↑92.8%
ECQA	LIME CC-Cos	Mean	08.05 ±0.12	08.10 ±0.12 ↑0.6%	**09.72** ±0.12 ↑20.7%	01.46 ±0.14	01.63 ±0.13 ↑11.6%	**02.47** ±0.14 ↑69.2%
ECQA	LIME CC-Cos	Best	08.53 ±0.13	08.57 ±0.13 ↑0.5%	**10.28** ±0.13 ↑20.5%	01.81 ±0.14	01.96 ±0.13 ↑8.3%	**02.80** ±0.14 ↑54.7%
ECQA	LIME CC-Sp	Worst	05.00 ±0.30	07.20 ±0.30 ↑44.0%	**07.86** ±0.29 ↑57.2%	10.04 ±0.31	10.50 ±0.32 ↑4.6%	**12.94** ±0.32 ↑28.9%
ECQA	LIME CC-Sp	Mean	18.22 ±0.23	20.22 ±0.23 ↑11.0%	**20.58** ±0.24 ↑13.0%	22.52 ±0.26	22.65 ±0.27 ↑0.6%	**24.71** ±0.27 ↑9.7%
ECQA	LIME CC-Sp	Best	31.30 ±0.28	32.86 ±0.29 ↑5.0%	**33.45** ±0.28 ↑6.9%	35.02 ±0.30	34.81 ±0.30 ↓0.6%	**36.50** ±0.29 ↑4.2%
ECQA		Acc.	67.03	67.03	67.68 ↑1.1%	62.44	62.44	62.99 ↑0.9%
ECQA	LIG CC-Cos	Worst	0.87 ± 0.03	01.35 ± 0.02 ↑55.2%	**01.36** ± 0.02 ↑56.3%	01.90 ± 0.03	01.97 ± 0.03 ↑3.7%	**02.54** ± 0.03 ↑33.7%
ECQA	LIG CC-Cos	Mean	01.22 ± 0.03	01.67 ± 0.02 ↑36.9%	**01.67** ± 0.02 ↑36.9%	02.32 ± 0.03	02.37 ± 0.03 ↑2.2%	**02.82** ± 0.03 ↑21.6%
ECQA	LIG CC-Cos	Best	01.55 ± 0.03	01.96 ± 0.03 ↑26.5%	**01.96** ± 0.02 ↑26.5%	02.72 ± 0.03	02.73 ± 0.04 ↑0.4%	**02.99** ± 0.03 ↑9.9%
ECQA	LIG CC-Sp	Worst	34.71 ± 0.44	34.71 ± 0.45	**36.59** ± 0.42 ↑5.4%	28.61 ± 0.45	29.04 ± 0.45 ↑1.5%	**33.30** ± 0.45 ↑16.4%
ECQA	LIG CC-Sp	Mean	42.52 ± 0.40	42.65 ± 0.41 ↑0.3%	**43.66** ± 0.39 ↑2.7%	36.72 ± 0.42	37.17 ± 0.41 ↑1.2%	**41.90** ± 0.41 ↑14.1%
ECQA	LIG CC-Sp	Best	49.87 ± 0.40	50.15 ± 0.40 ↑0.6%	**51.24** ± 0.38 ↑2.7%	44.53 ± 0.41	45.05 ± 0.41 ↑1.2%	**50.17** ± 0.41 ↑12.7%
ARC-Easy		Acc.	87.72	87.72	86.76 ↓1.1%	81.62	81.62	81.04 ↓0.7%
ARC-Easy	LIME CC-Cos	Worst	13.32 ±0.25	13.47 ±0.26 ↑1.1%	**14.97** ±0.24 ↑12.4%	05.58 ±0.32	05.58 ±0.30	**06.32** ±0.29 ↑13.2%
ARC-Easy	LIME CC-Cos	Mean	14.17 ±0.27	14.31 ±0.27 ↑1.0%	**15.89** ±0.25 ↑12.1%	06.13 ±0.33	06.11 ±0.31 ↓0.3%	**06.87** ±0.30 ↑12.1%
ARC-Easy	LIME CC-Cos	Best	15.03 ±0.28	15.15 ±0.29 ↑0.8%	**16.82** ±0.27 ↑11.9%	06.68 ±0.34	06.66 ±0.32 ↓2.9%	**07.42** ±0.31 ↑11.1%
ARC-Easy	LIME CC-Sp	Worst	-01.90 ±0.43	**01.84** ±0.44 ↑96.8%	01.76 ±0.44 ↑92.6%	07.97 ±0.45	08.99 ±0.44 ↑12.8%	**11.48** ±0.46 ↑44.0%
ARC-Easy	LIME CC-Sp	Mean	12.01 ±0.35	**15.01** ±0.38 ↑25.0%	14.88 ±0.37 ↑23.9%	20.72 ±0.39	21.53 ±0.39 ↑3.9%	**23.16** ±0.39 ↑11.8%
ARC-Easy	LIME CC-Sp	Best	25.72 ±0.43	**28.38** ±0.45 ↑10.3%	27.86 ±0.44 ↑8.3%	33.14 ±0.44	34.19 ±0.46 ↑3.2%	**34.80** ±0.41 ↑5.0%
ARC-Easy		Acc.	84.26	84.26	84.84 ↑0.7%	80.23	80.23	79.27 ↓1.2%
ARC-Easy	LIG CC-Cos	Worst	0.98 ± 0.05	**01.56** ± 0.04 ↑59.2%	01.54 ± 0.04 ↑57.1%	01.20 ± 0.04	01.25 ± 0.04 ↑4.2%	**01.71** ± 0.04 ↑42.5%
ARC-Easy	LIG CC-Cos	Mean	01.35 ± 0.05	**01.84** ± 0.04 ↑36.3%	01.82 ± 0.04 ↑34.8%	01.62 ± 0.04	01.66 ± 0.04 ↑2.5%	**02.07** ± 0.04 ↑27.8%
ARC-Easy	LIG CC-Cos	Best	01.68 ± 0.05	**02.11** ± 0.04 ↑25.6%	02.08 ± 0.04 ↑23.8%	02.01 ± 0.04	02.03 ± 0.04 ↑1.0%	**02.43** ± 0.04 ↑20.9%
ARC-Easy	LIG CC-Sp	Worst	37.40 ± 0.64	38.94 ± 0.66 ↑4.1%	**39.93** ± 0.65 ↑6.8%	33.77 ± 0.65	34.88 ± 0.67 ↑3.3%	**37.97** ± 0.65 ↑12.4%
ARC-Easy	LIG CC-Sp	Mean	45.55 ± 0.55	46.76 ± 0.57 ↑2.7%	**47.09** ± 0.58 ↑3.4%	43.15 ± 0.60	44.02 ± 0.61 ↑2.0%	**46.24** ± 0.60 ↑7.2%
ARC-Easy	LIG CC-Sp	Best	52.88 ± 0.52	53.63 ± 0.53 ↑1.4%	**53.79** ± 0.55 ↑1.7%	51.75 ± 0.60	52.52 ± 0.60 ↑1.5%	**53.74** ± 0.60 ↑3.8%

Table 2: In-Domain Performance: Models trained and evaluated on the same dataset (ECQA, ARC-Easy). Results are based on LIME and LIG attributions, reporting Worst, Mean, and Best self-consistency scores across 5 sampled explanations (×100). Accuracy is shown alongside. Each model is evaluated under three modes (B–B, B–T, T–T), with relative improvements over B–B shown in parentheses.

Training Procedure. We enable parameter-efficient adaptation using LoRA (Low-Rank Adaptation) (Hu et al., 2021), which inserts trainable low-rank matrices into the attention and feed-forward layers of the transformer. This allows updating only a small subset of parameters, greatly reducing compute and memory cost. All explanation generations use the same zero-shot prompting scheme as in Section 3.4 to ensure consistency between training and evaluation. Models are fine-tuned independently on their attribution-based preference datasets, with hyperparameters and LoRA configurations provided in Appendix C.

5 Empirical Evaluation

We assess the impact of DPO training across three Decider–Explainer configurations: B–B (Base Decider, Base Explainer), B–T (Base Decider, Tuned Explainer), and T–T (Tuned Decider, Tuned Explainer). The B–T setting is particularly noteworthy, as it evaluates whether the tuned model can function as a plug-in explainer while keeping the decision model unchanged.

5.1 In-Domain Evaluation

We begin with in-domain experiments, where models are fine-tuned and evaluated on the same dataset. This setting isolates the direct effect of DPO training on explanation self-consistency, and Table 2 reports task accuracy alongside self-consistency scores computed with both LIME and LIG in the three Decider-Explainer modes. Across both model sizes, DPO training consistently improves explanation quality in the **T–T mode**, with the strongest gains for the weakest explanations. On ECQA for LLaMA3.1-8B, Spearman correlation improves by 13% on mean and 57.2% on worst-case with LIME,

		Training Data	Acc.	CC-Cos (↑↓)		CC-Sp (↑↓)	
				B-T	T-T	B-T	T-T
CODAH	L3.1-8B	None	81.29	13.50 ± 0.28		19.46 ± 0.48	
		ECQA	77.70	13.39 ± 0.27 (↓ 0.8%)	14.05 ± 0.32 (↑ 4.1%)	18.30 ± 0.56 (↓ 6.0%)	21.58 ± 0.49 (↑ 10.9%)
		ARC-Easy	80.94	13.38 ± 0.27 (↓ 0.8%)	**14.29** ± 0.29 (↑ 5.9%)	18.79 ± 0.57 (↓ 3.4%)	**21.71** ± 0.50 (↑ 11.6%)
	L3.2-3B	None	73.90	08.04 ± 0.34		17.95 ± 0.57	
		ECQA	72.43	08.16 ± 0.35 (↑ 1.5%)	09.42 ± 0.31 (↑ 17.2%)	19.41 ± 0.54 (↑ 8.1%)	20.10 ± 0.55 (↑ 12.0%)
		ARC-Easy	74.63	08.02 ± 0.34 (↓ 0.2%)	**09.51** ± 0.31 (↑ 18.3%)	18.13 ± 0.57 (↑ 1.0%)	**20.52** ± 0.55 (↑ 14.3%)
ARC-Chal	L3.1-8B	None	76.15	16.26 ± 0.33		11.81 ± 0.45	
		ECQA	73.85	16.48 ± 0.33 (↑ 1.3%)	17.57 ± 0.32 (↑ 8.1%)	13.60 ± 0.49 (↑ 13.16%)	13.45 ± 0.49 (↑ 13.9%)
		ARC-Easy	73.46	16.49 ± 0.33 (↑ 1.4%)	**17.65** ± 0.31 (↑ 8.5%)	**13.87** ± 0.49 (↑ 17.4%)	13.31 ± 0.49 (↑ 12.7%)
	L3.2-3B	None	66.67	08.58 ± 0.46		19.51 ± 0.62	
		ECQA	66.67	08.46 ± 0.44 (↓ 1.4%)	**09.13** ± 0.41 (↑ 6.4%)	19.83 ± 0.62 (↑ 1.6%)	21.37 ± 0.59 (↑ 9.5%)
		ARC-Easy	67.06	08.49 ± 0.44 (↓ 1.0%)	09.08 ± 0.42 (↑ 5.8%)	19.83 ± 0.69 (↑ 1.6%)	**21.78** ± 0.59 (↑ 11.6%)

Table 3: Cross-Domain Performance: Models trained on source datasets (ECQA, ARC-Easy) and evaluated on target datasets (CODAH, ARC-Challenge). Results are based on LIME evaluations, showing mean Cosine similarity (CC-Cos) and mean Spearman correlation (CC-Sp) (×100), with relative improvements over base models in parentheses.

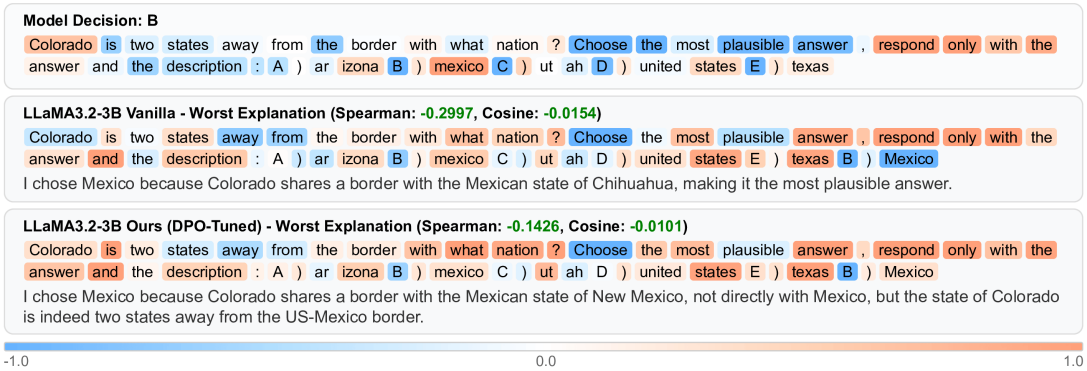


Figure 5: An example from ECQA shows attribution alignment for the LLaMA3.2-3B model's decision and its worst explanation across two variants. The model selects "B" (Mexico), with high-attribution tokens like "Colorado," "two," "states," and "Mexico" supporting its geographic reasoning. The DPO-tuned model generates a worst-case explanation that better reflects this rationale, noting Colorado's distance from Mexico, mirroring the decision attribution. In contrast, the vanilla model incorrectly references a border with "Chihuahua," unsupported by the decision attribution.

while LIG adds smaller gains of 2.7% and 5.4%. On ARC-Easy, LIME nearly doubles the worst-case score (+92.6%) and raises the mean by 23.9%, while LIG yields smaller but consistent improvements of 6.8% and 3.4%. The smaller LLaMA3.2-3B follows the same trend: on ECQA, LIME boosts mean and worst-case scores by 9.7% and 28.9%, while LIG raises them by 14.1% and 16.4%; on ARC-Easy, LIME increases the mean by 11.8% and worst-case by 44.0%, while LIG improves them by 7.2% and 12.4%. Results in the **B–T mode** also improve, though more moderately, reflecting the mismatch between a vanilla decider and a tuned explainer. The stronger gains in T–T highlight that self-consistency is highest when the same tuned model generates both decisions and explanations, aligning their attribution vectors. Across datasets and models, these improvements come with only minimal accuracy changes ($\leq 1.5\%$ in either direction), showing that DPO enhances self-consistency without harming predictive performance.

Cross-Metric Transfer. Although training is guided only by the Spearman correlation, we observe improvements in cosine similarity across both LIME and LIG. This cross-metric transfer suggests that optimizing for rank alignment not only stabilizes feature prioritization but also enhances directional attribution similarity. Notably, for LLaMA3.2-3B on ECQA, cosine similarity improvements (up to *92.8%*) exceed Spearman gains, indicating positive spillover effects. The consistency of these results across attribution methods further supports the robustness of our approach.

5.2 Cross-Domain Generalization

Table 3 reports cross-domain performance across evaluation modes. In the **T–T mode**, models fine-tuned on ECQA or ARC-Easy show consistent improvements in Spearman correlation. For LLaMA3.1-8B, gains range from 10.9–13.9%, with the strongest transfer from ARC-Easy to the more challenging ARC-Challenge (+13.9%). The smaller LLaMA3.2-3B also benefits, with gains between 9.5–14.3%. Cosine similarity shows a more moderate but still positive transfer: for LLaMA3.1-8B, improvements range from 4.1–6.4%, while for LLaMA3.2-3B, gains span 5.8–18.3%, including a notable improvement when transferring from ARC-Easy to CODAH. Complementary results in the **B–T mode** show weaker and sometimes mixed trends, indicating that the strongest gains arise when both the decision and explanation are generated by the tuned model in the T–T setting. In general, these findings suggest that our DPO-based approach encourages explanation strategies that generalize beyond the training domain.

5.3 Plausibility and Truthfulness Evaluation

To ensure that optimizing for self-consistency does not affect the naturalness or plausibility of explanations, we conducted a small-scale plausibility study (Appendix H). Two annotators evaluated 80 explanations across all models and datasets, judging (i) whether each was written in natural language and (ii) its plausibility on a 3-point scale. Nearly all explanations (98.8% for base and 100% for tuned models) were in natural language, with average plausibility scores of $2.92/3.0$ and $2.88/3.0$, respectively. These results indicate that DPO fine-tuning does not diminish plausibility or naturalness.

We further assessed whether improving self-consistency affects factual reliability by evaluating both the vanilla and DPO-tuned models on the TRUTHFULQA benchmark (Lin et al., 2021). This benchmark measures whether models produce factually correct answers to questions specifically designed to elicit false or misleading responses. Results (Appendix F) show comparable or slightly higher accuracy across both LLaMA variants, indicating that enhancing attributional self-consistency preserves overall truthfulness.

5.4 Cross-Method Consistency Analysis

We examine whether improving self-consistency under one attribution method transfers to others. Models fine-tuned using LIME-based alignment are evaluated with LIG-based metrics and vice versa, with additional evaluation using the sampling-based KSHAP (Lundberg and Lee, 2017) method. Results are reported in Appendix G. Across all settings, we observe minimal transfer: improvements largely remain within the training method, suggesting that each attribution approach captures a distinct notion of feature importance.

To understand this discrepancy, we further compare how LIME and LIG rank the same set of explanations across 10,880 ECQA examples. Their rankings exhibit only moderate agreement, rarely sharing the same top explanation (Top-1 ≈ 0.2) but often overlapping among the top three (Top-3 ≈ 0.6). This partial overlap indicates that perturbation-, gradient-, and sampling-based methods emphasize complementary aspects of model reasoning, a phenomenon also observed in prior work (Krishna et al., 2022; Neely et al., 2021, 2022). Together, these findings highlight that attribution-based self-consistency is not a single unified property but a multifaceted one, improving consistency under one definition does not guarantee improvements under another.

6 Conclusion

We introduce the *PSCB*, enabling the first large-scale study of how closely LLM explanations align with the evidence behind their decisions. Our analysis shows that *self-consistency* represents a distinct and interpretable aspect of model behavior, largely independent of answer correctness. Rank-based metrics such as Spearman correlation offer sharper and more stable signals for assessing this alignment. Leveraging these insights, we apply preference supervision to optimize attributional alignment, achieving consistent improvements in self-consistency across models, datasets, and attribution methods. The strongest gains appear when the same tuned model generates both decisions and explanations, indicating that shared representations enhance consistency. These improvements come without compromising task accuracy, natural language form, plausibility, or truthfulness. Overall, our results establish a scalable framework for quantifying and improving self-consistency, advancing language models whose explanations faithfully reflect the factors shaping their outputs and foster greater interpretability and trust.

Limitations

This work focuses on assessing and improving the self-consistency of natural language explanations given by LLMs. We report the following limitations:

Attribution Methods and Scalability. Attribution-based self-consistency analysis remains computationally demanding, which limits the range of methods that can be feasibly applied at scale. While theoretically principled approaches such as CC-SHAP (Parcalabescu and Frank, 2024) provide strong guarantees, they more than 4-minute per example, rendering large-scale benchmarking and preference-based training impractical. To balance rigor and scalability, our main experiments rely on the perturbation-based LIME method (Ribeiro et al., 2016), complemented by the gradient-based Layer Integrated Gradients (LIG) approach (Sundararajan et al., 2017). These choices enable consistent evaluation across tens of thousands of examples while still capturing complementary perspectives on feature importance.

Our cross-method study (Section 5.4) shows that attribution techniques often diverge in how they rank explanations, reflecting that each method encodes a distinct operational definition of input relevance. Consequently, improvements obtained under one attribution paradigm may not generalize to others, a limitation inherent to all attribution-based evaluations. This method dependence highlights both the difficulty of defining a single universal measure of explanation faithfulness and the value of our method as a scalable framework for systematically comparing and optimizing multiple attribution-based notions of self-consistency.

Human Evaluation. Self-consistency measures the internal alignment between a model's decision and its explanation through their attribution vectors. This property is inherently inaccessible to human annotators, as it requires knowledge of how the model distributes importance across input features, information even experts lack for large LLMs. Hence, our main evaluation remains attribution-based. However, to complement this internal analysis, we conducted a small-scale plausibility study in which human participants judged whether explanations produced by different models were expressed in natural language and rated their plausibility. While this provides an initial perspective on user-facing quality, evaluating whether higher self-consistency also leads to more *helpful* explanations from a user's perspective remains an open question. Future work may explore this direction through human studies to better understand the downstream utility of self-consistency for interpretability and trust.

Model and Task Scope. We focus exclusively on small to medium scale instruction-tuned LLMs in a multiple-choice question answering (MCQA) setup. This choice simplifies attribution computation and explanation prompting, but may limit generalization to other model families (e.g., base pretrained models, encoder-decoder architectures) and open-ended tasks (e.g., summarization, reasoning chains).

Offline vs. Online. To improve the LLM's self-consistency, we update it using DPO in an offline manner, creating a new model dedicated to providing consistent explanations. However, this approach introduces a trade-off: while the updated model generates more self-consistent explanations, it achieves only approximate self-consistency since it differs from the original model that generated the reference answers. Online learning could theoretically maintain perfect self-consistency by updating the model while continuously re-calculating attribution vectors for its own evolving outputs. However, this approach presents significant technical challenges, including the computational overhead of re-calculating the attributions at each training step and the potential instability in the learning dynamics, making offline training the more practical option despite its inherent approximation.

References

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for commonsenseqa: New dataset and models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065.

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. Faithfulness tests for natural language explanations. *arXiv preprint arXiv:2305.18029*.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations

for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Michael Chen, Mike D'Arcy, Alisa Liu, Jared Fernandez, and Doug Downey. 2019. CODAH: An adversarially-authored question answering dataset for common sense. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 63–69, Minneapolis, USA. Association for Computational Linguistics.

Yanda Chen, Chandan Singh, Xiaodong Liu, Simiao Zuo, Bin Yu, He He, and Jianfeng Gao. 2024. Towards consistent natural-language explanations via explanation-consistency finetuning. *arXiv preprint arXiv:2401.13986*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? *arXiv preprint arXiv:2010.04119*.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch. *ArXiv*, abs/2009.07896.

Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. 2022. The disagreement problem in explainable machine learning: A practitioner's perspective. *Trans. Mach. Learn. Res.*, 2024.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful? *arXiv preprint arXiv:2401.07927*.

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. 2019. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *arXiv preprint arXiv:2004.14546*.

Michael Neely, Stefan F. Schouten, Maurits J. R. Bleeker, and Ana Lucic. 2021. Order in the court: Explainable ai methods prone to disagreement. *ArXiv*, abs/2105.03287.

Michael Neely, Stefan F. Schouten, Maurits J. R. Bleeker, and Ana Lucic. 2022. A song of (dis)agreement: Evaluating the evaluation of explainable artificial intelligence in natural language processing. In *HHAI*.

Letitia Parcalabescu and Anette Frank. 2024. On measuring faithfulness or self-consistency of natural language explanations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6048–6089, Bangkok, Thailand. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290.

Korbinian Randl, John Pavlopoulos, Aron Henriksson, and Tony Lindgren. 2024. Evaluating the reliability of self-explanations in large language models. In *International Conference on Discovery Science*, pages 36–51. Springer.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? *arXiv preprint arXiv:1906.03731*.

Noah Y Siegel, Oana-Maria Camburu, Nicolas Heess, and Maria Perez-Ortiz. 2024. The probabilities also matter: A more faithful metric for faithfulness of free-text explanations in large language models. *arXiv preprint arXiv:2404.03189*.

Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, and 1 others. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*, 3.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Sarah Wiegreffe, Ana Marasović, and Noah A Smith. 2020. Measuring association between labels and free-text rationales. *arXiv preprint arXiv:2010.12762*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*.

Haiyan Zhao, Hanjie Chen, F. Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2023. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15:1 – 38.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. *arXiv preprint arXiv:2303.11315*.

A PSCB Results

			LLaMA3.1-8B		LLaMA3.2-3B	
			LIME	LIG	LIME	LIG
ECQA (#10882)	Acc.		71.11	66.51	65.85	64.83
	CC-Cos	Worst	07.70	01.36	01.30	01.82
		Mean	08.18	01.67	01.65	02.25
		Best	08.67	01.98	01.99	02.66
	CC-Sp	Worst	05.07	33.79	09.75	30.55
		Mean	18.47	41.70	22.28	38.72
		Best	31.76	49.11	34.64	46.54
ARC-E (#5197)	Acc.		87.01	82.87	81.90	80.69
	CC-Cos	Worst	13.31	01.49	05.32	01.65
		Mean	14.16	01.78	05.86	02.04
		Best	15.02	02.07	06.41	02.41
	CC-Sp	Worst	–01.06	38.81	07.26	36.63
		Mean	12.44	46.42	19.68	45.58
		Best	25.77	53.41	31.98	53.68
ARC-C (#2590)	Acc.		77.57	71.12	68.70	64.90
	CC-Cos	Worst	15.09	01.39	07.67	01.49
		Mean	16.07	01.67	08.29	01.90
		Best	17.08	01.94	08.93	02.29
	CC-Sp	Worst	–01.82	37.07	05.94	33.22
		Mean	11.48	44.87	18.44	41.43
		Best	24.73	52.11	30.89	49.12
Coda-H (#2776)	Acc.		83.39	–	75.48	72.05
	CC-Cos	Worst	12.87	–	07.49	01.91
		Mean	13.68	–	08.06	02.47
		Best	14.52	–	08.64	02.96
	CC-Sp	Worst	07.11	–	06.82	35.24
		Mean	19.97	–	19.39	44.88
		Best	32.66	–	31.95	53.60

Table 4: Evaluation results across tasks. (x100).

B Prompt Templates

Decision Prompt. To elicit the model's answer:

```
Question: {question text}
Choose the most plausible answer,
respond only with the answer and
the description:

A. {option A}
B. {option B}
C. {option C}
D. {option D}

Answer:
```

The model is expected to respond with a single letter corresponding to its chosen option and the answer content (e.g., "A. option A").

Explanation Prompt. To generate a justification for the selected answer, we prompt the model as follows:

```
Question: {question text}
Choose the most plausible answer,
respond only with the answer and
the description:

A. {option A}
B. {option B}
C. {option C}
D. {option D}
```

`Selected Answer: {model's answer}`

`Why did you make that choice? Explain briefly.`

`Explanation: {model's explanation}`

The explanation should rely only on the information provided in the question and answer choices.

C Implementation Details

Environment. All experiments were conducted using NVIDIA A100 80GB GPUs. We used PyTorch and Hugging Face Transformers. Captum was used for attribution computations. All models were accessed through the Hugging Face Hub and run in float16 mode.

Dataset Processing. Each multiple-choice QA dataset (ECQA, ARC-Easy, ARC-Challenge, CODAH) was converted into a unified format consisting of the question, 4–5 answer choices, the model's predicted answer, and $k = 5$ sampled post-hoc explanations. All datasets were split into training (70%), validation (20%), and test (10%) sets using a fixed random seed (42). We precomputed attribution vectors for the model's decision and each explanation, resulting in six LIME runs per scenario.

Prompting. Decisions were elicited using the following template: "`Choose the most plausible answer:`", followed by the answer options. Explanations were generated using the model's answer with the prompt: "`Why did you make that choice? Explain briefly.`" (see Appendix B for full prompt templates).

Text Generation. We used nucleus sampling with top-$p = 0.9$, temperature = 0.7, and a maximum generation length of 400 tokens. Generation was done in a zero-shot setting. Padding tokens were manually set to a new padding token for compatibility. For each input, we generated 5 explanations using fixed random seeds ranging from 42 to 46 to ensure diversity and reproducibility.

Attribution. Feature attributions were computed using LIME with 500 perturbation samples per example, using Captum's implementation. The reference input consisted of the pad token repeated to the input length. A manually defined list of formatting and punctuation tokens was excluded from attribution (see Appendix D).

Consistency Metrics. We computed cosine similarity and Spearman rank correlation between the attribution vectors of the model's decision and each explanation. These scores were used to rank explanations and construct attribution-based preference pairs.

DPO Fine-Tuning. We used Direct Preference Optimization (DPO) to fine-tune each model using preference pairs derived from attribution alignment scores. We sampled 5 explanations per instance, ranked them using Spearman correlation, and used the highest- and lowest-ranked as the preferred and rejected explanations, respectively. Fine-tuning was done with LoRA for parameter-efficient updates. We apply LoRA with a rank of 32 and scaling factor `lora_alpha` = 32, targeting all major projection layers (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`) and disabling dropout and bias adaptation. Gradient checkpointing is enabled via the `unsloth` backend for memory efficiency. All models were fine-tuned independently per dataset. We summarize the hyperparameters used for fine-tuning all models in Table 5.

D Skip Tokens for Attribution

To ensure attribution focuses on semantically meaningful input content, we exclude formatting and structure-related tokens from all attribution computations. For LLaMA models, we identify a set of *skip tokens* that should not be considered when measuring input importance. These tokens include both model-specific structural markers and general-purpose formatting symbols.

Structure Tokens. Based on the tokenizer vocabulary and architecture of **LLaMA3.1** and **LLaMA3.2**, we exclude the following structure tokens when present:

- `<|start_header_id|>`
- `<|end_header_id|>`
- `<|eot_id|>`
- `<|begin_of_text|>`
- `Ċ` (newline marker or formatting artifact)
- `Ġ->` (common artifact from tokenizer for arrows or prompt delimiters)

Hyperparameter	ECQA (L3.1)	ARC-Easy (L3.1)	ECQA (L3.2)	ARC-Easy (L3.2)
Epochs	10	10	10	10
Batch Size	16	16	16	16
Gradient Accumulation	8	8	8	8
Learning Rate	4.21×10^{-6}	4.65×10^{-6}	9.55×10^{-6}	6.32×10^{-6}
DPO Beta	5.13	5.64	8.44	8.84
Score Scale Factor	10	10	10	10
Optimizer	AdamW	AdamW	AdamW	AdamW

Table 5: Hyperparameters used for DPO fine-tuning across model–dataset pairs.

These tokens typically serve as formatting scaffolding or internal delimiters in system and chat prompts, and do not reflect actual semantic content from the question or explanation.

Usage. We apply this skip list to both the decision and explanation inputs before computing attribution scores. Tokens that match the above set (by string or token ID) are held fixed during perturbation and excluded from similarity calculations between attribution vectors.

Note. We do not exclude standard stop words or punctuation in our main experiments, as their contribution may still reflect the model's learned reasoning behavior. However, our framework allows toggling this behavior for ablation studies.

E Qualitative Examples Heatmaps

Figure 6 presents token-level attribution heatmaps for model decisions and explanations, using the LIME method. Each subfigure illustrates a different scenario from the ECQA dataset, comparing the best and worst explanations produced by vanilla models based on Spearman and Cosine alignment scores. Across examples, we observe that high-quality explanations (left) tend to emphasize tokens that align more closely with the model's decision rationale—for instance, highlighting location-specific cues like "eastern coast" or situational context like "board room." In contrast, low-ranking explanations (right) often shift focus to semantically irrelevant or misleading tokens, despite sounding plausible. These visualizations underscore the discriminative power of attribution-based metrics and highlight the variability in explanation quality for the same input.

F TruthfulQA Evaluation

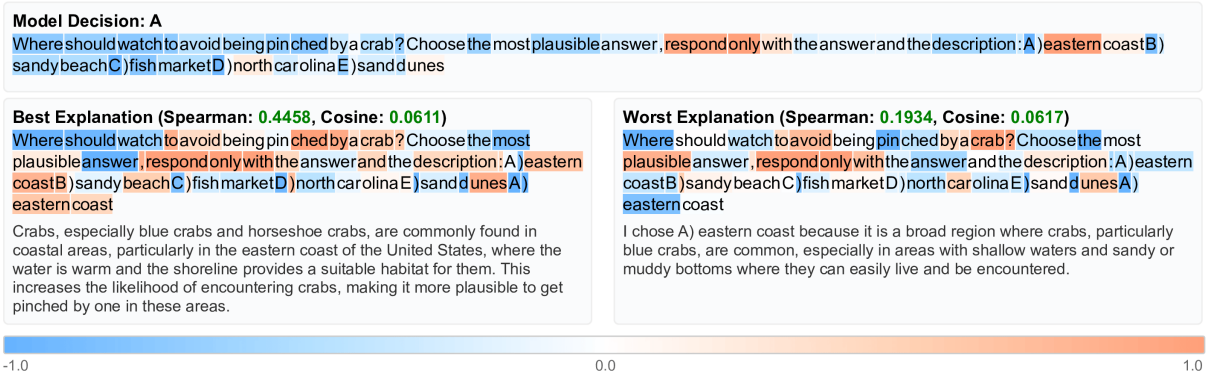
We further evaluate whether optimizing for self-consistency affects factual reliability using the TruthfulQA benchmark (Lin et al., 2021). Table 6 reports multiple-choice accuracy for the vanilla and DPO-tuned models under both LIME- and LIG-based consistency training. Across all settings, accuracy differences remain marginal ($< 1\%$), confirming that improving attributional self-consistency does not compromise factual correctness. In some cases (e.g., ARC-Easy–trained L3.1–8B), the DPO-tuned models slightly outperform their vanilla counterparts, suggesting that the enhanced alignment between decisions and explanations may even reinforce truthful reasoning. These results support that our optimization procedure preserves the model's faithfulness to facts while improving internal consistency.

	Model	Training Data	Acc. (%)	vs Vanilla
LIME	L3.1-8B	Vanilla	53.86	–
		ECQA	53.73	-0.13
		ARC-Easy	54.71	+0.85
	L3.2-3B	Vanilla	52.75	–
		ECQA	52.63	-0.12
		ARC-Easy	53.0	+0.25
LIG	L3.1-8B	Vanilla	53.86	–
		ECQA	54.10	+0.24
		ARC-Easy	53.98	+0.10
	L3.2-3B	Vanilla	52.75	–
		ECQA	53.49	+0.74
		ARC-Easy	53.12	+0.37

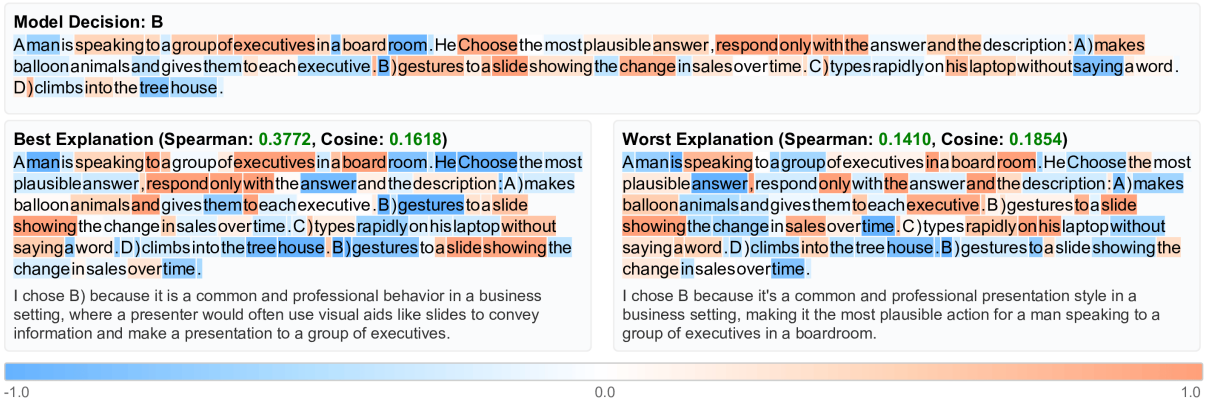
Table 6: TruthfulQA MCQA accuracy as a faithfulness test. Higher is better.

G Cross-Method Generalization

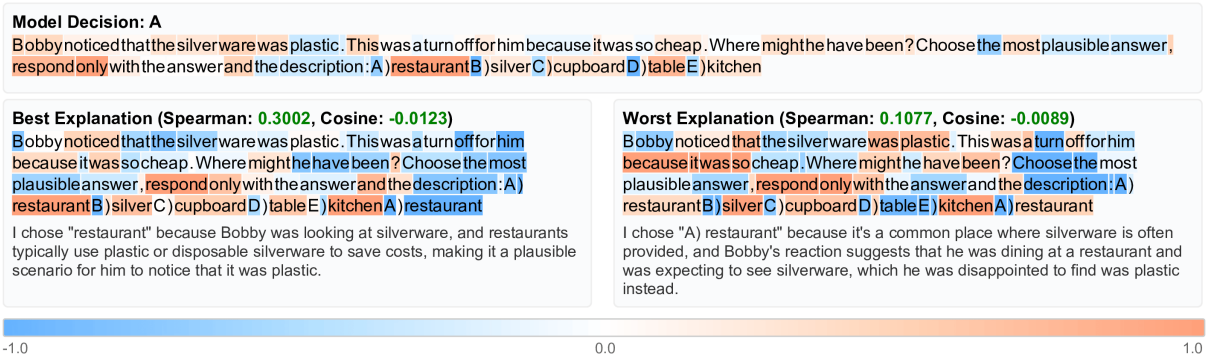
We further assess whether improvements in self-consistency transfer across attribution methods. Specifically, we train models using one attribution framework (e.g., LIG) and evaluate them using another (e.g., LIME), as shown in Table 7. This cross-method setup tests whether the model internalizes a general notion of alignment. Results indicate that DPO-tuned models exhibit limited transfer: improvements in one attribution method do not



(a) Vanilla LLaMA3.1, ECQA, LIME



(b) Vanilla LLaMA3.1, ECQA, LIME



(c) Vanilla LLaMA3.2, ECQA, LIME

Figure 6: Qualitative heatmap visualizations showing token-level attribution scores for model decisions and explanations using the LIME method.

reliably extend to another. For example, models trained with LIG show negligible or inconsistent gains when evaluated with LIME metrics, and vice versa. This outcome aligns with recent evidence that different attribution paradigms capture complementary aspects of model reasoning rather than identical importance structures.

		Training	Acc.	CC-Cos (↑↓)	CC-Sp (↑↓)
ECQA	L3.1-8B	None	68.40	**09.57** ± 0.31	17.68 ± 0.47
		ECQA	71.20	09.24 ± 0.32	17.73 ± 0.48
		ARC-E	70.00	09.28 ± 0.33	**18.08** ± 0.43
	L3.2-3B	None	68.00	0.33 ± 0.30	**19.36** ± 0.57
		ECQA	67.20	0.46 ± 0.31	19.23 ± 0.56
		ARC-E	66.80	**0.57** ± 0.30	19.27 ± 0.57
ARC-Easy	L3.1-8B	None	83.20	15.56 ± 0.34	**13.65** ± 0.49
		ECQA	82.80	**15.73** ± 0.36	13.17 ± 0.52
		ARC-E	82.80	15.51 ± 0.35	13.51 ± 0.52
	L3.2-3B	None	76.80	06.48 ± 0.44	17.30 ± 0.54
		ECQA	76.80	**06.49** ± 0.44	17.33 ± 0.55
		ARC-E	76.80	06.42 ± 0.44	**17.45** ± 0.56

Table 7: **cross-method generalization: LIG model using LIME evaluations** Values aggregate over 5 explanations per item (Mean). Δ is DPO - Vanilla.

H Plausibility Study

To confirm that optimizing for self-consistency did not compromise the readability of explanations, we conducted a small-scale plausibility study.

Participants. Three graduate students participated in the study.

Materials. We sampled 80 explanations in total: 5 examples from each condition defined by attribution method (LIME, LIG), dataset (ECQA, ARC-Easy), model (LLaMA3.1, LLaMA3.2), and variant (Base, DPO-Tuned). Each explanation was presented as full text together with the model's selected decision.

Procedure. The study was administered via Qualtrics, with questions randomized. For each explanation, annotators answered two questions: (i) whether the explanation was written in natural language (binary yes/no), and (ii) how plausible the explanation appeared on a 3-point Likert scale (1 = implausible, 3 = highly plausible).

Results. Nearly all explanations (99.4%) were judged to be natural language. The mean plausibility score was 2.90 out of 3. Ratings were broadly

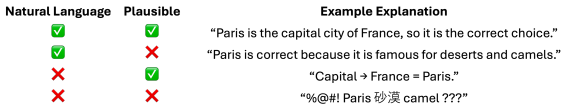


Figure 7: Illustrated example showen to participants of annotating

consistent across participants. These results confirm that explanations remain coherent and generally human-plausible, even after DPO fine-tuning.

Example Shown `Scenario: What is the capital of France? A) Rome B) Paris C) Madrid D) Berlin`
`Model's Decision: B) Paris`