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

Sahar Admoni[1], Ofra Amir[1], Assaf Hallak[2], Yftah Ziser[2,3]
[1]Technion – Israel Institute of Technology [2]Nvidia Research [3]University of Groningen
saharad@campus.technion.ac.il, oamir@technion.ac.il, {ahallak,yziser}@nvidia.com

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, and show that standard supervised fine-tuning on the same data fails to achieve comparable gains. These improvements generalize robustly across domains, paving 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).

[1]Code and data are available at https://github.com/saharad1/ConstLLM.

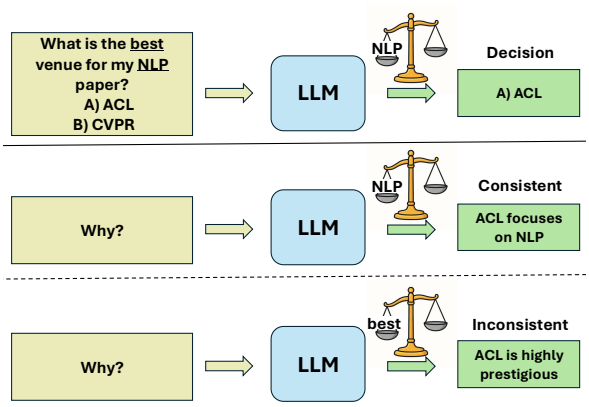


Figure 1: Illustration of explanation consistency using feature importance. **Top:** The LLM generates an answer where the word *NLP* in the prompt has high feature importance. **Middle:** A consistent explanation where *NLP* in the prompt also has high feature importance. **Bottom:** An inconsistent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

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).

Prior works apply counterfactual interventions

Findings of the Association for Computational Linguistics: ACL 2026, pages 987–1003
July 2-7, 2026

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 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 with the following contributions: (1) **Post-hoc Self-Consistency Bank (PSCB):** a large-scale benchmark linking over 85,000 decisions with 428,000 explanations and attribution vectors (LIME and LIG) across four QA datasets and two LLMs, enabling systematic evaluation of explanation–decision alignment. (2) **Empirical analysis:** The first large-scale study of attribution-based self-consistency, showing it is largely orthogonal to correctness and that Spearman rank correlation provides a more discriminative signal than cosine similarity, effectively separating high- and low-quality explanations. (3) **Preference-based optimization:** Attribution-based preference data from PSCB is used to fine-tune LLMs with DPO, yielding substantial in-domain gains and robust cross-domain generalization without degrading accuracy. (4) **Multidimensionality:** Improvements transfer across domains but not across attribution paradigms, revealing that different methods capture fundamentally distinct notions of input relevance, with direct implications for how the community evaluates attribution-based explanations.

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 $\sim$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). Other recent work targets related but distinct objectives: PEX (Zhao and Iii, 2025) evaluates whether an explanation linguistically supports a label by comparing label-explanation log-odds, and CCT (Siegel et al., 2024) computes population-level correlations between counterfactual interventions and explanation mentions. Both operate at the level of probabilistic or semantic consistency rather than input-level attribution alignment. Because these methods do not access or compare token-level attribution vectors, they measure a complementary dimension of explanation quality; a model can score well on probabilistic consistency while still exhibiting misaligned feature attributions between its decision and explanation. Our work focuses on *attributional* self-consistency within each decision, directly comparing the input evidence used for the answer with that invoked in the explanation, and provides the first scalable framework for both measuring and improving this property.

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:

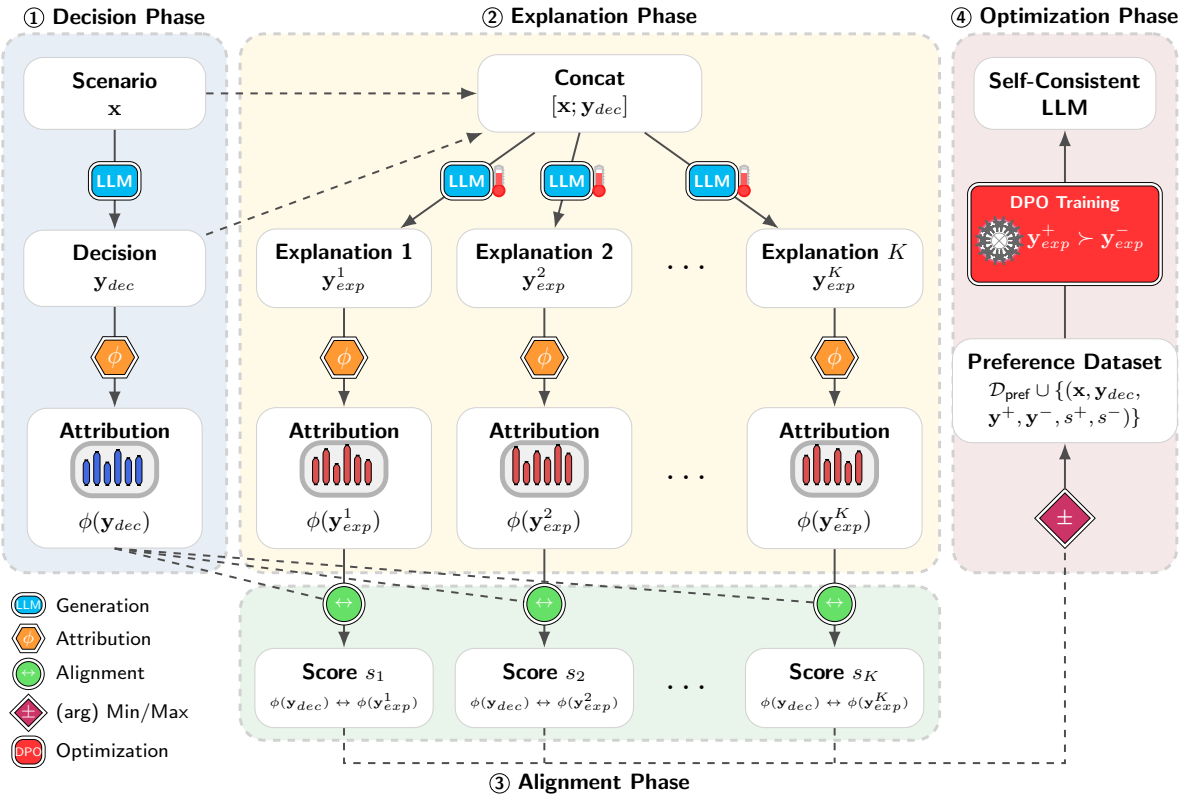


Figure 2: Overview of the PSCB pipeline. ① **Decision:** Given a multiple-choice QA instance, the LLM generates an answer together with attribution scores over the input. ② **Explanation:** The model is then prompted to produce K diverse explanations (via temperature sampling) conditioned on the question 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.

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 $\mathrm{SLP}(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \log P(y_t \mid \mathbf{x}, \mathbf{y}_{<t})$, where $P(y_t \mid \mathbf{x}, \mathbf{y}_{<t}) = \frac{\exp(\ell_t[y_t])}{\sum_{v \in V} \exp(\ell_t[v])}$, and ℓ_t denotes the model logits at step t conditioned on $(\mathbf{x}, \mathbf{y}_{<t})$. This scalar serves as the attribution target, indicating how much each token in $\mathbf{x}$ contributes to generating $\mathbf{y}$. For each output $\mathbf{y}$, 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 evaluate 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, using both the original Meta release and the UNSLOTH-optimized implementation (identical weights, minor decoding differences).

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 A.

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 C.

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). High values indicate that explanations highlight features aligned with the decision.

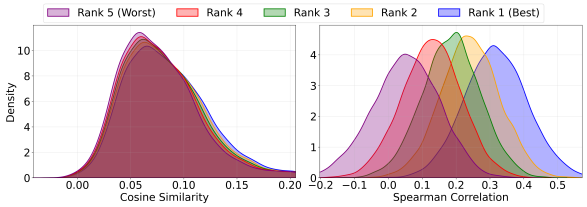


Figure 3: Smoothed distribution of self-consistency scores by explanation rank for the LLaMA3.1-8B model on ECQA. **Spearman correlation (right)** shows clear rank separation, reflecting strong sensitivity to explanation quality. **Cosine similarity (left)** shows substantial overlap across ranks, indicating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

3.5 Key Findings and Insights

We report self-consistency across all model–dataset pairs using both LIME and LIG attributions, measured by cosine similarity (CC-COS) and Spearman rank correlation (CC-SP). For each setting, we report the *worst*, *mean*, and *best* scores across five sampled explanations.

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),

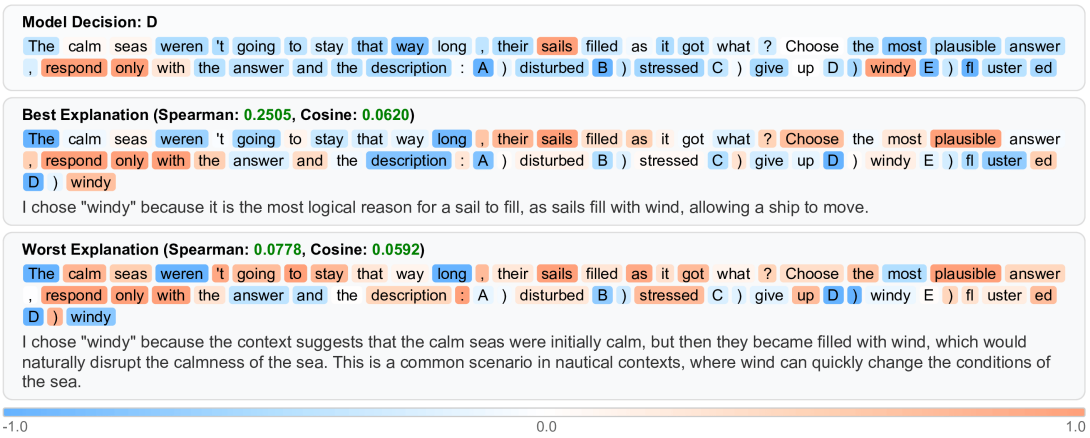


Figure 4: An example from the ECQA dataset shows attribution alignment for the LLaMA3.2-3B model's decision and its best ($Sp = 0.25$) and worst ($Sp = 0.08$) explanations. The model selects answer "D" ("windy"), with attribution heatmaps highlighting word-level influences (blue = negative, orange = 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.

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 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 self-consistency correlates with prediction correctness by comparing alignment scores for correct vs. incorrect predictions (Table 2 in Appendix). Differences are small and inconsistent across datasets and models, suggesting that self-consistency is largely orthogonal to accuracy.

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.

Training Procedure. We use LoRA (Hu et al., 2021) for parameter-efficient adaptation, inserting trainable low-rank matrices into the transformer's attention and feed-forward layers. All explanation generations follow the zero-shot prompting scheme of Section 3.4 to ensure train–eval consistency. Models are fine-tuned independently per dataset; hyperparameters and LoRA configurations are in Appendix B. A qualitative comparison of DPO-tuned and vanilla explanations is shown in Figure 5.

5 Empirical Evaluation

We assess the impact of DPO training across three Decider–Explainer configurations: B–B (Base De-

			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
CODAH (#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 1: PSCB self-consistency scores (×100) on the test split of each dataset, for both LLaMA models and both attribution methods (LIME, LIG). Worst, Mean, and Best are reported over 5 sampled explanations per item. CC-Cos: cosine similarity; CC-Sp: Spearman rank correlation between decision and explanation attribution vectors.

cider, 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.

Baseline Definition. Improvements are measured relative to the B–B baseline, with all other factors (architecture, prompting, decoding, attribution pipeline) held constant. The B–B configuration represents a fully untuned baseline, while B–T isolates the effect of tuning the explainer alone. Comparisons to T–T therefore isolate the effect of attribution-based preference optimization on the decision model. To our knowledge, no prior methods directly optimize attributional self-consistency. Methods such as PEX (Zhao and Iii, 2025) optimize explanation-label probabilistic support, which is complementary but operates at a different level of analysis: a model can achieve strong probabilistic consistency while its explanation still relies on different input features than its decision.

	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**
CODAH	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 2: 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.

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 3 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, 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

			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%
	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%
		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%
		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%
	LIG 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%
		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%
		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%
		Acc.	67.03	67.03	67.68 ↑1.1%	62.44	62.44	62.99 ↑0.9%
	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%
		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%
		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%
	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%
		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%
		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%
	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%
		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%
		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%
	LIG 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%
		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%
		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%
		Acc.	84.26	84.26	84.84 ↑0.7%	80.23	80.23	79.27 ↓1.2%
	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%
		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%
		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%
	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%
		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%
		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 3: 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 indicated by arrows.

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. To verify that these patterns extend beyond the LLaMA family, we apply the same pipeline to QWEN2.5-7B-INSTRUCT on ECQA (Appendix I). The trends hold: DPO improves worst-case Spearman (10.83→10.87) while preserving higher-ranked explanations (36.98→37.02), and cosine similarity improves across all ranks (e.g., worst: 5.55→5.57, best: 6.48→6.61). As with LLaMA, the strongest gains target the weakest explanations, suggesting attribution-guided preference optimization captures an architecturally general pattern.

SFT Baseline Comparison. To isolate the role of the contrastive training signal in DPO, we compare against a supervised fine-tuning (SFT) baseline trained on the same highest-ranked explanations that serve as the chosen examples in DPO, without exposure to rejected explanations. Table 5 reports results on ECQA for LLaMA3.1-8B with LIME attributions. SFT substantially degrades Spearman correlation in both the T–T and B–T settings: mean CC-Sp drops by 41.4% and 37.3%, respectively, relative to the B–B baseline. Cosine similarity also decreases in the B–T setting (−6.6%). In contrast, DPO improves Spearman correlation by 13.0% (T–T) and 11.0% (B–T), with cosine gains of up to 20.8%. These results demonstrate that simply maximizing the likelihood of high-consistency explanations is insufficient for improving self-consistency; the contrastive signal provided by DPO, comparing

		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 4: 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.

		B-B	SFT		DPO	
			B-T	T-T	B-T	T-T
	Acc.	70.34	71.53		70.34	69.70
CC-Cos	Worst	07.58	07.01	07.65	07.64	**09.16**
	Mean	08.05	07.52	08.19	08.10	**09.72**
	Best	08.53	08.04	08.74	08.57	**10.28**
CC-Sp	Worst	05.00	-02.66	-03.46	07.20	**07.86**
	Mean	18.22	11.42	10.68	20.22	**20.58**
	Best	31.30	25.38	24.47	32.86	**33.45**

Table 5: SFT vs. DPO on ECQA (LLaMA3.1-8B, LIME). SFT is trained on the highest-ranked explanations (by Spearman correlation) used as the chosen examples in DPO. SFT degrades Spearman correlation across all settings, while DPO consistently improves it.

preferred and rejected explanations, is essential for guiding the model toward more aligned reasoning.

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 4 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

Because self-consistency is defined in terms of internal attributional alignment between a model's decision and its explanation, it cannot be directly assessed by human annotators. Accordingly, our primary evaluation of self-consistency relies on attribution-based measures. To verify that optimizing for self-consistency does not degrade user-facing explanation quality, we conduct a small-scale human plausibility study as a sanity check (Appendix H). This study is intentionally scoped to assess surface-level quality rather than attributional faithfulness, which is inherently inaccessible to human annotators.

Three annotators evaluated 80 explanations sampled across all models and datasets, judging (i) whether each explanation was expressed in natural language and (ii) its surface-level plausibility on a 3-point scale. Nearly all explanations were judged to be natural language (98.8% for base models and 100% for DPO-tuned models), with average plausibility scores of $2.92/3.0$ and $2.88/3.0$, respectively. These results indicate that DPO fine-tuning preserves the naturalness and plausibility of explanations, and does not introduce artifacts that would reduce human interpretability.

We further evaluate factual reliability by testing both vanilla and DPO-tuned models on TRUTHFULQA (Lin et al., 2022), which probes a model's tendency to produce false or misleading answers under adversarial questioning. DPO-tuned models achieve comparable or slightly higher accuracy across both variants (Appendix F), showing that improving attributional self-consistency does not compromise factual accuracy.

5.4 Self-Consistency is Multidimensional

A key finding of this work is that attribution-based self-consistency is not a single unified property but a multidimensional one. Notably, while our method generalizes robustly across domains (Section 5.1), it does not transfer across attribution paradigms. 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, improvements largely remain within the training method, confirming that each attribution approach captures a fundamentally distinct notion of feature importance.

This outcome is consistent with the theoretical differences between these paradigms: LIME fits local linear surrogates via input perturbation, while Integrated Gradients accumulates sensitivity along a baseline path, and KSHAP estimates Shapley values through sampling. These methods encode different assumptions about how to decompose a prediction into token-level contributions, and prior comparative analyses confirm they often disagree in practice (Krishna et al., 2022; Neely et al., 2021, 2022). To quantify this, we 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).

Importantly, this does not indicate that our optimization exploits method-specific artifacts: although training uses only Spearman, the unoptimized cosine metric also improves strongly (up to 92.8% on ECQA, Section 5.1). Gaming method-specific noise would degrade unoptimized metrics, not improve them.

These findings carry broader implications for the attribution literature: practitioners should select the attribution method whose assumptions best match their evaluation goals, and our framework provides a general-purpose pipeline for optimizing self-consistency under any chosen definition of input relevance.

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 Spearman correlation offers a sharper and more stable signal than cosine for assessing this alignment. Leveraging it via preference supervision yields 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 the first scalable framework for quantifying and improving attributional self-consistency, advancing language models whose explanations are grounded in the same input evidence that drives their decisions.

Future Directions. Natural extensions include adapting the framework to open-ended generation by aggregating attributions over answer spans, jointly optimizing across multiple attribution paradigms given that they capture complementary notions of importance (Section 5.4), and an online variant that recomputes attributions during training to close the residual gap between the tuned model and the offline attribution signal it was optimized against. A complementary direction is testing whether higher self-consistency yields measurable user gains, e.g., better-calibrated 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 require more than 4 minutes 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.

As shown in Section 5.4, improvements under one attribution paradigm do not generalize to others, a limitation inherent to all attribution-based evaluations. This highlights both the difficulty of defining a universal measure of explanation faithfulness and the value of our framework for optimizing under any chosen attribution method.

Human Evaluation. Because attributional self-consistency depends on internal model signals inaccessible to human annotators, our primary evaluation is attribution-based. The plausibility study (Section 5) serves only as a sanity check on surface-level quality. Whether increased self-consistency translates into more *helpful* explanations from a user perspective remains an open question, and we view systematic human studies of downstream utility as an important direction for future work.

Model and Task Scope. Our study focuses on small to medium scale instruction-tuned LLMs evaluated in a multiple-choice question answering (MCQA) setting. We adopt MCQA as a controlled testbed for studying attributional self-consistency, as it provides a well-defined decision signal and enables reliable comparison between decision and explanation attributions across models. This design choice prioritizes attributional reliability and experimental control over task generality.

The proposed optimization framework itself is not tied to MCQA and can in principle be applied to open-ended decision settings, including free-form reasoning or summarization. However, extending PSCB-style attribution benchmarks to such tasks substantially increases attribution and evaluation complexity. While we include a lightweight cross-model-family validation, a systematic evaluation across diverse architectures and open-ended tasks remains an important direction for future work.

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.

Acknowledgments

Funded by the European Union (ERC, Convey, 101078158). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

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. In *Proceedings*

of the 61st Annual Meeting of the Association for Computational Linguistics (ACL).

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? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

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. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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.

Xiangjue Zhao and James Caverlee Iii. 2025. Pex: Improving faithfulness of large language models through explanation editing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

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

A 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.

B 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 A 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 C).

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.

SFT Baseline. To evaluate whether a standard supervised fine-tuning objective can improve self-consistency, we train an SFT baseline on the same highest-ranked explanations used as the chosen examples in DPO. The SFT model is fine-tuned to maximize the likelihood of these preferred explanations using the same LoRA configuration, training epochs, batch size, and optimizer as the DPO models. The learning rate is 6.95×10^{-6}. This isolates the effect of the contrastive DPO signal by removing the rejected explanation from the training procedure.

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 6.

C 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)

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

- Ġ-> (common artifact from tokenizer for arrows or prompt delimiters)

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.

D DPO vs. Vanilla Example

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., 2022). Table 7 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 factual accuracy 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 7: TruthfulQA MCQA accuracy as a factual reliability check. 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 8. This cross-

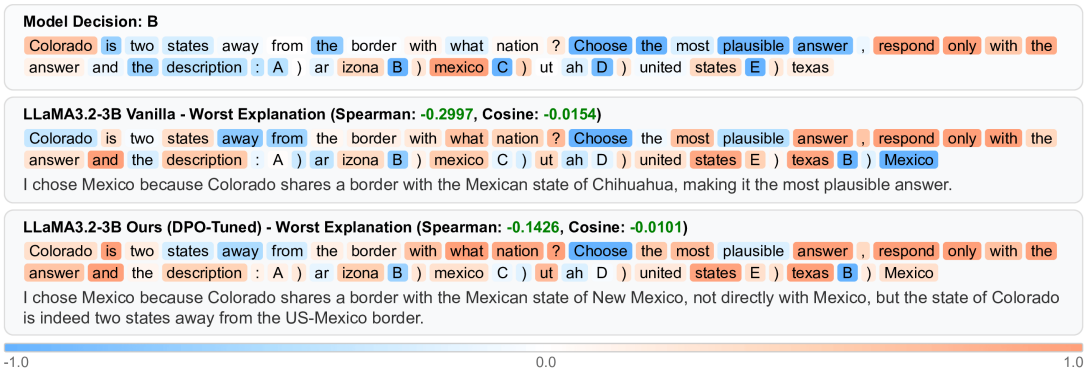


Figure 5: An example from ECQA shows attribution alignment for the LLaMA3.2-3B model's decision and its worst explanation across two variants. The model selects "B" (Mexico), with high-attribution tokens like "Colorado," "two," "states," and "Mexico" supporting its geographic reasoning. The DPO-tuned 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.

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

H Plausibility Study

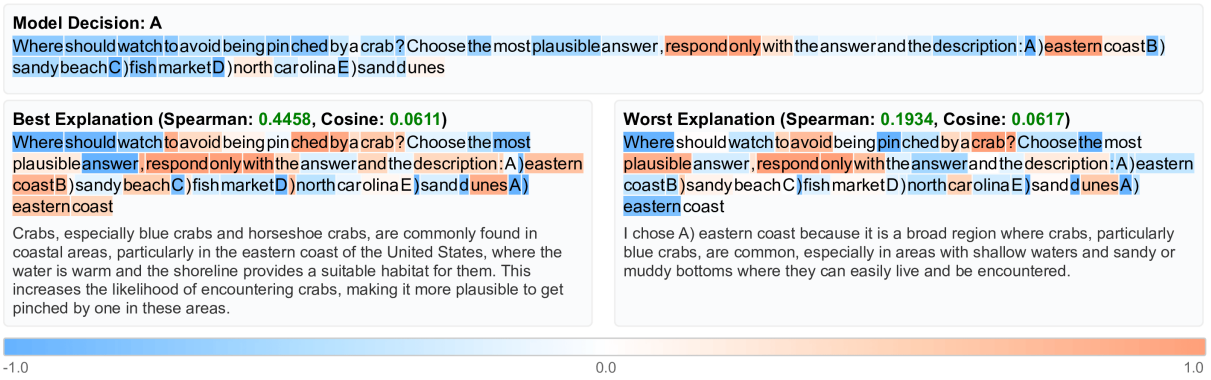
To confirm that optimizing for self-consistency did not compromise the readability of explanations, we conducted a small-scale plausibility study.

Participants. Three graduate students participated in the study.

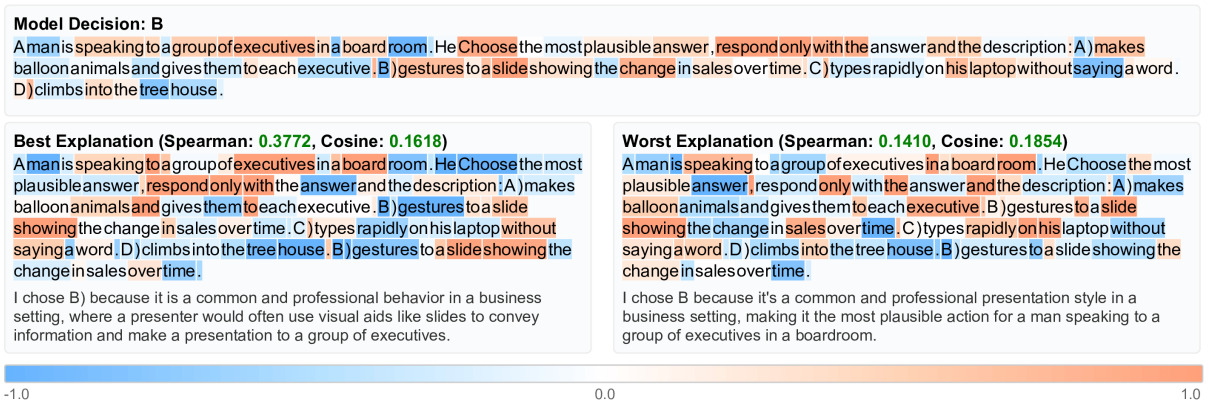
Materials. We sampled 80 explanations in total: 5 examples from each condition defined by attribution method (LIME, LIG), dataset (ECQA, ARC-Easy), model (LLaMA3.1, LLaMA3.2), and variant (Base, DPO-Tuned). Each explanation was presented as full text together with the model's selected decision.

Procedure. The study was administered via Qualtrics, with questions randomized. For each explanation, annotators evaluated two criteria: (i) whether the explanation was expressed in fluent natural language (binary yes/no), and (ii) how plausible the explanation appeared on a 3-point Likert scale (1 = implausible, 3 = highly plausible). Annotators were instructed not to consider alternative answers, external knowledge, or internal model reasoning. No comparison between explanations, ranking tasks, or faithfulness judgments were requested.

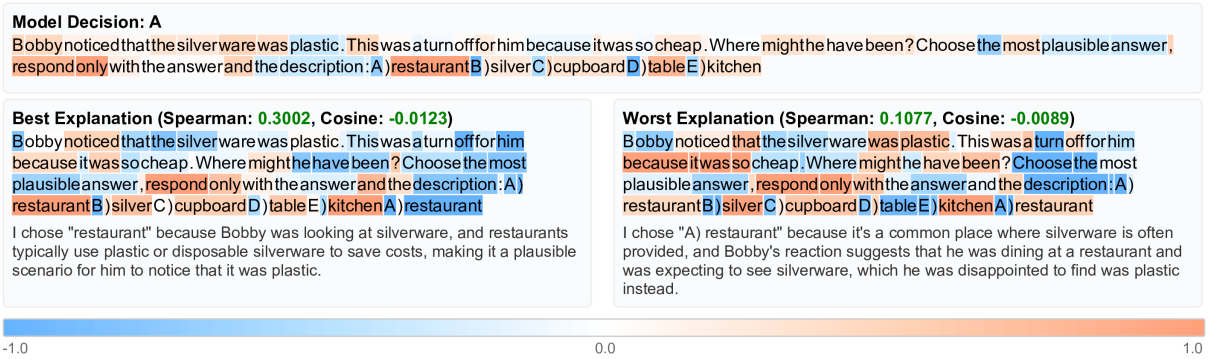
Results. Nearly all explanations were judged to be natural language (98.8% for base models and 100% for DPO-tuned models). The mean plausibility scores were $2.92/3.0$ for base models and $2.88/3.0$ for DPO-tuned models. Annotators showed high percent agreement across both criteria, with ratings broadly consistent across participants. These results confirm that explanations remain coherent and generally human-plausible, even after DPO fine-tuning. We emphasize that this study is intentionally scoped as a sanity check on surface-



(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.

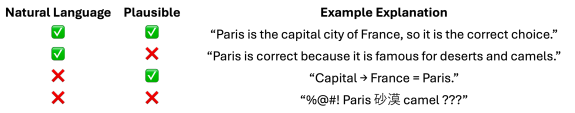


Figure 7: Illustrated example shown to participants during annotation.

level explanation quality, rather than a validation of attributional faithfulness, which depends on internal model signals inaccessible to human annotators.

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

I Cross-Model-Family Validation

To examine whether the observed self-consistency improvements are tied to the LLaMA architecture, we perform a lightweight cross-model-family validation on the ECQA dataset using QWEN2.5-7B-INSTRUCT. ECQA offers a controlled multiple-choice setting with a well-defined decision signal, allowing us to apply the same attribution, alignment, and evaluation pipeline used in the main experiments without modification.

Setup. We evaluate QWEN2.5-7B-INSTRUCT under the same Decider–Explainer configurations employed throughout the paper, comparing a vanilla model against a DPO-tuned variant trained using attribution-based preferences. Due to the substantial computational cost of constructing a full PSCB benchmark for an additional model family, this validation is restricted to ECQA. Unlike prior small-scale attribution studies, we nonetheless use the full ECQA test set rather than a reduced subset. Self-consistency is measured primarily using Spearman rank correlation, with cosine similarity reported as a secondary metric. All attribution methods, hyperparameters, and evaluation procedures are identical to those used for the LLaMA models, and all scores are reported on the same scale as in the main results.

Results. The results follow the same qualitative trends observed for the LLaMA family. Under Spearman correlation, DPO tuning improves self-consistency at the lower end of the explanation quality distribution, increasing the worst-case score from 10.829 to 10.871, while leaving the best-case score largely unchanged (36.980 to 37.021). The median Spearman score decreases slightly from 24.163 to 23.235, reflecting the same redistribution pattern seen in earlier experiments. Cosine similarity shows consistent improvements across all ranks, with increases in the worst (5.547 to 5.571), median (5.964 to 6.080), and best (6.478 to 6.607) explanations. As in the LLaMA experiments, the most reliable gains occur for the weakest explanations, while changes to higher-quality explanations remain modest.

Discussion. Although the absolute magnitude of the improvements is smaller than for the LLaMA models, the qualitative behavior closely mirrors the central findings of the paper. Attribution-guided preference optimization primarily strengthens low-quality explanations while largely preserving stronger ones, consistent with the intended objective of the method. This cross-model-family validation therefore provides evidence that the proposed framework captures an optimization pattern that is not specific to a single architecture, within the same operational definition of attributional self-consistency. We stress that this experiment is intentionally limited in scope and is intended as a sanity check rather than a replacement for a full multi-family PSCB construction.