Construct, Don't Discover: Dual-Stream Sentiment Decomposition for Interpretable Implicit Sentiment Analysis

Anonymous ACL submission

Abstract

Mechanistic opacity in language models leads to instability and systematic errors in implicit sentiment analysis (ISA) tasks involving complex pragmatics. Existing mechanistic interpretability approaches primarily follow a *post-hoc discovery* paradigm, which faces limitations when the prior concept is already known due to feature polysemy and weak causal binding. In this paper, we propose the Dual-Stream Sentiment Decomposition (DSSD) framework, which shifts mechanistic interpretability from post-hoc discovery to *supervised construction*. DSSD combines differential supervision with bidirectional geometric constraints to explicitly disentangle literal semantics from intended sentiment representations. Beyond improving ISA performance, DSSD provides an auditable geometric pathway for mechanistic analysis and causal intervention in pragmatic reasoning tasks. Across multiple architectures, DSSD achieves a near-orthogonal average subspace separation of 85.15° on literal–intent conflict samples, compared with a near-aligned 16.07° on non-conflict samples. Bidirectional causal intervention experiments further yield an average causal effect (ACE) of 0.88, demonstrating strong causal controllability. We evaluate DSSD on five cross-domain ISA benchmark datasets. DSSD achieves F1 improvements in 85.8% of 90 within-model comparisons and statistically significant gains in 70.8%, with an average F1 improvement of 16.76%.

1 Introduction

Transformer-based pretrained language models (PLMs) and large language models (LLMs) demonstrate strong performance on explicit sentiment analysis (ESA) tasks (Sharma et al., 2025). However, implicit sentiment analysis (ISA) tasks with complex pragmatics remain challenging (Lai et al., 2025; Liu et al., 2026a). Prior studies have shown that advanced language models, such as GPT-4 Turbo in few-shot chain-of-thought (CoT) settings

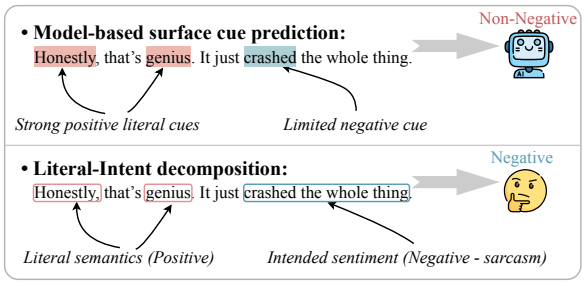

Figure 1: Surface cue bias remains an unresolved cause of model failure in ISA tasks with literal-intent conflict.

and PLMs with supervised fine-tuning (SFT), such as RoBERTa, perform significantly worse on ISA datasets than on ESA datasets. The F1 score gap can exceed 40% (Zhang et al., 2025). Figure 1 shows a typical model failure in an ISA task involving a literal–intent conflict. When such an error pattern occurs, it is unclear whether the failure arises from the model's inability to capture the intended sentiment or from literal signals overwhelming the intended signals. In this paper, we operationalize "literal semantics" as surface polarity signals supported by lexical items and explicit sentiment cues, and "intended sentiment" as target polarity signals determined by pragmatics, which may conflict with surface signals in implicit expressions.

From the perspective of trustworthy AI and model alignment, such error patterns are not merely classification failures; they may reflect a deeper mismatch between the model's internal understanding and human intended meaning, posing potential risks for safety alignment (Zheng et al., 2024). For example, a model may fail to identify harmful instructions when literal semantics conflict with pragmatic intent. In downstream applications, this mechanistic opacity can further undermine reliability and auditability in high-stakes domains such as public opinion monitoring and mental health analysis (Herrera-Poyatos et al., 2025).

Existing approaches provide two complementary yet incomplete directions. Task-oriented optimiza-

tion methods improve downstream performance but offer limited insight into the model's internal decision-making process (Kumar et al., 2023). In contrast, post-hoc feature discovery methods, often based on sparse autoencoders (SAEs), attempt to identify latent concepts after training. However, because SAEs are inherently unsupervised, they cannot be explicitly guided toward predefined prior concepts and often struggle with features that are not highly sparse or that exhibit substantial overlap (Marks et al., 2024; Gallifant et al., 2025). These limitations reduce their reliability as auditing tools for complex pragmatic reasoning tasks.

To address these limitations, we propose Dual-Stream Sentiment Decomposition (DSSD), a *supervised construction* framework for mechanistic interpretability in ISA. Rather than discovering latent features post hoc, DSSD explicitly constructs two parallel representation streams, the literal semantics and intended sentiment, and learns their geometric disentanglement through differential supervision and bidirectional geometric constraints. The framework is designed not only to improve task performance, but also to make the decomposition between surface meaning and pragmatic intent auditable and controllable. Inspired by the SAE-free steering approach of (Li et al., 2025), we further estimate steerable directions from dual-stream differential representations and perform inference-time interventions. This establishes a mechanistic evidence chain from geometric disentanglement to causal validation, enabling interpretable failure diagnosis and actionable insights for model improvement. We conduct experiments across diverse model families, including RoBERTa, DeBERTa, Llama-3.1-8B, Qwen-2.5-7B, and Phi-3-mini, and evaluate DSSD on five cross-domain benchmark datasets. Our contributions are as follows:

Framework. The DSSD framework is the first to introduce conditional geometric topology into semantic disentanglement. It shows that supervised signals can enforce the reconstruction of latent-space structure in language models.

Method. We propose an SAE-free causal intervention mechanism based on semantic rotation along a stream-difference vector, and demonstrate that the constructed geometric subspace is semantically correlated with the prior concept and supports controllable causal intervention.

Evaluation. In generalization experiments, we reveal geometric pattern differences across model paradigms when handling semantic conflicts: rigid orthogonality versus soft entanglement. Code and data are available in an anonymous repository.

2 Related Work

Implicit Sentiment Analysis and Sarcasm Understanding. Mainstream ISA approaches primarily rely on supervised fine-tuning (SFT) or reinforcement learning from human feedback (RLHF) (Sharma et al., 2025). Recent work has also explored prototype-based architectures to introduce intrinsic interpretability into SFT models (Wen and Rezapour, 2026). However, these methods often hard-code incongruity assumptions into supervision and validate explanations through limited case studies, restricting their generalizability. Prompt-based approaches instead aim to emulate human-like pragmatic reasoning. THOR introduces a three-step chain-of-thought prompting framework for sarcasm reasoning (Fei et al., 2023). Yao et al. (2025) further develops Bagging of Cues (BoC), which randomly samples cue combinations and aggregates per-sample predictions through majority voting, but treats each cue as interchangeable and offers no explicit reasoning trace.

Mechanistic Interpretability for Sentiment Representations. Sparse autoencoders (SAEs) have emerged as a prominent tool for identifying activation-space directions aligned with high-level concepts, motivated by the Linear Representation Hypothesis (Marks et al., 2024; Gallifant et al., 2025). This hypothesis posits that semantic concepts are encoded as approximately linear directions in activation space, while superposition causes these concepts to become entangled. SAEs attempt to reconstruct and isolate monosemantic features through unsupervised sparse dictionary learning (Park et al., 2024; Rimsky et al., 2024). Prior work further combines SAEs with linear probes, principal component analysis (PCA), and representation alignment techniques to study the relationship between representation geometry and sentiment prediction (Tigges et al., 2024; Reichman et al., 2025). Other studies construct local and global circuits associated with emotional expression to identify critical sentiment-related components (Hatua, 2025). LinguaLens extends SAE-based analysis to broader semantic and pragmatic phenomena (Jing et al., 2025). Collectively, these approaches improve the interpretability of sentiment representations, but remain limited in controllable and target-specific mechanistic auditing.

From Post-hoc Discovery to Supervised Con-

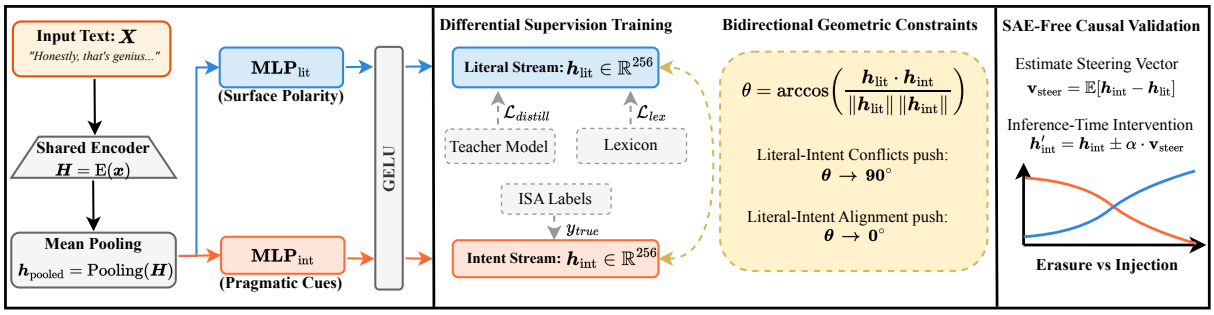

Figure 2: Proposed DSSD framework and SAE-free controlled representation intervention.

struction. Existing mechanistic interpretability approaches based on the SAE-driven post-hoc discovery paradigm face practical limitations when the prior concept is already known. In particular, they remain less effective for complex pragmatic reasoning, causal analysis, and controllable intervention. Although SAEs are effective at identifying concrete and atomic concepts (Zou et al., 2023), they struggle to recover features that are not highly sparse or that exhibit substantial overlap (Li et al., 2025). Sarcasm and other ISA phenomena are representative examples of complex pragmatics, where literal semantics and intended sentiment may conflict through dynamic and potentially non-linear interactions across multiple feature subspaces (Liu et al., 2026b). Even when SAEs successfully identify relevant features, using them for controllable steering remains challenging. SAE-based methods typically decompose activations into tens of thousands of latent features, making it difficult to identify the most causally relevant components (Li et al., 2025). Moreover, SAE training is computationally expensive and difficult to transfer across models or tasks (Molinari et al., 2024). We do not argue that SAEs are ineffective for mechanistic interpretability. Rather, we suggest that when the prior concept is known, such as literal–intent conflict in ISA, a supervised construction paradigm could provide a more direct, controllable, and auditable pathway than post-hoc feature discovery.

3 DSSD: Constructing Literal-Intent Decomposition

Existing methods for separating literal and intended sentiment remain limited. Some rely on lexicon-based methods and overlook the inherently subjective nature of intended sentiment (Liu et al., 2022). Others depend on cross-checking and self-evaluation among LLMs, which is costly and prone to hallucination (Liu et al., 2026b). To address these limitations, we design a dual-stream encoder architecture that learns to disentangle literal semantics and intended sentiment at the representation level, enabling auditable geometric analysis.

3.1 Dual-Stream Architecture for Learned Disentanglement

Architecture Overview. As shown in Figure 2, DSSD processes an input text X through a shared backbone encoder $E(\cdot)$ to obtain contextualized representations $H = E(X) \in \mathbb{R}^{n \times d}$. To unify the interface across different architectures, we apply mean pooling to aggregate H into a global vector $h_{\text{pooled}} \in \mathbb{R}^d$. This vector is then fed into two parallel projection heads, implemented as multilayer perceptrons (MLPs), MLP_{lit} and MLP_{int}, mapping the representation into a shared 256-dimensional space. For each MLP, we apply GELU activation, because it provides smoother gradients, which is beneficial for fine-grained representation learning required in our dual-stream setting (Hendrycks, 2016).

$$\begin{aligned} h_{\text{lit}} &= \text{GELU}\big(W_{\text{lit}}\, h_{\text{pooled}} + b_{\text{lit}}\big) \\ h_{\text{int}} &= \text{GELU}\big(W_{\text{int}}\, h_{\text{pooled}} + b_{\text{int}}\big) \end{aligned} \tag{1}$$

As shown in Eq. 1, W and b denote the learnable weight matrices and bias vectors of the linear layers. h_{lit} and h_{int} represent the literal semantic stream and the intended sentiment stream.

Then, under bidirectional geometric constraints, DSSD applies differential supervision to h_{lit} and h_{int}. This training process achieves dual-stream decomposition at the representation level. Algorithm 1 in Appendix A presents the complete implementation procedure of the DSSD architecture.

Differential Supervision Strategy. To induce functional specialization at the representation level, we adopt asymmetric supervision signals to training two streams. Force h_{int} to capture pragmatic truth while h_{lit} is anchored to surface polarity.

Let z_{int} and z_{lit} denote the raw logits produced by the classifiers of the intent stream and the literal stream. Each classifier uses a linear layer with the structure $\text{Linear}(256, 2)$. z_{int} denotes the classifier logits of the intent stream for the ISA label. In this paper, these logits correspond to binary sarcasm detection. The intent stream serves as the primary decision stream for the ISA task. Therefore, h_{int} should capture pragmatic shifts and override literal cues when a literal-intent conflict exists. We directly supervise this stream with the ground-truth label y_{true} from the sarcasm dataset. We then compute the Cross-Entropy (CE) loss over the classifier output of the intent stream. This ensures that z_{int} aligns with the final pragmatic meaning and that the model learns to resolve semantic conflicts.

$$\mathcal{L}_{\text{int}} = \mathcal{L}_{\text{CE}}\big(\text{Softmax}(z_{\text{int}}),\, y_{\text{true}}\big) \tag{2}$$

z_{lit} denotes the classifier logits of the literal stream for the explicit surface sentiment proxy. We construct a composite signal to anchor the literal stream to surface polarity. We are training a teacher model on an explicit sentiment dataset to provide a soft probability distribution $P_{teacher}$, and combined with a sentiment lexicon to provide pseudo-labels y_{lex}. As an explicit sentiment prior, this composite signal offers a 'Counter Factual Anchor' for the literal stream (Ren et al., 2026). Without it, the intent stream is hard to define 'conflict'. The literal loss combines knowledge distillation and lexical guidance. We selected Kullback-Leibler divergence (KL) to measure the gap between the literal stream's prediction and the teacher's distribution because it can transfer the semantic knowledge implied in the teacher's soft probabilities. While λ is a hyperparameter controlling the weight of each component. By optimizing these objectives, both streams will be penalized for deviating from their direction, forcing the shared encoder to maintain both representations simultaneously.

$$\begin{aligned}\mathcal{L}_{\text{lit}} = \lambda_{\text{distill}} \cdot \text{KL}(P_{\text{teacher}} \,\|\, \text{Softmax}(z_{\text{lit}})) \\ + \;\lambda_{\text{lex}} \cdot \text{CE}(\text{Softmax}(z_{\text{lit}}),\, y_{\text{lex}})\end{aligned} \tag{3}$$

Bidirectional Geometric Constraints. Differential supervision alone may not prevent the representation collapse. To explicitly enforce geometric disentanglement and shape the topology of the representation space, we introduce a bidirectional geometric constraint mechanism. The definition of the intersection angle θ between the two streams is shown in Eq. 4. Where ϵ is a small scalar for numerical stability.

$$\theta = \arccos\left(\frac{h_{\text{lit}} \cdot h_{\text{int}}}{\|h_{\text{lit}}\|\,\|h_{\text{int}}\| + \epsilon}\right) \tag{4}$$

Inspired by multi-task learning and domain adaptation, enforcing orthogonality across task-specific feature spaces is a common strategy to mitigate negative transfer (Chen et al., 2025b). In our setting, we defined geometric alignment as semantic redundancy and geometric orthogonality as information independence. When literal semantics are consistent with intended sentiment, the two streams should encode largely redundant information, and this corresponds to subspace alignment. When literal-intent conflict exists, we expect the intent stream capture pragmatic cues that are missing from the literal stream, which corresponds to subspace orthogonality.

Thus, we formulate the geometric loss $\mathcal{L}_{geo}$ to dynamically push the intersection angle based on the ground-truth label y_{true} in sarcastic dataset. When $y_{\text{true}} = 1$, push θ to orthogonality ($90°$), when $y_{\text{true}} = 0$, push θ to alignment ($0°$).

$$\mathcal{L}_{\text{geo}} = \begin{cases} |\cos(h_{\text{lit}},\, h_{\text{int}})|\,, & \text{if } y_{\text{true}} = 1 \\ 1 - \cos(h_{\text{lit}},\, h_{\text{int}})\,, & \text{if } y_{\text{true}} = 0 \end{cases} \tag{5}$$

When θ approaches $90°$, it does not mean the two streams are unrelated. In contrast, it indicates that the intent stream contains a 'stream-difference vector' Δh, which is independent of the literal stream. As shown in Eq. 6, when literal-intent conflict exists in the text, Δh should have a large magnitude and a distinctive direction.

$$h_{\text{int}} = h_{\text{lit}} + \Delta h \tag{6}$$

The final training objective combines the task-specific intent loss, the auxiliary literal supervision, and the geometric constraints.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{int}} + \lambda_{\text{lit}}\,\mathcal{L}_{\text{lit}} + \lambda_{\text{geo}}\,\mathcal{L}_{\text{geo}} \tag{7}$$

Where λ_{lit} and λ_{geo} are scalar hyperparameters that control the loss weights of the auxiliary literal supervision and the geometric constraints.

3.2 SAE-Free Steering for Causal Validation

After constructing the disentangled subspace, we can conduct causal interventions without relying on any specific SAEs. We define the semantic shift vector or steering vector $\mathbf{v}_{steer}$ as the expected difference between intent and literal representations over the text X.

$$\mathbf{v}_{\text{steer}} = \mathbb{E}_{x\sim\mathcal{D}}\big[\, h_{\text{int}}(x) - h_{\text{lit}}(x) \,\big] \tag{8}$$

During inference, we edit the intent representation by injecting or erasing $\mathbf{v}_{steer}$ and measure the resulting change in the predicted sarcasm probability. Erasing $\mathbf{v}_{steer}$ from sarcastic examples should reduce sarcasm confidence, while injecting it into non-sarcastic examples should increase sarcasm confidence. We use this intervention as a controlled behavioral test of whether the constructed stream-difference direction is decision-relevant. This test reveals whether the constructed geometry has functional influence on the model's decision boundary.

$$h'_{\text{int}} = h_{\text{int}} \pm \alpha \cdot \mathbf{v}_{steer} \tag{9}$$

Where the hyperparameter α is a dynamically adjusted scalar, we linearly sweep α over $[0.0, 2.0]$ to measure the causal response curve between intervention strength and the resulting predicted probabilities. To quantify the steering efficacy, we compute the Average Causal Effect (ACE), defined as the mean shift in predicted probability induced by maximal intervention, which can measure the magnitude of behavior change induced by representation editing in the constructed intent-stream space.

$$\begin{aligned}\text{ACE} = \mathbb{E}_{x\sim\mathcal{D}}\big[&P\big(y = s \,\big|\, h'_{\text{int}}(x;2)\big) \\ &-P\big(y = s \,\big|\, h'_{\text{int}}(x;0)\big)\big]\end{aligned} \tag{10}$$

4 Experiments

4.1 Experimental Setting

Dataset. Prior ISA studies often treat sarcasm as the most common form of implicit sentiment (Yacoub et al., 2024). Therefore, we adopt the News Headlines Dataset for Sarcasm Detection to evaluate subspace disentanglement, causal intervention, and ablation settings in the DSSD framework, which is a widely used formal textual style ISA dataset with human-annotated sarcasm labels (Misra and Arora, 2023). This formal style reduces label noise and informal expressions that are common in Twitter-based datasets (Liashuk, 2021). We hold out 15% of the data as the test set and split the remaining 85% evenly into training and validation.

For anchoring the literal stream, we use the Stanford Sentiment Treebank v2 (SST-2) to train the teacher model (Socher et al., 2013). SST-2 is built from movie reviews with sentiment polarity labels. Its distribution is dominated by explicit sentiment lexical cues. These cues serve as a proxy for context-independent lexical polarity. We use label-stratified sampling to split the data into 90% training and 10% validation. We further incorporate auxiliary signals from the VADER sentiment lexicon, forcing the literal stream to fit the statistical correlations between surface textual cues and sentiment labels (Hutto and Gilbert, 2014).

To further evaluate the task utility and cross-domain generalization of DSSD, we conduct tests on five public benchmark datasets from SarcasmBench (Zhang et al., 2025). These datasets differ in source, topic, language style, and size.

Model Selection and Baselines. To evaluate the generality and robustness of the DSSD, we selected five backbone models across different architectures, parameter scales, and training paradigms. RoBERTa-base is one of the most common BERT-style baselines in sentiment analysis (Liu et al., 2019). DeBERTa-v3-base includes an inherently disentangled attention mechanism, which may synergize with our explicit geometric constraints (He et al., 2021). We treat those encoder backbones as representative discriminative baselines.

Llama-3.1-8B-it, Qwen-2.5-7B-it, and Phi-3-mini-128k-it are our generative baselines. Llama-3.1 represents strong open-weight reasoning LLM (Grattafiori et al., 2024). We include Qwen-2.5 for its strong reasoning and multilingual understanding ability (Team, 2024). Phi-3-mini can examine whether geometric disentanglement also holds in the model with high information density but limited parameter capacity (Nguyen et al., 2025).

In the cross-domain tests, we add several strong baselines. DC-Net-RoBERTa is the strongest PLM baseline reported in SarcasmBench (Zhang et al., 2025). The Bagging of Cues (BoC) prompts is included as a recent state-of-the-art approach for sarcasm detection (Yao et al., 2025). ChatGPT-5.5 is a strong, closed-source general-purpose LLM baseline.

Implementation Details. We first train DSSD on five backbone models to construct literal and intent streams in the representation space. Then, we compute the intersection angle θ between the two streams to test whether DSSD induces the expected stable geometric topology. Next, we test the necessity of each component through ablation studies and anchor robustness checks. We further conduct controlled representation interventions to examine whether the learned steering vector supports auditable causal intervention. We compare DSSD steering with random, SAE-style, PCA, and other control directions, and use ACE and off-target drift

Model	$\theta_{\text{sarc}}(^\circ)$	$\theta_{\text{non}}(^\circ)$	$\Delta\theta(^\circ)$	σ^2_{within}	t-statistic	Cohen's d	$\|\Delta h_{\text{avg}}\|$
RoBERTa-base	87.08	3.49	83.59	143.60	304.58	6.98	16.96
DeBERTa-V3-base	**89.11**	5.31	**83.79**	127.58	**347.30**	**7.41**	17.45
Llama-3.1-8B-it	85.56	25.59	59.98	219.14	190.92	4.05	26.53
Qwen-2.5-7B-it	88.92	26.89	62.03	**336.82**	164.17	3.38	**27.02**
Phi-3-mini-128k-it	75.07	19.05	56.02	266.12	150.55	3.43	20.42

$n_{sarc} = 3802, n_{non} = 4494$. θ denotes the sample-level mean intersection angle over all instances in the corresponding class. $\|\Delta h_{\text{avg}}\|$ denotes the ℓ_2-norm of the mean semantic shift vector. σ^2_{within} denotes pooled within-class variance.

Table 1: Geometric separation statistics across diverse model architectures.

to evaluate intervention strength and specificity. Finally, we conduct cross-domain task utility tests to examine whether DSSD can serve as a controllable interpretability and auditing tool while preserving or improving performance on sarcasm detection. We fully fine tune the selected encoder-only models. Due to computational constraints, we fine tune the selected LLMs with QLoRA. We use 4-bit quantization and train LoRA adapters on a frozen backbone (Dettmers et al., 2023; Hu et al., 2022). We set $r = 8$ and $\alpha = 16$ for Llama-3.1-8B and Qwen-2.5-7B, $r = 16$ and $\alpha = 32$ for Phi-3-mini. For all LLMs, we follow each model's official chat template to match its instruction tuning format. For fair comparison, we train each backbone under the same configuration with 3 random seeds. The teacher model is trained for 3 epochs, and DSSD is trained for 5 epochs. The maximum sequence length is 64 and effective batch size at 64. The learning rate is 2×10^{-5} for encoder-only models, and 1×10^{-4} for LLMs. The loss weights are fixed as $\lambda_{\text{lit}} = 1.0$, $\lambda_{\text{geo}} = 0.2$, $\lambda_{\text{distill}} = 0.5$, and $\lambda_{\text{lex}} = 0.1$. We select these weights through a coarse grid search on the validation set. All models are optimized with AdamW and a linear scheduler. For the cross-domain tests, we follow the data splits, parameter settings, and prompt designs used in SarcasmBench. Appendix B provides the full implementation details.

4.2 Results and Discussion

Geometric Disentanglement and Stream Validity. To validate that DSSD successfully constructs geometrically disentangled subspaces, we measure the intersection angle θ between the literal and intent representations across all test samples. As Table 1 shows, the results confirm our core hypothesis, across all five architectures, the sarcastic samples contain literal-intent conflicts that exhibits near-orthogonal (θ_{sarc} approaching 90°). In contrast, non-sarcastic samples converge toward alignment (θ_{non} approaching 0°). The separation magnitude $\Delta\theta$ exceeds 55° across all models. Statistical significance is confirmed by independent two-sided Welch's t-tests, where all $t > 150$ with corresponding $p < 10^{-16}$, indicating that the observed separation is not due to chance. Cohen's $d > 3.0$ indicates practically large effect sizes beyond statistical significance. The $\|\Delta h_{\text{avg}}\|$ value supports the hypothesis in Eq. (6). The consistently large magnitudes across all architectures indicate that the intent stream does not simply copy the literal stream but encodes an operational stream-difference vector. The within-class variance σ^2_{within} and the Figure 4 show clear distributional differences across model paradigms. Encoder-only models present 'rigid orthogonality' with concentrated distributions and lower intra-class variance, which may reflect the discriminative nature of encoder architectures, since their representations are optimized for classification boundaries. In contrast, LLM distributions are much wider, indicating a 'soft entanglement pattern'. This may be related to their autoregressive pretraining objective, which encourages representations to preserve information for next-token prediction rather than enforce strict classification boundaries (Sander and Peyré, 2025).

UMAP has been used to visualize the latent semantic shift vectors from a global structural view (McInnes et al., 2018). We use the cosine metric to capture semantic directionality and set $n_{\text{neighbors}} = 50$ and $min_{\text{dist}} = 0.1$ to preserve the global topology. As shown in Appendix C, sarcastic and non-sarcastic samples form visually distinguishable clusters across all architectures. These results prove DSSD successfully induces conditional geometric topology across both discriminative and generative backbones, and reveal architectural differences in the learned geometric structure.

Ablation and Robustness. To determine whether the dual stream geometry is caused by the proposed DSSD design, rather than being an artifact of extra heads and task supervision, we compare a series DSSD ablation variants. All metrics are averaged over five models and three random seeds. In ad-

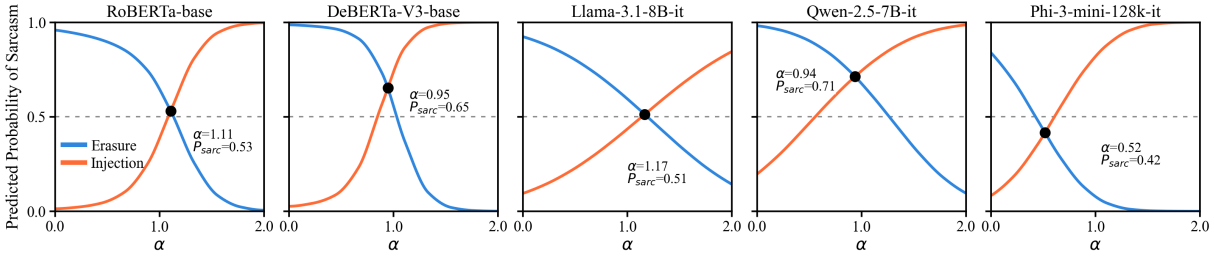

Figure 3: SAE-free controlled intervention curves using the DSSD stream-difference direction.

Variant	Macro-F1	$\Delta\theta$	GTS
Single-Head SFT/LoRA	94.26	–	–
Two-Head Same-Label	94.24	0.77	0.49
NoGeo ($\lambda_{\text{geo}} = 0$)	94.21	-3.28	0.53
SoftGeo ($\lambda_{\text{geo}} = 0.05$)	94.25	51.53	0.14
HardGeo ($\lambda_{\text{geo}} = 0.20$)	**94.36**	65.52	0.09
Orthogonal-only	94.26	2.16	0.49
Alignment-only	94.24	47.40	0.26
No-Literal-Anchor	94.19	**69.90**	**0.07**
Teacher-only Anchor	94.32	64.16	0.09
Lexicon-only Anchor	94.27	67.65	0.08

All values are averaged over five models and three random seeds, yielding 15 model-seed combinations.

Table 2: Compact component ablation summary.

Model	$\text{ACE}_{\text{Erasure}}$	$\text{ACE}_{\text{Injection}}$
RoBERTa-base	0.9522	**0.9830**
DeBERTa-V3-base	**0.9849**	0.9735
Llama-3.1-8B-it	0.7799	0.7499
Qwen-2.5-7B-it	0.8846	0.7895
Phi-3-mini-128k-it	0.8387	0.9173

All values are averaged over three random seeds for each model at the maximum intervention strength $\alpha = 2.0$.

Table 3: Average causal effect (ACE) under erasure and injection interventions.

dition to calibrated task performance and $\Delta\theta$, we defined a scalar geometry diagnostic, the Geometry Topology Score (GTS):

$$\text{GTS} = \frac{1}{2}\left(\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}|c_i| + \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}}(1-c_i)\right) \tag{11}$$

where $c_i = \cos(h_{\text{lit}}^{(i)}, h_{\text{int}}^{(i)})$, $\mathcal{S}$ is the sarcastic set, and $\mathcal{N}$ is the non-sarcastic set. Lower GTS means that sarcastic examples are closer to orthogonality and non-sarcastic examples are closer to alignment.

Table 2 shows that differential supervision alone does not induce the desired topology. The NoGeo variant achieves competitive F1 but negative $\Delta\theta$ and high GTS. A similar pattern is observed at the Two-Head baseline, suggesting that adding a classifier is insufficient to separate the two streams. In contrast, the bidirectional geometric constraint strongly improves topology formation. Higher λ_{geo} leads to higher $\Delta\theta$ and lower GTS. We also scan λ_{geo} from 0 to 2.0 to test sensitivity to geometry strength. As shown in Appendix D, increasing λ_{geo} improves geometric structure with only small task-performance changes. The one-sided variants further show that both constraint directions are necessary. Those results suggest that DSSD is robust across a range of geometry strengths and does not depend on a single setting. Although the No-Anchor variant obtains the lowest GTS, it shows the weakest F1. The full anchor ablation results (Appendix E) confirm our definition of the anchor as a semantic grounding mechanism, rather than a sole source of task accuracy.

SAE-Free Controlled Representation Intervention. The steering vector $\mathbf{v}_{\text{steer}}$ is estimated from the training set and tested on an unseen test set, which ensures that the observed causal effects reflect generalization of the learned geometric structure, rather than overfitting to specific instances.

Figure 3 shows the causal intervention curves across five architectures. The y-axis denotes the model confidence for the sarcastic class. When $\alpha = 0$, no intervention is applied, so the y-axis value reflects the baseline prediction confidence. As α increases, both erasure and injection show the expected monotonic trends across all five models.

The curve intersections with the decision boundary $P_{\text{sarc}} = 0.5$ discover that intervention symmetry depends on both architecture and class-specific geometric tightness. Models with strongly orthogonal sarcastic representations but more diffuse non-sarcastic distributions show injection-dominant asymmetry (DeBERTa-V3, Qwen-2.5-7B). In contrast, models with weaker orthogonality show erasure-dominant behavior (Phi-3-mini).

Table 3 reports high ACE values for bidirectional interventions across all models, indicating that the $\mathbf{v}_{\text{steer}}$ is strongly decision-relevant across architectures. However, high ACE alone may not be enough to prove a unique causal mechanism. As shown in Table 5, random and literal-sentiment

Model	Setting	IAC-V1	IAC-V2	iSarc.	Ghosh	SE18	Avg.
RoBERTa-base	SFT	68.48	80.59	70.71	88.32	71.30	75.88
	DSSD	69.48*	79.97	74.33*	88.78*	71.82*	76.88
DeBERTa-v3-base	SFT	68.53	81.90	77.50	89.02	73.67	78.12
	DSSD	69.72*	**82.49***	**77.90***	**90.14***	**73.99***	**78.85**
DC-Net-RoBERTa	SFT	67.97	80.14	72.87	88.06	71.65	76.14
	DSSD	**70.00***	80.89*	74.97*	88.63*	73.45*	77.59
Llama-3.1-8B-it	BoC	63.12	68.12	61.74	65.96	63.29	64.45
	SFT (LoRA)	**72.60**	**83.11**	73.12	88.04	77.38	78.85
	DSSD	68.07	82.70	**79.31***	**89.11***	**78.43***	**79.52**
Qwen-2.5-7B-it	BoC	63.19	66.89	53.47	59.00	63.74	61.26
	SFT (LoRA)	71.08	82.23	64.36	86.97	74.66	75.86
	DSSD	69.17	81.37	78.57*	88.83*	75.75*	78.74
Phi-3-mini-128k-it	BoC	68.21	69.48	64.21	69.65	65.81	67.45
	SFT (LoRA)	40.02	33.76	49.97	48.19	37.88	41.96
	DSSD	51.15	53.71	45.69	35.38	49.05	47.00
ChatGPT-5.5	K-shot	67.26	70.88	64.30	74.05	71.59	69.62

1. *indicates a statistically significant improvement ($p < 0.05$) over the strongest competitive baseline in the respective group.
2. Bold underlined values mark the best PLM result for each dataset. Blue bold underlined values mark the best LLM result for each dataset.

Table 4: Compact cross-domain task utility on five SarcasmBench datasets, we report Macro-F1 only.

Direction	ACE	OTD
DSSD	**0.9596**	**0.8981**
Random	0.0082	0.0196
Literal-sentiment	-0.0267	0.0246
PCA	0.9205	0.9706
CAA	0.9210	0.9709
Classifier weight	0.9210	0.9709
Shuffled-label	0.3525	0.4139
SAE-style	0.3220	0.3743

Table 5: Compact steering controls. OTD denotes the average off-target drift over Low-LDI and surface-polarity non-sarcastic control subsets.

directions produce near-zero ACE, whereas PCA, CAA, and classifier-weight directions also produce high ACE but show larger off-target drift. These controls suggest that DSSD is not the only decision-relevant editing direction, and also provide a compact supervised stream-difference direction that is stronger than irrelevant controls and directly tied to the constructed literal-intent geometry. Full control definitions and separate off-target scores are provided in Appendix G.

Task Utility and Cross-Domain Generalization. To test whether the constructed geometry translates into task utility beyond the main News Headlines setting, we further evaluate DSSD on SarcasmBench (Zhang et al., 2025). Because closed-source API models do not expose internal representations, DSSD is evaluated only on open-weight backbones.

Table 4 shows that DSSD improves task utility across both PLM and open-weight LLM settings. We conduct paired bootstrap resampling with $N = 10{,}000$ to test the one-sided hypothesis $\Delta\text{F1} > 0$ at $p < 0.05$. Across 120 within-model paired bootstrap tests, DSSD achieves positive ΔF1 in 85.8% of comparisons and statistically significant gains in 70.8%, with a mean $\Delta\text{F1} = 16.76$. Overall, these results support two complementary claims. DSSD is significantly stronger than prompt-based LLMs at ISA reasoning. It's also stronger than strong supervised PLM baselines while providing an auditable and controllable representation structure. The full results and 95% confidence intervals are provided in Appendix H.

The results also confirm a failure pattern reported by prior studies: advanced general-purpose LLMs do not necessarily outperform fine-tuned PLMs on ISA tasks (Zhang et al., 2025). In addition, complex prompting strategies often lead to degeneration at the generation level in LLMs (Holtzman et al., 2019; Welleck et al., 2019). Case studies are provided in Appendix F.

5 Conclusion

This paper presents DSSD, a supervised construction framework that addresses key limitations of post-hoc mechanistic interpretability approaches. By combining differential supervision with bidirectional geometric constraints, DSSD explicitly learns disentangled representations of literal semantics and intended sentiment, rather than discovering features from entangled activation spaces. Experiments across multiple architectures show that DSSD induces a conditional geometric topology with strong separation, large effect sizes, and predictable intervention behavior. Cross-domain evaluations also show consistent improvements in task performance and generalization. Overall, DSSD reframes mechanistic analysis in ISA from post-hoc feature discovery toward supervised and auditable representation construction.

6 Limitations

Despite its contributions, this work has several limitations. First, DSSD operationalizes literal semantics using explicit surface-polarity signals derived from SST-2 and VADER. While these signals provide useful and auditable supervision, they do not fully capture the broader linguistic notion of literal meaning. Accordingly, the literal stream should be interpreted as a surface-polarity anchor rather than a complete semantic representation.

Second, the current geometric design of DSSD is primarily tailored to binary conflict settings such as sarcasm detection. Real-world ISA scenarios may involve more complex pragmatic phenomena, including multi-polar sentiment, multimodal interactions, cross-cultural variation, and long-context discourse. Future work should extend the supervised construction toward more general label-conditional geometries and flexible subspace structures.

Third, representation intervention results should be interpreted as behavioral evidence under controlled manipulations, not as proof of a unique causal mechanism. Our control-direction experiments reduce this concern but cannot fully rule out all alternative discriminative directions.

Finally, DSSD requires access to internal model representations. While the framework demonstrates more stable and significant performance gains and intervention effects than prompt-engineering approaches on open-source models, it cannot currently be applied to closed-source systems such as ChatGPT-5.5. Future work should investigate how to better leverage the capabilities of closed-source LLMs, which often exhibit strong general reasoning ability, to achieve more reliable improvements on ISA tasks.

References

Ning Chen, Shen-Huan Lyu, Yanyan Wang, and Bin Tang. 2025a. Learning semantic boundaries: An adaptive structural loss for multi-label contrastive learning. In *2025 2nd International Conference on Intelligent Perception and Pattern Recognition (IPPR)*, pages 204–210.

Shengbing Chen, Dongmei Zhou, Chao Wei, Xingyang Li, Hongyu Yao, and Hao Luo. 2025b. Dual-stream collaborative network (dscn): Multimodal sentiment analysis via modality-invariant and modality-specific representation learning. *Neurocomputing*, page 131059.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. 2023. Reasoning implicit sentiment with chain-of-thought prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1171–1182, Toronto, Canada. Association for Computational Linguistics.

Jack Gallifant, Shan Chen, Kuleen Sasse, Hugo Aerts, Thomas Hartvigsen, and Danielle Bitterman. 2025. Sparse autoencoder features for classifications and transferability. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29939–29963, Suzhou, China. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Amartya Hatua. 2025. Mechanistic interpretability of gpt-2: Lexical and contextual layers in sentiment analysis. In *NeurIPS 2025 Workshop on Efficient Reasoning*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

D Hendrycks. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

David Herrera-Poyatos, Carlos Peláez-González, Cristina Zuheros, Andrés Herrera-Poyatos, Virilo Tejedor, Francisco Herrera, and Rosana Montes. 2025. An overview of model uncertainty and variability in llm-based sentiment analysis: challenges, mitigation strategies, and the role of explainability. *Frontiers in Artificial Intelligence*, Volume 8 - 2025.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.

Yi Jing, Zijun Yao, Hongzhu Guo, Lingxu Ran, Xiaozhi Wang, Lei Hou, and Juanzi Li. 2025. LinguaLens: Towards interpreting linguistic mechanisms of large

language models via sparse auto-encoder. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28232–28251, Suzhou, China. Association for Computational Linguistics.

Manohar Kumar, Aisha Aijaz, Omkar Chattar, Jainendra Shukla, and Raghava Mutharaju. 2023. Opacity, transparency, and the ethics of affective computing. *IEEE Transactions on Affective Computing*, 15(1):4–17.

Wenna Lai, Haoran Xie, Guandong Xu, and Qing Li. 2025. Multi-task learning with llms for implicit sentiment analysis: Data-level and task-level automatic weight learning. *IEEE Transactions on Knowledge and Data Engineering*.

Zihao Li, Xu Wang, Yuzhe Yang, Ziyu Yao, Haoyi Xiong, and Mengnan Du. 2025. Feature extraction and steering for enhanced chain-of-thought reasoning in language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10893–10913, Suzhou, China. Association for Computational Linguistics.

Xenia Liashuk. 2021. Informal language learning through interactive social media tools: current trends and practices. In *ICERI2021 Proceedings*, pages 5514–5520. IATED.

Chang Liu, Chen Gao, Yong Li, and Meng Wang. 2026a. Exploring implicit sentiment analysis through progressive prompting strategies. *IEEE Transactions on Computational Social Systems*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yiyi Liu, Yequan Wang, Aixin Sun, Xuying Meng, Jing Li, and Jiafeng Guo. 2022. A dual-channel framework for sarcasm recognition by detecting sentiment conflict. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1670–1680, Seattle, United States. Association for Computational Linguistics.

Ziqi Liu, Ziyang Zhou, Yilin Li, Mingxuan Hu, Yushan Pan, Zhijie Xu, and Yangbin Chen. 2026b. Sevade: Self-evolving multi-agent analysis with decoupled evaluation for hallucination-resistant sarcasm detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29573–29581.

Luke Marks, Amir Abdullah, Clement Neo, Rauno Arike, David Krueger, Philip Torr, and Fazl Barez. 2024. Interpreting learned feedback patterns in large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 36541–36566. Curran Associates, Inc.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Rishabh Misra and Prahal Arora. 2023. Sarcasm detection using news headlines dataset. *AI Open*, 4:13–18.

Marco Molinari, Davide Ghilardi, and Federico Belotti. 2024. Efficient training of sparse autoencoders for large language models via layer clustering.

Huu-Khanh Nguyen, Van-Viet Nguyen, Kim-Son Nguyen, Thi Minh-Hue Luong, Vinh T Nguyen, Duc-Quang Vu, and Cong Huu Nguyen. 2025. Fine-tuning mini language models for legal multiple-choice question answering: A comparative study of phi-3.5, qwen 2.5 and llama 3.2. In *Conference on Information Technology and its Applications*, pages 671–682. Springer.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Benjamin Reichman, Adar Avsian, and Larry Heck. 2025. Emotions where art thou: Understanding and characterizing the emotional latent space of large language models. *arXiv preprint arXiv:2510.22042*.

Jing Ren, Wenhao Zhou, Bowen Li, Mujie Liu, Nguyen Linh Dan Le, Jiade Cen, Liping Chen, Ziqi Xu, Xiwei Xu, and Xiaodong Li. 2026. Causal prompting for implicit sentiment analysis with large language models. *IEEE Transactions on Computational Social Systems*, pages 1–13.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Michael Sander and Gabriel Peyré. 2025. Towards understanding the universality of transformers for next-token prediction. In *International Conference on Learning Representations*, volume 2025, pages 86901–86920.

Neeraj Anand Sharma, ABM Shawkat Ali, and Muhammad Ashad Kabir. 2025. A review of sentiment analysis: tasks, applications, and deep learning techniques. *International journal of data science and analytics*, 19(3):351–388.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Curt Tigges, Oskar J. Hollinsworth, Atticus Geiger, and Neel Nanda. 2024. Language models linearly represent sentiment. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 58–87, Miami, Florida, US. Association for Computational Linguistics.

Jason Wei, Denny Zhou, and 1 others. 2022. Language models perform reasoning via chain of thought. *Goggle AI Blog*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Ximing Wen and Rezvaneh Rezapour. 2026. A transformer and prototype-based interpretable model for contextual sarcasm detection. In *The Proceedings for the 15th Workshop on Computational Approaches to Subjectivity, Sentiment Social Media Analysis (WASSA 2026)*, pages 278–288, Rabat, Morocco. Association for Computational Linguistics.

Ahmed Derbala Yacoub, Salwa Slim, and Amal Aboutabl. 2024. A survey of sentiment analysis and sarcasm detection: Challenges, techniques, and trends. *International journal of electrical and computer engineering systems*, 15(1):69–78.

Ben Yao, Yazhou Zhang, Qiuchi Li, and Jing Qin. 2025. Is sarcasm detection a step-by-step reasoning process in large language models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25651–25659.

Yazhou Zhang, Chunwang Zou, Zheng Lian, Prayag Tiwari, and Jing Qin. 2025. Sarcasmbench: Towards evaluating large language models on sarcasm understanding. *IEEE Transactions on Affective Computing*, 16(4):2560–2578.

Yue Zheng, Chip-Hong Chang, Shih-Hsu Huang, Pin-Yu Chen, and Stjepan Picek. 2024. An overview of trustworthy ai: Advances in ip protection, privacy-preserving federated learning, security verification, and gai safety alignment. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 14(4):582–607.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A DSSD Training Procedure

Algorithm 1 DSSD Training Procedure

Input: dataset $\mathcal{D}$ yielding mini-batches (X, y_{true}); encoder E; teacher model T; sentiment lexicon $\mathcal{L}$.
Parameters: loss weights $\lambda_{\text{lit}}, \lambda_{\text{geo}}, \lambda_{\text{distill}}, \lambda_{\text{lex}}$; learning rate η; small constant ϵ.
Output: trained DSSD parameters; prediction is made by $\text{Classifier}_{\text{int}}(h_{\text{int}})$.

Initialize $\text{MLP}_{\text{lit}}, \text{MLP}_{\text{int}}$ and $\text{Classifier}_{\text{lit}}, \text{Classifier}_{\text{int}}$
while not converged **do**
  for each mini-batch (X, y_{true}) in $\mathcal{D}$ **do**
    $H \leftarrow E(X)$
    $h_{\text{pooled}} \leftarrow \text{MeanPool}(H)$
    $h_{\text{lit}} \leftarrow \text{MLP}_{\text{lit}}(h_{\text{pooled}})$
    $h_{\text{int}} \leftarrow \text{MLP}_{\text{int}}(h_{\text{pooled}})$
    $z_{\text{lit}} \leftarrow \text{Classifier}_{\text{lit}}(h_{\text{lit}})$
    $z_{\text{int}} \leftarrow \text{Classifier}_{\text{int}}(h_{\text{int}})$
    $L_{\text{int}} \leftarrow \text{CE}(\text{Softmax}(z_{\text{int}}),\ y_{\text{true}})$
    $P_{\text{teacher}} \leftarrow \text{Softmax}(T(X))$
    $y_{\text{lex}} \leftarrow \mathbf{1}[\text{VADER}(X)_{\text{compound}} > 0]$
    $L_{\text{distill}} \leftarrow \text{KL}\big(P_{\text{teacher}} \,\|\, \text{Softmax}(z_{\text{lit}})\big)$
    $L_{\text{lex}} \leftarrow \text{CE}\big(\text{Softmax}(z_{\text{lit}}),\ y_{\text{lex}}\big)$
    $L_{\text{lit}} \leftarrow \lambda_{\text{distill}}\, L_{\text{distill}} + \lambda_{\text{lex}}\, L_{\text{lex}}$
    $L_{\text{geo}} \leftarrow 0$
    for $i \leftarrow 1$ to $|X|$ **do**
      $s_i \leftarrow \dfrac{h_{\text{lit}}^{(i)} \cdot h_{\text{int}}^{(i)}}{\|h_{\text{lit}}^{(i)}\|\,\|h_{\text{int}}^{(i)}\| + \epsilon}$
      if $y_{\text{true}}^{(i)} = 1$ **then**
        $L_{\text{geo}} \leftarrow L_{\text{geo}} + |s_i|$
      else
        $L_{\text{geo}} \leftarrow L_{\text{geo}} + (1 - s_i)$
      end if
    end for
    $L_{\text{geo}} \leftarrow L_{\text{geo}}/|X|$
    $L_{\text{total}} \leftarrow \mathcal{L}_{\text{int}} + \lambda_{\text{lit}}\, \mathcal{L}_{\text{lit}} + \lambda_{\text{geo}}\, \mathcal{L}_{\text{geo}}$
    Update trainable parameters using AdamW with learning rate η
  end for
end while
return trained parameters

B Implementation Details

All experiments in this study were conducted in a Python 3.12.3 environment with PyTorch 2.11.0 and CUDA 13.0, on the platform with Nvidia Rtx Pro 6000 GPU and an Intel Xeon(R) Platinum 8470Q CPU. Detailed experiment dates and API usage are reported in the README file of the anonymous repository. API usage under the CoT and BoC settings should be interpreted as a reference only, because it depends on parsing strategies, retry settings, and possible platform-level price changes.

B.1 Artifact Statement

We used several public dataset artifacts. The News Headlines Dataset for Sarcasm Detection is used as the main implicit-sarcasm benchmark and is listed on Kaggle as CC BY 4.0. SST-2 is used only as the literal-sentiment anchor. We access it through a Kaggle mirror listed as CC0, while the original Stanford Sentiment Treebank requests citation and does not provide a separate explicit license statement. Sarcasm Corpus V1/IAC-V1 and Sarcasm Corpus V2/IAC-V2 are used for cross-domain sarcasm evaluation. We did not identify standalone open-source licenses for these datasets, but their official NLDS pages indicate free research use. The Ghosh and Veale Twitter sarcasm dataset is treated as research-use only and subject to Twitter/X-derived-data restrictions. iSarcasmEval is released under the MIT License. SemEval-2018 Task 3 is released under CC BY-NC-SA 4.0 and academic/non-commercial use terms. Our DSSD CoT reasoning annotations are an author-created annotation layer for research reproduction and inherit the restrictions of the underlying examples.

We used the following model and API artifacts. RoBERTa-base and DeBERTa-V3-base are used for encoder fine-tuning and DSSD experiments. Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, and Phi-3-mini-128k-instruct are used as instruction-tuned LLM backbones and prompt baselines under their respective licenses. The OpenAI ChatGPT API model, OpenRouter routing for Llama, and SiliconFlow routing for Qwen are governed by their respective proprietary platform terms and applicable model-provider terms.

The code uses Python, Jupyter/IPython, NumPy, pandas, PyTorch, scikit-learn, Hugging Face Transformers, PEFT, Accelerate, bitsandbytes, and sentence-transformers under their respective open-source licenses. These tools are used for notebook execution, data processing, model loading, LoRA/QLoRA training, quantization, metrics, clustering, and statistical analysis. We do not redistribute pretrained weights, API keys, private credentials, predictions, logits, checkpoints, or cached outputs in the supplement.

B.2 Hyperparameter

Main Experiments and Ablation Studies. For the main DSSD experiments and the corresponding ablation studies, we remove empty, very short, and duplicated examples to improve geometric stability in high-dimensional spaces. This preprocessing also increases the signal-to-noise ratio of cosine similarity and reduces early oscillation in the geometric constraint loss caused by noisy vectors (Chen et al., 2025a).

We evaluated five backbone families: RoBERTa-

Item	Setting
Seeds	42, 43, 44
Max sequence length	64
Effective batch size	64
Evaluation batch size	128
Teacher epochs	3
Student epochs	5
Warmup ratio	0.06
Gradient clipping	1.0
Stream hidden dimension	256
Encoder learning rate	2×10^{-5}
LLM learning rate	1×10^{-4}
Optimizer	AdamW with linear learning-rate warmup and decay
LLM tuning	4-bit QLoRA with NF4 quantization and bfloat16 computation
LoRA rank/alpha	Qwen2.5 and Llama-3.1: $r = 8, \alpha = 16$; Phi-3: $r = 16, \alpha = 32$
LoRA dropout	0.05

Table 6: Core hyperparameters for the main DSSD experiments and ablations.

Dataset	Train	Dev	Test
IAC-V1	1214	304	417
IAC-V2	4031	1008	1481
Ghosh	32708	8677	1354
iSarcasmEval	821	280	299
SemEval-2018 Task 3	2910	924	784

Table 7: Dataset splits for the SarcasmBench cross-domain experiments.

base, DeBERTa-V3-base, Qwen2.5-7B-it, Llama-3.1-8B-it, and Phi-3-mini-128k-it. Each backbone was trained with three random seeds $\{42, 43, 44\}$ to reduce variance due to random initialization and data-order effects. The main experimental tables report the mean over these three seeds. The main hyperparameter settings are summarized in Table 6.

SarcasmBench Cross-Domain Benchmark Reproduction. For the cross-domain experiments, we strictly followed the original SarcasmBench reproduction protocol (Zhang et al., 2025). For each model or variant, we train with five random seeds $\{42, 123, 2024, 31415, 7\}$ following SarcasmBench, and report the mean over the five seeds. The dataset splits are shown in Table 7. We follow SarcasmBench and use the same data preprocessing steps, including text normalization, label mapping, and the removal of URLs and hashtags to avoid label leakage. The remaining hyperparameters are summarized in Table 8.

Item	Setting
Seeds	42, 123, 2024, 31415, 7
Max sequence length	512 (IAC-V1 and IAC-V2) 128 (Ghosh, iSarcasmEval, and SemEval-2018 Task 3)
Early stopping patience	5
Effective Batch size	64
Training epochs	30
Evaluation batch size	128
Learning rate	5×10^{-5}
Weight decay	0.01
Adam epsilon	1×10^{-8}
Dropout	0.5
Warmup ratio	0.06
Precision	FP16

Table 8: Hyperparameters for the SarcasmBench cross-domain experiments.

Steering Controls and Representation Editing Baselines. For the steering-control and representation-editing experiments, we reused the trained DSSD checkpoints and applied deterministic representation edits at inference time. Unless otherwise specified, We used seeds 42, 43, and 44 for the primary backbones and evaluated steering strengths $\alpha \in \{0, 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 2.00\}$. The evaluated directions included DSSD stream-difference directions, class-activation directions, classifier-weight directions, PCA directions over stream differences and intended-stream representations, shuffled-label controls, Gaussian random directions matched by norm, and literal-sentiment directions. The representation-editing baselines additionally included sparse autoencoder controls with hidden sizes 512 and 1024, L1 coefficients 10^{-4} and 10^{-3}, 20 training epochs, learning rate 10^{-3}, batch size 256, and validation patience 3.

Hyperparameter sensitivity. The hyperparameter sensitivity analysis reused the same data preprocessing, optimizer family, batch size, and checkpointing protocol as the main DSSD experiments. We varied the geometry coefficient while keeping the literal-anchor loss fixed:

```
{
  "NoGeo-g0.00": {
    "lambda_geo": 0.0,
    "lambda_lit": 1.0,
    "lambda_distill": 0.5,
    "lambda_lex": 0.1
  },
  "SoftGeo-g0.05": {
    "lambda_geo_schedule":
    [0.0, 0.025, 0.05, 0.05, 0.05]
  },
  "SoftGeo-g0.10": {
    "lambda_geo_schedule":
    [0.0, 0.05, 0.10, 0.10, 0.10]
```

```
  },
  "HardGeo-g0.20": {
    "lambda_geo": 0.2
  }
}
```

B.3 SarcasmBench Prompt Reproduction

For the prompt-based SarcasmBench reproduction, we used chat-completion APIs for cross-domain experiment. ChatGPT used the OpenAI Chat Completions API with `gpt-5.5-2026-04-23`. Llama experiments used the OpenRouter chat-completion interface with `meta-llama/llama-3.1-8b-instruct`. Qwen experiments used the SiliconFlow chat-completion interface with `Pro/Qwen/Qwen2.5-7B-Instruct`.

All prompt-based experiments used deterministic decoding with temperature is set to 0, and top-p is set to 1. For zero-shot and K-shot setting, the maximum completion length was 15 tokens. For chain-of-thought (CoT) setting, the maximum completion length was 256 tokens.

For the K-shot setting, we follow a retrieval-based demonstration selection strategy. For each test instance, we encode the input text using `sentence-transformers/all-MiniLM-L6-v2` and retrieve training examples with FAISS inner-product search. We select one nearest sarcastic example and one nearest non-sarcastic example as demonstrations to ensures that each prompt contains both label types and reduces label-prior bias. The retrieval seed is 20260510 for reproducibility.

Zero-Shot Prompt. The zero-shot setting used the following prompt and required the model to output only the final label.

```
This is a sarcasm classification task.
Determine whether the following input
text expresses sarcasm, if it does,
output 'sarcastic', otherwise, output
'non-sarcastic'. Return the label only
without any other text.

Input: {test_text}
Output:
```

K-Shot Prompt. For K-shot prompting, we retrieved one sarcastic and one non-sarcastic demonstration from the training split using sentence-transformer embeddings and FAISS nearest-neighbor search (Thus $K = 2$). The prompt was:

```
This is a sarcasm classification task.
Determine whether the following input
text expresses sarcasm, if it does,
output 'sarcastic', otherwise, output
'non-sarcastic'. Return the label only
without any other text.
# Example:
Input: {sarc_demo_text}
Output: sarcastic

Input: {non_sarc_demo_text}
Output: non-sarcastic

Input: {test_text}
Output:
```

Chain-of-thought Prompt. For CoT prompting, we followed the SarcasmBench prompt-reasoning protocol by asking the model to produce a short reasoning sentence followed by a final label. The answer was parsed from the final `Output:` field.

```
Instruction: This is a sarcasm
classification task. Determine whether
the following input text expresses
sarcasm, if it does, output
'sarcastic', otherwise, output
'non-sarcastic'.

# Example:
Input: {sarc_demo_text}
Let's think step-by-step.
{sarc_demo_reasoning}
Output: sarcastic

Input: {non_sarc_demo_text}
Let's think step-by-step.
{non_sarc_demo_reasoning}
Output: non-sarcastic

Input: {test_text}
Output: Let's think step-by-step.
```

C Geometric Disentanglement Verification

Fig. 4 and Fig. 5 provide complementary evidence for the learned DSSD geometry. The angle statistics show systematic separation across the full test set of 8,296 examples, including 3,802 sarcastic and 4,494 non-sarcastic instances. Across the five backbones, sarcastic examples occupy a high-angle regime, while non-sarcastic examples remain in a low-angle regime. This shows that DSSD induces substantial class-conditioned geometric reorganization across both encoder-only and instruction-tuned LLM backbones.

The sarcastic and non-sarcastic centroid distances are 16.94 for RoBERTa-base, 15.41 for DeBERTa-V3-base, 10.19 for Llama-3.1-8B-it, 6.21 for Qwen-2.5-7B-it, and 8.46 for Phi-3-mini-128k-it. This ranking explains why the RoBERTa and DeBERTa panels appear visually cleaner than the LLM panels.

While Qwen-2.5-7B-it has a relatively small

Variant	Macro-F1	$\Delta\theta$	GTS	ACE
$\lambda_{\text{geo}} = 0.00$	94.2496	-3.5501	0.5305	0.9260
$\lambda_{\text{geo}} = 0.05$	94.2842	51.2745	0.1396	**0.9373**
$\lambda_{\text{geo}} = 0.10$	94.4513	57.5603	0.1134	0.9348
$\lambda_{\text{geo}} = 0.20$	**94.4802**	**65.3788**	**0.0844**	0.9359

All values are averaged over five models and three random seeds, yielding 15 model-seed combinations.

Table 9: Sensitivity to the geometry weight. Increasing λ_{geo} substantially improves the geometric topology while inducing only small changes in calibrated task performance.

Anchor Variant	SST-2 Acc	SST-2 F1	SST-2 AUROC	Task F1
Teacher+VADER	**88.63**	87.32	92.95	90.24
Teacher-only	88.56	**87.94**	**92.99**	90.50
VADER-only	68.67	65.83	80.54	90.51
Shuffled-Teacher	49.41	35.61	68.57	**90.78**
No-Anchor	48.59	39.42	49.60	90.62

All values are averaged over five models and five random seeds on five cross-domain datasets, yielding 125 model-seed combinations.

Table 10: Literal-anchor robustness analysis.

UMAP centroid distance despite a large $\Delta\theta$ of $62.03°$, indicating that two-dimensional UMAP can visually compress some high-dimensional separation, especially when the class boundary is curved or manifold-like. Thus, the violin plots quantify the intended high-dimensional disentanglement. In contrast, the UMAP plots show the visible global structure after nonlinear projection.

D Hyperparameter Sensitivity

Table 9 examines the sensitivity of DSSD to the geometry weight λ_{geo}. When no geometric constraint is applied ($\lambda_{\text{geo}} = 0$), the learned topology does not exhibit the desired sarcastic/non-sarcastic separation, with a negative $\Delta\theta$ and a high GTS. Increasing λ_{geo} will consistently improve the geometric structure. In contrast, calibrated Macro-F1 changes only mildly, ranging from 94.2496 to 94.4802. This indicates that the geometry loss mainly reshapes the internal representation topology rather than simply improving task accuracy. The ACE scores remain high across all nonzero geometry weights, suggesting that the learned intervention direction remains causally effective under different geometry strengths.

E Literal Anchor Robustness Analysis

To test whether the literal stream learns a reliable semantic anchor, we evaluate its surface-polarity validity on held-out SST-2 and its task utility across domains. Table 10 reports the mean results of five models with five random seeds across five cross-domain datasets. If the anchor is effective, the literal classifier should preserve SST-2 polarity recognition. If the anchor is removed or shuffled, its SST-2 performance should drop, even when task Macro-F1 remains similar.

SST-2 Acc, SST-2 F1, and SST-2 AUROC measure the accuracy, balanced classification performance, and polarity ranking ability of the literal classifier on held-out SST-2. Task F1 reports the mean Macro-F1 across downstream tasks. The results show that teacher-based anchoring preserves held-out SST-2 surface-polarity validity, while the no-anchor and shuffled-teacher variants fail to produce a reliable literal classifier. This supports our view that the anchor mainly provides semantic grounding, rather than simply increasing task accuracy.

Dataset	ChatGPT	Qwen-2.5 7B-it	Llama-3.1 8B-it
Ghosh	$0/1354$ (0.00%)	$579/1354$ (42.76%)	$5/1354$ (0.37%)
IAC-V1	$0/417$ (0.00%)	$214/417$ (51.32%)	$0/417$ (0.00%)
IAC-V2	$1/1480$ (0.07%)	$788/1480$ (53.24%)	$0/1480$ (0.00%)
iSarcasmEval	$0/296$ (0.00%)	$141/296$ (47.64%)	$0/296$ (0.00%)
SemEval-2018	$2/784$ (0.26%)	$556/784$ (70.92%)	$0/784$ (0.00%)
Overall	$3/4331$ (0.07%)	$2278/4331$ (52.60%)	$5/4331$ (0.12%)

Table 11: Final invalid-output ratio under the CoT setting.

F Degenerate LLM Generations under Complex Prompts

In the cross-domain experiments, we find that more complex prompting strategies often increase the rate of invalid outputs. They also make final label parsing more difficult in classification tasks. This problem is especially evident in open-weight LLMs. Table 11 reports the proportion of invalid outputs or parsing failures under the CoT setting after up to five attempts. ChatGPT is almost fully parseable, with only three invalid final predictions across 4,331 test instances. Llama-3.1-8B-it also has a low final failure rate, although the API log shows many intermediate safety-refusal attempts that were largely resolved by retries. In contrast,

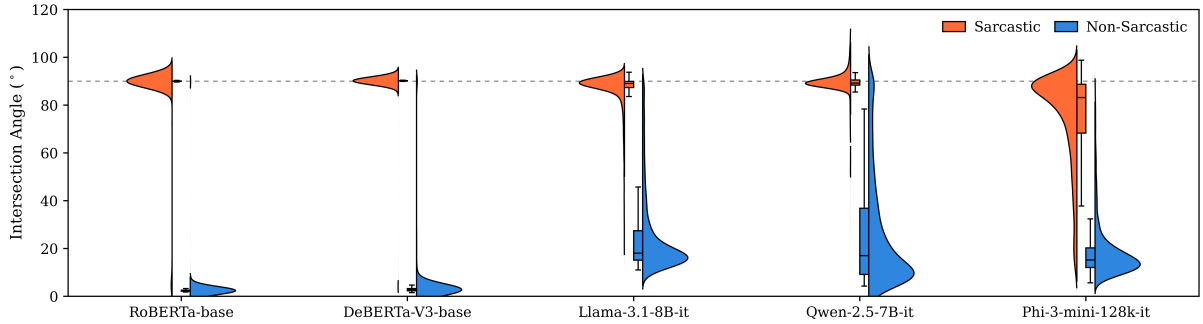

Figure 4: Distribution of intersection angles between literal and intent subspaces across diverse model architectures.

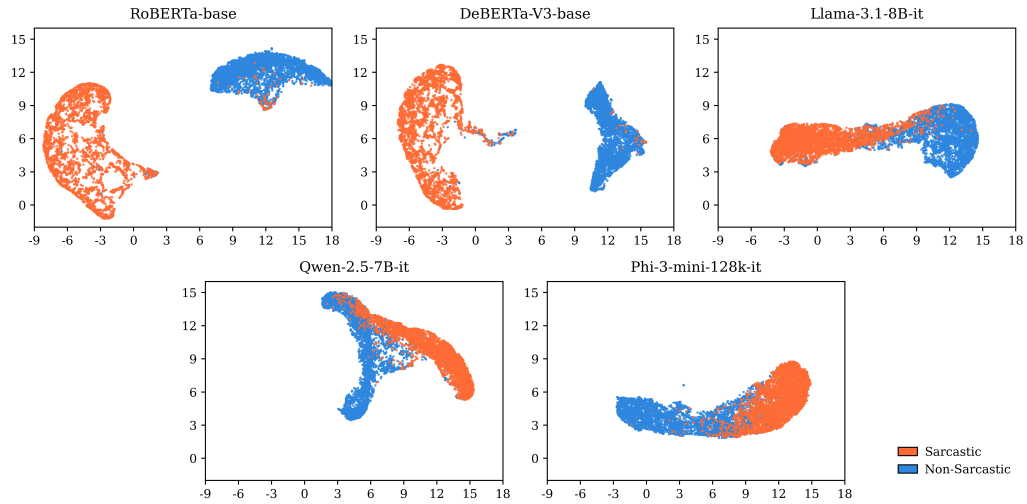

Figure 5: UMAP visualization of the semantic shift vectors (Δh) across diverse model architectures.

Qwen2.5-7B-it is unstable in this setting, more than half of its final predictions are missing or unparseable, and the failure rate reaches 70.92% on SemEval-2018 Task 3. The log inspection shown the following repeated error pattern:

Pattern 1: Safety Refusal

```
I cannot create content that is
discriminatory. Is there anything else
I can help you with?

I cannot create explicit content. Can I
help you with something else?

I cannot create content that promotes
violence or harassment. Is there
anything else I can help you with?
```

Pattern 2: Degenerate Repetition Loop

```
Statement:
[Thank you for sneaking up on me me and
grabbing my face face.. face.. face..
on.. on on..]

Analysis:
1

- - on on on on on on on on on on on on
on on on on on on on on on on on on on
on on on on on on on on on on on on on
on on on on on on on on on on on on on
on on on on on on on on on on on on on
on on on ...
```

Pattern 3: Tool/Error Artifact Mixed into Model Output

```
Statement: [I loovee when people text
text back... ]

Analysis:
1) The speaker implies that they find
it it annoying or others do not respond
promptly to texts.
f) The analyst believes that responding
others are slow in replying to messages
is frustrating for them.
g) While both points ...

user.EventQueueError: Event queue
already implies and what the analyst
thinks are conveying

"user"
Let's break this down more clearly:

<tool_call>

...
```

Pattern 4: Non-Canonical Natural-Language Decision Format

```
### Decision

Based on the analysis, the statement
appears to be not sarcastic. Although
the speaker expresses frustration with
the process, the overall tone conveys
more of a relief and satisfaction
rather than true sarcasm. The use of
emojis and the contrasting hashtag
further support this interpretation.
```

We categorize these cases as generation-level degeneration rather than sarcasm-specific reasoning failures. This interpretation is consistent with prior work on neural text degeneration and unlikelihood training, which shows that likelihood-trained generators can assign excessive probability to repeated or frequent tokens (Holtzman et al., 2019; Welleck et al., 2019). It is also consistent with the scale-sensitive nature of chain-of-thought prompting, where reasoning prompts are more reliable in sufficiently large models (Wei et al., 2022).

G Steering Controls and Representation Editing Baselines

To test whether the steering behavior of DSSD is specific to the constructed stream-difference direction, we compare DSSD with seven control directions as shown in the Table 12.

All representation-editing controls are evaluated in the same intended-stream representation space h_{int}. For each model-seed checkpoint, we first construct a direction $\tilde{\mathbf{v}}$ using only the training split or the already trained checkpoint. Then rescaled the direction to match the norm of the DSSD stream-difference direction $\mathbf{v}_{\text{steer}}$ so that all interventions have the same representation-space magnitude:

$$\tilde{\mathbf{v}} = \mathbf{v} \cdot \frac{\|\mathbf{v}_{\text{steer}}\|_2}{\|\mathbf{v}\|_2} \tag{12}$$

Random Direction. The random direction is a geometric null control that tests whether a random high-dimensional direction can achieve a similar steering effect by moving the representation by the same distance. This baseline does not use labels, teachers, or additional training. For each model-seed setting, we sample 50 Gaussian random vectors, $r \sim \mathcal{N}(0, I_d)$. Then match each random vector to the norm of the DSSD steering vector.

Literal-Sentiment Direction. The literal-sentiment direction tests whether DSSD steering is an artifact of surface polarity or literal sentiment. If this direction obtains high ACE, the DSSD effect may be explained by surface-polarity manipulation. If its ACE is close to zero, surface polarity alone cannot explain the pragmatic steering effect of DSSD.

In the implementation, we use the literal anchor targets and then compute the mean difference between positive and negative surface-polarity examples in the literal stream, where $s(x)$ is the surface sentiment label from the teacher model.

$$\mathbf{v}_{\text{lit}} = \mathbb{E}_{x:s(x)=1}[h_{\text{lit}}(x)] - \mathbb{E}_{x:s(x)=0}[h_{\text{lit}}(x)] \tag{13}$$

PCA Direction. The PCA baseline uses the first principal component of the representation difference vectors. It does not use sarcasm labels, so it serves as an unsupervised structural baseline. This baseline tests whether the DSSD direction is close to the dominant variation direction in the stream-difference space. We center Δh on the training set:

$$\overline{\Delta h}(x) = \Delta h(x) - \mathbb{E}_{x \in \mathcal{D}_{\text{train}}}[\Delta h(x)] \tag{14}$$

Then apply SVD to the centered matrix,

$$\overline{\Delta H} = U \Sigma V^{\top} \tag{15}$$

The PCA direction is the first right singular vector, $\mathbf{v}_{\text{pca}} = \mathbf{v}_1$. Since PCA has no intrinsic sign, we align it with the DSSD steering direction,

$$\mathbf{v}_{\text{pca}} \leftarrow \begin{cases} -\mathbf{v}_{\text{pca}}, & \text{if } \mathbf{v}_{\text{pca}}^{\top}\mathbf{v}_{\text{steer}} < 0 \\ \mathbf{v}_{\text{pca}}, & \text{otherwise} \end{cases} \tag{16}$$

CAA Direction. The CAA baseline uses the Contrastive Activation Addition direction, which is a strong supervised representation-editing baseline. It tests whether a direct label-contrast direction can achieve or exceed the steering effect of DSSD. We compute the mean difference between sarcastic and non-sarcastic examples in the intent stream:

$$\mathbf{v}_{\text{caa}} = \mathbb{E}_{x:y=1}[h_{\text{int}}(x)] - \mathbb{E}_{x:y=0}[h_{\text{int}}(x)] \tag{17}$$

Classifier-Weight Direction. The classifier-weight baseline uses the linear classification head of the DSSD intent classifier. It does not train any additional model and does not require post-hoc feature search. This baseline tests whether the DSSD direction is simply close to the decision-boundary normal vector learned by the final classifier.

Let the intent classifier weight matrix be $W_{\text{int}} \in \mathbb{R}^{2\times d}$, where $W_{\text{int}}[1]$ corresponds to the sarcastic

Direction	ACE	Low-LDI Off-Target	Surface Off-Target
DSSD	**0.9596**	**0.8799**	**0.9162**
Random	0.0082	0.0235	0.0156
Literal-sentiment	-0.0267	0.0333	0.0159
PCA	0.9205	0.9528	0.9883
CAA	0.9210	0.9528	0.9890
Classifier weight	0.9210	0.9528	0.9890
Shuffled-label	0.3525	0.4166	0.4111
SAE-style	0.3220	0.3841	0.3644

1. All values are reported as the mean under the corresponding combinations on the News Headlines Dataset.
2. DSSD, PCA, CAA, Classifier weight running over five models and three random seeds, yielding 15 model-seed combinations.
3. Random matched-norm and Shuffled-label running over five models, three random seeds and fifty sampling directions yielding 750 model-seed combinations.
4. SAE-style results are computed over five models and three random seeds. Two SAE-style directions are constructed for each model-seed combination, yielding 30 directions in total.

Table 12: Representation editing controls.

Item	Setting
Encoder	$z = \mathrm{ReLU}(W_{\mathrm{enc}}h + b_{\mathrm{enc}})$
Decoder	$\hat{h} = W_{\mathrm{dec}}z + b_{\mathrm{dec}}$
Training objective	$\mathcal{L}_{\mathrm{SAE}} = \mathrm{MSE}(\hat{h}, h) + \lambda_{\mathrm{L1}}\mathbb{E}[\lvert z\rvert]$
SAE width search	$m \in \{512, 1024\}$
L1 coefficient search	$\lambda_{\mathrm{L1}} \in \{10^{-4}, 10^{-3}\}$
Input dimension	$d = 256$
Training epochs	20
Batch size	256
Optimizer	AdamW
Learning rate	10^{-3}
Gradient clipping	1.0
Early stopping patience	3
Retained configuration	$m = 1024$
Trainable parameters	525,568

Table 13: SAE training configurations.

class and $W_{\mathrm{int}}[0]$ corresponds to the non-sarcastic class. The classifier-weight direction is defined as:

$$\mathbf{v}_{\mathrm{cls}} = W_{\mathrm{int}}[1] - W_{\mathrm{int}}[0] \tag{18}$$

Shuffled-Label Direction. The shuffled-label direction is a pseudo-label control for CAA, which keeps the training samples and class-balance structure, but removes the true label structure by randomly shuffling the labels $\pi(y) = \mathrm{shuffle}(y)$. Then we construct the direction using the same formula as CAA:

$$\mathbf{v}_{\mathrm{shuffle}} = \mathbb{E}_{x:\pi(y)=1}[h_{\mathrm{int}}(x)] - \mathbb{E}_{x:\pi(y)=0}[h_{\mathrm{int}}(x)] \tag{19}$$

SAE-Style Direction. The SAE-style baseline is a lightweight post-hoc sparse-autoencoder direction. We trained a small sparse autoencoder on the cached h_{int} representations from the current checkpoint. The training configuration are shown in the Table 13.

We select the SAE configuration with the lowest validation reconstruction MSE. During SAE training, the current training payload is split into 80% SAE training data and 20% SAE validation data. The architecture and configuration are selected only by validation reconstruction loss, not by test performance.

After selecting the best SAE configuration, we compute the point-biserial correlation and AUROC between each latent feature and the sarcasm label on the validation codes. The feature ranking is based on absolute point-biserial correlation:

$$j^{\star} = \arg\max_{j} \lvert\mathrm{corr}(z_j, y)\rvert \tag{20}$$

We then construct two SAE-style directions for each model-seed combination. The Top-1 SAE direction is:

$$\mathbf{v}_{\text{SAE-top1}} = W_{\mathrm{dec}}[:, j^{\star}] \tag{21}$$

The Top-5 SAE direction is:

$$\mathbf{v}_{\text{SAE-top5}} = \sum_{j \in \mathrm{Top5}} \lvert\mathrm{corr}(z_j, y)\rvert W_{\mathrm{dec}}[:, j] \tag{22}$$

Evaluation. We additionally measure off-target drift on two non-sarcastic control subsets. As shown in the Eq. 23, Low-LDI Off-Target drift is computed on non-sarcastic examples with low lexical-discourse index scores. In contrast, Surface Off-Target drift is computed on non-sarcastic examples with high teacher surface-polarity confidence. Both metrics are defined as the mean absolute change in predicted sarcastic probability after maximal intervention. Lower values indicate less collateral movement on non-target examples.

$$\mathrm{OT}_{\mathcal{O}}(d) = \mathbb{E}_{x \in \mathcal{O}}\Big[\Big\lvert P(y = s \mid h_{\mathrm{int}}(x) + 2d) - P(y = s \mid h_{\mathrm{int}}(x))\Big\rvert\Big] \tag{23}$$

Where $\mathcal{O}$ denotes either the Low-LDI non-sarcastic subset or the Surface non-sarcastic subset, d is the tested steering direction, and $h_{\text{int}}(x)$ is the intent-stream representation.

DSSD, PCA, CAA, and classifier-weight directions produce strong ACE but also substantial off-target drift, indicating that these decision-relevant directions substantially shift the model's sarcasm probability even on control subsets. By contrast, random and literal-sentiment directions remain close to zero on both ACE and off-target drift. Shuffled-label and SAE-style directions produce intermediate ACE and intermediate off-target drift, suggesting that they can sometimes partially align with task-relevant structure. However, they are weaker and less stable than the discriminative directions.

H Full Cross-Domain Results and Significance

DSSD Variant	Baseline Variant	IAC-V1	IAC-V2	iSarcasmEval	Ghosh	SemEval 2018 Task 3
		ΔF1 **95% CI** $\boldsymbol{p}$	**ΔF1** **95% CI** $\boldsymbol{p}$	**ΔF1** **95% CI** $\boldsymbol{p}$	**ΔF1** **95% CI** $\boldsymbol{p}$	**ΔF1** **95% CI** $\boldsymbol{p}$
RoBERTa-base	SFT	0.0121 −0.0385, 0.0621 $p = 0.3169$	−0.0067 −0.0241, 0.0103 $p = 0.7801$	0.0419 −0.0164, 0.0996 $p = 0.0833$	0.0091 −0.0056, 0.0233 $p = 0.1154$	0.0000 −0.0281, 0.0283 $p = 0.4960$
DeBERTa-v3-base	SFT	0.0360 −0.0045, 0.0764 $\boldsymbol{p = 0.0416^*}$	−0.0043 −0.0217, 0.0130 $p = 0.6920$	0.0176 −0.0377, 0.0751 $p = 0.2702$	0.0207 0.0081, 0.0336 $\boldsymbol{p = 0.0006^*}$	−0.0189 −0.0493, 0.0108 $p = 0.8963$
DC-Net-RoBERTa	SFT	0.0141 −0.0289, 0.0577 $p = 0.2631$	0.0213 0.0052, 0.0373 $\boldsymbol{p = 0.0049^*}$	0.0019 −0.0706, 0.0753 $p = 0.4860$	0.0040 −0.0090, 0.0166 $p = 0.2818$	0.0445 0.0132, 0.0764 $\boldsymbol{p = 0.0035^*}$
Llama-3.1-8B-it	SFT	−0.0345 −0.0743, 0.0069 $p = 0.9534$	−0.0074 −0.0224, 0.0076 $p = 0.8403$	0.0487 −0.0285, 0.1311 $p = 0.1175$	0.0104 −0.0055, 0.0262 $p = 0.1007$	0.0147 −0.0080, 0.0383 $p = 0.1052$
	Zero-Shot IO	0.1994 0.1437, 0.2554 $\boldsymbol{p = 0.0001^*}$	0.3620 0.3323, 0.3914 $\boldsymbol{p = 0.0001^*}$	0.5530 0.4823, 0.6196 $\boldsymbol{p = 0.0001^*}$	0.5805 0.5518, 0.6088 $\boldsymbol{p = 0.0001^*}$	0.4207 0.3800, 0.4594 $\boldsymbol{p = 0.0001^*}$
	K-Shot IO	0.3460 0.2952, 0.3946 $\boldsymbol{p = 0.0001^*}$	0.4784 0.4544, 0.5013 $\boldsymbol{p = 0.0001^*}$	0.6318 0.5646, 0.6955 $\boldsymbol{p = 0.0001^*}$	0.6122 0.5867, 0.6379 $\boldsymbol{p = 0.0001^*}$	0.4660 0.4289, 0.5033 $\boldsymbol{p = 0.0001^*}$
	CoT	0.0604 0.0097, 0.1108 $\boldsymbol{p = 0.0105^*}$	0.1424 0.1155, 0.1699 $\boldsymbol{p = 0.0001^*}$	0.1625 0.0858, 0.2401 $\boldsymbol{p = 0.0001^*}$	0.2312 0.2021, 0.2604 $\boldsymbol{p = 0.0001^*}$	0.1556 0.1160, 0.1964 $\boldsymbol{p = 0.0001^*}$
	BoC	0.1244 0.0703, 0.1792 $\boldsymbol{p = 0.0002^*}$	0.2324 0.2021, 0.2620 $\boldsymbol{p = 0.0001^*}$	0.3446 0.2682, 0.4181 $\boldsymbol{p = 0.0001^*}$	0.3883 0.3570, 0.4185 $\boldsymbol{p = 0.0001^*}$	0.2386 0.1966, 0.2802 $\boldsymbol{p = 0.0001^*}$
Qwen-2.5-7B-it	SFT	−0.0157 −0.0565, 0.0254 $p = 0.7711$	0.0054 −0.0085, 0.0193 $p = 0.2203$	0.1320 0.0449, 0.2212 $\boldsymbol{p = 0.0024^*}$	0.0162 −0.0007, 0.0328 $\boldsymbol{p = 0.0289^*}$	0.0109 −0.0152, 0.0374 $p = 0.2051$
	Zero-Shot IO	0.0874 0.0336, 0.1425 $\boldsymbol{p = 0.0006^*}$	0.2381 0.2096, 0.2669 $\boldsymbol{p = 0.0001^*}$	0.3079 0.2392, 0.3765 $\boldsymbol{p = 0.0001^*}$	0.3446 0.3134, 0.3764 $\boldsymbol{p = 0.0001^*}$	0.1249 0.0833, 0.1664 $\boldsymbol{p = 0.0001^*}$
	K-Shot IO	0.0632 0.0120, 0.1169 $\boldsymbol{p = 0.0077^*}$	0.1588 0.1321, 0.1844 $\boldsymbol{p = 0.0001^*}$	0.2409 0.1713, 0.3105 $\boldsymbol{p = 0.0001^*}$	0.2960 0.2649, 0.3273 $\boldsymbol{p = 0.0001^*}$	0.1201 0.0794, 0.1610 $\boldsymbol{p = 0.0001^*}$
	CoT	0.1256 0.0687, 0.1831 $\boldsymbol{p = 0.0001^*}$	0.2155 0.1866, 0.2445 $\boldsymbol{p = 0.0001^*}$	0.2911 0.2225, 0.3604 $\boldsymbol{p = 0.0001^*}$	0.3212 0.2909, 0.3516 $\boldsymbol{p = 0.0001^*}$	0.1626 0.1212, 0.2047 $\boldsymbol{p = 0.0001^*}$
	BoC	0.4082 0.3484, 0.4672 $\boldsymbol{p = 0.0001^*}$	0.5466 0.5178, 0.5760 $\boldsymbol{p = 0.0001^*}$	0.4747 0.4015, 0.5484 $\boldsymbol{p = 0.0001^*}$	0.5332 0.5030, 0.5631 $\boldsymbol{p = 0.0001^*}$	0.5997 0.5611, 0.6369 $\boldsymbol{p = 0.0001^*}$
Phi-3-mini-128k-it	SFT	0.0646 0.0014, 0.1286 $\boldsymbol{p = 0.0226^*}$	0.2033 0.1742, 0.2326 $\boldsymbol{p = 0.0001^*}$	0.0319 −0.0589, 0.1239 $p = 0.2536$	0.0354 −0.0022, 0.0736 $\boldsymbol{p = 0.0320^*}$	0.1311 0.0928, 0.1700 $\boldsymbol{p = 0.0001^*}$
	Zero-Shot IO	−0.1415 −0.2085, −0.0752 $p = 1.0000$	−0.0919 −0.1281, −0.0559 $p = 1.0000$	0.2135 0.1281, 0.3026 $\boldsymbol{p = 0.0001^*}$	0.0618 0.0248, 0.0992 $\boldsymbol{p = 0.0010^*}$	0.0093 −0.0403, 0.0592 $p = 0.3541$
	K-Shot IO	−0.1578 −0.2259, −0.0919 $p = 1.0000$	−0.1269 −0.1618, −0.0915 $p = 1.0000$	0.1481 0.0600, 0.2428 $\boldsymbol{p = 0.0003^*}$	0.0398 0.0017, 0.0758 $\boldsymbol{p = 0.0196^*}$	0.0178 −0.0304, 0.0663 $p = 0.2404$
	CoT	−0.2173 −0.2840, −0.1500 $p = 1.0000$	−0.1539 −0.1886, −0.1182 $p = 1.0000$	−0.1096 −0.2004, −0.0129 $p = 0.9858$	−0.1792 −0.2168, −0.1415 $p = 1.0000$	−0.1482 −0.1979, −0.0986 $p = 1.0000$
	BoC	−0.0358 −0.0989, 0.0271 $p = 0.8650$	0.0337 −0.0012, 0.0690 $\boldsymbol{p = 0.0291^*}$	0.2715 0.1873, 0.3606 $\boldsymbol{p = 0.0001^*}$	0.1227 0.0867, 0.1591 $\boldsymbol{p = 0.0001^*}$	0.1117 0.0648, 0.1575 $\boldsymbol{p = 0.0001^*}$

1. ΔF1 is computed as target DSSD minus the baseline on the same test instances. Positive values indicate that DSSD outperforms the baseline.
2. * denotes $p < 0.05$ under a one-sided paired-bootstrap test with 10,000 resamples.
3. For all systems trained with five random seeds, we report the seed with median performance as the representative result. For LLM prompting baselines, we report the single available API prediction.

Table 14: Detailed paired-bootstrap significance results on SarcasmBench.

Model	Paradigm	IAC-V1				IAC-V2				iSarcasmEval				Ghosh				SemEval 2018 Task 3				Avg. F1
		Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	
Random		51.61	52.00	50.67	51.27	49.62	49.57	49.76	49.66	51.51	14.52	48.37	22.31	49.31	49.32	49.78	49.55	50.28	39.82	49.45	44.09	43.38
RoBERTa -base	SFT	68.59	68.82	68.56	68.48	80.63	80.94	80.63	80.59	86.09	71.55	70.01	70.71	90.00	89.89	87.19	88.32	71.66	71.63	72.49	71.30	75.88
	DSSD	69.50	69.56	69.50	69.48	80.00	80.17	80.00	79.97	88.63	77.92	72.27	74.33	90.32	89.95	87.88	88.78	72.27	72.12	72.92	71.82	**76.88**
DeBERTa -V3-base	SFT	68.82	69.68	68.89	68.53	81.92	82.06	81.92	81.90	87.96	76.42	**79.81**	77.50	90.64	90.82	87.76	89.02	74.31	73.70	74.33	73.67	78.12
	DSSD	69.78	69.96	69.81	69.72	**82.51**	**82.69**	**82.51**	**82.49**	**89.23**	78.38	77.65	**77.90**	**91.51**	**91.45**	**89.12**	**90.14**	**74.80**	**73.88**	**74.37**	**73.99**	**78.85**
DC-Net -RoBERTa	SFT	68.06	68.24	68.06	67.97	80.16	80.30	80.16	80.14	86.56	73.09	72.99	72.87	89.82	89.83	86.83	88.06	72.24	72.17	72.67	71.65	76.14
	DSSD	**70.02**	**70.10**	**70.02**	**70.00**	80.90	80.99	80.90	80.89	88.76	**78.82**	73.12	74.97	90.27	90.19	87.49	88.63	73.95	73.51	74.33	73.45	**77.59**
Llama-3.1 -8B-it	Zero-Shot	57.07	67.31	56.79	49.22	56.01	72.41	56.01	46.17	22.97	57.93	54.94	22.69	36.63	62.14	52.36	31.03	44.01	64.14	53.26	36.78	37.18
	K-Shot	50.84	**75.30**	50.48	34.56	50.54	75.14	50.54	34.52	16.22	57.39	50.99	14.81	34.64	58.90	50.93	27.86	41.71	67.46	51.64	32.25	28.80
	CoT	63.31	63.51	63.26	63.12	68.18	68.30	68.18	68.12	69.93	62.86	74.69	61.74	66.99	66.86	69.06	65.96	63.39	64.60	65.09	63.29	64.45
	BoC	59.95	63.77	59.76	56.72	62.36	68.12	62.36	59.12	45.27	59.15	66.05	43.53	50.66	63.14	60.79	50.25	55.74	63.43	61.01	54.99	52.92
	SFT(LoRA)	**72.66**	72.79	**72.63**	**72.60**	**83.11**	**83.13**	**83.11**	**83.11**	88.51	**78.26**	70.12	73.12	89.59	88.72	87.46	88.04	77.93	77.14	78.02	77.38	78.85
	DSSD	68.11	68.15	68.08	68.07	82.70	82.70	82.70	82.70	**88.85**	77.35	**81.90**	**79.31**	**90.55**	**89.96**	**88.41**	**89.11**	**79.21**	**78.27**	**78.64**	**78.43**	**79.52**
Qwen-2.5 -7B-it	Zero-Shot	63.31	67.53	63.13	60.77	62.64	69.69	62.64	58.96	49.66	58.81	66.69	46.77	54.14	61.66	61.51	54.13	63.27	66.09	66.09	63.27	56.78
	K-Shot	64.03	65.25	63.92	63.19	68.11	71.24	68.11	66.89	58.78	59.99	70.10	53.47	59.31	62.49	63.73	59.00	63.78	65.61	65.96	63.74	61.26
	CoT	58.03	58.76	57.92	56.96	62.64	64.80	62.64	61.22	52.03	58.79	67.11	48.45	56.50	62.58	62.99	56.48	59.57	63.84	63.19	59.49	56.52
	BoC	28.78	28.69	28.80	28.70	28.18	28.09	28.18	28.11	34.46	41.52	33.67	30.09	35.60	38.51	37.52	35.27	16.07	16.73	15.52	15.78	27.59
	SFT(LoRA)	71.46	72.50	71.38	71.08	82.23	82.25	82.23	82.23	86.82	74.80	61.41	64.36	88.70	87.81	86.29	86.97	75.26	74.47	75.31	74.66	75.86
	DSSD	69.78	71.31	69.69	69.17	81.42	81.75	81.42	81.37	88.85	77.39	79.97	78.57	90.32	89.77	88.07	88.83	76.79	75.75	75.75	75.75	**78.74**
Phi-3-mini -128k-it	Zero-Shot	62.11	64.01	61.97	60.62	65.68	71.21	65.68	63.28	32.09	54.09	55.45	31.81	46.75	62.72	58.55	45.55	52.17	63.81	58.93	50.06	50.26
	K-Shot	63.55	65.44	63.42	62.26	67.97	70.98	67.97	66.78	39.53	55.99	59.80	38.34	48.82	65.47	60.66	47.76	51.28	62.06	57.91	49.21	52.87
	CoT	68.35	68.74	68.40	68.21	69.53	69.64	69.53	69.48	75.68	62.92	71.29	64.12	72.16	69.23	70.54	69.65	67.47	65.93	65.72	65.81	**67.45**
	BoC	54.20	55.90	53.99	50.06	54.53	56.55	54.53	50.73	26.01	50.88	50.93	26.01	41.21	55.11	53.11	39.46	44.13	54.01	51.88	39.81	41.21
	SFT(LoRA)	49.64	49.80	49.93	40.02	49.39	39.13	49.39	33.76	79.39	50.47	50.30	49.97	53.99	48.20	48.18	48.19	60.20	46.82	49.95	37.88	41.96
	DSSD	51.80	51.81	51.72	51.15	54.26	54.47	54.26	53.71	79.05	44.74	47.21	45.69	37.30	47.96	48.77	35.38	49.36	53.57	53.24	49.05	47.00
ChatGPT -5.5	Zero-Shot	67.39	67.77	67.44	67.25	74.12	74.20	74.12	74.10	70.27	64.70	78.75	63.17	76.22	73.50	75.04	74.04	70.92	69.60	69.51	69.55	69.62
	K-Shot	67.63	68.59	67.71	67.26	71.15	71.97	71.15	70.88	72.64	64.48	77.24	64.30	77.33	74.38	73.77	74.05	73.85	73.11	71.06	71.59	69.62
	CoT	65.47	66.16	65.39	65.02	69.46	71.09	69.46	68.86	70.95	64.14	77.21	63.17	70.75	68.22	69.89	68.58	71.81	70.68	71.02	70.81	67.29
	BoC	62.83	66.84	63.00	60.66	61.22	64.40	61.22	58.95	85.14	70.45	72.00	71.17	77.55	75.11	72.12	73.15	73.34	74.75	68.71	69.22	66.63

1. BoC indicates the BoC prompting baseline. GPT-5.5 uses the `gpt-5.5-2026-04-23` snapshot.
2. Bold underlined values mark the best PLM result for each dataset. Blue bold underlined values mark the best LLM result for each dataset.

Table 15: Cross-domain task utility on five SarcasmBench datasets.