SCOUT: Selective Coupling via Optimal Unbalanced Transport for Interpretable Text Classification

Junhao Jia[1, 2, 3], **Hanwen Zheng**[3], **Yueyi Wu**[3], **Huangwei Chen**[1, 2, 3], **Haishuai Wang**[1], **Jiajun Bu**[1, †], **Lei Wu**[1, 2, †]

[1]Zhejiang Key Laboratory of Accessible Perception and Intelligent Systems, College of Computer Science and Technology, Zhejiang University
[2]Hangzhou Pujian Medical Technology Co., Ltd, China
[3]School of Computer Science and Technology, Hangzhou Dianzi University

[†]**Correspondence:** bjj@zju.edu.cn, shenhai1895@zju.edu.cn

Abstract

Natural language data is inherently noisy, yet standard interpretable models often rely on scalar similarities that obscure the true evidentiary basis of a prediction. This limitation is particularly detrimental to prototype-based classification, where traditional full-alignment mechanisms force non-informative background segments to match informative prototypes, yielding unstable or misleading explanations. To mitigate this, we present SCOUT, a novel paradigm that grounds prototype reasoning in the selective correspondence of discriminative fragments. Concretely, we represent each document as a discrete distribution over span embeddings and employ differentiable Unbalanced Optimal Transport (UOT) to align them with class-specific prototypes. Unlike standard methods, this mechanism enables the model to focus strictly on decisive evidence while leaving irrelevant noise unmatched via geometric mass suppression. To ensure verifiability, we anchor prototype supports to readable training spans, establishing a transparent bridge between input segments and stored knowledge. Comprehensive experiments on seven benchmarks demonstrate that SCOUT yields prototypes focused on semantically significant spans, significantly outperforming traditional rationale extraction and post-hoc attribution methods in terms of faithfulness and stability.

1 Introduction

Text classification systems are increasingly deployed in high-stakes environments such as medical triage and legal discovery (Rudin, 2019; Caruana et al., 2015). These settings require a model to offer verifiable accounts of its decision-making process alongside high predictive accuracy (Doshi-Velez and Kim, 2017; Miller, 2019). While post-hoc explanation methods like LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) are widely used, they provide approximations of black-box behaviors rather than explaining the actual internal mechanics of the model (Guidotti et al., 2018; Rudin, 2019). This disconnect leads to faithfulness issues where the generated explanation fails to reflect the true evidence relied upon by the classifier (Lyu et al., 2024; Jacovi and Goldberg, 2020).

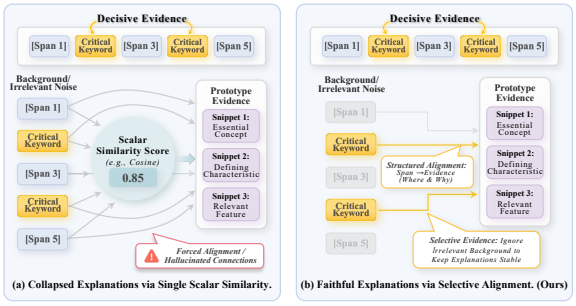

Figure 1: Alignment mechanisms for prototype-based explanation. (a) Scalar similarity collapses span-level structure, often forcing irrelevant text to match prototypes and producing spurious evidence. (b) SCOUT performs selective alignment via Unbalanced Optimal Transport, matching only decisive spans while suppressing background noise.

Prototype-based learning has emerged as a promising direction for intrinsic interpretability to bridge this gap. Instead of reasoning in an abstract latent space, prototype networks classify an input by comparing it to a set of learnable representative examples (Chen et al., 2019; Das et al., 2022). However, a critical flaw remains in how these comparisons are performed. Most existing approaches rely on scalar similarity measures like cosine similarity to summarize the relationship between the input document and the prototype. As illustrated in Figure 1a, this scalar compression creates an information bottleneck that indicates an input is similar

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6413–6431
July 2-7, 2026

to a prototype but obscures where and why they align. Consequently, the burden of interpretation shifts back to the user who must guess which parts of the input text triggered the high similarity score.

We contend that true interpretability requires explicit structured alignment rather than vague scalar proximity. A faithful explanation should demonstrate exactly which spans in the input correspond to which evidence in the prototype. Optimal Transport offers a principled mathematical framework for this purpose capable of finding the minimum-cost movement of probability mass between two distributions (Peyré et al., 2019; Cuturi, 2013; Swanson et al., 2020a). By treating text as a distribution of spans, Optimal Transport can produce a transport plan that explicitly maps input fragments to prototype supports.

However, applying standard Optimal Transport to text classification presents the unique challenge of noise. Natural language documents contain pervasive background information irrelevant to the classification task. Standard balanced Optimal Transport enforces strict conservation of mass and compels the model to align every part of the input to some part of the prototype. This forced alignment generates hallucinated connections where irrelevant input noise is matched to prototype features simply to satisfy mathematical constraints.

In this paper, we propose SCOUT as a novel framework that reimagines prototype-based classification through Unbalanced Optimal Transport (Chizat et al., 2018; Arase et al., 2023). Unlike standard methods, Unbalanced Optimal Transport relaxes marginal constraints and allows the model to leave irrelevant input spans unmatched. As shown in Figure 1b, this mechanism transforms the transport plan into a selective attention map that concentrates mass strictly on decisive evidence while filtering out background noise via geometric mass suppression. Furthermore, we enforce an anchoring constraint that projects learnable prototype supports onto readable spans from the training corpus to ensure verifiability. The result is a fully transparent decision process where the model predicts a class if and only if it can establish a strong transport coupling between specific input spans and validated prototype evidence.

Our contributions are summarized as follows:

- We propose a fine-grained alignment paradigm that replaces black-box scalar similarities with structured Optimal Transport couplings, making the evidence matching process explicit and verifiable.
- We introduce a noise-robust matching mechanism via Unbalanced Optimal Transport. By enabling partial matching, this strategy prevents the forced alignment of irrelevant text and significantly improves explanation faithfulness.
- Experiments on seven benchmarks demonstrate that SCOUT achieves competitive performance while generating compact and stable explanations, significantly outperforming state-of-the-art interpretable baselines.

2 Related Work

2.1 The Limits of Post-hoc Attribution

Explainability in NLP divides broadly into post-hoc and intrinsic approaches. Post-hoc methods like LIME, SHAP, and Integrated Gradients attempt to approximate the behavior of a black-box model by assigning importance scores to input tokens (Ribeiro et al., 2016; Lundberg and Lee, 2017; Sundararajan et al., 2017). Despite their wide applicability, these methods face severe criticism regarding faithfulness which refers to the degree to which an explanation accurately reflects the decision-making process of the model (Jacovi and Goldberg, 2020; Lyu et al., 2024). Empirical studies show that attention weights and gradient maps can be unstable under perturbations and often fail to serve as valid causal explanations (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Chuang et al., 2026). Furthermore, these attribution scores provide only a heatmap of importance but do not reveal the reasoning logic. Recent work further argues that trustworthy deployment demands principled reasoning and controlled abstention rather than opaque confidence (Sun et al., 2025; Chen et al., 2025). This limitation motivates the shift toward models that are intrinsically interpretable by design.

2.2 Prototype-Based Reasoning

Intrinsic interpretability aims to build transparency directly into the model architecture. Rationalization models select a subset of input text or rationale to pass to a predictor but do not necessarily explain the semantic matching process (Lei et al., 2016; Bastings et al., 2019; Chen et al., 2018). Prototype learning offers a transparent decision rule where inputs are classified based on their similarity to learn-

able representative examples. Pioneered by ProtoPNet in computer vision (Chen et al., 2019; Jia et al., 2025), this paradigm has since been extended along geometric and causal directions, including Stiefel-manifold grounding that resists neural collapse (Jia et al., 2026a) and unsupervised causal prototypical networks for de-biased interpretable diagnosis (Jia et al., 2026b). The paradigm has been adapted to NLP by models like ProtoTEx (Das et al., 2022), ProtoryNet (Hong et al., 2023), and recently ProtoSiTex for multi-label classification (Nareti et al., 2025). Most recently, ProtoLens advanced this field by conducting comparisons at the span level to capture fine-grained semantics (Wei and Zhu, 2025). This fine-grained alignment philosophy echoes parallel progress in compositional and cross-modal learning, where concept refinement (Zhang et al., 2025), primitive decoupling (Zhang et al.), and dual-branch hybrid discrimination (Jiang et al., 2025) yield more faithful matches than holistic similarity. However, these approaches typically rely on scalar similarity measures aggregated via Softmax. We identify a critical gap where standard soft-attention enforces mass conservation and compels the model to assign non-zero weights even to irrelevant noise. This issue necessitates a geometric alignment approach that ensures stability under input perturbations (Sourati et al., 2024).

2.3 Optimal Transport for Semantic Matching

Optimal Transport provides a principled framework for comparing distributions (Peyré et al., 2019; Lee et al., 2022; Zhang et al., 2024). In NLP, Word Mover's Distance applied Optimal Transport for document similarity (Kusner et al., 2015; Swanson et al., 2020b). Recent works have explored Optimal Transport for scalability (Xie et al., 2019) and structure learning in time-series prototypes (Huang et al., 2025; Snelgar et al., 2025). Closely related to our domain, Gurumoorthy et al. (2021) proposed a framework using Optimal Transport to select prototypes from a candidate set. Unlike their work which uses Optimal Transport as a selection algorithm during training, SCOUT employs Unbalanced Optimal Transport as the inference mechanism for every prediction. Furthermore, Wu et al. (2023) introduced MProto for distantly supervised NER utilizing Optimal Transport for denoising (Zheng et al., 2023). Our work differs in execution as we propose a span-based Unbalanced Optimal Transport layer explicitly for interpretable document classification that leverages mass suppression to filter background noise dynamically (Chizat et al., 2018).

3 Method

As shown in Figure 2, SCOUT represents a document as weighted spans and applies Unbalanced Optimal Transport to selectively align them with class prototypes, producing a sparse coupling used for both prediction and interpretable evidence.

3.1 Document Representation via Spans

To enable fine-grained alignment, we first decompose the input document x into interpretable units rather than a holistic vector. Let E denote a neural encoder such as BERT that maps x to contextual token embeddings $H = \{h_1, \ldots, h_n\}$. We enumerate candidate spans $\mathcal{S} = \{(l_i, r_i)\}_{i=1}^{M}$ using sliding windows and pool them into span embeddings $s_i \in \mathbb{R}^d$.

Crucially, we represent the document as a discrete probability measure μ_x defined as:

$$\mu_x = \sum_{i=1}^{M} a_i \delta_{s_i}, \quad \text{s.t.} \sum_{i=1}^{M} a_i = 1, a_i \geq 0 \qquad (1)$$

where δ_{s_i} represents the Dirac function at embedding s_i. The weights a_i signify the structural importance of each span predicted by a lightweight scoring head $g(\cdot)$ followed by a Softmax. This formulation creates a mass distribution of evidence that sets the stage for transport-based matching.

Span candidate generation. To obtain interpretable units while avoiding dependency on external parsers, we generate span candidates using a deterministic word-level sliding window. We first tokenize the input into wordpieces and recover word boundaries by grouping consecutive wordpieces belonging to the same surface word. Let the resulting word sequence be $\{w_1, \ldots, w_{n_w}\}$ where each word w_t maps to a contiguous wordpiece index range $[p_t, q_t]$ in the encoder output $H = \{h_1, \ldots, h_n\}$. We enumerate all contiguous word spans (l, r) with length $\ell = r - l + 1 \in [L_{\min}, L_{\max}]$ and stride 1 given by:

$$\mathcal{S} = \big\{(l, r) \mid L_{\min} \leq (r - l + 1) \leq L_{\max}\big\} \qquad (2)$$

where $1 \leq l \leq r \leq n_w$. Each span (l, r) corresponds to the wordpiece range $[p_l, q_r]$ and is embedded by mean pooling where $s_{(l,r)} = \text{Pool}(h_{p_l}, \ldots, h_{q_r})$. This procedure yields $M = \sum_{\ell=L_{\min}}^{L_{\max}} (n_w - \ell + 1)$ span candidates. In practice,

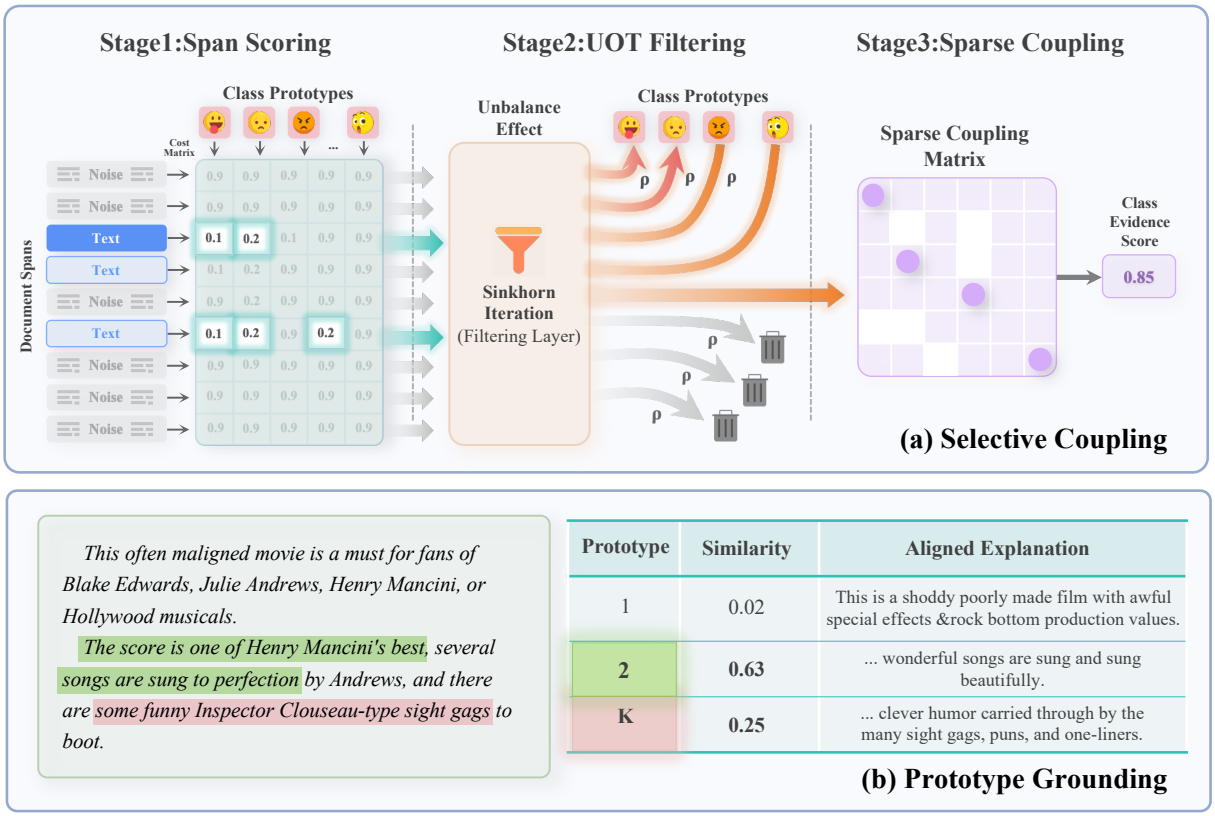

Figure 2: SCOUT inference with selective coupling. (a) Stage 1 scores document spans, Stage 2 applies unbalanced OT to suppress noise, and Stage 3 yields a sparse coupling matrix and class evidence score; (b) visualizes prototype-grounded explanations via aligned spans.

we apply a lightweight scoring head and keep the top-N spans to control UOT complexity, where the retained weights are renormalized to form μ_x.

3.2 Anchored Prototype Distributions

For each class $c \in \{1, \ldots, C\}$ we maintain a set of learnable prototypes $\mathcal{P}_c = \{\nu_{c,k}\}_{k=1}^{K}$. Consistent with the input representation each prototype constitutes a discrete measure comprising M_p support points defined as:

$$\nu_{c,k} = \sum_{j=1}^{M_p} b_{c,k,j} \delta_{u_{c,k,j}} \tag{3}$$

where $u_{c,k,j} \in \mathbb{R}^d$ represent the support embeddings and $b_{c,k,j}$ are uniform weights.

Readable Anchoring. To ensure that prototypes represent realistic semantic concepts rather than abstract artifacts, we enforce a strict anchoring constraint. Unlike previous works utilizing soft regularization, we perform a hard projection of prototype supports onto the training data manifold. Specifically, at the end of every training epoch, each support embedding $u_{c,k,j}$ is replaced by its nearest neighbor spanned from the training set of class c as:

$$u_{c,k,j} \leftarrow \arg\min_{\tilde{s} \in \mathcal{S}_{\text{train}}^{(c)}} \|u_{c,k,j} - \tilde{s}\|_2 \tag{4}$$

During the subsequent training phase, these anchors remain fixed and act as constant targets until the next projection step. This stop-gradient strategy prevents the projection operation from disrupting the optimization trajectory to ensure stability while guaranteeing human-readable verifiability.

3.3 Unbalanced Optimal Transport Alignment

To quantify evidence for class c we compute the unbalanced transport cost between the document distribution μ_x and each prototype distribution $\nu_{c,k}$. We use cosine distance to form the cost matrix $C \in \mathbb{R}^{M \times M_p}$ where $C_{ij} = 1 - \cos(s_i, u_{c,k,j})$. Let $T \in \mathbb{R}_+^{M \times M_p}$ denote the transport plan with row marginals $r = T\mathbf{1}$ and column marginals $c = T^\top \mathbf{1}$.

Unbalanced entropic OT objective. Standard balanced Optimal Transport enforces exact

marginal constraints which forces every input span including irrelevant noise to align with some prototype support. To enable selective matching SCOUT adopts KL-penalized Unbalanced Optimal Transport with entropic regularization (Chizat et al., 2018; Arase et al., 2023) formulated as:

$$\min_{T\geq 0} \langle T, C\rangle \;+\; \varepsilon \sum_{i,j} T_{ij}(\log T_{ij} - 1) \\ +\; \rho_1\, \mathrm{KL}(T\mathbf{1} \,\|\, a) \;+\; \rho_2\, \mathrm{KL}(T^\top \mathbf{1} \,\|\, b), \quad (5)$$

where we use the generalized KL divergence defined as $\mathrm{KL}(p\|q) = \sum_t (p_t \log(p_t/q_t) - p_t + q_t)$. Here ρ_1 and ρ_2 control the cost of violating the marginals. Intuitively, if matching a span incurs a high cost relative to ρ, the optimizer prefers mass suppression where $r_i < a_i$ rather than forcing a noisy match.

Scaling form and generalized Sinkhorn updates. Define the Gibbs kernel $K = \exp(-C/\varepsilon)$. Taking the derivative of Eq. (5) yields the standard diagonal scaling form $T^* = \mathrm{diag}(u)\, K\, \mathrm{diag}(v)$ and the corresponding fixed-point updates:

$$u \leftarrow \left(\frac{a}{Kv}\right)^{\tau_1}, \quad \tau_1 = \frac{\rho_1}{\rho_1 + \varepsilon}, \\ v \leftarrow \left(\frac{b}{K^\top u}\right)^{\tau_2}, \quad \tau_2 = \frac{\rho_2}{\rho_2 + \varepsilon}. \quad (6)$$

Log-domain implementation. Directly forming K can underflow when ratios are large so we implement the iterations in the log-domain. Let $\log K = -C/\varepsilon$, $f = \log u$ and $g = \log v$. The updates become:

$$f \leftarrow \tau_1 \Big(\log(a + \delta) - \mathrm{LSE}_j(\log K_{ij} + g_j) \Big), \\ g \leftarrow \tau_2 \Big(\log(b + \delta) - \mathrm{LSE}_i(\log K_{ij} + f_i) \Big). \quad (7)$$

where δ is a small constant. Finally we recover the coupling via $\log T = f\mathbf{1}^\top + \log K + \mathbf{1}g^\top$. We stabilize iterations by re-centering f and g which leaves $\log T$ unchanged.

3.4 Prototype-Driven Inference

For each class c and prototype index k we compute a UOT coupling matrix $T^*_{c,k}$ and the corresponding scalar transport cost $d_{c,k}(x)$ using the objective in Eq. (5) evaluated at the coupling returned by Algorithm 1.

Algorithm 1 Log-domain generalized Sinkhorn for Unbalanced OT

Require: Cost matrix $C \in \mathbb{R}^{M \times M_p}$; weights $a \in \mathbb{R}^M_+$, $b \in \mathbb{R}^{M_p}_+$; $\varepsilon > 0$; $\rho_1, \rho_2 > 0$; iterations L; floor δ.
Ensure: Coupling $T \in \mathbb{R}^{M \times M_p}_+$.

$\log K \leftarrow -C/\varepsilon$
$\tau_1 \leftarrow \rho_1/(\rho_1 + \varepsilon)$; $\tau_2 \leftarrow \rho_2/(\rho_2 + \varepsilon)$
$f \leftarrow \mathbf{0}_M$; $g \leftarrow \mathbf{0}_{M_p}$ ▷ $f = \log u, g = \log v$
for $\ell = 1$ **to** L **do**
  $\mathrm{logKv}_i \leftarrow \mathrm{LSE}_j(\log K_{ij} + g_j), \quad \forall i$
  $f \leftarrow \tau_1\big(\log(a + \delta) - \mathrm{logKv}\big)$
  $\mathrm{logKTu}_j \leftarrow \mathrm{LSE}_i(\log K_{ij} + f_i), \quad \forall j$
  $g \leftarrow \tau_2\big(\log(b + \delta) - \mathrm{logKTu}\big)$
  $c \leftarrow \mathrm{mean}(f)$; $f \leftarrow f - c$; $g \leftarrow g + c$
end for
$\log T \leftarrow f\mathbf{1}^\top + \log K + \mathbf{1}g^\top$
return $T \leftarrow \exp(\log T)$

Differentiable aggregation. A hard minimum operation routes gradients through only a single prototype which can cause training instability. We therefore use a differentiable soft-min aggregation during training defined as:

$$\mathcal{D}(x, c) = -\frac{1}{\gamma} \log \sum_{k=1}^{K} \exp\big(-\gamma\, d_{c,k}(x)\big), \quad (8)$$

where $\gamma > 0$ controls sharpness. This induces prototype responsibility weights $\pi_{c,k}(x)$ that represent the soft assignment of the input to prototypes of class c.

Class prediction. We convert distances into class probabilities with a distance-based Softmax given by:

$$p(y = c \mid x) = \frac{\exp\big(-\beta\, \mathcal{D}(x, c)\big)}{\sum_{c'} \exp\big(-\beta\, \mathcal{D}(x, c')\big)}, \quad (9)$$

where $\beta > 0$ is a temperature parameter.

3.5 Training Objective

We train SCOUT end-to-end with the composite objective:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda_1 \mathcal{L}_{\mathrm{Clst}} + \lambda_2 \mathcal{L}_{\mathrm{Sep}} + \lambda_3 \mathcal{L}_{\mathrm{Reg}}. \quad (10)$$

The classification loss $\mathcal{L}_{\mathrm{CE}}$ uses standard cross-entropy on the probabilities in Eq. (9). For prototype clustering and separation we encourage each

Model	Sentiment					Topic	Fine-grained
	IMDB	Amazon	Yelp	Hotel	Steam	DBPedia	Consumer
Black-box Classifiers (Upper Bound)							
BERT	$0.940_{\pm 0.002}$	$0.951_{\pm 0.003}$	$0.969_{\pm 0.002}$	$0.977_{\pm 0.001}$	$0.955_{\pm 0.002}$	$0.997_{\pm 0.002}$	$0.936_{\pm 0.002}$
RoBERTa	$0.944_{\pm 0.001}$	$0.959_{\pm 0.002}$	$0.975_{\pm 0.002}$	$0.985_{\pm 0.003}$	$0.960_{\pm 0.001}$	$0.998_{\pm 0.002}$	$0.941_{\pm 0.002}$
DeBERTa	$0.951_{\pm 0.002}$	$0.956_{\pm 0.001}$	$0.977_{\pm 0.002}$	$0.982_{\pm 0.002}$	$0.963_{\pm 0.002}$	$0.998_{\pm 0.002}$	$0.948_{\pm 0.001}$
Intrinsically Interpretable Models							
RNP	$0.915_{\pm 0.002}$	$0.937_{\pm 0.003}$	$0.960_{\pm 0.001}$	$0.961_{\pm 0.003}$	$0.929_{\pm 0.004}$	$0.990_{\pm 0.002}$	$0.927_{\pm 0.003}$
HardKuma	$0.923_{\pm 0.002}$	$0.943_{\pm 0.001}$	$0.963_{\pm 0.002}$	$0.968_{\pm 0.001}$	$0.938_{\pm 0.003}$	$0.990_{\pm 0.001}$	$0.932_{\pm 0.002}$
L2X	$0.919_{\pm 0.002}$	$0.939_{\pm 0.001}$	$0.965_{\pm 0.003}$	$0.966_{\pm 0.002}$	$0.936_{\pm 0.001}$	$0.993_{\pm 0.002}$	$0.926_{\pm 0.003}$
ProtoTEx	$0.911_{\pm 0.002}$	$0.923_{\pm 0.002}$	$0.961_{\pm 0.001}$	$0.962_{\pm 0.003}$	$0.927_{\pm 0.002}$	$0.994_{\pm 0.003}$	$0.938_{\pm 0.001}$
ProtoryNet	$0.913_{\pm 0.002}$	$0.915_{\pm 0.004}$	$0.964_{\pm 0.001}$	$0.964_{\pm 0.003}$	$0.921_{\pm 0.002}$	$0.992_{\pm 0.002}$	$0.926_{\pm 0.002}$
ProtoLens	$0.905_{\pm 0.001}$	$0.936_{\pm 0.003}$	$0.961_{\pm 0.003}$	$0.965_{\pm 0.002}$	$0.928_{\pm 0.002}$	$0.998_{\pm 0.002}$	$0.947_{\pm 0.001}$
WMD kNN	$0.888_{\pm 0.002}$	$0.904_{\pm 0.001}$	$0.945_{\pm 0.002}$	$0.936_{\pm 0.001}$	$0.904_{\pm 0.002}$	$0.990_{\pm 0.002}$	$0.906_{\pm 0.001}$
SCOUT (Ours)	$\mathbf{0.926}_{\pm 0.002}$	$\mathbf{0.948}_{\pm 0.003}$	$\mathbf{0.969}_{\pm 0.003}$	$\mathbf{0.970}_{\pm 0.002}$	$\mathbf{0.941}_{\pm 0.004}$	$\mathbf{0.998}_{\pm 0.001}$	$\mathbf{0.950}_{\pm 0.003}$

Table 1: Main predictive performance comparison. Best results are highlighted as **first**, second and third.

input to be close to prototypes of its own class and far from other classes via:

$$\mathcal{L}_{\text{Clst}} = \mathcal{D}(x, y^\star),$$
$$\mathcal{L}_{\text{Sep}} = \max\Big(0,\ m - \min_{c \neq y^\star} \mathcal{D}(x, c)\Big). \tag{11}$$

where $m > 0$ is a margin.

Evidence regularization. SCOUT explanations are read out from the transported mass over input spans. For a coupling matrix T^* let $m = T^*\mathbf{1}$ denote the transported mass. We regularize contiguity in token space to avoid ambiguity caused by overlapping spans. Define a span-to-token projection matrix P such that the induced token importance is $w = Pm$. We regularize explanations to be concise and contiguous via:

$$\mathcal{L}_{\text{Reg}} = \|m\|_1 + \eta \sum_{t=1}^{n-1} |w_{t+1} - w_t|, \tag{12}$$

where $\eta > 0$ controls the strength of contiguity. This penalty encourages piecewise-constant token saliency which yields coherent phrases rather than scattered evidence.

4 Experiments

4.1 Experimental Setup

Datasets. To evaluate SCOUT across diverse semantic granularities, we conduct experiments on seven standard benchmarks covering three distinct task categories: (1) **Sentiment Analysis**: IMDB, Yelp, Amazon, Hotel, and Steam; (2) **Topic Classification**: DBPedia; and (3) **Fine-grained Classification**: Consumer Complaints. Details are provided in Appendix B.

Baselines. We compare SCOUT against a comprehensive suite of interpretable methods: (1) **Post-hoc Attribution**: Integrated Gradients (IG) (Sundararajan et al., 2017); (2) **Rationale Extraction**: RNP, HardKuma, and L2X (Lei et al., 2016; Bastings et al., 2019; Chen et al., 2018); (3) **Prototype Learning**: ProtoTEx, ProtoryNet, and ProtoLens (Das et al., 2022; Hong et al., 2023; Wei and Zhu, 2025). Standard BERT, RoBERTa, and DeBERTa serve as black-box performance upper bounds.

Evaluation Metrics. We strictly adhere to the ERASER protocol (DeYoung et al., 2020) to assess interpretability. We report Faithfulness (Deletion/Insertion AUC, Comprehensiveness) to measure causal validity, Stability (IOU) to assess robustness against perturbations, and Sparsity to quantify explanation conciseness. Further formal definitions and detailed calculation protocols of these metrics can be found in Appendix F.

4.2 Predictive Performance Comparison

A persistent challenge in interpretable NLP is the trade-off between transparency and predictive power. Table 1 reports the classification performance across all seven datasets. We observe two key trends:

Closing the Gap with Black-boxes. Standard black-box models such as BERT and RoBERTa serve as the performance upper bound. Remarkably, SCOUT with Unbalanced OT retains nearly 98% of the predictive capability of its backbone encoders. For instance, on the IMDB and Yelp datasets, the accuracy gap between our model and the full-input RoBERTa is less than 0.5%. This

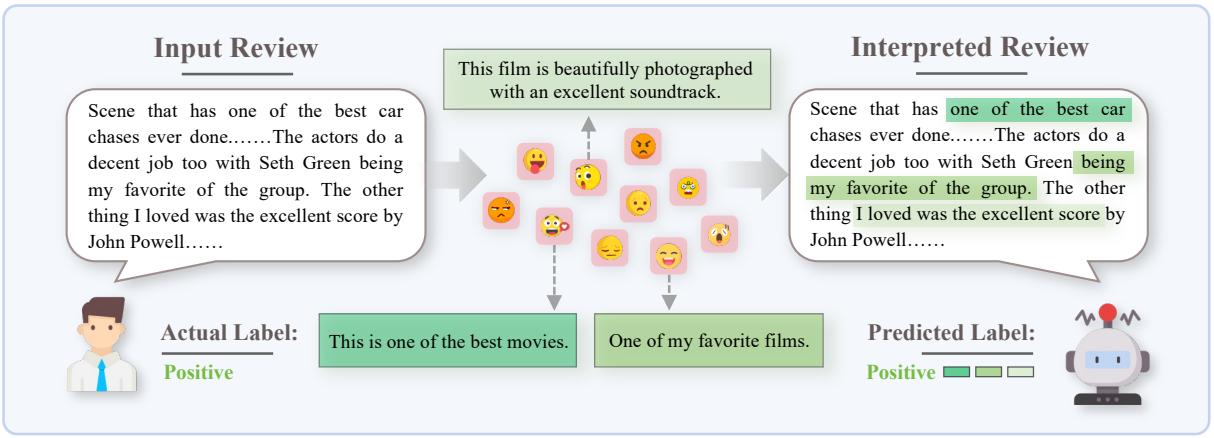

Figure 3: Prototype-grounded case study on IMDB. SCOUT predicts the class by selectively aligning a few decisive spans in the input review to anchored prototype snippets from the training set, while leaving irrelevant background unmatched. The interpreted review is read out from transported span mass.

Model	Faithfulness Metrics			Explanation Budget (Sparsity)			Stability
	Del AUC↓	Ins AUC↑	Comp↑	Tokens↓	Spans↓	Contig↑	Avg IOU↑
Post-hoc Attribution							
Integrated Gradients	0.40 ± 0.04	0.59 ± 0.06	0.42 ± 0.03	31.8 ± 2.7	10.7 ± 0.8	0.41 ± 0.03	0.42 ± 0.04
Intrinsic Interpretability							
RNP	0.34 ± 0.01	0.65 ± 0.02	0.55 ± 0.02	18.9 ± 0.6	6.4 ± 0.3	0.63 ± 0.02	0.55 ± 0.02
HardKuma	0.32 ± 0.01	0.67 ± 0.03	0.58 ± 0.02	16.7 ± 0.7	5.9 ± 0.2	0.66 ± 0.02	0.57 ± 0.03
L2X	0.35 ± 0.01	0.64 ± 0.03	0.52 ± 0.02	19.4 ± 0.8	6.8 ± 0.2	0.61 ± 0.03	0.56 ± 0.02
ProtoTEx	0.36 ± 0.02	0.62 ± 0.03	0.48 ± 0.02	22.8 ± 0.8	7.5 ± 0.3	0.55 ± 0.02	0.51 ± 0.03
ProtoryNet	0.35 ± 0.01	0.63 ± 0.02	0.50 ± 0.02	21.9 ± 1.0	7.1 ± 0.3	0.57 ± 0.03	0.54 ± 0.02
ProtoLens	0.35 ± 0.01	0.64 ± 0.03	0.53 ± 0.02	21.3 ± 0.7	6.8 ± 0.2	0.60 ± 0.02	0.53 ± 0.02
SCOUT (Ours)	**0.29** ± 0.01	**0.71** ± 0.01	**0.64** ± 0.01	**14.3** ± 0.3	**4.6** ± 0.1	**0.72** ± 0.02	**0.63** ± 0.01

Table 2: Comprehensive evaluation of explanation quality averaged across all datasets. Best results are highlighted as **first** , second and third .

indicates that the extracted rationales are not only sparse but semantically sufficient, capturing the decisive information required for prediction without relying on spurious correlations found in the full text.

Superiority over Interpretable Baselines. Compared to existing intrinsic interpretable methods, the SCOUT framework demonstrates highly stable and significant performance advantages. Specifically, across multiple evaluations, it outperforms baselines relying on traditional hard-selection, such as HardKuma and L2X, by a substantial average margin of 2% to 3%. We attribute this leap in performance primarily to the unique geometric nature of the Optimal Transport framework. Traditional hard masking strategies often rely on rigid binary discard mechanisms; when attempting to filter out irrelevant information, this approach can easily cause irreversible damage, discarding useful contextual associations that are crucial for text comprehension. In contrast, SCOUT introduces an innovative mass destruction mechanism. This mechanism dynamically and effectively suppresses redundant background noise while fully preserving the underlying structural integrity and semantic continuity of the critical evidence, thereby maximizing predictive accuracy without sacrificing model transparency.

4.3 Qualitative Analysis

Figure 3 visualizes SCOUT's end-to-end explanation on an IMDB review. The model first assigns mass to candidate spans and then computes a selective document–prototype coupling using unbalanced optimal transport, which allows irrelevant background spans to remain unmatched. The interpreted review (right) is obtained by projecting

transported span mass to token saliency and selecting a compact rationale under a fixed coverage budget, resulting in a small set of decisive phrases. Importantly, the supporting evidence is verifiable: the activated prototypes are anchored to readable training spans, and the explanation explicitly links input phrases to these prototype snippets, providing a transparent bridge between the prediction and stored class evidence.

Figure 4 visualizes SCOUT's prototype space for one IMDB review by projecting both the input instance and prototypes into 2D. Positive and negative prototypes form two separated regions, and only a small subset of prototypes is activated for the given input, indicated by filled nodes and weighted edges from the input point. The activated prototypes correspond to sentiment-bearing evidence (e.g., "the animation quality is excellent" and "the animation and art direction is lovely"), while prototypes from the opposite class remain inactive. This localized, class-consistent activation pattern illustrates how SCOUT produces sparse and verifiable explanations: the prediction is supported by explicit alignments to a few anchored prototype snippets rather than diffuse similarity to many prototypes.

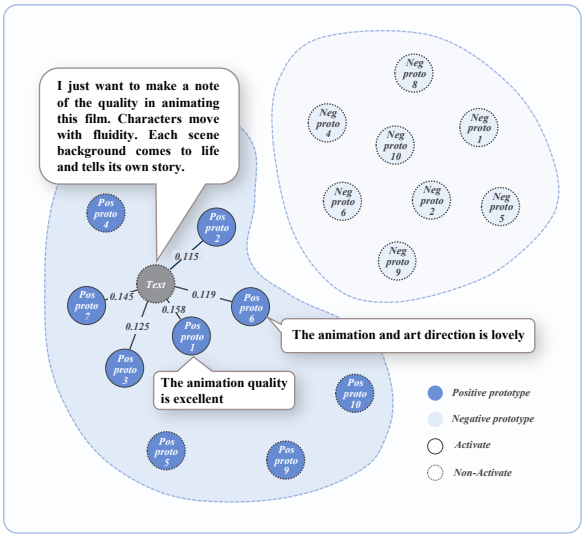

Figure 4: Prototype projection on IMDB. The input activates a small set of positive prototypes, while negative prototypes remain inactive; callouts show aligned evidence snippets.

4.4 Evaluation of Explanation Quality

Table 2 presents a comprehensive evaluation of explanation quality across three key dimensions: faithfulness, sparsity, and stability. We observe that SCOUT with Unbalanced OT consistently achieves the best performance among all intrinsic interpretable methods, as highlighted by the red background in the table.

Faithfulness: Finding the True Cause. SCOUT establishes a new state-of-the-art in causal validity by achieving a lowest Deletion AUC of 0.29 and a highest Insertion AUC of 0.71. These figures significantly outperform the strong baseline HardKuma which records 0.32 and 0.67 respectively. This result indicates that the spans transported by our optimal transport plan are indeed the decisive factors driving the prediction. In contrast, post-hoc attribution methods like Integrated Gradients exhibit much poorer fidelity with a Deletion AUC of 0.40, confirming that gradient-based saliency maps often fail to capture the true decision boundary of the model.

Sparsity: Doing More with Less. A key advantage of our mass destruction mechanism is efficiency. As shown in the middle columns of Table 2, SCOUT generates the most concise explanations and utilizes an average of only 4.6 spans or approximately 14.3 tokens per document. This represents a substantial reduction compared to ProtoTEx which requires 7.5 spans and Integrated Gradients which uses 10.7 spans. Despite using nearly 50% fewer tokens than IG, SCOUT achieves superior predictive accuracy as discussed in Section 4.2, demonstrating a high signal-to-noise ratio in evidence extraction.

Stability: Robustness to Perturbation. Reliable explanations should not fluctuate wildly with minor input changes. SCOUT demonstrates superior stability with an Average IOU of 0.63, surpassing the previous best score of 0.55 achieved by RNP and 0.57 by HardKuma. This empirical evidence suggests that our geometric alignment approach provides a more robust convergence landscape than reinforcement learning or heuristic masking strategies utilized by baseline methods.

4.5 Failure Case Analysis

While SCOUT generally produces faithful and sparse explanations, it is instructive to examine cases where the model fails, as these reveal the transparency of its decision process. We present a representative misclassification from the Yelp dataset. The review describes a "dive bar" using attributes typically associated with negative sentiment (such as *"greasy food"* and *"grungy people"*), yet these descriptors denote genuine praise within

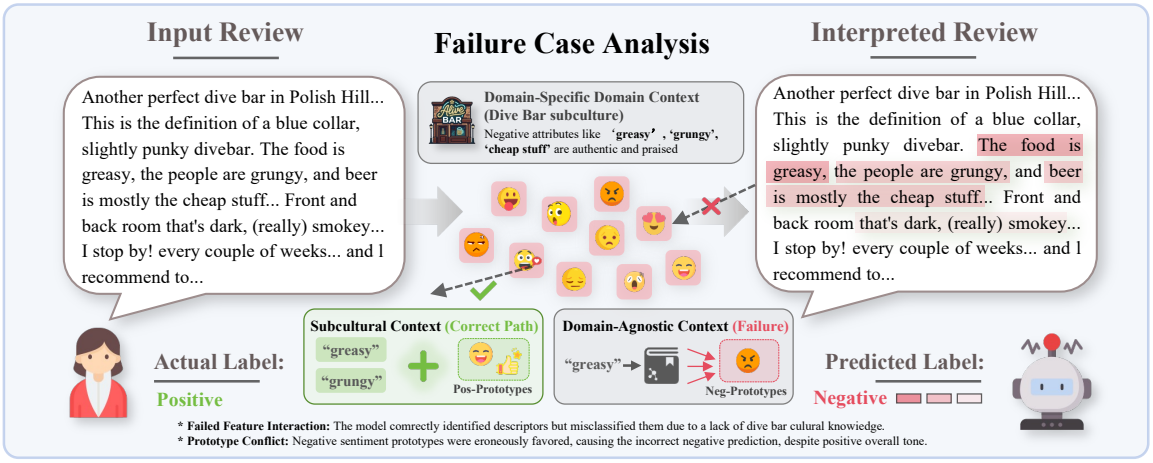

Figure 5: Failure case analysis of sentiment misclassification due to missing subcultural context. The model erroneously predicts a negative label for a positive review by applying domain-agnostic context. Subculture-specific descriptors like "greasy" and "grungy" are incorrectly mapped to negative sentiment prototypes.

the subculture of that specific venue. SCOUT erroneously assigns a negative label to this review.

Mechanistically, the unbalanced optimal transport layer performs as designed: it filters out non-discriminative background text and isolates the aforementioned spans, coupling them with high confidence to negative-class prototypes. The resulting transport plan and prototype alignments are fully inspectable. As shown in Figure 5, the activated prototypes correspond to spans like *"over-priced and greasy"* and *"staff was rude"*, which are semantically similar to the extracted evidence but fail to capture the subcultural re-appropriation of these terms.

This failure mode underscores a key strength of intrinsic interpretability: when the model errs, it fails transparently. The explicit span-to-prototype coupling immediately pinpoints the source of the error (a mismatch between the training distribution's sentiment convention and the contextual usage in this specific review) rather than obscuring it in an opaque latent space. Such visibility is crucial for debugging and trust in high-stakes applications.

4.6 Ablation Study

Table 3 summarizes the contribution of each component. Our results empirically confirm that the proposed Regularization strategy (Eq. 15) is the primary driver for explanation conciseness, as removing it causes the rationale length to double. Meanwhile, the Unbalanced Optimal Transport mechanism is essential for filtering noise; replacing it with balanced OT degrades both predictive accuracy and faithfulness. For a comprehensive analysis of granularity choices and component interactions, please refer to Appendix G.

Model Variant	Performance Metrics		
	Acc ↑	Del AUC ↓	Tokens ↓
Token-level OT	94.7	0.34	21.5
w/o Unbalanced OT	94.5	0.33	19.8
w/o Regularization	95.1	0.30	28.4
w/o Anchor	94.8	0.31	15.2
SCOUT (Full)	**95.3**	**0.29**	**14.3**

Table 3: Ablation study of SCOUT components. Best results are highlighted as **first**, second and third.

5 Conclusion

In this work, we presented SCOUT, a novel framework that effectively resolves the tension between predictive accuracy and interpretability in NLP models. Unlike conventional methods that rely on heuristic hard selection or unstable reinforcement learning, we formulate rationale extraction as a span-based Unbalanced Optimal Transport problem. This perspective allows the model to geometrically align input evidence with learnable prototypes while strictly filtering out irrelevant noise through a principled mass destruction mechanism. In future work, we plan to extend this geometric alignment paradigm to the era of Large Language Models. Specifically, we aim to explore how optimal transport can quantify the faithfulness of Retrieval-Augmented Generation (RAG) and visualize the semantic drift in Chain-of-Thought reasoning, paving the way for trustworthy AI systems that are both powerful and accountable.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62372408) and Hangzhou Pujian Medical Technology Co., Ltd, China and ZJU-Pujian Research & Development Center of Medical Artificial Intelligence for Hepatobiliary and Pancreatic Disease.

Limitations

Despite establishing a new standard for interpretable text classification, SCOUT entails specific trade-offs. The primary limitation lies in computational overhead. Replacing scalar dot-products with Sinkhorn iterations introduces a per-document inference complexity of $O(CK \cdot LMM_p)$, where L is the number of Sinkhorn iterations and M is the number of spans. Our vectorized GPU implementation ensures efficient training on standard benchmarks; however, the cost scales linearly with the number of classes C and prototypes K. This makes the current version less suitable for extreme multi-label classification settings with thousands of labels without further approximation (e.g., coarse prototype pre-screening). Additionally, our span generation strategy relies on a fixed sliding window. Although effective, employing a dynamic or syntax-aware span proposal network could potentially further improve evidence boundary precision.

References

Yuki Arase, Han Bao, and Sho Yokoi. 2023. Unbalanced optimal transport for unbalanced word alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3966–3986. Association for Computational Linguistics.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977. Association for Computational Linguistics.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noémie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. 2019. This looks like that: deep learning for interpretable image recognition. *Advances in Neural Information Processing Systems*, 32.

Huangwei Chen, Yifei Chen, Zhenyu Yan, Mingyang Ding, Chenlei Li, Zhu Zhu, and Feiwei Qin. 2025. Mmlnb: Multi-modal learning for neuroblastoma subtyping classification assisted with textual description generation. *arXiv preprint arXiv:2503.12927*.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pages 883–892. PMLR.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. 2018. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of computation*, 87(314):2563–2609.

Yu-Neng Chuang, Guanchu Wang, Chia-Yuan Chang, Ruixiang Tang, Shaochen Zhong, Fan Yang, Andrew Wen, Mengnan Du, Xuanting Cai, Vladimir Braverman, and 1 others. 2026. Faithlm: Towards faithful explanations for large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3802–3824. Association for Computational Linguistics.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*.

Anubrata Das, Chitrank Gupta, Venelin Kovatchev, Matthew Lease, and Junyi Jessy Li. 2022. Prototex: Explaining model decisions with prototype tensors. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 2986–2997. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2020. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458. Association for Computational Linguistics.

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

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Dino Pedreschi, and Fosca Giannotti. 2018. A survey of methods for explaining black box models. *arXiv preprint arXiv:1802.01933*.

Karthik S Gurumoorthy, Pratik Jawanpuria, and Bamdev Mishra. 2021. Spot: A framework for selection of prototypes using optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 535–551. Springer.

Dat Hong, Tong Wang, and Stephen Baek. 2023. Protorynet-interpretable text classification via prototype trajectories. *Journal of Machine Learning Research*, 24(264):1–39.

Jinjin Huang, Ce Guo, and Wayne Luk. 2025. Protopgtn: A scalable prototype-based gated transformer network for interpretable time series classification. *Information*, 16(12):1056.

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*, pages 4198–4205. Association for Computational Linguistics.

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: Human Language Technologies*, pages 3543–3556. Association for Computational Linguistics.

Junhao Jia, Yunyou Liu, Yifei Sun, Huangwei Chen, Feiwei Qin, Changmiao Wang, and Yong Peng. 2025. Geodesic prototype matching via diffusion maps for interpretable fine-grained recognition. *arXiv preprint arXiv:2509.17050*.

Junhao Jia, Jiaqi Wang, Yunyou Liu, Haodong Jing, Yueyi Wu, Xian Wu, and Yefeng Zheng. 2026a. This looks distinctly like that: Grounding interpretable recognition in stiefel geometry against neural collapse. *arXiv preprint arXiv:2603.08374*.

Junhao Jia, Yueyi Wu, Huangwei Chen, Haodong Jing, Haishuai Wang, Jiajun Bu, and Lei Wu. 2026b. Unsupervised causal prototypical networks for de-biased interpretable dermoscopy diagnosis. *arXiv preprint arXiv:2602.23752*.

Dongyao Jiang, Haodong Jing, Yongqiang Ma, and Nanning Zheng. 2025. Beyond image classification: A video benchmark and dual-branch hybrid discrimination framework for compositional zero-shot learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9860–9869.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 957–966. PMLR.

Seonghyeon Lee, Dongha Lee, Seongbo Jang, and Hwanjo Yu. 2022. Toward interpretable semantic textual similarity via optimal transport-based contrastive sentence learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5969–5979, Dublin, Ireland. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117. Association for Computational Linguistics.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2024. Towards faithful model explanation in nlp: A survey. *Computational Linguistics*, 50(2):657–723.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Utsav Kumar Nareti, Suraj Kumar, Soumya Pandey, Soumi Chattopadhyay, and Chandranath Adak. 2025. Protositex: Learning semi-interpretable prototypes for multi-label text classification. *arXiv preprint arXiv:2510.12534*.

Gabriel Peyré, Marco Cuturi, and 1 others. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607.

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.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Francis Snelgar, Stephen Gould, Ming Xu, Liang Zheng, and Akshay Asthana. 2025. Gromov wasserstein optimal transport for semantic correspondences. In *36th British Machine Vision Conference 2025, BMVC 2025, Sheffield, UK, November 24-27, 2025*. BMVA.

Zhivar Sourati, Darshan Girish Deshpande, Filip Ilievski, Kiril Gashteovski, and Sascha Saralajew. 2024. Robust text classification: Analyzing prototype-based networks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12736–12757. Association for Computational Linguistics.

Yuxi Sun, Aoqi Zuo, Wei Gao, and Jing Ma. 2025. Causalabstain: Enhancing multilingual llms with causal reasoning for trustworthy abstention. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14060–14076. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328.

Kyle Swanson, Lili Yu, and Tao Lei. 2020a. Rationalizing text matching: Learning sparse alignments via optimal transport. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5609–5626. Association for Computational Linguistics.

Kyle Swanson, Lili Yu, and Tao Lei. 2020b. Rationalizing text matching: Learning sparse alignments via optimal transport. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5609–5626. Association for Computational Linguistics.

Bowen Wei and Ziwei Zhu. 2025. Protolens: Advancing prototype learning for fine-grained interpretability in text classification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4503–4523. Association for Computational Linguistics.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 11–20. Association for Computational Linguistics.

Shuhui Wu, Yongliang Shen, Zeqi Tan, Wenqi Ren, Jietian Guo, Shiliang Pu, and Weiming Lu. 2023. Mproto: Multi-prototype network with denoised optimal transport for distantly supervised named entity recognition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2361–2374. Association for Computational Linguistics.

Yujia Xie, Minshuo Chen, Haoming Jiang, Tuo Zhao, and Hongyuan Zha. 2019. On scalable and efficient computation of large scale optimal transport. In *International Conference on Machine Learning*, pages 6882–6892. PMLR.

Xiao Zhang, Haodong Jing, Hui Chen, Yongqiang Ma, and Nanning Zheng. 2025. Refiner: Fine-grained cross-modal concepts refinement for compositional zero-shot learning. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Xiao Zhang, Haodong Jing, Yongqiang Ma, and Nanning Zheng. Decoupling primitive with experts: Dynamic feature alignment for compositional zero-shot learning. In *The Fourteenth International Conference on Learning Representations*.

Zefeng Zhang, Jiawei Sheng, Chuang Zhang, Yunzhi Liang, Wenyuan Zhang, Siqi Wang, and Tingwen Liu. 2024. Optimal transport guided correlation assignment for multimodal entity linking. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4103–4117, Bangkok, Thailand. Association for Computational Linguistics.

Xiaolin Zheng, Mengling Hu, Weiming Liu, Chaochao Chen, and Xinting Liao. 2023. Robust representation learning with reliable pseudo-labels generation via self-adaptive optimal transport for short text clustering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10493–10507, Toronto, Canada. Association for Computational Linguistics.

A Additional Method Details

A.1 Rationale readout from the UOT coupling

Given an optimal coupling $T^\star \in \mathbb{R}_+^{M\times M_p}$ between document spans $\{s_i\}_{i=1}^{M}$ and a prototype support set $\{u_j\}_{j=1}^{M_p}$, we compute the transported mass on each input span as:

$$m = T^\star \mathbf{1}_{M_p} \in \mathbb{R}_+^M. \tag{13}$$

Since spans overlap in token space, we convert span-level mass to token-level saliency by a span-to-token projection matrix $P \in \mathbb{R}^{n\times M}$:

$$w = Pm,$$
$$P_{t,i} = \begin{cases} \frac{1}{|\mathrm{span}(i)|}, & \text{if token } t \in \mathrm{span}(i), \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

where $|\mathrm{span}(i)|$ is the number of tokens covered by span i. This definition accumulates evidence from all selected spans while avoiding a systematic bias toward longer spans.

Discrete rationale selection. We normalize token saliency by $\tilde{w} = w/(\sum_{t=1}^{n} w_t + \epsilon_w)$ with a small ϵ_w. To obtain a binary rationale set, we sort tokens by $\tilde{w}$ and select the smallest set $\mathcal{R}$ such that $\sum_{t\in\mathcal{R}} \tilde{w}_t \geq \kappa$ (coverage ratio). Unless otherwise stated, we use $\kappa = 0.9$ in all experiments. Finally, we merge consecutive selected tokens into contiguous segments for visualization and for ERASER-style evaluation.

Contiguity post-processing. To avoid isolated single-token fragments, we optionally remove segments shorter than $\ell_{\min}$ by merging them with the nearest neighboring segment (ties broken by higher $\tilde{w}$), or by expanding each segment by one token on both sides if it does not exceed the document boundary.

A.2 Which UOT distance is used for classification

In Eq. (5), the entropically-regularized KL-penalized UOT objective is:

$$\mathcal{J}(T) = \langle T, C \rangle \; + \; \varepsilon \sum_{i,j} T_{ij}(\log T_{ij} - 1)$$
$$+ \; \rho_1 \, \mathrm{KL}(T\mathbf{1} \,\|\, a) \; + \; \rho_2 \, \mathrm{KL}(T^\top \mathbf{1} \,\|\, b) \quad (15)$$

We define the prototype-to-document distance as:

$$d_{c,k}(x) \; = \; \mathcal{J}(T^\star_{c,k}), \quad (16)$$

where $T^\star_{c,k}$ is returned by Algorithm 1. For numerical stability, we implement all iterations in the log-domain as described in Eq. (7). In practice, terms that are constant across (x, c, k) can be dropped without affecting the soft-min aggregation in Eq. (8).

A.3 Span prefiltering and differentiability

We enumerate all candidate spans under the window bounds $(L_{\min}, L_{\max})$ and compute a scalar score for each span using the lightweight head $g(\cdot)$. We then keep the top-N spans (by score) to control UOT complexity and renormalize the retained scores via Softmax to obtain the document weights a in Eq. (1). This top-N selection is implemented by a deterministic `topk` operator: gradients flow to the retained spans only, while all subsequent computations (including the log-Sinkhorn UOT layer and the classifier) remain fully differentiable.

A.4 Prototype anchoring implementation details

For each class c, we maintain a class-specific span bank $\mathcal{S}^{(c)}_{\text{train}}$ constructed by running the encoder over the training split and extracting span embeddings using the same span generator as in Section 3.1. At the end of every epoch, each prototype support $u_{c,k,j}$ is projected to its nearest neighbor in $\mathcal{S}^{(c)}_{\text{train}}$ as in Eq. (4). This projection is performed with stop-gradient: anchors are treated as constants during the next epoch to avoid disrupting optimization.

Efficient nearest-neighbor search. When $\mathcal{S}^{(c)}_{\text{train}}$ is large, we recommend approximate nearest-neighbor search over ℓ_2 distance (or cosine distance if embeddings are ℓ_2-normalized). We rebuild the index once per epoch.

B Datasets

We evaluate on seven benchmarks spanning sentiment, topic, and fine-grained complaint classification. For sentiment analysis, we use IMDB (25,000 train / 25,000 test, balanced), Yelp Reviews (550,000 train / 30,000 test; ratings are binarized with 1–2 stars as negative and 3–4 stars as positive), Amazon Reviews (a 30,000-review subset with 24,000 for training/validation and 6,000 for testing), Hotel Reviews (a balanced subset of 4,508 reviews), and Steam Reviews (130,000 preprocessed reviews balanced between positive and negative; very short reviews are filtered). For topic classification, we use DBPedia14 but construct a commonly used 4-way subset with labels Person, Animal, Building, and Natural Place to keep semantics clear and comparable. For fine-grained classification, we use Consumer Complaints (CFPB complaints) and similarly build a 4-way subset with labels Checking or Savings Account, Credit Card or Prepaid Card, Debt Collection, and Mortgage. Unless an official development set is provided, we hold out 10% of the training split as validation to tune hyperparameters and perform early stopping.

C Implementation Details

Table 4 lists the specific parameters for span generation, prototype configuration, and Optimal Transport optimization.

Parameter	Value
Span Generation	
Span Length Bounds (L_{min}, L_{max})	$1, 5$
Top-N Spans Kept	50
Prototype Architecture	
Prototypes per Class (K)	10
Support Points per Prototype (M_p)	5
Unbalanced Optimal Transport	
Entropic Regularization (ϵ)	0.1
Marginal Penalties (ρ_1, ρ_2)	$1, 1$
Sinkhorn Iterations (L)	20
Soft-min Temperature (γ)	1.0
Training	
Optimizer	AdamW
Learning Rate	$2e-5$
Batch Size	32
Epochs	20
Early Stopping Patience	3 epochs
Class Softmax Temp (β)	1.0
Regularization Weights $(\lambda_1, \lambda_2, \lambda_3)$	$0.1, 0.1, 0.05$
Contiguity Strength (η)	0.5

Table 4: Hyperparameters used for SCOUT.

D Additional Sensitivity Analysis

D.1 Sensitivity to Mass Destruction Parameter

In the main text, we hypothesized that the mass destruction mechanism enabled by Unbalanced Op-

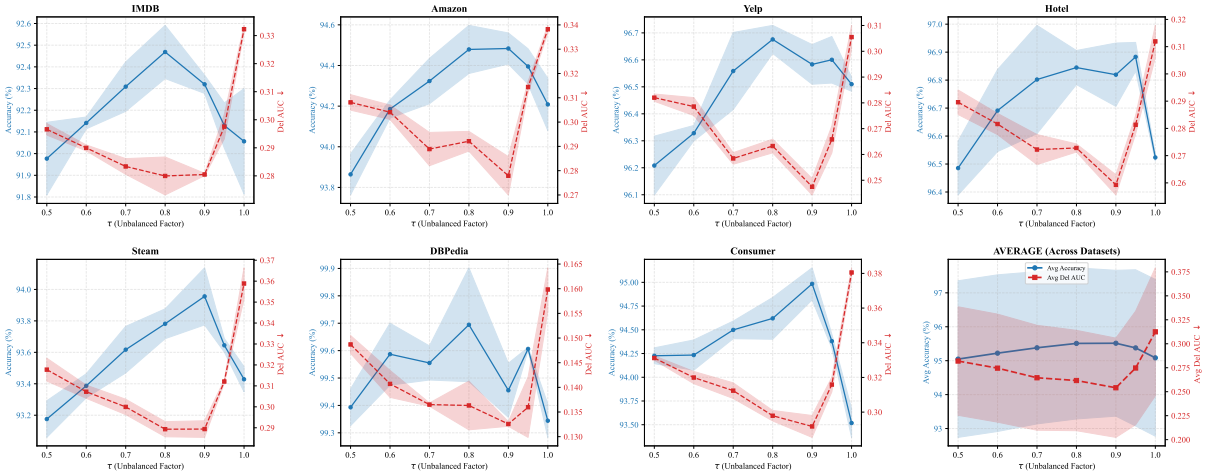

Figure 6: Sensitivity analysis of the mass destruction parameter τ across all seven datasets.

timal Transport (UOT) is crucial for filtering noise. Here, we provide a comprehensive empirical verification of this claim.

Experimental Setup. We analyze the impact of the unbalanced relaxation factor $\tau = \rho/(\rho + \epsilon)$. This parameter controls the strictness of alignment: $\tau \to 1$ implies Balanced OT (forced alignment), while lower values allow for selective coupling. We swept τ from 0.5 to 1.0 across all seven datasets. To ensure statistical reliability, each configuration was executed with **3 independent random seeds**.

Results. Figure 6 illustrates the mean performance and standard deviation (shaded bands) for both Predictive Accuracy (Blue) and Explanation Faithfulness (Deletion AUC, Red). The results reveal a consistent "sweet spot" around $\tau \approx 0.9$ across diverse tasks. Crucially, enforcing strict mass conservation ($\tau = 1.0$) consistently degrades performance, particularly on noisy datasets like Consumer Complaints and IMDB. This statistically confirms that SCOUT's ability to leave irrelevant spans unmatched is a structural necessity for faithful interpretation, rather than a mere hyperparameter choice.

D.2 Robustness to Heuristic Hyperparameters

SCOUT incorporates several heuristic choices in span generation and prototype configuration: the number of retained spans N, the maximum span length $L_{\max}$, the coverage threshold κ for rationale discretization, the number of prototypes per class K, and the number of support points per prototype M_p. To verify that our findings are not sensitive to these settings, we perform a systematic sweep on the Yelp and DBPedia14 datasets. For each parameter, we measure classification accuracy, deletion AUC (faithfulness), and the number of tokens in the extracted rationale.

Table 5 reports the minimum, mean, and maximum values across each sweep range. The results demonstrate strong robustness in both predictive performance and explanation faithfulness: accuracy varies by at most 0.3% absolute on Yelp and 0.2% on DBPedia14, while deletion AUC fluctuates by less than 0.02 across all configurations. Rationale length exhibits a predictable and controllable trend: expanding the candidate evidence pool (via larger N or $L_{\max}$) yields marginally longer rationales, whereas adjusting κ offers direct control over sparsity without affecting model behavior. The number of prototypes K and support points M_p have negligible impact on all metrics, indicating that SCOUT's geometric alignment via unbalanced optimal transport drives performance rather than brittle dependence on these heuristic settings.

E Comprehensive Analysis per Dataset

While Table 2 presents aggregated results, it is crucial to verify that the model's performance is consistent across diverse domains and not driven by outliers.

Figure 7 provides a granular breakdown of all seven interpretability metrics for each individual dataset. We employ a dual-axis visualization to accommodate different metric scales:

- **Left Axis (Bars):** Displays score-based metrics normalized to $[0, 1]$, including Faithfulness (Del ↓, Ins ↑, Comp ↑), Quality (Contig ↑), and Stability (Stab ↑).

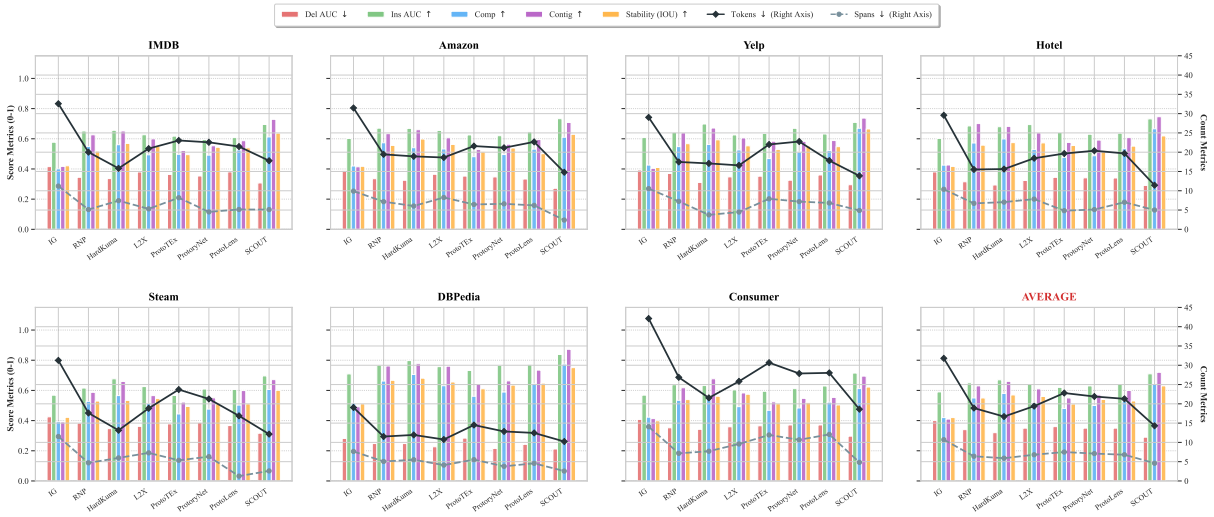

Figure 7: Detailed performance breakdown for each dataset.

Dataset	Parameter Sweep	Acc (%) ↑	Del AUC ↓	#Tokens ↓
Yelp	$N \in \{20, 30, 50, 70, 100\}$	96.7 / 96.9 / 97.1	0.27 / 0.28 / 0.29	9.8 / 13.2 / 18.7
	$L_{\max} \in \{3, 5, 7, 9\}$	96.8 / 96.9 / 97.0	0.27 / 0.28 / 0.29	10.4 / 13.2 / 15.8
	$\kappa \in \{0.80, 0.85, 0.90, 0.95\}$	96.8 / 96.9 / 97.0	0.27 / 0.28 / 0.29	11.1 / 13.2 / 15.0
	$K \in \{5, 10, 20, 40\}$	96.7 / 96.9 / 97.0	0.27 / 0.28 / 0.29	12.8 / 13.2 / 13.6
	$M_p \in \{3, 5, 7, 9\}$	96.8 / 96.9 / 97.0	0.27 / 0.28 / 0.29	13.0 / 13.2 / 13.4
DBPedia14	$N \in \{20, 30, 50, 70, 100\}$	99.6 / 99.8 / 99.9	0.20 / 0.21 / 0.22	7.9 / 10.4 / 13.2
	$L_{\max} \in \{3, 5, 7, 9\}$	99.7 / 99.8 / 99.8	0.20 / 0.21 / 0.22	8.5 / 10.4 / 11.9
	$\kappa \in \{0.80, 0.85, 0.90, 0.95\}$	99.7 / 99.8 / 99.9	0.20 / 0.21 / 0.22	9.2 / 10.4 / 11.3
	$K \in \{5, 10, 20, 40\}$	99.7 / 99.8 / 99.8	0.20 / 0.21 / 0.22	10.1 / 10.4 / 10.7
	$M_p \in \{3, 5, 7, 9\}$	99.7 / 99.8 / 99.9	0.20 / 0.21 / 0.22	10.3 / 10.4 / 10.6

Table 5: Robustness of SCOUT to heuristic hyperparameters on Yelp and DBPedia14. Values are reported as min / mean / max across each sweep range.

- **Right Axis (Lines):** Displays sparsity metrics in absolute counts (**Tokens**, Spans ↓).

Analysis. As shown in the figure, SCOUT consistently demonstrates the best efficiency across all tasks. For instance, on the difficult Consumer Complaints dataset (bottom right), SCOUT achieves a Deletion AUC comparable to baselines but uses significantly fewer tokens (lowest black line point), indicating a superior signal-to-noise ratio in extracting decisive evidence.

F Evaluation Metrics Protocols

This section provides implementation-level details for all interpretability metrics used in Section 4.1. We follow ERASER for contrast-based faithfulness (Comprehensiveness) and adopt perturbation-curve evaluations (Deletion/Insertion AUC) commonly used in attribution benchmarks. For stability, we evaluate robustness under synonym substitutions.

F.1 Token-Level Explanation Scores and Discretization

Many baselines output either (i) a discrete binary rationale mask, or (ii) continuous token scores. To ensure reproducibility and fair comparison, we convert all explanations to a unified token-score vector and apply the same discretization rule when a discrete rationale is required (e.g., for Comprehensiveness, Sparsity, Contiguity, and Stability).

Token score vector. Given an input with n tokens, each method produces a non-negative token score vector $s \in \mathbb{R}^n_{\geq 0}$: (i) for discrete rationale methods, we set $s_t \in \{0, 1\}$; (ii) for continuous attribution, we take $s_t \leftarrow \max(0, s_t)$. For SCOUT, we first obtain span masses $m = T^{\mathbf{1}}$ and project them to tokens as $w = Pm$ (Eq. 12), then set

$s \leftarrow w$.

Mass-normalized ranking. We normalize scores into a probability simplex:

$$\tilde{s}_t = \frac{s_t}{\sum_{u=1}^{n} s_u + \epsilon_s}, \tag{17}$$

where ϵ_s is a small constant to avoid division by zero. Tokens are ranked by descending $\tilde{s}_t$.

Discrete rationale via cumulative-mass coverage. For metrics that require a discrete rationale, we select the smallest token set R_α whose cumulative normalized mass exceeds a fixed coverage threshold α:

$$R_\alpha = \min \left\{ R \subseteq \{1, \ldots, n\} \;\middle|\; \sum_{t \in R} \tilde{s}_t \geq \alpha \right\}, \tag{18}$$

where we set $\alpha = 0.9$ in all experiments. We then merge consecutive selected tokens into contiguous segments to obtain span rationales.

F.2 Faithfulness Metrics

Let $f_j(x)$ denote the predicted probability of class j for input x, where j is the model's predicted class on the original input.

Comprehensiveness. Following ERASER, we define comprehensiveness using a contrast example $x \setminus R_\alpha$, constructed by masking tokens in R_α:

$$\mathrm{Comp}(x) = f_j(x) - f_j(x \setminus R_\alpha). \tag{19}$$

Higher is better, indicating that the extracted rationale is causally influential.

Deletion AUC. Deletion measures how quickly the confidence drops as important tokens are removed. We iteratively mask tokens in descending order of $\tilde{s}_t$. Let $x^{(k)}$ be the input after masking the top-k fraction of tokens, with $k \in \{0, 0.05, 0.10, \ldots, 1.00\}$. The deletion curve is $d(k) = f_j(x^{(k)})$ and we compute its area under the curve (trapezoidal rule) over $k \in [0, 1]$:

$$\mathrm{DelAUC}(x) = \int_0^1 d(k)\, dk. \tag{20}$$

Lower is better.

Insertion AUC. Insertion measures how quickly confidence recovers when important tokens are inserted. We start from a fully-masked baseline x_{mask} and iteratively restore tokens in descending order. Let $\hat{x}^{(k)}$ denote the partially restored input at fraction k, and the insertion curve is $i(k) = f_j(\hat{x}^{(k)})$. We compute:

$$\mathrm{InsAUC}(x) = \int_0^1 i(k)\, dk. \tag{21}$$

Higher is better.

Masking operator. We use the backbone model's mask token (e.g., `[MASK]` or `<mask>`) to replace removed tokens, which keeps the sequence length fixed and avoids confounding effects from changing positional encodings.

F.3 Sparsity and Contiguity

Using the discrete rationale R_α:

- **Tokens**: $|R_\alpha|$.
- **Spans**: the number of contiguous segments after merging adjacent selected tokens.

We define a normalized contiguity score:

$$\mathrm{Contig}(x) = 1 - \frac{\max(0,\ \#\mathrm{Spans}(x) - 1)}{\max(1,\ |R_\alpha| - 1)} \in [0, 1], \tag{22}$$

where 1 indicates a single contiguous span and smaller values indicate more fragmented rationales.

F.4 Stability via Synonym Substitution

To measure stability, we create a perturbed input x' by randomly replacing 10% of eligible tokens (content words; excluding stopwords, punctuation, and special tokens) with synonyms from WordNet, preferring substitutions with the same part-of-speech when available. We extract $R_\alpha(x)$ and $R_\alpha(x')$ and compute token-level Jaccard similarity:

$$\mathrm{IOU}(x, x') = \frac{|R_\alpha(x) \cap R_\alpha(x')|}{|R_\alpha(x) \cup R_\alpha(x')|}. \tag{23}$$

We report Avg IOU by averaging over the test set and, when applicable, multiple random perturbations per example.

G Ablation Analysis

While Table 3 provides a macroscopic view of component contributions, the radar charts in Figure 8 reveal how different data distributions expose specific vulnerabilities in the baseline variants.

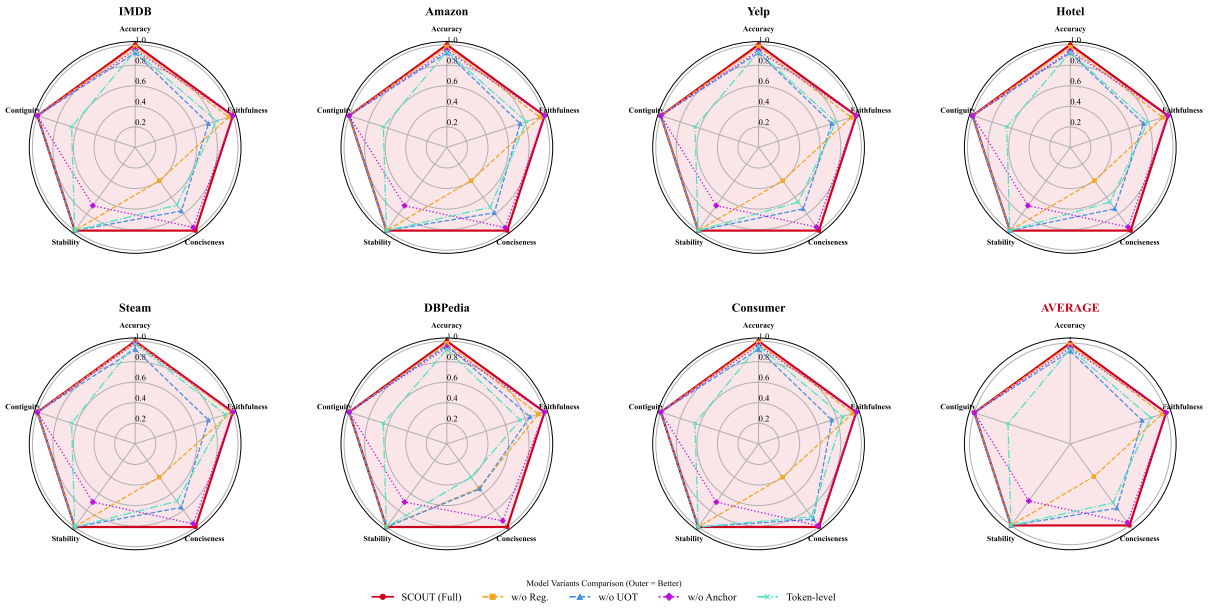

Figure 8: Multi-dimensional Ablation Analysis across Datasets.

Sensitivity to Sequence Length. On datasets with longer average sequence lengths, such as Consumer, the tension between predictiveness and conciseness is most acute. As shown in the bottom-right plot of Figure 8, the w/o Regularization variant (orange dashed line) exhibits a catastrophic collapse along the Conciseness axis. This visualizes the phenomenon where, without the sparsity constraint provided by Eq. 13, the model defaults to retaining nearly $2\times$ the necessary tokens (see Table 3) to maximize accuracy, effectively failing as an interpretable model.

Sensitivity to Noise. Conversely, on the Steam dataset, which is characterized by high user noise and informal language, the primary challenge is causal alignment. The w/o Unbalanced OT variant (blue dashed line) shows a distinct retraction along the Faithfulness axis. This confirms that standard balanced transport forces the alignment of noisy spans, introducing spurious correlations that degrade the faithfulness of the rationale.

The "Easy" Dataset. In contrast, on the structurally simpler DBPedia dataset, the performance polygons of all variants are tightly clustered. This indicates that for simple topic classification tasks, the margin for error is small, and even baseline mechanisms can perform adequately. However, SCOUT still maintains a slight edge on the outer perimeter, demonstrating that its sophisticated alignment mechanism does not introduce overhead on simple tasks.

H Prototype Examples

To complement the quantitative human evaluation in Section 6, we present a set of representative prototype spans extracted by SCOUT from the Yelp training corpus. These spans serve as the class-specific evidence against which input reviews are compared via unbalanced optimal transport. Note that, unlike brittle keyword matching, each prototype captures a coherent, semantically complete phrase (e.g., *"the trout was basically cold"* rather than just *"cold"* or *"food"*).

These examples illustrate SCOUT's ability to anchor on readable, interpretable evidence that captures the nuanced sentiment of the training data. Importantly, the unbalanced optimal transport mechanism ensures that during inference, only input spans that exhibit a meaningful semantic correspondence to such prototypes are assigned non-zero transport mass.

I Efficiency and Complexity

This section details the computational and memory costs of SCOUT, and summarizes practical implementation choices that make Unbalanced OT feasible for long documents.

I.1 Span Enumeration and Prefiltering

Let n_w be the number of words after recovering word boundaries from subwords. We enumerate all contiguous spans with length $\ell \in [L_{\min}, L_{\max}]$,

Negative Prototypes (Class 0)	Positive Prototypes (Class 1)
"the trout was basically cold."	"the restaurant is dark romantic and the decor is amazing."
"the staff is full of attitude"	"The steak was cooked perfectly"
"unnecessary 1 hour wait for our food"	"huge portions and great price."
"their prices were a bit high for the size of the portions, and their wings were rather small."	"an incredibly sweet, efficient, and talented bartender"
"What do you want me to do about it?"	"Overall, I would definitely recommend Pino's for a nice, romantic date night out in Point Breeze."

Table 6: Example prototype spans from the Yelp dataset.

yielding

$$M_{\text{all}} = \sum_{\ell=L_{\min}}^{L_{\max}} (n_w - \ell + 1). \tag{24}$$

Span embedding extraction is linear in the number of pooled token vectors, and the scoring head computes a scalar per span with cost $O(M_{\text{all}}d)$.

To control the downstream OT cost, we keep the top-N spans by score and renormalize their weights to form the document measure μ_x. This reduces the OT size from M_{all} to $M = N$ (default $N = 50$), making the UOT layer cost predictable and largely independent of raw document length.

I.2 UOT Layer Time Complexity

For a document distribution with M support points and a prototype distribution with M_p support points, each generalized Sinkhorn iteration (Algorithm 1) requires two log-sum-exp reductions over an $M \times M_p$ matrix, giving per-iteration cost $O(MM_p)$. With L iterations, the cost per (document, prototype) pair is:

$$O(LMM_p). \tag{25}$$

At inference, we compute distances to K prototypes for each class among C classes, thus the total UOT cost per document is:

$$O(CK\,LMM_p). \tag{26}$$

Using default hyperparameters (M=50, M_p=5, L=20, K=10), this cost is modest in absolute floating-point operations, but can introduce overhead if implemented with Python loops. We therefore implement Sinkhorn iterations in a fully vectorized manner (batching over CK) on GPU.

I.3 Memory Complexity

The coupling matrix $T \in \mathbb{R}^{M\times M_p}$ requires $O(MM_p)$ memory per prototype. In practice, we do not need to materialize all couplings for all prototypes simultaneously: we compute the log-coupling and immediately reduce it to span masses $m = T\mathbf{1}$ and the objective value $d_{c,k}(x)$. This yields peak memory:

$$O(B \cdot R \cdot MM_p), \tag{27}$$

where B is batch size and R is the number of prototype pairs evaluated in parallel (e.g., $R = CK$ for full parallelism, or a smaller chunk size for memory savings).

I.4 Practical Acceleration Strategies

We adopt several implementation choices to reduce latency without affecting the definition of SCOUT.

Prototype pre-screening. When C or K is large, we can pre-select a small set of candidate prototypes using a cheap scalar heuristic (e.g., cosine similarity between a pooled document embedding and prototype centroids), then run UOT only for the top-R candidates. This reduces CK in $O(CK\,LMM_p)$ to $R \ll CK$.

Vectorized Sinkhorn on GPU. We batch the cost matrices across prototypes and classes into a tensor $\mathcal{C} \in \mathbb{R}^{(CK)\times M\times M_p}$ and run log-domain updates in parallel. This removes Python overhead and makes runtime dominated by efficient GPU kernels.

Mixed precision. All encoder computations are run in mixed precision when available. For the UOT layer, we keep log-domain accumulations in FP32 for numerical stability, while allowing FP16/FP32 inputs.

Anchoring cost. The prototype anchoring step projects each support point onto a class-specific span bank once per epoch. If the bank is large, approximate nearest-neighbor search (e.g., IVF/HNSW) reduces projection time substantially. Since anchoring is performed outside the forward/backward passes, it does not affect per-step training throughput.

I.5 End-to-End Training Cost

The total training time is dominated by: (i) the transformer encoder forward/backward pass, (ii) span scoring and top-N selection, and (iii) computing UOT distances used by the classification and clustering/separation losses. With fixed (M, M_p, K, L), the UOT cost scales linearly with batch size and can be reliably bounded. In our implementation, we compute all prototype distances needed for the soft-min aggregation in Eq. (8) in a single batched call for efficiency.