LEARNING TO EXPLAIN WHERE MODELS DISAGREE

Anonymous authors
Paper under double-blind review

ABSTRACT

Aggregate benchmark performance often does not reveal *where* two classifiers disagree. We study how to turn a small set of labeled agreement and disagreement examples into a concise natural-language description that transfers to new inputs. We introduce SEAM , an explanation policy trained with reinforcement learning to maximize pairwise simulatability: the accuracy with which an observer can use an explanation to predict whether the classifiers agree on an unseen input. The reward uses automatically obtained agreement between models and requires no human-written explanations or preference labels. Across 15 structured and text classification tasks held out from training, SEAM achieves 0.75 simulatability, compared with 0.69 for GPT-4o and 0.51 for its untrained base model. Its advantage persists for observer models excluded from reward computation. On 4 held-out multimodal tasks, a jointly trained vision-language variant achieves 0.68 simulatability. The advantage measured by the LLM observer also appears for human participants: in a controlled study with 60 participants, SEAM explanations support 74.0% agreement-prediction accuracy, compared with 55.4% for GPT-4o explanations. These results show that direct optimization can produce transferable natural-language summaries of pairwise model behavior.[1]

1 INTRODUCTION

Machine learning practitioners increasingly choose among models that look similarly strong on benchmarks, but benchmark scores hide *where* models differ. Two models with similar accuracy can fail on very different inputs, and a single aggregate metric cannot determine which model to trust in which setting. Practitioners instead need explanations that characterize *when and where* models behave differently that are interpretable and actionable.

Prior works have explored different aspects of this problem. `ModelDiff` (Shah et al., 2022) characterized difference between two learning algorithms by identifying input transformations that affect one model's prediction but not the other's. `SLED` (Malladi et al., 2025) learns synthetic samples that highlight behavioral differences between models and prompts LLMs to explain these differences. The problem of presenting differences between competing models in natural language consists of two sub-problems: (1) *Sample Discovery*, which identifies the inputs or transformations where differences between the models become apparent, and (2) *Explanation Generation*, which creates explanations to better capture the patterns exhibited by these samples. Existing methods primarily focus on Sample Discovery, while treating Explanation Generation as a generic prompting step.

A separate line of work shows that explanation generation can be optimized. `MaNtLE` (Menon et al., 2023) learns explanations from human-written rationales, and `OPeX` (Malladi & Srivastava, 2026) uses reinforcement learning to produce explanations that help an observer predict a single model's behavior. However, these methods target individual models, and prior work on model comparison has largely focused on structured or text-only settings. This leaves two open challenges: optimizing explanations of *differences* between models (rather than explaining each separately), and extending them to multimodal models where disagreements may depend jointly on text and images. Given examples where two classifiers agree and disagree, can an explanation predict where they will disagree on new inputs?

We call this objective contrastive diagnosticity. An explanation has high contrastive diagnosticity when an observer, whether an LLM or a person, can use it to predict whether two classifiers

[1] Code, data, and model weights will be released upon first publication.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

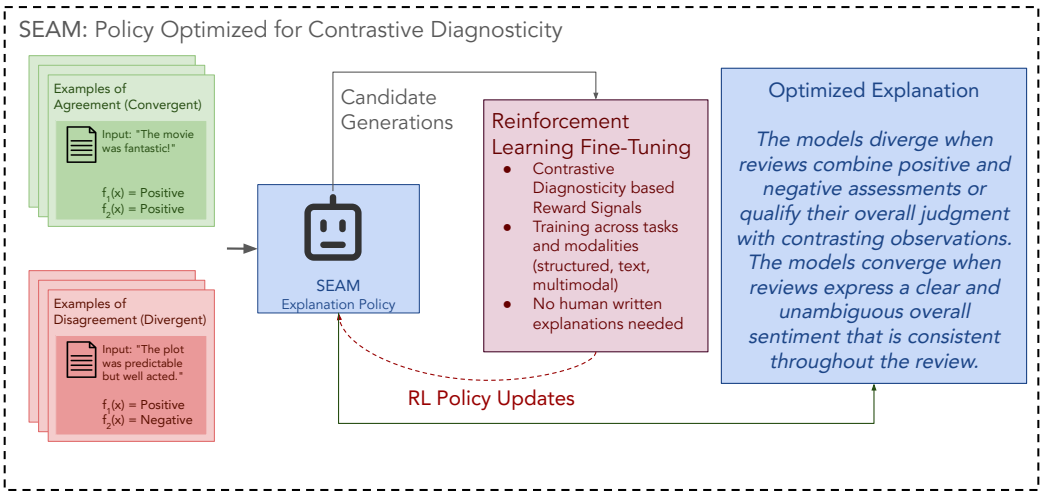


Figure 1: Overview of SEAM . Given two competing models and their convergent/divergent behaviors, SEAM learns explanations optimized for contrastive diagnosticity, enabling observers to identify model behavior on unseen inputs.

will agree on an input without seeing either classifier's prediction. Following Malladi & Srivastava (2026), we evaluate this through *recoverability*, the observer's accuracy on the examples used to generate the explanation, and *simulatability*, its accuracy on disjoint examples. Simulatability is our primary measure because it tests whether the description extends to unseen inputs. The target is limited to agreement between classifiers: it does not establish which classifier is correct or explain either classifier's internal reasoning. Agreement labels can be obtained directly from classifier predictions, so this objective can be optimized without human-written explanations or preferences.

We introduce **SEAM** , a natural language explainer optimized through reinforcement learning to generate explanations for precise qualitative analysis and comparison of ML models for the same task. We introduce two main variants: SEAM , optimized for text-based and structured classification tasks, and SEAM-VL , extended to multimodal (vision-language) tasks. Across a diverse suite of structured and text-based tasks, SEAM explanations achieve a recoverability score of 85% and a simulatability score of 75%, while SEAM-VL achieves 78% recoverability and 68% simulatability on multimodal tasks. We further find that SEAM explanations significantly outperform chain-of-thought (CoT)-based explanations from thinking models. Finally, SEAM explanations enable human users to achieve 18% higher accuracy on new, unseen samples than strong LLM baselines.

This work makes three contributions: (1) **Direct optimization of pairwise model-difference explanations**. We formulate contrastive diagnosticity as an observer's ability to use a natural-language explanation to predict agreement between two models, and introduce SEAM, which optimizes this objective with reinforcement learning using automatically obtained agreement labels. (2) **Multimodal model comparison**. We introduce SEAM-VL, extending the same framework to vision-language classifiers and showing that explanations can capture behavioral differences that depend on text, images, or both. (3) **Generalization beyond the training setting**. We show that the resulting explanations transfer to held-out tasks and classifier pairs, remain effective under observers excluded from training, and improve human agreement prediction on unseen examples.

2 SEAM

Given examples on which two classifiers agree or disagree, our goal is to produce a natural language explanation that allows an observer to anticipate their agreement on new inputs. We take the examples as given and focus on optimizing the explanation itself. We first define the pairwise agreement task and its observer-based evaluation. We then describe the explanation policy and the reinforcement learning procedure used to optimize it.

2.1 Problem Formulation

Our setting follows Malladi et al. (2025), but focuses on explanation optimization rather than sample discovery. Let $f_1, f_2 : \mathcal{X} \to \mathcal{Y}$ be two classifiers for the same task $\mathcal{T}$. For an input $x \in \mathcal{X}$, we define their agreement relation as

$$z(x) = \begin{cases} C & \text{if } f_1(x) = f_2(x), \\ D & \text{if } f_1(x) \neq f_2(x), \end{cases} \tag{1}$$

where C and D denote convergence and divergence. Equivalently, we define the spaces of convergent and divergent samples as:

$$\mathcal{X}_C = \{x \in \mathcal{X} : z(x) = C\}, \qquad \mathcal{X}_D = \{x \in \mathcal{X} : z(x) = D\}. \tag{2}$$

For a fixed task and classifier pair, we construct a conditioning set $S = S_C \cup S_D$, where $S_C \subseteq \mathcal{X}_C$ and $S_D \subseteq \mathcal{X}_D$ contain equal numbers of convergent and divergent examples. An explanation policy π_θ receives the labeled examples and generates a natural-language explanation

$$e \sim \pi_\theta(\cdot \mid S). \tag{3}$$

The explanation should characterize input patterns associated with agreement and disagreement. Given e and a new input x, an observer should be able to predict $z(x)$ without seeing either classifier's prediction. We note that the prediction target here is limited to the agreement relation between the classifiers. It does not identify which classifier is correct or explain the classifiers' internal reasoning.

2.2 Observer-Based Diagnosticity

Many established explanation metrics evaluate whether a description is faithful to the predictions of a single model (Jacovi & Goldberg, 2020). They do not directly assess whether a description captures the agreement pattern between two models. We use observer accuracy on agreement prediction to operationalize this pairwise property, which we call *contrastive diagnosticity*. An observer O, which may be a language model or a human, receives an explanation e and an input x and predicts

$$\hat{z}_O(x, e) = O(x, e) \in \{C, D\}. \tag{4}$$

The observer does not receive $f_1(x)$ or $f_2(x)$. These predictions are used only to compute the reference label $z(x)$. For an evaluation set A, we define the observer-specific contrastive diagnosticity of e as

$$\mathrm{CD}_O(e; A) = \frac{1}{|A|} \sum_{x \in A} \mathbb{I}[\hat{z}_O(x, e) = z(x)]\,. \tag{5}$$

This quantity depends on the explanation, observer, and evaluation set. Because our evaluation sets contain equal numbers of convergent and divergent examples, CD_O is a balanced agreement-prediction accuracy with chance performance of 0.5.

Recoverability. Following prior work on behavioral faithfulness and recoverability (Jacovi & Goldberg, 2020; Malladi & Srivastava, 2026), we measure whether the observer can recover the agreement pattern among the examples used to generate the explanation:

$$\mathrm{Rec}_O(e; S) = \mathrm{CD}_O(e; S). \tag{6}$$

Recoverability measures how well the explanation accounts for its conditioning evidence. It does not test whether the explanation extends beyond that evidence.

Simulatability. Following simulatability-based evaluation of explanations (Hase & Bansal, 2020; Malladi & Srivastava, 2026), we evaluate the explanation on a disjoint held-out set V from the same task and classifier pair:

$$\mathrm{Sim}_O(e; V) = \mathrm{CD}_O(e; V), \qquad S \cap V = \emptyset. \tag{7}$$

Simulatability is our primary measure because it tests whether the explanation captures an agreement pattern that extends to previously unseen inputs. Unless otherwise stated, we use

`Qwen3-VL-32B-Instruct` (Bai et al., 2025a) as the observer. This model supplies both the training reward and the primary automatic evaluation scores. The shared observer makes large-scale evaluation possible, but also creates the possibility that the learned explanations are tailored to that observer. Therefore, in §4, we re-evaluate generated explanations with alternative LLM observers and with human participants, and ensure that the explanations transfer to new observers. Equation 7 also defines the human evaluation, with participant judgments replacing the LLM prediction.

2.3 Explanation Policy

Unless otherwise stated, SEAM is initialized from `Qwen2.5-VL-32B-Instruct` (Bai et al., 2025b). Unlike `OPeX` (Malladi & Srivastava, 2026), which optimizes explanations of individual model behavior, SEAM generates a single explanation of the agreement pattern between two models. Each policy input contains N convergent and N divergent examples from one task and classifier pair. We linearize each example together with its agreement label and instruct the policy to infer patterns that distinguish the two groups and extend beyond the provided examples. The generated explanation is a concise paragraph following the form "The models diverge when ... The models converge when ...". The complete generation prompt is provided in § A.1.

Each input concerns a single task and classifier pair, while training spans many task–classifier combinations. This exposes the policy to varied agreement patterns and encourages transfer to new tasks and unseen trained classifier instances.

2.4 Reinforcement Learning Fine-Tuning

We train SEAM using multi-task reinforcement learning fine-tuning over classifier pairs from structured and text classification tasks. The structured portion contains six datasets and five classical classifier families. The text portion contains five datasets and six pretrained language-model classifier families. The complete task list, classifier configurations, pair-selection procedure, and training details are provided in Appendix A.3.

Training instances. For each task $\mathcal{T}$ and classifier pair (f_1, f_2), we sample a balanced conditioning set containing 20 convergent and 20 divergent examples:

$$S = S_C \cup S_D, \qquad |S_C| = |S_D| = 20. \tag{8}$$

We independently sample a disjoint reward set $V = V_C \cup V_D$ containing another 20 examples of each type. The policy observes S but never observes V. Each tuple $(\mathcal{T}, f_1, f_2, S, V)$ constitutes one training instance. We construct 15,000 such instances across the structured and text tasks, including synthetic structured examples generated following Malladi et al. (2025).

Because both S and V are tied to a fixed task and classifier pair, each explanation describes the agreement pattern for that pair. Variation across training instances exposes the policy to different tasks, feature spaces, classifier families, and agreement boundaries.

Learning objective. For each training instance, the policy samples a candidate explanation $e \sim \pi_\theta(\cdot \mid S)$. We score the explanation using its simulatability under the training observer O_{train}:

$$R(e; V) = \text{Sim}_{O_{\text{train}}}(e; V) = \frac{1}{|V|} \sum_{x \in V} \mathbb{I}[O_{\text{train}}(x, e) = z(x)]\,. \tag{9}$$

The reward measures whether the observer can apply the explanation to inputs withheld from the policy. The agreement labels are obtained directly from f_1 and f_2, so training requires no human-written explanations or preference annotations.

We optimize the policy using Group Relative Policy Optimization (GRPO) (Shao et al., 2024). For each conditioning set, GRPO samples a group of candidate explanations and updates the policy using their relative rewards. We apply Low-Rank Adaptation (LoRA) (Hu et al., 2021) to the policy model and use the GRPO KL penalty to limit deviation from the pretrained model. Full optimization and training details and parameters are reported in Appendix A.3.

We train two principal variants that differ in their exposure to synthetic data. SEAM-R is trained only on real structured and text examples. SEAM additionally includes synthetic structured exam-

ples generated following Malladi et al. (2025). This comparison isolates whether exposure to synthetic evidence improves transfer when explanations are later conditioned on synthetic samples. Our vision-language variants use the same observer-based objective with multimodal classifier pairs, as detailed in Appendix A.3. Training selects explanations that a fixed observer can apply to held-out inputs. Our experiments test whether these gains persist on new tasks, under observers that supplied no training reward, and for human users.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

We evaluate whether the explanation policies learned through RLFT capture behavioral differences between classifiers and generalize beyond the training examples and tasks. We evaluate SEAM explanations on classifiers trained on a diverse set of structured, text-based, and multimodal tasks. These tasks are distinct from the RLFT training tasks, resulting in evaluation on new classifiers unseen during the training of SEAM . For details of the training and evaluation tasks, see Appendix A.3.

We train classifiers, construct classifier pairs, and evaluate their behavioral differences. Across our evaluations and analyses, we compare SEAM against different subsets of baselines, including `Llama-3.1-8B-Instruct`, `Llama-3.3-70B-Instruct` (Grattafiori et al., 2024), `Qwen2.5-VL-32B-Instruct` (Bai et al., 2025b), `Qwen3-VL-32B-Instruct` (Bai et al., 2025a), `DeepSeek-R1-Distill-Llama-70B` (Guo et al., 2025), `GPT-4o` (Hurst et al., 2024), and `OPeX` (Malladi & Srivastava, 2026). These baselines span different model families, scales, architectures, and post-training approaches. Comparing SEAM with `Qwen2.5-VL-32B-Instruct`, its underlying model, isolates the effect of RLFT. For details on classifier training, see Appendix A.3

3.2 EVALUATION PIPELINE

For each evaluation task $\mathcal{T}$ and classifier pair (f_1, f_2), we construct an evaluation instance using new tasks and classifier pairs. We randomly sample a conditioning set $S = S_D \cup S_C$ containing 20 divergent samples $S_D \subseteq \mathcal{X}_D$ and 20 convergent samples $S_C \subseteq \mathcal{X}_C$, linearize them as described in § 2.3, and generate an explanation e. We then evaluate e on a disjoint held-out set $V = V_D \cup V_C$ containing 30 divergent and 30 convergent samples from the same task and classifier pair. The prompt is fixed across all experiments. We compute *recoverability* on S and *simulatability* on V using the observer LLM O and evaluation procedure described in § 2.2. Across all evaluations involving SEAM , we use `Qwen3-VL-32B-Instruct` as O, which is distinct from the `Qwen2.5-VL-32B-Instruct` base model used by SEAM . Given e and x, O predicts whether the classifiers converge or diverge, with accuracy against the ground-truth label $z(x)$ giving the corresponding score. Using the same conditioning samples, prompt, observer, and evaluation tasks across models provides a controlled comparison, while the disjoint V measures simulatability on unseen samples. For details on the prompt, see AppendixA.1.

4 RESULTS AND ANALYSES

Evaluation on Structured and Text Classification Tasks We compare the contrastive diagnosticity of explanations generated by SEAM and the baselines across structured and text-based classification tasks (Table 1). On average, SEAM achieves 0.85 recoverability and 0.75 simulatability, outperforming the baselines by 12–30 and 6–24 percentage points, respectively (paired t-test, $p < 0.01$). Compared with concatenated individual-model explanations from `OPeX`, SEAM also achieves higher scores, indicating that directly optimizing for behavioral differences is more effective than combining independently optimized explanations. We also find that SEAM achieves higher simulatability than decision trees and random forests. The strong simulatability scores indicate that SEAM captures behavioral patterns that extend beyond the conditioning examples, enabling its explanations to characterize when classifiers converge or diverge on unseen inputs. For further details, see Appendix A.4, Tables 9 and 11

	GPT-4o	SEAM	OPeX	Qwen2.5-VL-32B	Qwen3-32B	Llama-3.3-70B
Tic-Tac-Toe Endgame	**0.73 ± 0.11**	0.68 ± 0.16	0.54 ± 0.10	0.50 ± 0.11	0.54 ± 0.11	0.59 ± 0.10
Balance Scale	0.64 ± 0.15	**0.70 ± 0.14**	0.54 ± 0.10	0.47 ± 0.07	0.51 ± 0.15	0.58 ± 0.12
Magic Gamma	0.64 ± 0.06	**0.81 ± 0.15**	0.55 ± 0.10	0.50 ± 0.03	0.56 ± 0.03	0.61 ± 0.07
Wine Quality	0.63 ± 0.12	**0.77 ± 0.14**	0.53 ± 0.08	0.50 ± 0.06	0.56 ± 0.11	0.60 ± 0.08
Car Evaluation	**0.90 ± 0.16**	0.79 ± 0.14	0.47 ± 0.11	0.54 ± 0.10	0.66 ± 0.11	0.60 ± 0.15
Adult	0.73 ± 0.19	**0.81 ± 0.17**	0.54 ± 0.10	0.57 ± 0.14	0.64 ± 0.14	0.62 ± 0.17
Bank Marketing	0.81 ± 0.15	**0.87 ± 0.13**	0.54 ± 0.08	0.49 ± 0.12	0.67 ± 0.12	0.72 ± 0.12
Banknote Authentication	0.71 ± 0.12	**0.79 ± 0.16**	0.44 ± 0.11	0.53 ± 0.11	0.59 ± 0.13	0.65 ± 0.15
Rice	0.65 ± 0.03	**0.88 ± 0.15**	0.54 ± 0.15	0.54 ± 0.10	0.55 ± 0.00	0.61 ± 0.06
Nursery	0.66 ± 0.08	**0.71 ± 0.14**	0.42 ± 0.09	0.49 ± 0.09	0.56 ± 0.11	0.58 ± 0.07
AG News	0.65 ± 0.12	**0.68 ± 0.13**	0.58 ± 0.09	0.51 ± 0.09	0.55 ± 0.09	0.58 ± 0.12
CoLA	0.71 ± 0.14	**0.74 ± 0.15**	0.61 ± 0.11	0.55 ± 0.13	0.59 ± 0.14	0.68 ± 0.11
IMDb Reviews	0.55 ± 0.11	0.63 ± 0.10	0.57 ± 0.08	0.50 ± 0.07	0.53 ± 0.10	**0.64 ± 0.09**
PAWS	0.64 ± 0.12	**0.70 ± 0.13**	0.53 ± 0.11	0.50 ± 0.10	0.54 ± 0.11	0.59 ± 0.10
SST-2	0.65 ± 0.10	**0.69 ± 0.13**	0.70 ± 0.12	0.50 ± 0.07	0.56 ± 0.10	0.62 ± 0.08
Average	0.69 ± 0.15	**0.75 ± 0.16**	0.54 ± 0.12	0.51 ± 0.10	0.57 ± 0.12	0.62 ± 0.12

Table 1: Simulatability scores for explanations of classifier behavioral differences across structured and text-based classification tasks. Best results per row are shown in bold. Scores are averaged over 80 runs and reported as mean ± standard deviation.

	SEAM-VL	SEAM	SEAM-V	Qwen2.5-VL-32B	Qwen3-VL-32B
recoverability					
GQA	**0.77 ± 0.12**	0.64 ± 0.11	0.41 ± 0.15	0.54 ± 0.10	0.55 ± 0.13
MMSD 2.0	**0.80 ± 0.10**	0.66 ± 0.07	0.54 ± 0.12	0.56 ± 0.10	0.59 ± 0.09
Hateful Memes	**0.77 ± 0.09**	0.69 ± 0.12	0.56 ± 0.10	0.61 ± 0.12	0.62 ± 0.10
Crisis MMD	**0.79 ± 0.10**	0.69 ± 0.10	0.53 ± 0.09	0.60 ± 0.10	0.61 ± 0.10
Avg	**0.78 ± 0.11**	0.67 ± 0.10	0.51 ± 0.13	0.58 ± 0.11	0.59 ± 0.11
simulatability					
GQA	**0.61 ± 0.15**	0.57 ± 0.14	0.42 ± 0.14	0.47 ± 0.11	0.45 ± 0.10
MMSD 2.0	**0.68 ± 0.15**	0.61 ± 0.09	0.52 ± 0.10	0.52 ± 0.09	0.53 ± 0.11
Hateful Memes	**0.80 ± 0.11**	0.74 ± 0.12	0.57 ± 0.11	0.61 ± 0.10	0.61 ± 0.11
Crisis MMD	**0.65 ± 0.14**	0.60 ± 0.14	0.54 ± 0.11	0.50 ± 0.13	0.54 ± 0.12
Avg	**0.68 ± 0.16**	0.63 ± 0.14	0.51 ± 0.13	0.53 ± 0.12	0.53 ± 0.12

Table 2: recoverability and simulatability scores for explanations of classifier behavioral differences on multimodal classification tasks. Best results per row are shown in bold. Scores are reported as mean ± standard deviation.

Extending SEAM to Multimodal Settings We extend SEAM to multimodal settings with two variants: SEAM-VL , trained jointly on structured, text-based, and multimodal tasks, and SEAM-V , trained exclusively on multimodal tasks. Both use the RLFT framework described in § 2.4, with multimodal classifier pairs providing the behavioral differences to be explained. As shown in Table 2, on full multimodal inputs, SEAM-VL achieves 0.78 recoverability and 0.68 simulatability, compared with 0.67 and 0.63 for SEAM , which receives no multimodal training, and 0.51 and 0.51 for SEAM-V . Both SEAM-VL and SEAM significantly outperform the `Qwen2.5-VL-32B` base model on both metrics (paired t-test, $p < 0.01$). These results show that multimodal training improves performance on multimodal inputs, while the higher scores of SEAM-VL compared to SEAM-V suggest that broader training exposure also provides useful variation. For details on multimodal training and evaluation setup, see Appendix A.3.

To examine whether the explanation policy benefits from both modalities, we vary the conditioning inputs while keeping evaluation fixed to the full text-and-image inputs. As shown in Table 3, conditioning on both text and images achieves 0.78 recoverability and 0.68 simulatability, compared with 0.74/0.58 with text-only inputs and 0.72/0.59 with image-only inputs. The improvement with both modalities provides evidence that the policy uses complementary information from the text and image when generating explanations.

Different Explanation Budgets We next investigate how the explanation budget affects the contrastive diagnosticity of SEAM . We train separate SEAM variants with maximum explanation budgets of 16, 32, 64, 128, 256, and 512 tokens and evaluate each variant on recoverability and simulata-

Conditioning Inputs	Recoverability	Simulatability
Only Text	0.74 ± 0.09	0.58 ± 0.13
Only Image	0.72 ± 0.10	0.59 ± 0.12
Text + Image	$\mathbf{0.78 \pm 0.11}$	$\mathbf{0.68 \pm 0.16}$

Table 3: Recoverability and simulatability of SEAM-VL when explanations are generated from different input modalities. Evaluation is performed using both text and image inputs in all conditions.

Budget	512	256	128	64	32	16
Sim.	0.75 ± 0.16	0.71 ± 0.13	0.61 ± 0.14	0.56 ± 0.13	0.51 ± 0.12	0.50 ± 0.08

Table 4: We train separate SEAM variants with different explanation budgets and evaluate each variant on simulatability. Reducing the budget from 512 to 256 tokens results in a modest decrease in simulatability, while further reductions progressively degrade performance.

bility. As shown in Table 4, we observe that reducing the budget from 512 to 256 tokens results in only a modest decrease in both recoverability and simulatability, with scores decreasing from 0.85 to 0.75 and from 0.75 to 0.71, respectively. However, further reducing the budget leads to progressively larger decreases in both metrics, with performance approaching chance levels at 16 tokens. These results suggest that SEAM does not require the full 512-token budget to achieve strong contrastive diagnosticity, but explanations require a minimum length to effectively capture and communicate the behavioral differences between classifiers. We also find that the 256-token variant of SEAM outperforms `GPT-4o` (paired t-test, $p < 0.05$) in terms of simulatability. For further details, see Appendix A.4 Figure 2).

Syntactic Structure We compare the structure of explanations generated by SEAM and the baseline language models. SEAM produces more rules and conjunctive rules per explanation, with 4.49 and 3.03 on average compared to 2.00–2.84 and 1.24–2.06 for the baselines. However, its average maximum number of conjuncts per rule (3.47) is comparable to Llama-3.3-70B (3.43) and Qwen2.5-VL-32B (3.23), indicating that the additional rules reflect broader coverage rather than substantially more complex individual rules. This is also reflected in the number of atomic conditions: SEAM produces 9.91 per explanation, compared to 5.52 for Llama-3.3-70B, 6.90 for Qwen2.5-VL-32B, 3.87 for DeepSeek-R1-70B, and 4.75 for GPT-4o. SEAM also produces more divergence and convergence rules, with 2.42 and 2.41 per explanation compared to substantially fewer for the baselines. Its Div./Conv. Balance of 1.03 is comparable to the baselines (1.01–1.03), indicating that the additional rules cover both behavioral outcomes rather than disproportionately focusing on divergence. Overall, SEAM explanations are highly precise and explicitly rule-based, focusing on concrete feature conditions rather than unnecessary conceptual abstraction, guesswork, or extended reasoning. Their larger number of rules and atomic conditions suggests that they recover a broader set of behavioral differences between the classifiers, while the bounded complexity of individual rules keeps the explanations interpretable and easy to apply. All analysis used deterministic syntactic parsing; details are in Appendix A.4. For explanation examples, see Appendix A.4 Table 12. For further details on results, see Appendix Table 13

Qualitative Analysis We examine representative cases (Table 5) where the explanations generated by `GPT-4o` and SEAM succeed or fail to provide diagnostic descriptions of model differences. On the Wine Quality task, SEAM provides a more precise and operational explanation than `GPT-4o`. It identifies explicit thresholds on `chlorides` and `density`, specifying that divergence occurs when both values are below 0.05 and 0.995, respectively, while convergence occurs when either exceeds its corresponding threshold. In contrast, `GPT-4o` describes broader variability involving acidity, residual sugar, and sulfur dioxide without specifying concrete conditions that distinguish divergent from convergent samples. SEAM also achieves higher diagnosticity, suggesting that explicit feature-level conditions can provide a more useful characterization of model differences. The Car Evaluation task provides a contrasting case. Although SEAM produces a richer explanation involving `maint`, `safety`, `doors`, `persons`, and `lug_boot`, `GPT-4o` achieves higher diagnosticity despite reducing the distinction largely to `safety`. This suggests that greater detail does not necessarily improve diagnosticity: explicitly enumerating many conditions may make it harder for an

	GPT-4o	SEAM
Car Evaluation	The models diverge when the safety feature is high, regardless of other feature values, indicating that high safety is a critical factor causing prediction differences between models. The models converge when the safety feature is low or medium, suggesting that lower safety levels lead to consistent predictions across models, regardless of variations in other features.	The models diverge when the combination of `maint` (maintenance cost) and `safety` (safety rating) is such that one is high or very high while the other is medium or high, regardless of other features, or when `doors` is 4 and `persons` is more, or when `doors` is 5more and `lug_boot` is big. The models converge when `safety` is low, or when `maint` is low and `doors` is 2, or when `doors` is 5more and `persons` is more, or when `doors` is 4 and `lug_boot` is small, or when `doors` is 3 and `persons` is 2, or when `doors` is 5more and `lug_boot` is med, or when `doors` is 4 and `persons` is 4, or when `doors` is 2 and `persons` is 4, or when `doors` is 3 and `persons` is more, or when `doors` is 5more and `persons` is 4.
Wine Quality	The models diverge when there is a higher variability in the balance between fixed acidity and volatile acidity, coupled with a wider range of residual sugar levels and total sulfur dioxide concentrations, leading to inconsistencies in density and alcohol content. The models converge when these features exhibit more consistent ratios, particularly with moderate fixed acidity and volatile acidity levels, stable residual sugar, and total sulfur dioxide concentrations, resulting in more predictable density and alcohol content.	The models diverge when the `chlorides` value is relatively low (typically below 0.05) and the `density` value is relatively low (typically below 0.995), regardless of other feature values. The models converge when the `chlorides` value is relatively high (typically above 0.05) or the `density` value is relatively high (typically above 0.995), or both, regardless of other feature values. These conditions consistently separate the divergent and convergent samples, with no exceptions in the provided data.

Table 5: Qualitative comparison of explanations generated by GPT-4o and SEAM on the Car Evaluation and Wine Quality tasks. On Wine Quality, SEAM achieves higher diagnosticity by providing explicit, actionable feature-level conditions, whereas GPT-4o gives broader qualitative patterns. Conversely, on Car Evaluation, GPT-4o achieves higher diagnosticity, while the greater number of rules in SEAM may make the explanation harder for observers to navigate and apply.

Conditioning Set	GPT-4o	SEAM	SEAM-R	Qwen2.5-VL-32B-Instruct
Real	0.69 ± 0.15	$\mathbf{0.75 \pm 0.16}$	0.70 ± 0.15	0.51 ± 0.10
Synthetic (`SLED`)	0.70 ± 0.15	$\mathbf{0.77 \pm 0.17}$	0.69 ± 0.18	0.55 ± 0.14

Table 6: Average simulatability scores across all tasks when explanations are conditioned on real or SLED-generated synthetic samples and evaluated on held-out real samples.

observer to navigate the explanation and determine which rule applies to a given input. Thus, an explanation can be more comprehensive yet less effective for downstream diagnosis. For further limitations of our work, see Appendix A.4.

Explanations Conditioned on Synthetic Samples In practical settings, access to representative real-world samples may be limited. We therefore evaluate whether explanations conditioned on `SLED`-generated synthetic samples (Malladi et al., 2025) transfer to real-world data. As shown in Table 6, SEAM achieves a simulatability score of 0.77 with synthetic conditioning, matching its 0.75 score with real conditioning. This provides evidence that SEAM can extract behavioral patterns from synthetic samples that remain predictive on the real data distribution. Moreover, SEAM outperforms SEAM-R under synthetic conditioning (0.77 vs. 0.69), demonstrating the benefit of synthetic-data exposure during RLFT. For full results, see Appendix A.4 Table 10

Transfer across Observer Models All explanation policies were trained using rewards computed by `Qwen3-VL-32B-Instruct`. We then froze the generated explanations and evaluated them using four observer models that did not provide training rewards: `Qwen3-32B`, `GPT-4o`,

`GPT-4o-mini`, and `Llama-3.3-70B-Instruct`. Recoverability and simulatability remain stable across observers, varying by at most two percentage points (0.83–0.85 for recoverability and 0.73–0.75 for simulatability). The gains over the baselines also remain consistent across observers. When evaluated by `Llama-3.3-70B-Instruct`, SEAM achieves 0.84 recoverability and 0.74 simulatability, compared with 0.74 and 0.70, respectively, for `GPT-4o`, preserving the advantage of SEAM under an independent observer. These results indicate that the learned explanations transfer beyond the specific evaluator used to compute rewards during model training. For details, see Appendix Figure 3b.

Transfer to Human Observers We evaluate the downstream usefulness of SEAM explanations with human users by comparing them against GPT-4o on the *Rice* classification task. We recruit 60 participants through Prolific and randomly assign 30 participants to each condition. For each evaluation instance, SEAM and GPT-4o receive the same set of inducing samples from a fixed task–classifier pair and generate their respective explanations. Participants in both conditions then classify the same set of 30 disjoint, previously unseen samples from that task–classifier pair as either *convergent* or *divergent*, without access to either classifier's predictions. Thus, the only difference between the two conditions is the model used to generate the explanation. Participants using SEAM explanations achieve 74% accuracy, compared to 55.4% with GPT-4o explanations (two-way t-test, $p < 0.01$), demonstrating that SEAM explanations enable more accurate identification of classifier behavior on unseen inputs. For further details, see Appendix A.4.

Effect of Policy Model Scale We examine whether RLFT improves contrastive diagnosticity across policy model scales by comparing SEAM , initialized from `Qwen2.5-VL-32B-Instruct`, and SEAM-LLAMA , initialized from `Llama-3.1-8B-Instruct`, with their respective base models. RLFT improves the Qwen-based model from 0.55 to 0.85 in recoverability and from 0.51 to 0.75 in simulatability, while improving the Llama-based model from 0.51 to 0.72 and from 0.49 to 0.67, respectively. Despite its substantially smaller size, SEAM-LLAMA achieves 0.72 recoverability and 0.67 simulatability, showing that the gains from contrastive-diagnosticity-driven RLFT extend across policy model scales. For details, see Appendix Figure 3a,

Comparison with Reasoning-Based Language Models We next compare SEAM with reasoning-based language models to evaluate whether explicit reasoning improves their ability to characterize behavioral differences between classifiers. SEAM achieves a recoverability score of 0.85 and a simulatability score of 0.75, compared to 0.71 and 0.67, respectively, for `DeepSeek-R1-Distill-Llama-70B`. SEAM explanations significantly outperform explanations generated by a thinking model (paired t-test, $p < 0.01$). These results indicate that directly optimizing explanations for contrastive diagnosticity enables SEAM to more effectively capture and communicate behavioral differences between classifiers than reasoning-based generation.

5 CONCLUSION

We introduced SEAM , an explanation policy that uses reinforcement learning to characterize how the behaviors of two competing models differ. We formalized *contrastive diagnosticity* as the ability of an explanation to identify whether models will converge or diverge on an input, and evaluated it through recoverability and simulatability. Across structured and text-based classification tasks, SEAM achieves 0.85 recoverability and 0.75 simulatability, substantially outperforming general-purpose and CoT-based language-model baselines. Our analyses further show that these gains arise from generating explicit, feature-level rules that generalize beyond the samples used to construct an explanation, rather than from simply producing longer or more elaborate descriptions. The benefits of contrastive-diagnosticity-driven RLFT also extend across policy model scales, remain robust across observer models, and persist when explanations are conditioned on synthetic samples. We further extend SEAM to multimodal model comparison through SEAM-VL , achieving 0.78 recoverability and 0.68 simulatability across multimodal tasks. Finally, a user study demonstrates that the resulting explanations are not only effective for automated observers but also improve human's ability to predict model behavior on unseen inputs. These results suggest that directly optimizing explanations for their downstream diagnostic utility can produce concise and actionable descriptions of model differences.

AI USE STATEMENT

In this work, we used generative AI tools for minor assistance with software code and grammar editing. We have not used generative AI tools for any other required-disclosure tasks, including developing the research methodology, designing experiments, formulating hypotheses, interpreting results, or conducting qualitative analysis. All research ideas, methodology, experiments, analyses, claims, and citations were developed by the authors. All AI-assisted code and text edits were reviewed by the authors, and we take full responsibility for the final content of this work, including all text, claims, analyses, and artifacts.

ETHICS STATEMENT

This work includes a human evaluation conducted with 60 participants recruited through Prolific. The study involved a low-risk task in which participants predicted whether pairs of classifiers would converge or diverge on previously unseen examples based solely on generated explanations, without access to the classifiers' predictions. The study did not collect sensitive or personally identifying information from participants. Participants were fairly compensated for their time at a rate substantially higher than applicable minimum wage requirements. The study was designed solely to evaluate the usefulness of model-generated explanations and did not involve potentially harmful interventions, sensitive personal data, or high-risk applications. We take responsibility for the ethical conduct and reporting of the study and for the responsible use of the data and models involved.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of our methodology, experimental setup, model training, and evaluation procedures to facilitate reproduction of our results. Details necessary to replicate all experiments and model training can be found in § 2, 3.1 and Appendix A.3, including the relevant training configurations, datasets, classifier construction, evaluation protocols, and implementation details.

REFERENCES

Firoj Alam, Ferda Ofli, and Muhammad Imran. Crisismmd: Multimodal twitter datasets from natural disasters. In *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*, June 2018.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL https://doi.org/10.5281/zenodo.5297715. If you use this software, please cite it using these metadata.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association*

for Computational Linguistics, pp. 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747/.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language, 2017. URL https://arxiv.org/abs/1703.04009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Aaron Grattafiori et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Peter Hase and Mohit Bansal. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5540–5552, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.491. URL https://aclanthology.org/2020.acl-main.491/.

Chenhao He and Pramit Saha. Investigating yolo models towards outdoor obstacle detection for visually impaired people, 2023. URL https://arxiv.org/abs/2312.07571.

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

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering, 2019. URL https://arxiv.org/abs/1902.09506.

Aaron Hurst et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4198–4205, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL https://aclanthology.org/2020.acl-main.386/.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The uci machine learning repository. https://archive.ics.uci.edu, 2024.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes, 2021. URL https://arxiv.org/abs/2005.04790.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019. URL https://arxiv.org/abs/1908.03557.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Advaith Malladi and Shashank Srivastava. Directly optimizing natural language explanations for behavioral faithfulness: Simulatability and recoverability. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=8fAh8C27Hj.

Advaith Malladi, Rakesh R Menon, Yuvraj Jain, and Shashank Srivastava. Explaining differences between model pairs in natural language through sample learning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 28737–28759, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1462. URL https://aclanthology.org/2025.emnlp-main.1462/.

Rakesh Menon, Kerem Zaman, and Shashank Srivastava. MaNtLE: Model-agnostic natural language explainer. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13493–13511, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.832. URL https://aclanthology.org/2023.emnlp-main.832/.

Kai Nakamura, Sharon Levy, and William Yang Wang. Fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 6149–6157, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.755/.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, 2005.

Shahriar Parvez. multiclass-sentiment-analysis-dataset. https://huggingface.co/datasets/Sp1786/multiclass-sentiment-analysis-dataset, 2023.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, 2016. URL https://arxiv.org/abs/1505.04870.

Libo Qin, Shijue Huang, Qiguang Chen, Chenran Cai, Yudi Zhang, Bin Liang, Wanxiang Che, and Ruifeng Xu. Mmsd2.0: Towards a reliable multi-modal sarcasm detection system, 2023. URL https://arxiv.org/abs/2307.07135.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In John DeNero, Mark Finlayson, and Sravana Reddy (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 97–101, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-3020. URL https://aclanthology.org/N16-3020/.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL https://arxiv.org/abs/1910.01108.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Harshay Shah, Sung Min Park, Andrew Ilyas, and Aleksander Madry. Modeldiff: A framework for comparing learning algorithms, 2022. URL https://arxiv.org/abs/2211.12491.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model, 2022. URL https://arxiv.org/abs/2112.04482.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019. In the Proceedings of ICLR.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, 2018.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019.

Xiao Xu, Chenfei Wu, Shachar Rosenman, Vasudev Lal, Wanxiang Che, and Nan Duan. Bridgetower: Building bridges between encoders in vision-language representation learning, 2024. URL https://arxiv.org/abs/2206.08657.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In *Proc. of EMNLP*, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*, 2019.

A Appendix

A.1 Prompts

Table 7 and Table 8 present the explainer and observer prompts used across our experiments and model families. We selected these prompts through multiple iterations based on their consistent performance.

A.2 Related Work

A large body of prior research has focused on generating explanations for classification rationale at the instance level. LIME (Ribeiro et al., 2016) explains individual predictions by fitting an interpretable model locally around a given input, using input perturbations to identify the features that most influence the prediction, while Anchors (Ribeiro et al., 2018) uses high-precision rules to identify sufficient conditions under which a model's prediction remains stable, helping users understand and predict behavior on unseen instances. In contrast, another line of work focuses on explaining classifier rationale at the global level. `MaNtLE` (Menon et al., 2023) generates natural-language explanations from sets of input-prediction pairs rather than feature attributions or rules, and is trained through supervised fine-tuning on a diverse set of explanations. `OPeX` (Malladi & Srivastava, 2026) instead uses reinforcement learning to optimize natural-language explanations for the behavioral faithfulness of individual models, directly optimizing explanation quality without relying on human-written explanation annotations.

As mentioned in Section § 1, explaining differences between models for a task can be viewed as two complementary problems: sample discovery and explanation optimization. Most existing work focuses on sample discovery. `ModelDiff` (Shah et al., 2022) addresses sample discovery by identifying distinguishing input transformations that induce different predictions from models trained with different learning algorithms. It leverages data attribution and datamodels to identify subpopulations where models behave differently, but does not generate natural-language explanations of these differences. `SLED` (Malladi et al., 2025) extends model comparison to natural-language explanations by using gradient-based optimization to synthesize divergent and convergent samples, which are then provided to an LLM to generate an explanation. Thus, its explanation generation is primarily driven by the quality of the discovered samples rather than by explicitly optimizing the explanation itself.

System Instruction	You generate concise natural-language explanations of the high-level patterns that distinguish divergent samples from convergent samples. Output only the explanation paragraph.
Prompt	You are given a set of samples: {samples} Each sample is labeled either divergent or convergent. Divergent samples are those where two machine learning models produce different predictions. Convergent samples are those where the models produce the same predictions. Your task is to infer the underlying high-level patterns that distinguish divergent samples from convergent samples. Important requirements: • Do NOT describe, summarize, or enumerate the individual samples. • Do NOT simply restate feature values appearing in the examples. • Infer the deeper abstract feature relationships and interactions that consistently separate the two groups. • Identify the generalizable rule or structure underlying divergence and convergence. • The explanation must apply to new, unseen samples with similar characteristics. • The divergence conditions must hold for all divergent samples and none of the convergent samples. • The convergence conditions must hold for all convergent samples and none of the divergent samples. • Focus only on patterns present in the data. • Do not use external or real-world knowledge. • Generate a single concise but precise paragraph. • Use the exact format: "The models diverge when ... The models converge when ..." • Output ONLY the explanation paragraph and no additional text, formatting, warnings, or notes.

Table 7: System instruction and prompt used for explanation generation.

System Instruction	You are an expert pattern analyst. Given an explanation describing the high-level patterns distinguishing divergent and convergent samples, classify each input sample as either "divergent" or "convergent". Output only one word: divergent or convergent.
Prompt	Given the following explanation: {explanation} Classify this input sample based on the explanation: {sample} Requirements: • Predict whether the sample is divergent or convergent. • Use only the provided explanation and input features. • The classification must follow the abstract high-level patterns described in the explanation. • Generalize beyond surface-level feature matching. • Output exactly one word: divergent or convergent. • Do not output any additional text, reasoning, punctuation, or formatting.

Table 8: System instruction and prompt used by the observer language model for downstream convergence–divergence classification.

Reinforcement learning has become an important paradigm for optimizing language models toward desired behaviors. Proximal Policy Optimization (PPO) (Schulman et al., 2017) uses a clipped surrogate objective to constrain policy updates and has been widely used for reinforcement learning from human feedback. Direct Preference Optimization (DPO) (Rafailov et al., 2024) directly opti-

mizes language models from preference pairs, avoiding explicit reward modeling and online policy optimization. More recently, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) estimates relative advantages from groups of sampled responses, enabling optimization with scalar rewards without a separate value model. These methods provide general mechanisms for optimizing language-model behavior using reward or preference signals. Our work builds on this paradigm to optimize natural-language explanations according to their ability to communicate behavioral differences between competing models.

Positioning of Our Work. Existing approaches address different components of model explanation and comparison. LIME and Anchors explain individual model predictions, while `MaNtLE` and `OPeX` optimize natural-language explanations of individual model behavior. `ModelDiff` and `SLED` instead focus on identifying samples that reveal behavioral differences between models, with `SLED` additionally using LLMs to describe these differences. Our work brings these directions together by directly optimizing natural-language explanations of behavioral differences between competing models, using reinforcement learning and a downstream diagnosticity objective. Unlike prior work, we further extend this formulation to multimodal model comparison.

A.3 Training and Evaluation Details

Training Data. We construct the training data across structured, text-based, and multimodal classification tasks. For structured classification, we train five classifier families: Logistic Regression, Decision Trees, Random Forests, Support Vector Classifiers (SVCs), and Multi-Layer Perceptrons (MLPs). These classifiers are trained on the following tasks for the UCI Machine Learning repository (Kelly et al., 2024): *Abalone, Census Income, Occupancy Detection, Mushroom, Thyroid,* and *Bike Sharing*. In addition to real structured data, SEAM and SEAM-VL also use synthetic samples generated by SLED (Malladi et al., 2025) for the same structured tasks.

For text classification, we train classifiers based on six pretrained language model families: `BERT-base-uncased`(Devlin et al., 2019), `DistilBERT`(Sanh et al., 2020), `RoBERTa` (Liu et al., 2019), `XLM-RoBERTa` (Conneau et al., 2020), `GPT-2` (Radford et al., 2019), and `GPT-Neo` (Black et al., 2021). These classifiers are trained on *Rotten Tomatoes Movie Reviews (Pang & Lee, 2005), GLUE-MNLI (Williams et al., 2018), GLUE-QQP (Wang et al., 2019), Hate Speech Detection (Davidson et al., 2017),* and *Sentiment Analysis* (Parvez, 2023).

For multimodal classification, we train classifiers based on `VisualBERT` (Li et al., 2019), `CLIP` (Radford et al., 2021), `FLAVA` (Singh et al., 2022), `BridgeTower` (Xu et al., 2024), and `BLIP` (Li et al., 2022) on *SNLI-VE (Xie et al., 2019), Flickr30k (Plummer et al., 2016), COCO (Lin et al., 2015), Fakeddit (Nakamura et al., 2020),* and *Obstacle Detection* (He & Saha, 2023).

Construction of Training Instances. For each task, we first train each classifier on a randomly selected subset of approximately 2,000 examples from the training data. We then use the trained classifiers to label the full dataset. For every pair of classifiers trained on the same task, we construct contrastive examples based on their predictions: an example x is *divergent* if $f_1(x) \neq f_2(x)$ and *convergent* if $f_1(x) = f_2(x)$. We denote the corresponding sets of examples by $\mathcal{X}_D$ and $\mathcal{X}_C$, respectively.

For each task $\mathcal{T}$ and classifier pair (f_1, f_2), we randomly sample a conditioning set $S = S_D \cup S_C$, containing 20 divergent samples $S_D \subseteq \mathcal{X}_D$ and 20 convergent samples $S_C \subseteq \mathcal{X}_C$. Each sample is linearized as described in § 2.3 and provided to the explanation policy together with its behavioral label. The full system prompt and generation instructions are given in § A.1. We additionally construct a disjoint set containing 20 convergent and 20 divergent examples, which is used to evaluate the generated explanation during training through the simulatability reward. The policy only observes the conditioning set and does not receive the labels of these held-out examples as input.

For each task, we randomly sample up to 1,000 training instances from the available classifier-pair and sample combinations. For smaller datasets, such as Abalone, where fewer valid instances are available, we use the available instances rather than oversampling the task. We then aggregate the resulting instances across tasks to construct the training set. This procedure provides variation across

classifier pairs while ensuring that each explanation is conditioned on examples from a single task and classifier pair.

SEAM is trained on 5,000 real structured instances, 5,000 synthetic structured instances, and 5,000 text-based instances. The synthetic structured instances are constructed using SLED-generated samples from the structured training tasks. SEAM-VL uses the same training data as SEAM and additionally includes 5,000 multimodal instances. SEAM-R uses the same training data as SEAM but excludes the 5,000 synthetic structured instances. Finally, SEAM-V is trained exclusively on the 5,000 multimodal instances.

Evaluation Data. For evaluation, we use tasks that are completely held out from training. Within each modality, we use the same classifier families as in training, but train new classifier instances on the held-out evaluation tasks. These classifier instances and their behavioral differences are therefore not observed during training.

For structured classification, we evaluate on the following tasks from the UCI Machine Learning repository (Kelly et al., 2024): *Tic-Tac-Toe Endgame, Balance Scale, Magic Gamma, Wine Quality, Car Evaluation, Adult, Bank Marketing, Banknote Authentication, Rice,* and *Nursery*. For text classification, we evaluate on *AG News (Zhang et al., 2015), CoLA (Warstadt et al., 2018), PAWS (Zhang et al., 2019; Yang et al., 2019), IMDb Reviews (Maas et al., 2011),* and *SST-2* (Socher et al., 2013). For multimodal classification, we evaluate on *MMSD 2.0 (Qin et al., 2023), Hateful Memes (Kiela et al., 2021), CrisisMMD (Alam et al., 2018),* and *GQA* (Hudson & Manning, 2019). For *GQA*, we restrict evaluation to the yes/no questions.

For each evaluation task and classifier pair, we construct a conditioning set containing 20 divergent and 20 convergent examples. The generated explanation is then evaluated on a separate, disjoint set of 30 divergent and 30 convergent examples from the same task and classifier pair. Recoverability is measured on the conditioning set, while simulatability is measured on the disjoint held-out set. This evaluation protocol tests whether the generated explanation captures the behavioral relationship between the classifiers and whether that relationship extends beyond the examples used to construct the explanation.

SEAM Training Details. We fine-tune SEAM using Group Relative Policy Optimization (GRPO) with parameter-efficient LoRA fine-tuning. The policy model is initialized from Qwen2.5-VL-32B-Instruct. The main training hyperparameters are:

- **Learning rate:** 5×10^{-5}
- **Optimizer:** 8-bit paged AdamW
- **Adam coefficients:** $\beta_1 = 0.9$, $\beta_2 = 0.99$
- **Weight decay:** 0.1
- **Learning-rate schedule:** cosine, with 10% warmup
- **GRPO KL coefficient:** 0.04
- **Number of epochs:** 1
- **Generations per prompt:** 8
- **Maximum new tokens:** 512
- **Maximum gradient norm:** 1

We apply LoRA with rank $r = 16$ and $\alpha = 16$ to the language model. The adapters are applied to the attention projections (q_{proj}, k_{proj}, v_{proj}, and o_{proj}) and the MLP projections ($\text{gate}_{\text{proj}}$, up_{proj}, and $\text{down}_{\text{proj}}$). The vision encoder is kept frozen, while the language-model layers are fine-tuned. We use gradient checkpointing through Unsloth to reduce memory requirements. Training is performed on H100 GPUs.

The reward is based on simulatability. For each generated explanation, a separate `Qwen3-VL-32B-Instruct` observer classifies the held-out examples as either divergent or convergent. The evaluator is run deterministically with temperature 0, and the resulting classification accuracy is used as the reward. The accuracy is linearly rescaled around chance performance, such

	GPT-4o	SEAM	OPeX	Qwen2.5-VL-32B	Qwen3-32B	Llama-3.3-70B
Tic-Tac-Toe Endgame	**0.69 ± 0.15**	0.68 ± 0.14	0.56 ± 0.09	0.52 ± 0.10	0.57 ± 0.11	0.60 ± 0.10
Balance Scale	0.65 ± 0.14	**0.71 ± 0.15**	0.53 ± 0.09	0.51 ± 0.08	0.51 ± 0.16	0.60 ± 0.13
Magic Gamma	0.69 ± 0.07	**0.86 ± 0.10**	0.54 ± 0.08	0.50 ± 0.03	0.54 ± 0.04	0.62 ± 0.09
Wine Quality	0.62 ± 0.10	**0.82 ± 0.12**	0.55 ± 0.09	0.49 ± 0.08	0.55 ± 0.10	0.60 ± 0.07
Car Evaluation	**0.94 ± 0.08**	0.86 ± 0.10	0.47 ± 0.10	0.61 ± 0.11	0.72 ± 0.12	0.65 ± 0.15
Adult	0.82 ± 0.16	**0.91 ± 0.09**	0.52 ± 0.09	0.63 ± 0.16	0.70 ± 0.15	0.69 ± 0.16
Bank Marketing	0.84 ± 0.12	**0.93 ± 0.10**	0.53 ± 0.07	0.52 ± 0.13	0.69 ± 0.13	0.74 ± 0.13
Banknote Authentication	0.77 ± 0.13	**0.82 ± 0.17**	0.49 ± 0.11	0.53 ± 0.12	0.66 ± 0.12	0.70 ± 0.16
Rice	0.65 ± 0.03	**0.91 ± 0.11**	0.58 ± 0.12	0.54 ± 0.08	0.55 ± 0.02	0.61 ± 0.05
Nursery	0.81 ± 0.09	**0.87 ± 0.10**	0.41 ± 0.11	0.55 ± 0.09	0.64 ± 0.11	0.57 ± 0.06
AG News	0.74 ± 0.10	**0.89 ± 0.07**	0.57 ± 0.12	0.55 ± 0.08	0.63 ± 0.09	0.65 ± 0.09
CoLA	0.75 ± 0.15	**0.94 ± 0.08**	0.64 ± 0.13	0.60 ± 0.12	0.64 ± 0.14	0.71 ± 0.16
IMDb Reviews	0.63 ± 0.13	**0.77 ± 0.11**	0.56 ± 0.08	0.58 ± 0.12	0.58 ± 0.09	0.70 ± 0.13
PAWS	0.59 ± 0.16	**0.76 ± 0.11**	0.52 ± 0.09	0.45 ± 0.12	0.50 ± 0.14	0.54 ± 0.14
SST-2	0.75 ± 0.15	**0.98 ± 0.04**	0.67 ± 0.13	0.59 ± 0.12	0.67 ± 0.12	0.74 ± 0.14
Average	0.73 ± 0.16	**0.85 ± 0.14**	0.54 ± 0.12	0.55 ± 0.12	0.61 ± 0.13	0.65 ± 0.14

Table 9: Recoverability scores for explanations of classifier behavioral differences across structured and text-based classification tasks. Best results per row are shown in bold. Scores are averaged over 80 runs and reported as mean ± standard deviation.

		Simulatability Scores			
Task	**Conditioning Set**	**GPT-4o**	**SEAM**	**SEAM-R**	**Qwen2.5-VL-32B-Instruct**
Bank Marketing	Real	0.81 ± 0.15	**0.87 ± 0.13**	0.77 ± 0.13	0.49 ± 0.12
	Synthetic (SLED)	0.70 ± 0.17	**0.83 ± 0.20**	0.68 ± 0.25	0.39 ± 0.22
Tic Tac Toe Endgame	Real	**0.73 ± 0.11**	0.68 ± 0.16	0.66 ± 0.18	0.50 ± 0.11
	Synthetic (SLED)	0.65 ± 0.07	**0.75 ± 0.13**	0.67 ± 0.09	0.55 ± 0.06
Magic Gamma	Real	0.64 ± 0.06	**0.81 ± 0.15**	0.70 ± 0.13	0.50 ± 0.03
	Synthetic (SLED)	0.64 ± 0.03	**0.82 ± 0.17**	0.72 ± 0.22	0.55 ± 0.02
Car Evaluation	Real	**0.90 ± 0.16**	0.79 ± 0.14	0.76 ± 0.13	0.54 ± 0.10
	Synthetic (SLED)	0.74 ± 0.14	0.72 ± 0.22	**0.74 ± 0.14**	0.55 ± 0.07
Nursery	Real	0.66 ± 0.08	**0.71 ± 0.14**	0.63 ± 0.09	0.49 ± 0.09
	Synthetic (SLED)	0.68 ± 0.16	**0.69 ± 0.19**	0.68 ± 0.09	0.59 ± 0.12
Banknote Authentication	Real	0.71 ± 0.12	**0.79 ± 0.16**	0.72 ± 0.15	0.53 ± 0.11
	Synthetic (SLED)	0.79 ± 0.14	**0.82 ± 0.15**	0.77 ± 0.13	0.63 ± 0.13
Rice	Real	0.65 ± 0.03	**0.88 ± 0.15**	0.77 ± 0.14	0.54 ± 0.10
	Synthetic (SLED)	0.64 ± 0.15	**0.79 ± 0.12**	0.68 ± 0.14	0.50 ± 0.14
Balance Scale	Real	0.64 ± 0.15	**0.70 ± 0.14**	0.64 ± 0.13	0.47 ± 0.07
	Synthetic (SLED)	**0.78 ± 0.17**	0.73 ± 0.15	0.61 ± 0.23	0.58 ± 0.14
Wine Quality	Real	0.63 ± 0.12	**0.77 ± 0.14**	0.72 ± 0.15	0.50 ± 0.06
	Synthetic (SLED)	0.42 ± 0.03	**0.94 ± 0.04**	0.70 ± 0.19	0.61 ± 0.14
Average	Real	0.69 ± 0.15	**0.75 ± 0.16**	0.70 ± 0.15	0.51 ± 0.10
	Synthetic (SLED)	0.70 ± 0.15	**0.77 ± 0.17**	0.69 ± 0.18	0.55 ± 0.14

Table 10: Simulatability of explanations conditioned on real or SLED-generated synthetic samples and evaluated on held-out real-world samples. SEAM achieves comparable simulatability when conditioned on real and synthetic samples, providing evidence that it can extract behavioral patterns from synthetic samples that remain predictive on the real data distribution. The improvement of SEAM over SEAM-R under synthetic conditioning further demonstrates the benefit of synthetic-data exposure during RLFT. Scores are reported as mean ± standard deviation, averaged across 80 runs.

that the reward is $2(a - 0.5)$, where a is the evaluator accuracy. Thus, chance-level performance receives a reward of 0, while higher-quality explanations receive positive rewards. The policy is optimized over eight sampled generations for each prompt. Across the entire training process for all variants of SEAM , `Qwen3-VL-32B-Instruct` is used consistently as the observer.

A.4 Additional Analyses and Details

User Study Details We conduct a controlled user study to evaluate whether explanations generated by SEAM can be applied by human users to identify convergence and divergence on previously unseen inputs. The study compares SEAM against GPT-4o while keeping the inducing samples and downstream evaluation samples identical across conditions. For each evaluation instance, we sample a set of inducing samples from a fixed task–classifier pair and provide this exact same set to both SEAM and GPT-4o to generate their respective explanations. We then independently sample

	Recoverability	Simulatability
Decision Tree	0.77	0.57
Random Forests	0.96	0.55
SEAM	0.84	0.78

Table 11: Recoverability and Simulatability scores for decision trees, random forest, and SEAM explanations averaged across all structured tasks.

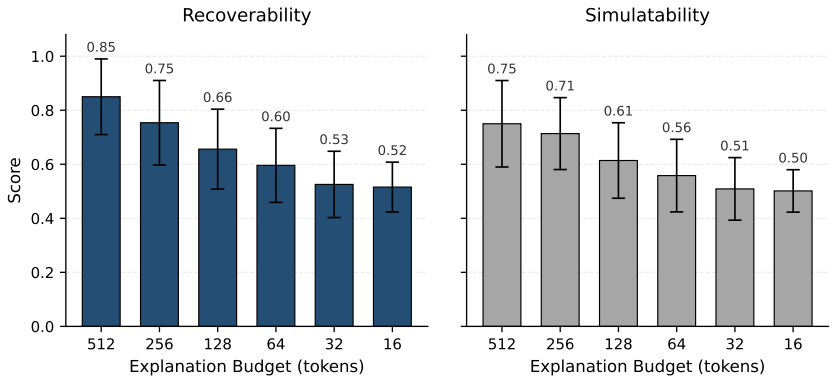


Figure 2: Effect of the explanation token budget on SEAM . We train separate SEAM variants with different explanation budgets and evaluate each variant on recoverability and simulatability. Reducing the budget from 512 to 256 tokens results in only a modest decrease in both metrics, while further reductions progressively degrade performance. This suggests that explanations require a minimum length to effectively convey behavioral differences between classifiers.

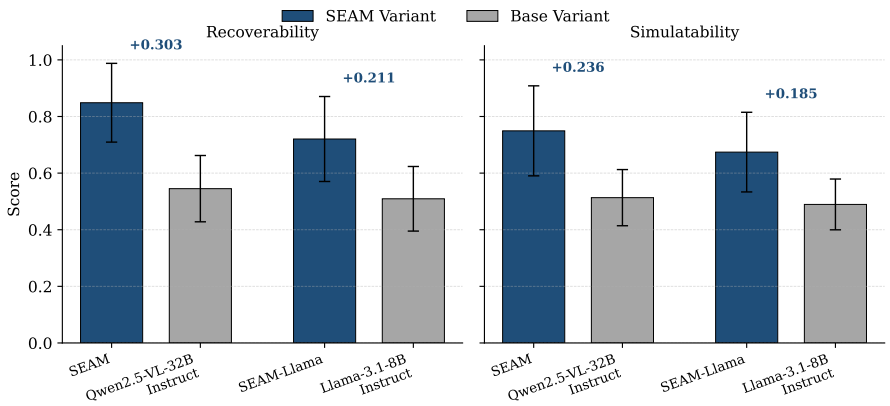
a disjoint set of 30 previously unseen samples from the same task–classifier pair. Both participant groups are evaluated on this exact same set of unseen samples and classify each sample as either *convergent* or *divergent* based solely on the explanation provided to them. Participants do not have access to the predictions of either classifier. Thus, the explanation generation model is the only difference between the two experimental conditions.

We recruit 60 participants through Prolific and randomly assign them to two equally sized groups of 30 participants. Participants are compensated at a rate of $12 per hour. Participants are required to be at least undergraduate-level, fluent in English, and have a 90% approval rate on Prolific. One group receives explanations generated by GPT-4o, while the other receives explanations generated by SEAM . Both groups receive explanations generated from the same inducing samples and perform the downstream classification task on the same previously unseen samples, ensuring that the explanation generation model is the only experimental difference.

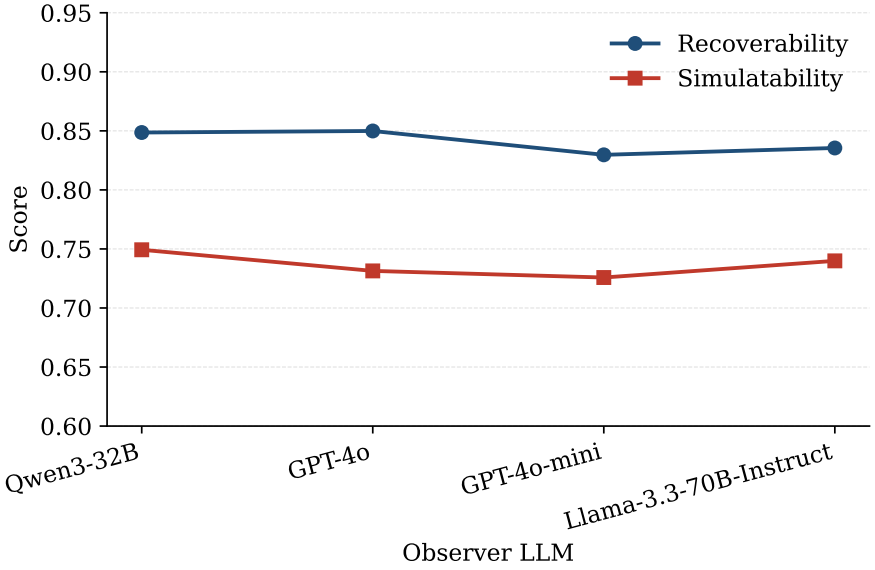
Participants using SEAM explanations achieve an average accuracy of 74% on the unseen samples, compared to 55.4% for participants using GPT-4o explanations. This difference is statistically significant (two-way t-test, $p < 0.01$). Participants in the GPT-4o condition also provided qualitative feedback indicating that the "explanations could be far clearer" and that "it was confusing at first, I had to read the explanations like 3 times." These responses provide additional qualitative evidence that differences in how the behavioral distinctions are communicated can affect their usefulness to downstream users.

Details on Qualitative Analysis The rule-count statistics in Table **??** are computed automatically using deterministic parsing rather than manual annotation or an LLM judge. We first segment each generated explanation into sentences using spaCy's `en_core_web_sm` model. A sentence is treated as a rule sentence if its text contains the substring `diverg` or `converg` (case-insensitive), capturing references such as "diverges," "divergence," "converges," and "convergence." Sentences that contain neither keyword are excluded from the rule-count analysis.

For each rule sentence, we use spaCy's dependency parse to identify coordinated conditions. Specifically, we count tokens whose dependency relation is `conj` and whose part-of-speech tag is one of



(a) recoverability and simulatability scores for explanations generated by SEAM and SEAM-LLAMA , compared against their respective base models, Qwen2.5-VL-32B-Instruct and Llama-3.1-8B-Instruct. The figure reports performance on both the samples used to elicit the explanations (recoverability) and previously unseen samples (simulatability), with error bars denoting one standard deviation. RL fine-tuning substantially improves both metrics for both policy model families, demonstrating that our training procedure consistently improves contrastive diagnosticity and that these benefits extend to substantially smaller policy models.



(b) recoverability and simulatability scores across different observer LLMs. Both metrics remain relatively stable across observer models, indicating that the evaluation is not overly sensitive to the choice of observer.

Figure 3: Impact of policy scale and Observer LLM on contrastive diagnosticity of explanations

VERB, AUX, or ADJ. The number of conditions in a rule is then defined as the number of detected conjunctions plus one for the initial condition. Thus, a rule containing three coordinated conditions, such as "diverges when A and B and C," contributes three atomic rules.

From these parsed rules, we compute the total number of rules, the number containing conjunctions, the maximum number of chained conditions in a single rule, the total number of atomic rules, and separate counts of divergence and convergence rules. A sentence containing both "diverg" and "converg" is counted in both respective categories. All statistics are produced by the same deterministic procedure for every explanation, with no human annotation or LLM-based judgment. The reported values are averages across explanations.

Limitations Our reinforcement learning objective optimizes solely for downstream diagnostic performance, but other properties of explanations, such as simplicity, readability, conciseness, and hu-

Task	SEAM	GPT-4o
Rice Classification	The models diverge when `Major_Axis_Length` is between approximately 185 and 195, `Minor_Axis_Length` is between 85 and 95, `Eccentricity` is between 0.87 and 0.91, and `Extent` is between 0.55 and 0.75. The models converge when these features fall outside these ranges, with `Major_Axis_Length` below 170 or above 200, `Minor_Axis_Length` below 75 or above 95, `Eccentricity` below 0.85 or above 0.91, or `Extent` below 0.55 or above 0.75.	The models diverge for samples with higher eccentricity and a wider range of extent values, indicating more elongated shapes with varying compactness. Divergent samples also tend to have a more balanced ratio between major and minor axis lengths. The models converge for lower eccentricity and more consistent extent values, with a more pronounced difference between major and minor axis lengths.
Bank Marketing	The models diverge when `housing` is `no`, regardless of other features such as age, job, marital status, education, balance, or contact. The models converge when `housing` is `yes`, irrespective of variations in other features.	The models diverge when samples combine high campaign numbers with no housing loan, suggesting a financial profile associated with inconsistent predictions. The models converge for lower campaign numbers, often with housing loans present, corresponding to a more straightforward financial profile.

Table 12: Qualitative comparison of SEAM and GPT-4o explanations generated from the same set of samples. SEAM provides explicit feature-level rules and thresholds for distinguishing model divergence and convergence, while GPT-4o describes broader qualitative patterns.

Metric	Llama-3.3-70B	Qwen2.5-VL-32B	DeepSeek-R1-70B	SEAM	GPT-4o
Number of Rules	2.00	2.84	2.00	**4.49**	2.13
Conjunctive Rules	1.67	2.06	1.24	**3.03**	1.41
Maximum Conjuncts per Rule	3.43	3.23	2.30	3.47	2.79
Atomic Conditions	5.52	6.90	3.87	**9.91**	4.75
Divergence Rules	1.02	1.44	1.02	**2.42**	1.08
Convergence Rules	1.00	1.42	1.01	**2.41**	1.06
Divergence-to-Convergence Rule Ratio	1.02	1.03	1.01	1.03	1.02

Table 13: Qualitative analysis of generated explanations. Values are averages per explanation.

man preference, may also be important. Incorporating these properties as additional reward signals could provide a more comprehensive objective for explanation quality. Furthermore, our evaluation is limited to classification tasks and does not consider generation-based tasks, where defining behavioral agreement between models may require different formulations. Finally, our binary convergence/divergence formulation provides only a coarse characterization of model differences: it does not indicate which model is correct, the severity of a disagreement, or the underlying reasoning of either model