From Numbers to Narratives: Goal-Oriented Summarization of Machine Learning Model Differences

Nam Hyeon-Woo [1] **Tae-Hyun Oh** [2] **Zeynep Akata** [3 4 5] **Stephan Alaniz** [6]

Abstract

Non-experts can now obtain natural-language explanations of how two ML models differ by feeding numerical results to an LLM-based agent. However, naively prompting an LLM often omits critical conditions, and non-experts often cannot easily detect these omissions, which can mislead downstream conclusions. We formulate this as goal-oriented summarization and propose CONDENSER, an iterative method that optimizes an explanation against two objectives: Completeness (faithfulness to observed differences) and Density (informativeness per unit length). CONDENSER+ extends CONDENSER with an LLM-based Explorer that actively selects conditions to evaluate. On four settings of increasing complexity (Colored MNIST, Fitzpatrick17k, Dollar Street, and open-condition gender classification), our methods produce concise and complete explanations. They also support downstream LLM tasks, prompt refinement on CelebA and subset benchmarking on Flickr30K, where measurable improvements further validate the effectiveness of our method. Goal-oriented summarization thus yields explanations that are concise, complete, and useful for downstream LLM tasks.

1. Introduction

Recent advances in large language models (LLMs) and agentic AI systems (Yao et al., 2022; Jimenez et al., 2024) such as Claude Code and OpenAI Codex now let non-experts obtain natural-language explanations of how two ML models differ by feeding numerical results to an LLM. Product managers and domain specialists increasingly rely on such explanations to interpret model behavior and inform decisions, even without deep ML expertise. For these explanations to be useful, they must be *faithful*: models often behave differently across input *conditions* such as attributes, demographic subgroups, or environmental factors (Geirhos et al., 2019; 2020; Sagawa et al., 2020; Hendrycks & Dietterich, 2019), so an explanation that omits or distorts condition-level differences can mislead. Yet a non-expert reader has little means to detect what the LLM left out. We illustrate the task and the trade-offs across explainers in Fig. 1.

Producing such an explanation is hard: an LLM must identify which differences matter and describe them faithfully, often in a few sentences. The most common approach is one-shot prompting, where an LLM is asked to *"Summarize numerical results between models,"* and the output is taken as the final explanation. While convenient, this approach has two limitations. First, such summaries often omit important condition-level differences, resulting in incomplete explanations. Second, the one-shot framing assumes that the relevant conditions have already been identified and evaluated, leaving no way to discover them in the first place. To address both limitations, we reformulate the task as an explicit optimization problem we call *goal-oriented summarization*. The explanation itself becomes the target, optimized against two objectives, COMPLETENESS (faithfulness to observed condition-level differences) and DENSITY (informativeness per unit length).

We optimize this with an iterative procedure in which an LLM, the *Refiner*, rewrites the current explanation at each step based on score feedback over a subset of conditions. The basic version, CONDENSER, draws this subset uniformly at random. Its extension CONDENSER+ replaces random sampling with an LLM-based *Explorer* that actively selects which conditions to evaluate. In the open-condition setting, where no predefined condition set is available, the Explorer also synthesizes new evaluation samples. See Fig. 2 for an overview of the procedure.

We evaluate both methods on four settings of increasing complexity. Colored MNIST (Arjovsky et al., 2019) serves as a controlled benchmark with ground-truth conditions. Fitzpatrick17k (Groh et al., 2021; 2022) and Dollar

[1]Electrical Engineering, POSTECH, South Korea [2]School of Computing, KAIST, South Korea [3]Technical University of Munich, Germany [4]Helmholtz Munich, Germany [5]Munich Center for Machine Learning, Germany [6]LTCI, Télécom Paris, Institut Polytechnique de Paris, France. Correspondence to: Stephan Alaniz <stephan.alaniz@telecom-paris.fr>.

Accepted to the 2nd Workshop on Compositional Learning at ICML 2026, Seoul, South Korea. *Copyright 2026 by the author(s).*

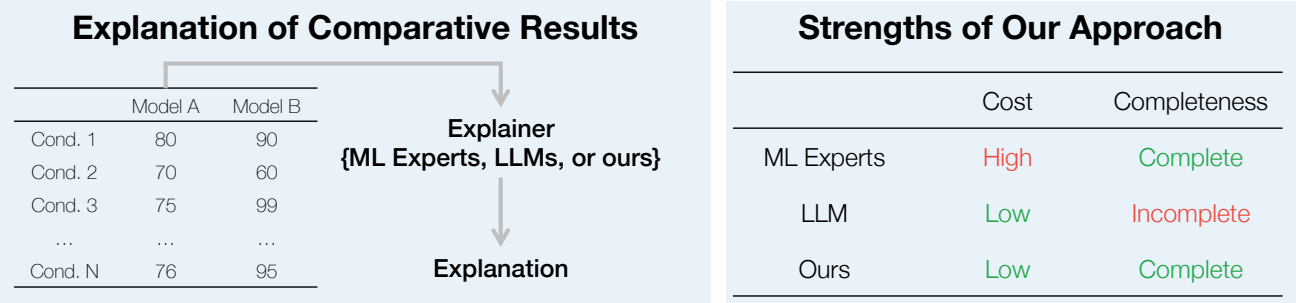


Figure 1. **Task setup and approach landscape.** Given numerical comparative results between two ML models, an Explainer (ML experts, LLMs, or our method) produces a natural-language explanation of the differences (left). Across explainers, ML experts can produce complete explanations but at high cost (human-hours), naive LLMs are low-cost (a single API call) but often incomplete, and our method aims to achieve both at moderate cost (right).

Street (Gaviria Rojas et al., 2022) cover real-world settings in dermatology and geographic diversity. The fourth, an open-condition gender classification setting, requires the method to operate without any predefined condition vocabulary. Beyond evaluating the explanations themselves, we further show that they enable downstream LLM tasks such as prompt refinement on CelebA and subset-level benchmarking on Flickr30K. Our contributions are:

1. Goal-oriented summarization for explaining how two ML models differ, formulated as an optimization problem with two objectives (COMPLETENESS and DENSITY);

2. CONDENSER and CONDENSER+, iterative methods for solving this optimization problem, with CONDENSER+ adding an LLM-based Explorer for active condition selection;

3. Experiments across four settings of increasing complexity, from Colored MNIST to open-condition gender classification, demonstrating that our methods produce concise and complete explanations;

4. Two downstream applications of the generated explanations, prompt refinement on CelebA and subset-level benchmarking on Flickr30K.

2. Related Work

Comparative Analysis. Several comparative analysis methods have been proposed. Jhamtani & Berg-Kirkpatrick (2018) describe differences between image pairs, while Dunlap et al. (2024) focus on image sets. Chiquier et al. (2025) generate images with subtle differences while preserving identity. VibeCheck (Dunlap et al., 2025) evaluates qualitative differences between LLM outputs. Whereas these works compare visual or textual artifacts (image pairs, image sets, or LLM generations), our method takes numerical performance across conditions as input. It jointly identifies the conditions under which two models diverge and expresses those differences concisely, with explicit objectives for measuring explanation quality. Structured slice discovery methods such as Domino (Eyuboglu et al., 2022) automatically identify data slices on which a single model underperforms. Our open-condition Explorer is related but compares two models, searching for conditions under which they diverge rather than slices where one model fails.

Textual Optimization. Retraining LLMs is computationally expensive; therefore, many approaches instead optimize the input text. Methods such as TextGrad (Yuksekgonul et al., 2025), DSPy (Khattab et al., 2024), and Verbalized Machine Learning (Xiao et al., 2025b) iteratively optimize various texts such as prompts or captions. Three aspects distinguish our method from prior textual optimization. (i) The optimization target is a structured comparison artifact (a model-difference explanation), rather than an arbitrary prompt or caption. (ii) Our objectives, COMPLETENESS and DENSITY, are grounded in the input data (per-condition performance), rather than a free-form quality criterion. (iii) The Explorer extends the textual optimization loop to *what to evaluate*, not just *how to phrase*, including condition synthesis in the open-condition setting.

Reward and Preference Modeling. Optimizing over performance differences is central to reward and preference modeling (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022; Rafailov et al., 2023). In these approaches, a scalar reward bridges the gap between preferred and less preferred outputs. In contrast, our method is distinguished by both its target and its objective. We optimize a natural language explanation for performance differences across diverse conditions. Evaluation paradigms (Zheng et al., 2023; Chiang et al., 2024) also automate the mapping of model discrepancies. However, they only provide numerical comparison results, whereas our goal is to produce faithful textual explanations that non-experts can read.

LLM Evaluator. LLMs are increasingly employed as evaluators to reduce time, labor, and cost (Verga et al., 2024;

Kim et al., 2024b; Hackl et al., 2023; He et al., 2024; Liu et al., 2023). He et al. (2024) use LLMs as annotators, while Liu et al. (2023) propose an LLM-based evaluation framework for natural language generation. Bills et al. (2023) simulate neural activations using LLMs. We similarly leverage LLMs as evaluators, but introduce a novel role. Rather than scoring generations along predefined quality axes, our LLM judge answers per-condition questions conditioned on the explanation, enabling us to quantify how faithfully the explanation captures the underlying comparative results.

Synthetic Data. With advances in diffusion models, synthetic images now achieve high quality across diverse applications (Kim et al., 2024a; Ye-Bin et al., 2024; Augustin et al., 2022; Jeanneret et al., 2022). In our open-condition setting, no predefined condition set exists; the Explorer proposes conditions in natural language and a diffusion model (Esser et al., 2024) synthesizes evaluation samples on the fly, enabling model comparison without any curated dataset.

3. Method

Task Formulation. Let $\mathcal{C}$ denote a set of conditions (e.g., input attributes or environmental factors), each expressed in natural language, and let a_c^A and a_c^B be the performance of models A and B on condition $c \in \mathcal{C}$ (e.g., accuracy for data with c). We denote by $\mathbf{a}_{\mathcal{C}}^A$ and $\mathbf{a}_{\mathcal{C}}^B$ the corresponding performance vectors over $\mathcal{C}$. The one-shot prompting approach treats the task as

$$e = \text{summarization}(\mathcal{C}, \mathbf{a}_{\mathcal{C}}^A, \mathbf{a}_{\mathcal{C}}^B), \tag{1}$$

where an LLM is prompted to compress the conditions $\mathcal{C}$ and the corresponding numerical results into a single explanation e. A key limitation of this formulation is that it places no constraint on the quality of e. The resulting explanation comes with no guarantee and may omit important condition-level differences. We reformulate Eq. (1) as *goal-oriented summarization*, which makes the quality of e an explicit optimization target:

$$\begin{aligned} \underset{e}{\text{maximize}} \quad & \textsc{score}(e) \\ \text{s.t.} \quad & e = \text{summarization}(\mathcal{C}, \mathbf{a}_{\mathcal{C}}^A, \mathbf{a}_{\mathcal{C}}^B). \end{aligned} \tag{2}$$

In addition, $\mathcal{C}$ itself need not be predefined. In many practical settings, the relevant conditions are not known in advance. We call this the *open-condition* setting (Sec. 4.3.3), where conditions are proposed on the fly rather than drawn from a fixed set. The remainder of this section first describes the iterative methods (Sec. 3.1). Specifically, CONDENSER operates on a given $\mathcal{C}$, while CONDENSER+ additionally builds up $\mathcal{C}$ on the fly. We then present the SCORE function that these methods optimize, composed of two objectives, COMPLETENESS and DENSITY (Sec. 3.2).

3.1. Method: CONDENSER and CONDENSER+

We describe the iterative optimization procedure that implements goal-oriented summarization (Fig. 2). At each iteration, *Refiner* updates the current explanation based on score feedback over a subset of conditions. In CONDENSER+, an additional *Explorer* module selects which conditions to evaluate next, rather than relying on uniform random sampling. We refer to the variant without the Explorer as CONDENSER, and the variant with the Explorer as CONDENSER+.

Refiner. Each iteration operates on a subset $\mathcal{S} \subseteq \mathcal{C}$ of K conditions. The Refiner takes the current explanation, its COMPLETENESS and DENSITY scores (defined in Sec. 3.2), $\mathcal{S}$, and the per-condition performance values $\mathbf{a}_{\mathcal{S}}^A, \mathbf{a}_{\mathcal{S}}^B$, and generates 3 candidate explanations (Algorithm 1). We then update the top-1 to the candidate with the highest COMPLETENESS score. The DENSITY score, included as input, tells the Refiner how information-dense the current explanation is, encouraging it to generate denser candidates. The Refiner prompt also instructs the LLM to keep candidates concise, *e.g., "Keep each candidate as a single paragraph, concise enough ..."*

LLM-based Explorer (in CONDENSER+). Rather than sampling conditions uniformly at random, the LLM-based Explorer actively decides which conditions to evaluate next based on the current explanation and the history of previously explored conditions. The Explorer operates in two modes. In the *closed-condition* setting, it selects conditions directly from the predefined condition set $\mathcal{C}$. In the *open-condition* setting (Sec. 4.3.3), no predefined $\mathcal{C}$ is available; the Explorer instead proposes new conditions in natural language and instructs a conditional generator (e.g., Stable Diffusion) to synthesize evaluation samples, without committing to a fixed set.

Variants. CONDENSER and CONDENSER+ follow the same iterative procedure (Algorithm 1), differing only in how the subset $\mathcal{S}$ is selected: uniformly at random from $\mathcal{C}$ for CONDENSER, and via the Explorer for CONDENSER+.

3.2. Score: Completeness and Density

A good explanation of comparative results should satisfy two properties. First, it must be *complete*. The explanation should faithfully convey the condition-level performance differences $\{a_c^A - a_c^B\}_{c \in \mathcal{C}}$ without omitting critical information. Second, it must be *dense*. Every part of the explanation should carry meaningful information. We quantify these two properties as COMPLETENESS and DENSITY, respectively.

Completeness. If the explanation fails to reflect the condition-level differences, the reader may misunderstand which conditions favor which model. We quantify this as

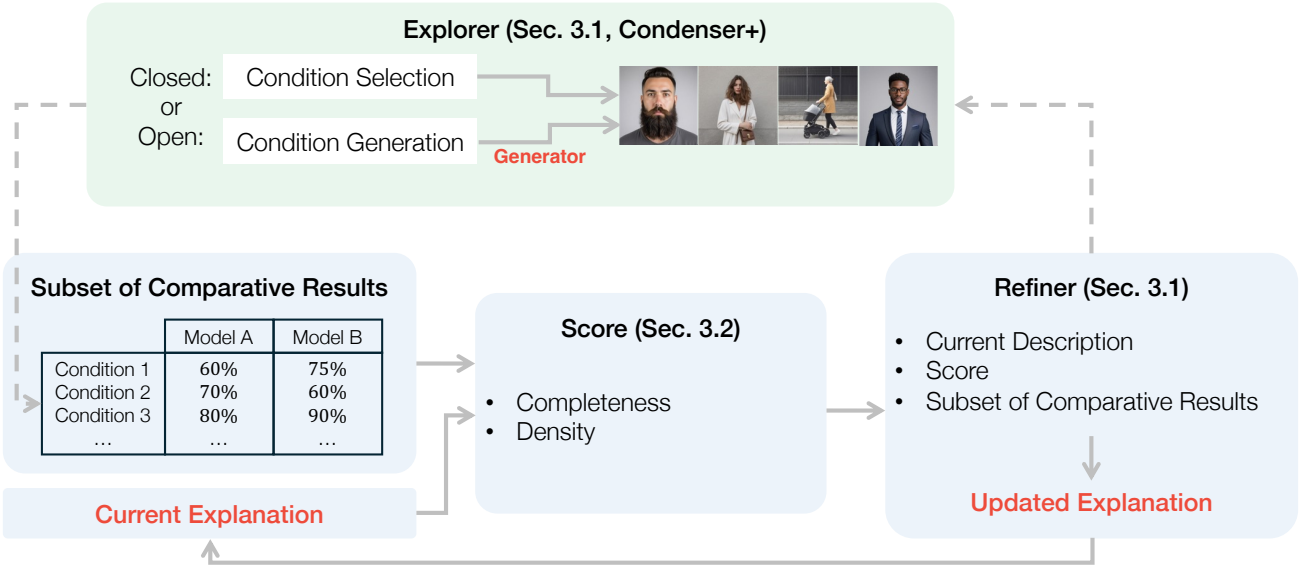


Figure 2. **CONDENSER and CONDENSER+.** Given a subset of comparative results and the current explanation, the Score module computes Completeness and Density. The Refiner takes the explanation, its scores, and the subset to produce an updated explanation. In CONDENSER+, the Explorer actively selects conditions from an existing set (closed setting) or generates new evaluation samples via a conditional generator (open setting).

the Pearson correlation, over all conditions $c \in \mathcal{C}$, between (i) an LLM's prediction of which model performs better under c given only the explanation, and (ii) the actual performance difference on c. Specifically, as illustrated in the upper panel of Fig. 3, given an explanation e, we prompt the LLM to predict which model performs better under each condition c and denote this prediction as $l_c \in \{-1, 0, +1\}$. COMPLETENESS is then defined as:

$$\text{COMPLETENESS}(e) = \text{CORRELATION}(\{l_c\}_{c\in\mathcal{C}}, \{a_c^A - a_c^B\}_{c\in\mathcal{C}}) \quad (3)$$

Intuitively, COMPLETENESS measures whether the explanation alone is sufficient to recover the condition-level differences. If e faithfully captures the condition-level differences, the LLM can correctly infer which model is better at each c, yielding a high COMPLETENESS. For example, in Fig. 3, the explanation achieves COMPLETENESS $= 0.64$, reflecting alignment between the LLM predictions and the actual condition-level differences.

Density. We introduce DENSITY that encourages each part of the explanation to be informative. As shown in Fig. 3, we instruct an LLM to produce a shortened version e' of the explanation, reducing it to approximately half its length while preserving its meaning, and measure the change in COMPLETENESS between the original and the shortened version:

$$\text{DENSITY}(e) = \text{COMPLETENESS}(e) - \text{COMPLETENESS}(e') \quad (4)$$

A positive DENSITY indicates that shortening reduces completeness, meaning e is *information-dense*. A negative DENSITY indicates that the shortened version is more complete, suggesting that the original contained misleading content that the shortening removed. A higher DENSITY thus indicates that more of the explanation contributes to completeness. DENSITY is intentionally defined as a *relative* quantity. By measuring how COMPLETENESS changes after shortening, it captures whether each part of the explanation contributes information, independent of the absolute COMPLETENESS scale of e.

4. Experiment

Metric and LLM Backbones. We report two metrics. *Classification accuracy* measures whether an LLM correctly predicts which model performs better under each condition given the generated explanation. *Word count* measures the length of the explanation. For generation, we use Gemini-2.5-Flash, GPT-5-mini, and Claude-3.5-Haiku. For evaluation, we use a different set of models: Gemini-3-Flash, GPT-5.2, and Claude-Haiku-4.5. We report the average accuracy across all three. This separation ensures the explanation is not biased toward generation models. In Sec. 4.2, we validate the metric by measuring inter-judge agreement and self-preference bias.

Experiment Overview. In Sec. 4.1, we conduct controlled experiments on the Colored MNIST dataset. In Sec. 4.3, we extend to real-world settings: Fitzpatrick17k, Dollar Street, and open-condition gender classification. We further demonstrate two downstream applications of the generated explanations: prompt refinement for zero-shot classification and subset-level benchmarking in Sec. 4.4.

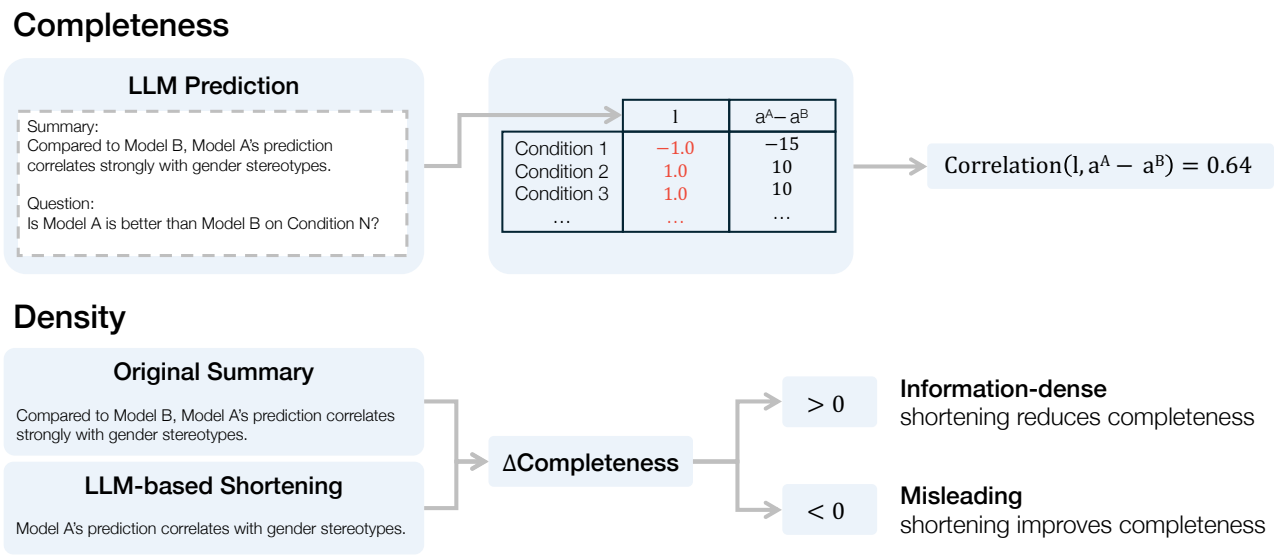


Figure 3. **Completeness and Density objectives.** Completeness quantifies how well an LLM can recover model performance differences from a summary. Density measures the change in Completeness after LLM-based shortening, rewarding explanations that are information-dense.

4.1. Controlled Experiment on Colored MNIST

4.1.1. Setting

We conduct controlled experiments on the Colored MNIST dataset (Bahng et al., 2020; Arjovsky et al., 2019). Model A is trained on data where digits 0–4 are biased toward red while digits 5–9 remain unbiased; Model B is trained with the opposite structure, where digits 5–9 are biased toward blue while digits 0–4 remain unbiased. By construction, Model A struggles on digits 0–4 under non-red colors and performs well on 5–9, while Model B shows the inverse pattern. This asymmetric design provides a ground truth along the digit axis against which we can precisely evaluate whether the generated explanations capture the underlying model discrepancies.

We report three reference points. The SUMMARY and HUMAN baselines are natural-language explanations; the NUMERICAL baseline feeds the raw table directly to the evaluator and serves as an upper bound. The SUMMARY baseline applies one-shot prompting as in Eq. (1). The full table of per-condition performance differences is provided to the LLM, and the output is taken as the final explanation. The NUMERICAL baseline provides the evaluator LLM with the full table of per-condition performance differences between Model A and Model B. Since this directly exposes the condition-level differences that the evaluation questions probe, it serves as an upper bound under our protocol. The HUMAN baseline is a natural-language explanation written by an author with full knowledge of the training setup described above. It represents what an expert who knows the experimental design can produce, and serves as a realistic target for automated methods.

Table 1. **Ablation on Colored MNIST.** ✓ indicates the component is enabled.

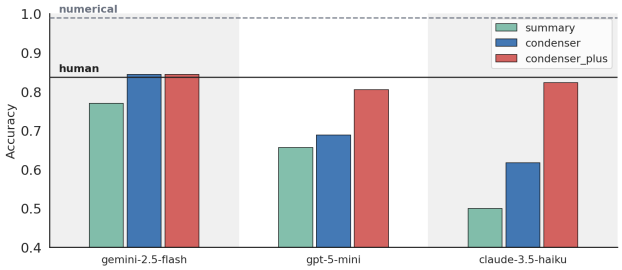
Concise	Density	Word Count	Accuracy
✗	✗	577	0.894
✗	✓	137	0.773
✓	✗	81	0.816
✓	✓	118	0.844

4.1.2. Results

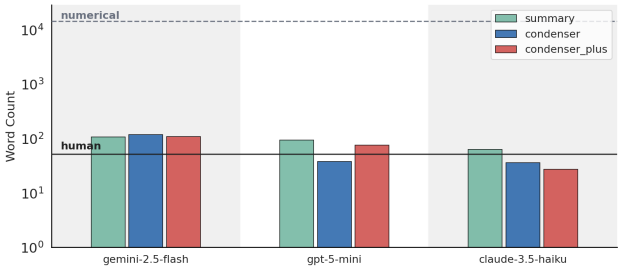
Accurate Goal-Oriented Summarization. As shown in Fig. 4a, CONDENSER and CONDENSER+ outperform the naive SUMMARY on all evaluated LLM backbones. This indicates that simply prompting an LLM to summarize is insufficient. Optimization produces explanations that more faithfully reflect the underlying performance differences.

More Complete Without Increasing Length. As shown in Fig. 4b, CONDENSER and CONDENSER+ maintain or reduce word count relative to the baseline. Together with the accuracy gain, this shows that goal-oriented optimization produces explanations that are more compact.

Comparable to the Human Reference. CONDENSER and CONDENSER+ approach the accuracy of the human reference, suggesting that goal-oriented optimization can recover much of the condition-level structure that a domain-aware author identifies, without hand-authored explanations.



(a) Accuracy (Higher is better.)



(b) Word count (Lower is better.)

Figure 4. **Colored MNIST results.** Accuracy and word count across LLM backbones and methods on Colored MNIST. Dashed line (NUMERICAL) denotes the upper bound where the evaluator receives the full per-condition performance table; solid line (HUMAN) denotes an explanation written by an author with full knowledge of the training setup.

Ablation. We ablate two components of the Refiner: the conciseness instruction in the prompt and the Density signal passed as additional guidance. Table 1 reports word count and accuracy on Colored MNIST under each combination. With neither component, explanations grow long and trivially achieve high accuracy by listing all per-condition results (577 words). The conciseness prompt makes the explanation short, and Density then restores the information that conciseness alone removes.

Qualitative Result. Fig. 5 contrasts the naive SUMMARY baseline with CONDENSER+ on the same comparative results. The baseline conflates digit-level and color-level effects, while CONDENSER+ isolates the digit axis as the true source of model differences, matching the asymmetric training design.

4.2. Judge Reliability Analysis

Inter-Judge Agreement. Table 2 reports Kendall's W and pairwise agreement across the three judges on Colored MNIST. Agreement is consistently high on informative explanations: the numerical upper bound ($W = 0.987$), the human reference ($W = 0.983$), and explanations produced by CONDENSER/CONDENSER+ (typically $W > 0.85$). It drops sharply on the naive SUMMARY baseline for some backbones (e.g., $W = 0.337$ and $W = 0.567$). We interpret this pattern as supportive of the metric rather than a reliability concern. When an explanation is ambiguous or incomplete, reasonable judges genuinely disagree on which model is better under each condition. When an explanation faithfully captures the differences, judges converge. Because the same three judges achieve $W > 0.98$ on the numerical and human references, we attribute the drop on the SUMMARY baseline to ambiguity in the explanation, not to judge-level noise.

SUMMARY

Model B demonstrates superior performance across most color and angle variations, consistently achieving higher accuracy rates compared to Model A. While Model A shows strong performance for specific digit classes and color conditions, Model B exhibits more robust and stable classification capabilities, particularly for digits 0–4. The performance gap is most pronounced in green, yellow, magenta, and cyan color ranges.

CONDENSER+

The analysis shows Model A excels with mid to high-range digits, while Model B demonstrates superior performance, especially in recognizing low-numbered and transitional digits across different contexts.

Figure 5. **Qualitative comparison on Colored MNIST.** CONDENSER+ correctly identifies the asymmetric split that reflects the training design, while the SUMMARY baseline conflates digit and color effects.

Table 2. **Inter-judge agreement on Colored MNIST.** W is Kendall's coefficient of concordance; G/P/C are pairwise agreements between Gemini-3 (G), GPT-5.2 (P), and Haiku-4.5 (C). Agreement drops on the naive SUMMARY baseline.

Method	Generator	W	Pairwise agreement		
			G↔P	G↔C	P↔C
Numerical	—	0.987	0.990	0.970	0.980
Human	—	0.983	0.990	0.950	0.950
SUMMARY	gemini-2.5-flash	0.966	0.940	0.890	0.910
	gpt-5-mini	0.567	0.620	0.710	0.570
	claude-3.5-haiku	0.337	0.810	0.620	0.690
CONDENSER	gemini-2.5-flash	0.894	0.960	0.890	0.870
	gpt-5-mini	0.710	0.700	0.760	0.640
	claude-3.5-haiku	1.000	1.000	1.000	1.000
CONDENSER+	gemini-2.5-flash	0.958	0.980	0.950	0.970
	gpt-5-mini	0.810	0.800	0.810	0.820
	claude-3.5-haiku	0.847	0.880	0.890	0.870

Self-Preference Bias. We next check whether judges favor explanations produced by models in their own family. For each (generator, judge) pair, we compute the accuracy gap $\Delta = \text{acc}_{\text{same-family}} - \text{acc}_{\text{cross-family}}$, where same-family accuracy uses the judge in the same model family as the generator and cross-family accuracy averages over the remaining two judges. Table 3 reports Δ across generators and methods. Gemini and Claude families exhibit negligible self-preference, while the GPT family shows a substantial

Table 3. **Self-preference analysis on Colored MNIST.** $\Delta = \text{acc}_{\text{same}} - \text{acc}_{\text{cross}}$. Positive Δ indicates that the same-family judge scores the generator's explanations higher than cross-family judges. Red marks positive Δ (greater self-preference); green marks negative.

Method	Generator	Accuracy		Δ
		Same	Cross	
SUMMARY	gemini-2.5-flash	0.734	0.787	−0.053
	gpt-5-mini	0.883	0.543	+0.340
	claude-3.5-haiku	0.532	0.484	+0.048
CONDENSER	gemini-2.5-flash	0.883	0.824	+0.059
	gpt-5-mini	0.883	0.590	+0.293
	claude-3.5-haiku	0.617	0.617	+0.000
CONDENSER+	gemini-2.5-flash	0.851	0.840	+0.011
	gpt-5-mini	0.851	0.782	+0.069
	claude-3.5-haiku	0.830	0.819	+0.011

positive bias for GPT-5.2 as a judge of GPT-5-mini generations. Interestingly, this gap is smaller under CONDENSER+, possibly because diverse condition selection makes the explanation less tied to any single generator's writing style. We mitigate the bias by averaging accuracy across three judges from different families, so that a single self-preferring judge cannot dominate the reported numbers.

4.3. Effectiveness Across Real-World Settings

Having validated our method in the controlled Colored MNIST setting, we now examine whether goal-oriented summarization generalizes to realistic model comparisons. We apply CONDENSER+ to three increasingly realistic settings. Quantitative results across LLM backbones for Fitzpatrick17k and Dollar Street are reported in Fig. 11.

4.3.1. FITZPATRICK17K

We evaluate on Fitzpatrick17k (Groh et al., 2021; 2022), a dermatology dataset of skin disease images annotated with both disease category and Fitzpatrick skin tone. We compare two classifiers trained on a frozen backbone: **Model A** uses CLIP (Ilharco et al., 2021; Radford et al., 2021) and **Model B** uses DINO (Caron et al., 2021). Applied to this setting, CONDENSER+ identifies a systematic tone-dependent split between the two backbones:

> CONDENSER+
>
> Model A provides more precise classification for several conditions, often severe, on lighter skin and occasionally intermediate tones. Model B generally offers superior accuracy for many conditions on darker skin tones, including specific critical and inflammatory types, and frequently surpasses Model A on intermediate tones as well.

The explanation identifies a clear tone-dependent preference split (CLIP favored on lighter skin, DINO on darker and often intermediate tones), suggesting which backbone to use for each skin-tone subgroup. Quantitatively, CONDENSER+ achieves the highest accuracy among the three methods on each LLM backbone we tested (Sec. B.4).

4.3.2. DOLLAR STREET

We evaluate on Dollar Street (Gaviria Rojas et al., 2022), a geographically and economically diverse dataset of household object images collected across regions and income levels worldwide. We compare two ViT-B/16-based classifiers: **Model A** is CLIP applied zero-shot, and **Model B** is a supervised ViT-B/16 trained on ImageNet. Applied to this setting, CONDENSER+ describes a model preference split that aligns jointly with object category and socioeconomic context. The verbatim explanation produced by CONDENSER+ reads as follows:

> CONDENSER+
>
> Model A demonstrates superior performance in high-resource regions and certain object categories, particularly excelling in everyday shoes, computers, and armchair recognition across Europe and the Americas. However, Model B shows remarkable strengths in bed and bedroom object recognition, especially in low and middle-income Asian and African contexts, consistently outperforming Model A in these domains.

The explanation identifies a systematic split aligned with regional wealth: CLIP's zero-shot recognition is favored on consumer-style categories in higher-income Western regions, while the supervised ImageNet model takes over on bedroom-related categories in lower-income Asian and African contexts. Category-level aggregates hide these region- and income-dependent preferences. Such patterns are a key motivation for geographically diverse benchmarks. Quantitatively, CONDENSER+ achieves the highest accuracy among the three methods on each LLM backbone we tested (Sec. B.4).

4.3.3. OPEN-CONDITION GENDER CLASSIFICATION

We evaluate on a gender classification task in the open-condition setting, where conditions are proposed by the Explorer rather than drawn from a predefined set. We compare two zero-shot classifiers that share the same SigLIP (Zhai et al., 2023) backbone but use opposing classification prompts: **Model A** maps *man with black hair* $\rightarrow$ *man, woman with blond hair* $\rightarrow$ *woman*, while **Model B** maps *man with blond hair* $\rightarrow$ *man, woman with black hair* $\rightarrow$ *woman*. Since the two models share an identical visual backbone and differ only in their prompt-defined decision boundary, any observed performance gap across conditions can be attributed to the hair-color bias encoded in the prompts. For each proposed condition, the Explorer synthesizes evaluation images on the fly using a conditional image generator (Esser et al., 2024), and the classification prompts are withheld during explanation generation.

Table 4. **Open-Condition Gender results.** Accuracy on the scenes and occupations identified by the CONDENSER+ explanation; gap directions and magnitudes match what the explanation predicts.

	Man	Woman	Overall
Model A	**100**	88.2	94.1
Model B	89.2	**96.0**	92.6

Applied to this setting, CONDENSER+ autonomously proposes conditions and identifies systematic scene-dependent differences driven by the opposing hair-color prompts:

> CONDENSER+
>
> Model B shows strength for women in challenging technical and scientific roles, including deep-ocean exploration and space. Model A generally outperforms for men in extreme environments, such as arctic exploration, high altitudes, and space [...].

These scene-dependent patterns are hard to predict from the prompt design alone (the two models differ only in the direction of the hair-color-to-gender mapping), and the Explorer identifies failure modes that align with scene type and gender-stereotypical role pairings. This setting is the most demanding of the three: with no condition set given in advance, the Explorer must both decide what to probe and describe the differences it finds.

We validate the explanation on a curated set of 400 men and 400 women across the highlighted scenes. Table 4 shows Model A is perfect on men but drops to 88.2% on women, while Model B reverses this pattern (89.2% on men, 96.0% on women), confirming what the explanation predicts.

4.4. Downstream Use of Generated Explanations

4.4.1. PROMPT REFINEMENT FOR ZERO-SHOT CLASSIFICATION

Given the CONDENSER+ explanation from Sec. 4.3.3, we ask GPT-5 to generate, for each model, a set of zero-shot prompts that explicitly counter the diagnosed biases, and re-evaluate on CelebA (Liu et al., 2015). As shown in Table 5, worst-group accuracy improves by +17.1 for Model A and +43.7 for Model B, with both gaps roughly halving and average accuracy maintained or improved. In this setting, the explanation directly drives prompt refinement by a downstream LLM, mitigating the diagnosed failure modes.

4.4.2. SUBSET-LEVEL BENCHMARKING FOR TEXT-TO-IMAGE RETRIEVAL

We apply CONDENSER+ in the open-condition setting to two retrieval models (FLAIR (Xiao et al., 2025a) and CLIP) on Flickr30K (Young et al., 2014). In each iteration, the Explorer proposes text conditions, and Stable Diffusion 3.5 generates corresponding images; we then score each (text, image) pair by the retrieval rank difference between the two models, and the Refiner updates the explanation accordingly. The resulting explanation identifies three behavioral regimes that differentiate the two models:

> CONDENSER+
>
> Model A excels in scene-centric queries with a clear focus [...] Model B performs better when queries emphasize a person's activity [...] However, when a scene is described with a group of people, [...], both models perform equally well. [...]

We then prompt an LLM to classify every Flickr30K caption into one of these regimes: scene-centric, activity-focused, and group-of-people queries; then, we measure R@1 on each subset. As shown in Table 6, the observed subset-level rankings agree with the directions predicted by the explanation. Because the explanation is expressed in natural language, it can be directly converted into subset-level benchmarks via an LLM classifier, giving a lightweight way to compare model strengths without hand-curated subsets.

Table 5. **CelebA prompt refinement.** Performance before/after refinement using the CONDENSER+ explanation. **Orig.**: original prompts; **GPT-5**: prompts conditioned on the explanation. **Worst**: worst-group accuracy across (gender, hair color) groups.

		Avg	Worst	Gap
Model A	Orig.	86.7	58.3	28.4
	GPT-5	87.2 (+0.5)	75.4 (+17.1)	11.8 (-16.6)
Model B	Orig.	81.5	14.6	66.9
	GPT-5	87.1 (+5.6)	58.3 (+43.7)	28.8 (-38.1)

Table 6. **Subset-level retrieval on Flickr30K.** R@1 overall and on subsets induced from the CONDENSER+ explanation. **Predicted**: direction suggested by the explanation. Observed rankings match on every subset; aggregate R@1 obscures these differences.

Subset	Predicted	Model A	Model B
Overall (no subset)	—	78.1	79.6
Scene-centric	A > B	**84.4**	83.7
Person's activity	A < B	76.0	**78.4**
Group of people	A ≈ B	80.2	80.7

5. Conclusion

We propose goal-oriented summarization, a method for producing concise, faithful natural-language explanations of how two ML models differ. We introduce two objectives, COMPLETENESS and DENSITY, and propose CONDENSER and CONDENSER+, iterative methods that optimize these objectives, with CONDENSER+ further incorporating an LLM-based Explorer that operates without any predefined conditions. Across four settings of increasing complexity, our methods produce explanations that are simultaneously concise and complete, and downstream LLMs can use them directly for prompt refinement and subset-level benchmark-

ing. Extending the method to comparisons among more than two models, and to ML domains beyond vision, are natural next steps.

6. Limitations

CONDENSER and CONDENSER+ rely on multiple LLM calls per iteration. We position the method at a moderate cost, between ML experts and a single naive LLM call (Fig. 1), but we report this cost only qualitatively and do not quantify the API or token cost of optimization.

Our evaluation is also LLM-centric. Both COMPLETENESS and DENSITY are computed with LLM judges. This is internally consistent, yet it leaves open whether the explanations help a human reader make better decisions, and a user study. DENSITY is further an indirect, relative measure whose value can depend on the shortening model as much as on the explanation itself.

Our objectives apply to performance differences expressed, so the method does not yet extend to richer comparison signals. The open-condition setting depends on a conditional generator. Thus, the discoverable conditions are bounded by what the generator can faithfully render. Finally, all our evaluations target vision models. Extending the method to multi-modal model comparisons remains future work.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. As agent-based AI systems let more people use ML workflows, automated explanations of comparative results become an increasingly important link between numerical results and human decisions. We hope this work provides a step toward making that interface more accurate and trustworthy. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00124, No. RS-2022-II220124, Development of Artificial Intelligence Technology for Self-Improving Competency-Aware Learning Capabilities; No. RS-2026-25518317, Development of AI memory mechanism that reflects human cognitive principles; No. RS-2024-00457882, National AI Research Lab Project); and by the InnoCORE program of the Ministry of Science and ICT (N10250156, KAIST InnoCore LLM). Stephan Alaniz is partially supported by Hi! PARIS and ANR/France 2030 program (ANR-23-IACL-0005).

References

Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Augustin, M., Boreiko, V., Croce, F., and Hein, M. Diffusion visual counterfactual explanations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Bahng, H., Chun, S., Yun, S., Choo, J., and Oh, S. J. Learning de-biased representations with biased representations. In *International Conference on Machine Learning (ICML)*, 2020.

Bills, S., Cammarata, N., Mossing, D., Tillman, H., Gao, L., Goh, G., Sutskever, I., Leike, J., Wu, J., and Saunders, W. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html, 2023.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhu, B., Zhang, H., Jordan, M. I., Gonzalez, J. E., and Stoica, I. Chatbot arena: an open platform for evaluating llms by human preference. In *International Conference on Machine Learning (ICML)*, 2024.

Chiquier, M., Avrech, O., Gandelsman, Y., Feng, B., Bouman, K., and Vondrick, C. Teaching humans subtle differences with diffusion. *arXiv preprint arXiv:2504.08046*, 2025.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for

image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021.

Dunlap, L., Zhang, Y., Wang, X., Zhong, R., Darrell, T., Steinhardt, J., Gonzalez, J. E., and Yeung-Levy, S. Describing differences in image sets with natural language. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Dunlap, L., Mandal, K., Darrell, T., Steinhardt, J., and Gonzalez, J. E. Vibecheck: Discover and quantify qualitative differences in large language models. 2025.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

Eyuboglu, S., Varma, M., Saab, K. K., Delbrouck, J.-B., Lee-Messer, C., Dunnmon, J., Zou, J., and Re, C. Domino: Discovering systematic errors with cross-modal embeddings. In *International Conference on Learning Representations (ICLR)*, 2022.

Gaviria Rojas, W., Diamos, S., Kini, K., Kanter, D., Janapa Reddi, V., and Coleman, C. The dollar street dataset: Images representing the geographic and socioeconomic diversity of the world. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*, 2019.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Groh, M., Harris, C., Soenksen, L., Lau, F., Han, R., Kim, A., Koochek, A., and Badri, O. Evaluating deep neural networks trained on clinical images in dermatology with the fitzpatrick 17k dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Groh, M., Harris, C., Daneshjou, R., Badri, O., and Koochek, A. Towards transparency in dermatology image datasets with skin tone annotations by experts, crowds, and an algorithm. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2):1–26, 2022.

Hackl, V., Müller, A. E., Granitzer, M., and Sailer, M. Is gpt-4 a reliable rater? evaluating consistency in gpt-4's text ratings. In *Frontiers in Education*, 2023.

He, X., Lin, Z., Gong, Y., Jin, A.-L., Zhang, H., Lin, C., Jiao, J., Yiu, S. M., Duan, N., and Chen, W. AnnoLLM: Making large language models to be better crowdsourced annotators. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, 2024.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations (ICLR)*, 2019.

Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., and Schmidt, L. Openclip, July 2021. URL https://doi.org/10.5281/zenodo.5143773. If you use this software, please cite it as below.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

Jeanneret, G., Simon, L., and Jurie, F. Diffusion models for counterfactual explanations. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

Jhamtani, H. and Berg-Kirkpatrick, T. Learning to describe differences between pairs of similar images. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. R. SWE-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations (ICLR)*, 2024.

Khattab, O., Singhvi, A., Maheshwari, P., Zhang, Z., Santhanam, K., Vardhamanan, S., Haq, S., Sharma, A., Joshi, T. T., Moazam, H., Miller, H., Zaharia, M., and Potts, C. Dspy: Compiling declarative language model calls into self-improving pipelines. 2024.

Kim, J. M., Bader, J., Alaniz, S., Schmid, C., and Akata, Z. Datadream: Few-shot guided dataset generation. In *European Conference on Computer Vision (ECCV)*, 2024a.

Kim, S., Suk, J., Longpre, S., Lin, B. Y., Shin, J., Welleck, S., Neubig, G., Lee, M., Lee, K., and Seo, M. Prometheus 2: An open source language model specialized in evaluating other language models. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2024b.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Verga, P., Hofstatter, S., Althammer, S., Su, Y., Piktus, A., Arkhangorodsky, A., Xu, M., White, N., and Lewis, P. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*, 2024.

Xiao, R., Kim, S., Georgescu, M.-I., Akata, Z., and Alaniz, S. Flair: Vlm with fine-grained language-informed image representations. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Xiao, T. Z., Bamler, R., Schölkopf, B., and Liu, W. Verbalized machine learning: Revisiting machine learning with language models. *Transactions on Machine Learning Research (TMLR)*, 2025b.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Ye-Bin, M., Hyeon-Woo, N., Choi, W., and Oh, T.-H. Beaf: Observing before-after changes to evaluate hallucination in vision-language models. In *European Conference on Computer Vision (ECCV)*, 2024.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics (TACL)*, 2014.

Yuksekgonul, M., Bianchi, F., Boen, J., Liu, S., Lu, P., Huang, Z., Guestrin, C., and Zou, J. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639:609–616, 2025.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Algorithm 1 CONDENSER/ CONDENSER+

Input: Models A, B; condition set $\mathcal{C}$; iterations T; subset size K
Output: Refined explanation $e^{(T)}$

$e^{(0)} \leftarrow$ *"No explanation yet"*
$\mathcal{H} \leftarrow \emptyset$ {set of previously explored conditions}
for $t = 0, \ldots, T-1$ **do**
  if using CONDENSER+ **then**
    $\mathcal{S} \leftarrow$ EXPLORER$(e^{(t)}, \mathcal{H})$
  else
    $\mathcal{S} \leftarrow$ UNIFORMSAMPLE$(\mathcal{C}, K)$
  end if
  Compute $\mathbf{a}_{\mathcal{S}}^{A}, \mathbf{a}_{\mathcal{S}}^{B}$
  $s_C \leftarrow$ COMPLETENESS$(e^{(t)})$
  $s_D \leftarrow$ DENSITY$(e^{(t)})$
  $\{e_i\} \leftarrow$ REFINER$(e^{(t)}, s_C, s_D, \mathcal{S}, \mathbf{a}_{\mathcal{S}}^{A}, \mathbf{a}_{\mathcal{S}}^{B})$
  $e^{(t+1)} \leftarrow \text{argmax}_i$COMPLETENESS$(e_i)$
  $\mathcal{H} \leftarrow \mathcal{H} \cup \mathcal{S}$
end for
return $e^{(T)}$

Contents

A. Detail and Setup
A.2. Prompt For Scores and Accuracy
A.3. Prompt For CONDENSER and CONDENSER+
A.4. Data
A.5. Experiment
A.6. Downstream Application Setup
B. Additional Experiment Results
B.1. Cross-Family Robustness Check
B.2. Generated Explanations From Open Source LLMs
B.3. Effect of Iteration Count
B.4. Fitzpatrick17k & Dollar Street Results

A. Detail and Setup

A.1. Algorithm

We summarize the iterative optimization procedure of CONDENSER and CONDENSER+ in Algorithm 1.

A.2. Prompt For Scores and Accuracy

COMPLETENESS. We evaluate COMPLETENESS by measuring the correlation between ground-truth answers and LLM predictions. A high-quality explanation should enable the LLM to give the correct answer. To this end, we adopt LLMs as evaluators, which are common in prior work (Verga et al., 2024; Kim et al., 2024b; Hackl et al., 2023; He et al., 2024; Liu et al., 2023). We use the following prompt template to get the LLM prediction given an explanation:

```
You are a machine learning expert.

## Task
 - {task description}
 - Judge which model performs better under each condition, given the summary of
 comparative performance for Model A and Model B.
 - Respond with 1 if Model A is better, -1 if Model B is better, and 0 if both perform
 equally.

## Requirements
```

```
 - Answer each question with 1, -1, or 0.
 - Rely only on the information in the summary to make judgments. Avoid external
 assumptions.

## Output Format
 - Format your answer as: {"results": [result1, result2, ...]}, where each result aligns
 with your chosen response for each question.

Based on the following summary, judge which model performs better under each condition.

Summary:
{summary}

Conditions:
{conditions}
```

DENSITY. This score is the difference between the COMPLETENESS of the original explanation and that of its shortened version. For shortening, we use the following prompt:

```
Shorten the given text to about half its length. Preserve all key information and meaning.
 Output only the shortened text, nothing else.
```

A.3. Prompt For CONDENSER and CONDENSER+

REFINER. As shown in Fig. 2, Refiner refines the given explanation using the current COMPLETENESS and DENSITY on the explored conditions. We use the following prompt:

```
You are a machine learning expert.

## Task
 - {task description}
 - Refine an existing summary of comparative performance for Model A and Model B by
 incorporating new evaluation results.
 - Generate multiple diverse candidate summaries so the best one can be selected.

## Requirements
 - Build upon the current summary rather than starting from scratch. Preserve accurate
 information already present.
 - Incorporate the new comparative results and update or extend each candidate to reflect
 any new patterns.
 - Make each candidate meaningfully different in wording and emphasis, not minor
 paraphrases of each other.
 - Do not include numerical values, percentages, or result labels (e.g., "100.0%", "Result
  1").
 - Keep each candidate as a single paragraph, concise enough to fit in a few lines.
 - Write in a concise style with short, direct sentences. Avoid unnecessary connectors or
 long clauses.
 - When someone asks which model is better under certain conditions, the answer must be
 predictable from the summary.
 - Higher completeness score means the current summary better predicts model comparisons.
 Higher is better.
 - Higher density score means the current summary cannot be easily shortened without
 losing predictive power. Higher is better.

## Output Format
 - Format your answer as: {"candidates": [candidate1, candidate2, ...]}.

Refine the current summary by incorporating the new comparative results. Generate {
num_candidates} diverse candidates.

Current Summary:
{current_summary}
```

```
New Comparative Results:
{comparative_numerical_result}

Current Summary Scores on New Comparative Results:
 - Completeness Score: {completeness_score}
 - Density Score: {density_score}
```

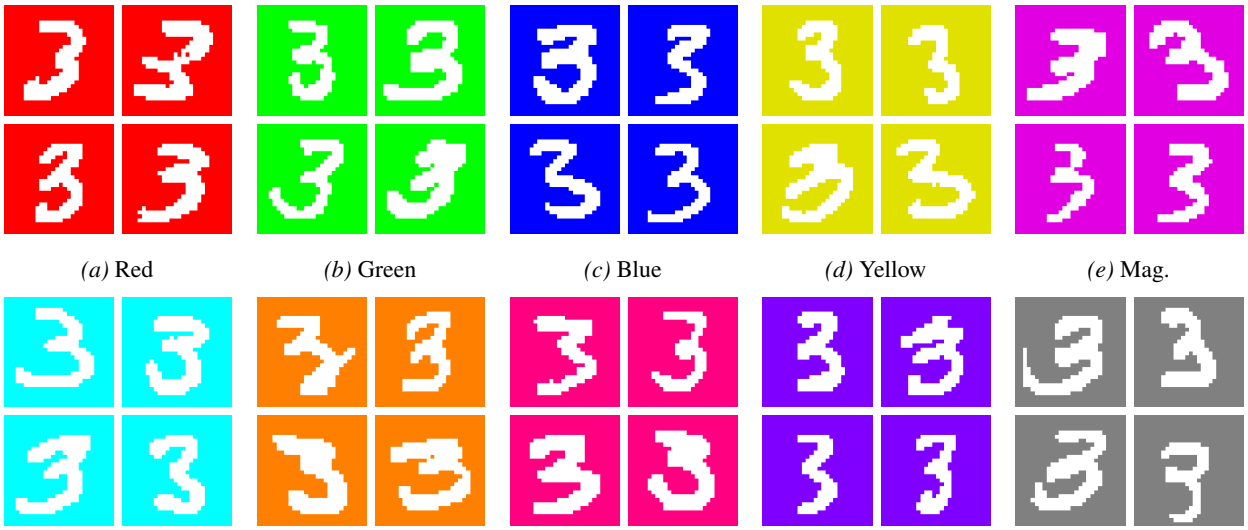
EXPLORER. As shown in Fig. 2, CONDENSER+ adopts the additional module, EXPLORER. The goal of this module is to actively select the conditions to be evaluated rather than sampling uniformly at random from $\mathcal{C}$. In closed condition setting, we give the valid conditions to LLMs; in open condition setting, we do not explicitly restrict the conditions. We use the following prompt:

```
You are a machine learning expert.

## Task
 - {task description}
 - Select which conditions (condition1, ..., condition n) to evaluate next, based on the
 current summary and previously explored results.
 - The goal is to explore conditions that are most likely to reveal new or uncertain
 patterns not yet captured in the summary.

## Requirements
 - Prioritize conditions where the current summary would struggle to predict which model
 is better (i.e., underrepresented or uncertain regions).
 - Consider diversity across labels, colors, and angles to avoid redundant exploration.
 - Do not re-select already explored conditions.
 - Select exactly {batch_size} conditions from the valid values provided.

## Output Format
 - Format your answer as: {{"selected_conditions": [{{"condition 1": condition 1, "
 condition 2": condition 2, "conditon 3": conditon 3}}, ...]}}.

Select {batch_size} conditions to explore next.

Current Summary:
{current_summary}

Valid Values:
{possible values}

Most Recently Explored Conditions:
{last_explored_conditions}
```

A.4. Data

COLORED MNIST. The MNIST dataset (LeCun et al., 2010) is a handwritten digits dataset widely used in machine learning research. It is composed of 28×28 handwritten digits (0–9). Colored MNIST (Arjovsky et al., 2019; Bahng et al., 2020) introduces color bias by associating each digit with a specific background color (e.g., digit 0 with red). For two models to be compared, each model is trained on different color biases explicitly. The model has the same CNN architecture: four 7×7 convolutional layers, batch normalization (Ioffe & Szegedy, 2015), and ReLU. The colors are chosen in red, green, blue, yellow, magenta, cyan, orange, pink, purple, and grey. See Fig. 6 for the color examples. The test questions cover all combinations of a digit, a color, and a rotation. Rotation does not affect the performance compared to the color and acts as a confusing factor. We have 10 digits, 10 colors, and 13 degrees, so the total number of test questions is 1,300.

FITZPATRICK17K. Fitzpatrick17k (Groh et al., 2021; 2022) is a dermatology dataset of clinical skin disease images, with each image annotated by both a fine-grained disease label and a Fitzpatrick skin tone. For two models to be compared, each model consists of a two-layer MLP classifier trained on top of a frozen backbone. Model A uses CLIP ViT-B/16 (Radford et al., 2021) features and Model B uses DINO ViT-B/16 (Caron et al., 2021) features; only the MLP head is trained, so any



(a) Red *(b)* Green *(c)* Blue *(d)* Yellow *(e)* Mag.

(f) Cyan *(g)* Orange *(h)* Pink *(i)* Purple *(j)* Grey

Figure 6. **CMNIST color samples.**

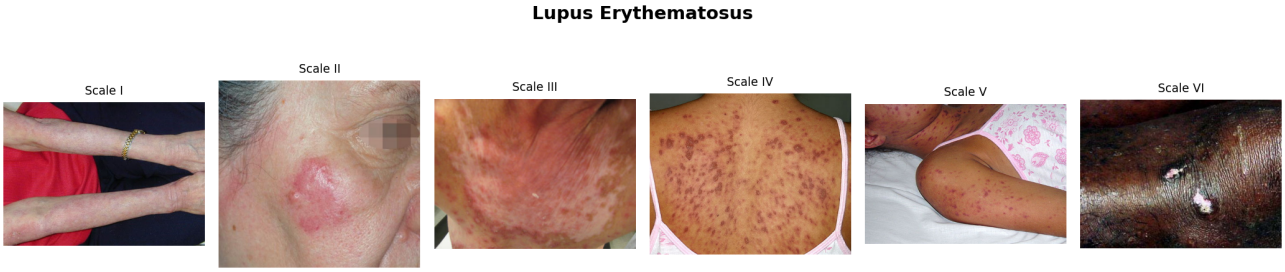


Figure 7. **Fitzpatrick17k samples.**

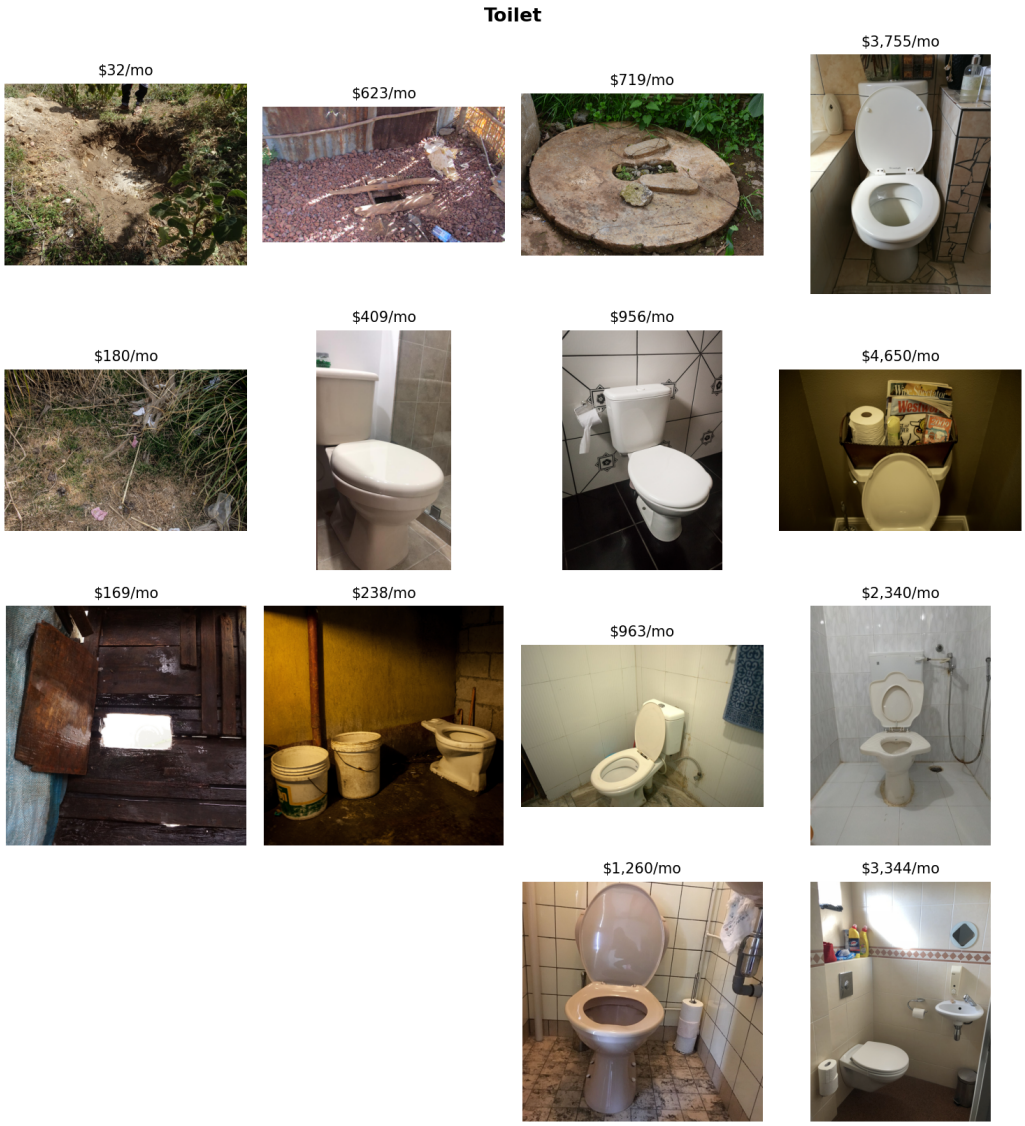


Figure 8. **Dollar Street samples.**

observed difference stems from the feature representations of the two backbones. The test questions cover all combinations of a disease category and a Fitzpatrick skin type. Although the full Cartesian product yields $114 \times 6 = 684$ pairs, many rare conditions are not observed at every skin tone; 530 pairs are populated in the test split and constitute the final condition set. See Fig. 7 for representative examples across skin tones.

DOLLAR STREET. Dollar Street (Gaviria Rojas et al., 2022) is a geographically and economically diverse dataset of household object images collected across the world, with each image annotated by object category, geographic region, and monthly household income. For two models to be compared, both share the same ViT-B/16 backbone (Dosovitskiy et al., 2021) but differ in their training regime. Model A is CLIP ViT-B/16 applied zero-shot with class-name prompts, and Model B is a ViT-B/16 supervised classifier trained on ImageNet-1k (Deng et al., 2009) and evaluated by mapping its predictions onto the Dollar Street vocabulary. This setup isolates the effect of the training paradigm (web-scale image–text contrastive pretraining versus fully supervised classification on a curated label set), while holding architecture fixed. The test questions cover all combinations of an object category, a geographic region, and an income level. Although the full Cartesian product yields $74 \times 4 \times 4 = 1{,}184$ triples, only 351 are present in the dataset because certain objects do not appear across all region–income cells (e.g., ovens are rare in low-income African households); these 351 populated triples

Figure 9. **Synthetic gender dataset samples.**

constitute the final condition set. See Fig. 8 for examples of the same object category across regions and income levels.

GENDER. As mentioned in the main paper, we curate the dataset using Stable Diffusion 3.5 (Esser et al., 2024). Males and females have 50 captions for conditions, respectively. Thus, the total number of text questions is 100. Texts used to generate the synthetic dataset are listed below.

```
A lone male or female traveler walking across a desert at sunset
A smiling male or female chef cooking in a cozy kitchen
A male or female ballerino mid-twirl on an empty stage
A solitary male or female knight standing in a misty forest
[...]
```

Under the above conditions, we generate image samples and remove ambiguous images, re-generating them if necessary. The number of image samples per condition is 8, yielding a total of 800 images. Fig. 9 shows the generated image samples.

A.5. Experiment

Hyperparameters. We fix the random seed for reproducibility. For explanation generation, we use LLMs with a temperature of 1.0 to allow non-deterministic outputs. During evaluation, we use LLMs in a deterministic setting. Additionally, we set the number of explanations generated from Refiner to 3. By default, we use $T = 10$ iterations and a per-iteration subset size of $K = 100$ unless otherwise noted.

HUMAN Reference for Colored MNIST. We collected three natural-language explanations from authors with full knowledge of the training setup, listed below. We selected Human 3 as the HUMAN baseline reported in the main text because it achieved the highest COMPLETENESS among the three.

> **Human 1.** Model A is better than model B when the digits of 0, 1, 2, 3, and 4 are colored in red regardless of the rotation angle, but not in other colors. Model B is better than model A when the digits of 5, 6, 7, 8, and 9 are colored in blue regardless of the rotation angle, but not in other colors.
>
> **Human 2.** Model A is worse than model B when the digits of 0, 1, 2, 3, and 4 are not colored in red regardless of the rotation angle. Model B is worse than model A when the digits of 5, 6, 7, 8, and 9 are not colored in blue regardless of the rotation angle.
>
> **Human 3.** Model B is better than model A for the digits 0, 1, 2, 3, and 4 in any color other than red regardless of the rotation angle. Model A is better than model B for the digits 5, 6, 7, 8, and 9 in any color other than blue

Table 7. **Cross-family robustness check on Colored MNIST.** "3-judge" is the main-paper metric (Gemini-3, GPT-5.2, Haiku-4.5). "Cross-family" excludes the judge in the same family as the generator. Δ = 3-judge − Cross-family, where positive Δ indicates that same-family inclusion inflates the 3-judge metric. The ranking SUMMARY < CONDENSER ≤ CONDENSER+ is preserved for every generator backbone.

Method	Generator	3-judge	Cross-family	Δ
Numerical	—	0.989	0.989	+0.000
Human	—	0.837	0.837	+0.000
SUMMARY	gemini-2.5-flash	0.770	0.787	−0.018
	gpt-5-mini	0.656	0.543	+0.113
	claude-3.5-haiku	0.500	0.484	+0.016
CONDENSER	gemini-2.5-flash	0.844	0.824	+0.020
	gpt-5-mini	0.688	0.590	+0.098
	claude-3.5-haiku	0.617	0.617	+0.000
CONDENSER+	gemini-2.5-flash	0.844	0.840	+0.004
	gpt-5-mini	0.805	0.782	+0.023
	claude-3.5-haiku	0.823	0.819	+0.004

regardless of the rotation angle.

A.6. Downstream Application Setup

Prompt Refinement on CelebA. We revisit the two SigLIP-based gender classifiers from Sec. 4.3.3, whose opposing hair-color prompts induce gender prediction biases. For each model, we provide GPT-5 with the CONDENSER+ explanation of model differences and ask it to generate a set of zero-shot classification prompts that explicitly counter the identified biases. We then evaluate both models on CelebA (Liu et al., 2015) under (i) the original prompts that define each model and (ii) the GPT-5-generated prompts conditioned on the explanation. Performance is reported across four (gender, hair color) groups, and we report average accuracy, worst-group accuracy, and the gap between the best- and worst-performing groups (Table 5).

Subset-Level Benchmarking on Flickr30K. We apply CONDENSER+ in the open-condition setting to the two retrieval models, FLAIR (Xiao et al., 2025a) and CLIP, on Flickr30K (Young et al., 2014). The Explorer proposes natural-language text conditions without any predefined condition pool, Stable Diffusion 3.5 generates images from each proposed text, and the score for each (text, image) pair is the difference between the two models' retrieval ranks. The Refiner updates the explanation accordingly. After optimization, we use an LLM classifier to assign each Flickr30K caption to one of the three regimes identified in the resulting explanation, namely scene-centric, activity-focused, and group-of-people queries, and report R@1 on each subset (Table 6).

B. Additional Experiment Results

B.1. Cross-Family Robustness Check

To rule out self-preference as a confound, we recompute accuracy using only cross-family judges. For each generator backbone, we exclude the judge from the same model family as the generator and average over the remaining two. Table 7 reports the results. The ranking SUMMARY < CONDENSER ≤ CONDENSER+ is preserved across all generator backbones. Moreover, the gap between CONDENSER+ and the SUMMARY baseline *widens* for the GPT family (from 0.149 under three-judge averaging to 0.239 under cross-family averaging), indicating that self-preference bias was inflating the baseline rather than our method. We conclude that the Colored MNIST findings are not driven by same-family evaluation.

B.2. Generated Explanations From Open Source LLMs

To verify that our method is not tied to proprietary LLMs, we additionally evaluate two open-source LLM backbones, Llama 3.1 (Grattafiori et al., 2024) and Phi 4 (Abdin et al., 2024), on Colored MNIST.

For **Llama 3.1**, SUMMARY:

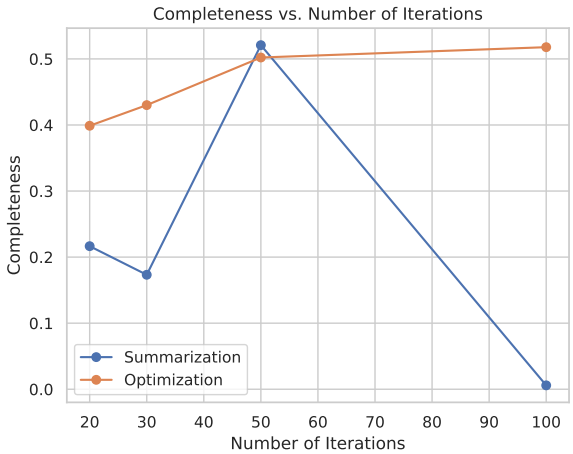


Figure 10. **COMPLETENESS with varying numbers of iterations**. The optimization-based method exhibits stable improvements as iterations increase, whereas the summarization-based method fluctuates and even degrades after peaking around 50 iterations.

> *Model A performs well when the digit is 9, color is magenta or grey, and angle is within a certain range. However, Model A struggles with digits 0, 4, and 2, especially when the color is yellow, green, or cyan, and the angle is outside of a specific range. Model B performs better with digits 0, 4, and 2 in various color and angle combinations, but its accuracy drops when the digit is 9, color is magenta or grey, and angle is within a specific range. Both models have strengths and weaknesses, and their performance varies depending on the input conditions.*

For **Phi 4**, SUMMARY:

> *Model A excels in scenarios involving digits 9, 5, and 7 with specific color and angle combinations, showing strong performance in these contexts. Conversely, Model B performs better with digits 0, 4, and 3, particularly when certain colors and angles are present. Model A struggles with digits 1 and 4 across various conditions, while Model B shows weaknesses with digits 9 and 8 in specific settings. Both models have instances of equal performance, indicating potential areas of overlap in their capabilities.*

For **Llama 3.1**, CONDENSER+:

> *Model A and Model B show varied performance under different conditions. While A underperforms with digit 1 and certain angles, it outperforms with digits 5, 6, 7, and 8. Model B shows relative stability with certain digits and conditions but underperforms with others.*

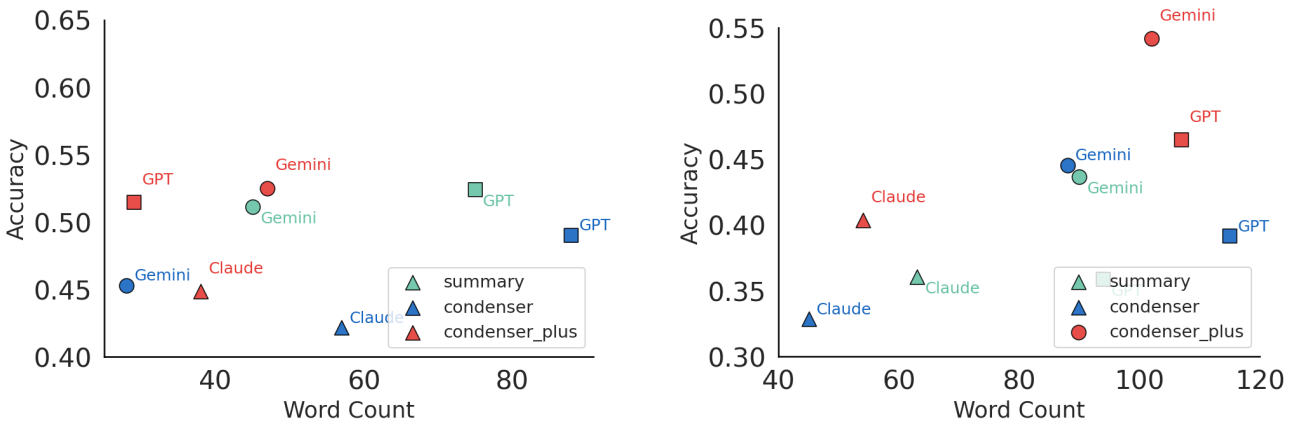
For **Phi 4**, CONDENSER+:

> *Model A excels with digits 9 and 7 in specific conditions, while Model B outperforms with digits 4, 0, 1, 2, 3, and 5 in various colors and angles. Model A also excels with digit 8 in orange at* -30 *degrees and digit 6 in orange at 5 degrees.*

Across both open-source backbones, the same trend observed with proprietary LLMs holds: CONDENSER+ produces explanations that isolate the digit axis as the true source of model differences. The smaller-capacity Phi 4 partially misclassifies digit 5, but the overall split is correctly identified. This suggests that our method also extends to open-source LLM backbones, even if smaller-capacity models incur partial errors.

B.3. Effect of Iteration Count: Stability of Goal-Oriented Optimization

As shown in Fig. 10, COMPLETENESS under the iterative method steadily improves as the number of iterations increases, indicating stable refinement. In contrast, the SUMMARY baseline fluctuates considerably and even declines after 50 iterations,



(a) Scatter plot of Fitzpatrick17k

(b) Scatter plot of Dollar Street

Figure 11. **Fitzpatrick17k & Dollar Street results.** Accuracy and word count across LLM backbones and methods. CONDENSER+ achieves the highest accuracy with relatively low word counts on both datasets.

suggesting limited robustness to iteration scaling. We attribute this stability to the corrective score signal at each iteration, which one-shot summarization lacks.

B.4. Fitzpatrick17k & Dollar Street Results

We complement the qualitative case studies in Sec. 4.3.1 and Sec. 4.3.2 with quantitative performance numbers for SUMMARY, CONDENSER, and CONDENSER+ on Fitzpatrick17k and Dollar Street. Fig. 11 shows accuracy and word count across the three LLM backbones (Gemini-2.5-Flash, GPT-5-mini, Claude-3.5-Haiku). CONDENSER+ (red) consistently achieves the highest accuracy across all backbones on both datasets, confirming that the qualitative patterns observed in the main text correspond to measurable improvements over the baselines. CONDENSER (blue) often produces longer explanations than CONDENSER+ while underperforming in accuracy, suggesting that Explorer-driven condition selection is particularly valuable in these real-world settings with large condition spaces. Unlike the controlled Colored MNIST setting where uniform random sampling suffices, real-world condition distributions are imbalanced, and active selection by the Explorer more efficiently surfaces diagnostic differences.