DeltaPrompts: Escaping the Zero-Delta Trap in Multimodal Distillation

Jaehun Jung **Hyunwoo Kim** **Brandon Cui** **Ximing Lu**
David Acuna **Prithviraj Ammanabrolu** **Yejin Choi**

NVIDIA Research

Abstract

Distillation enables compact Vision-Language Models (VLMs) to obtain strong reasoning capabilities, yet the prompts driving this process are typically chosen via simple heuristics or aggregated from off-the-shelf datasets. We reveal a critical inefficiency in this approach: up to 69% of the prompts in standard chart / document reasoning datasets are effectively *zero-delta*, meaning the teacher and student already induce the exact same answer distribution. Training on these prompts provides minimal learning signal, causing student improvement to rapidly saturate regardless of data scale. To escape the zero-delta trap, we return to first principles: distillation fundamentally minimizes distributional divergence, and thus a prompt is valuable only if it exposes a functional capability gap between the teacher and student. We quantify this gap through *answer divergence* (Δ), demonstrating that non-zero divergence is critical for effective scaling. Building on this insight, we propose a staged synthesis pipeline that repurposes existing datasets as seeds, actively targeting student failure modes to produce better prompts. The result is **DELTAPROMPTS**, a diverse dataset of 200k synthetic, high-divergence reasoning problems. We evaluate DELTAPROMPTS across three distinct settings: on-policy distillation with the target teacher-student pair, transfer to a novel model family without regenerating the data, and off-policy fine-tuning of a non-reasoning model. Across all scenarios, DELTAPROMPTS drives substantial gains, yielding up to 15% relative improvement even on top of a highly-optimized reasoning model (*e.g.,* Qwen3-VL-8B-Thinking)—averaged over 10 benchmarks spanning chart, document and perception-centric reasoning.

1 Introduction

Distillation plays a pivotal role in modern Vision-Language Models, offering compact models an efficient path to learn from a stronger teacher [4, 54]. A growing body of work has advanced the algorithmic foundations of this process—from stabilizing divergence objectives [13, 30, 61] to on-policy variants that mitigate distribution shift [2, 74, 41]—and has seen particular success on reasoning-heavy tasks. But across these advancements, comparatively little attention has been paid to the *prompts* themselves: in practice, reasoning problems are often collected from a standard array of datasets [59, 73, 37, 71], either selected by researchers or via heuristics such as difficulty. It remains an open question whether the off-the-shelf prompts are indeed suitable for closing the capability gap, and if not, how to systematically improve them.

In this work, we study what makes a prompt effective for multimodal distillation—and how to construct better ones at scale. We argue that a prompt is valuable only insofar as it exposes a *capability gap* between teacher and student—when the two models induce effectively the same answer distribution on the input, the distillation signal reveals little about what the student lacks.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

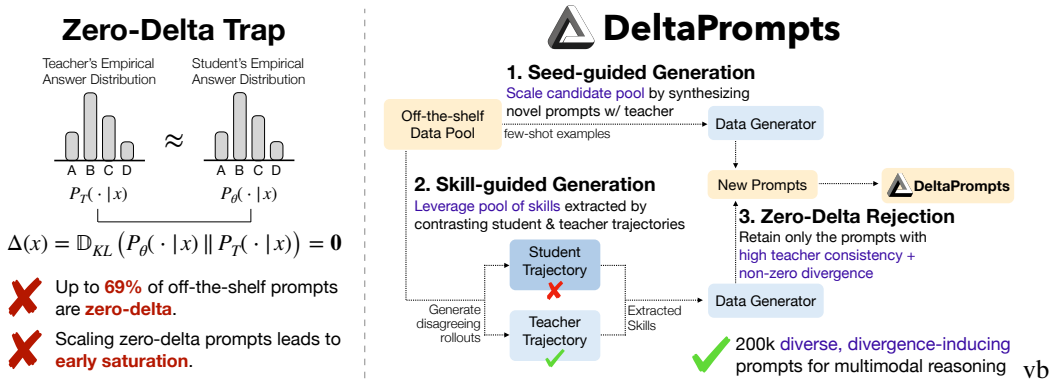


Figure 1: **Overview of DELTAPROMPTS**. We first identify the *zero-delta trap*: for majority of the prompts in chart & document reasoning datasets, the teacher and student already produce an identical answer distribution. We show that distillation on **these zero-delta prompts reaches early saturation in student performance** (§2.3). We then synthesize **DELTAPROMPTS, leveraging teacher model itself to find the divergence-inducing prompts, and using off-the-shelf data as seeds rather than a direct training resource** (§3).

From this perspective, the quality of a distillation corpus depends not on how many raw samples it contains, but on *how often it surfaces the prompts where the teacher and student actually disagree.*

To make this idea practical, we measure the divergence between the discrete answer distribution induced by the two models given a prompt, which we call **answer divergence** (Δ). Despite its simplicity—sidestepping the noisiness and cost of sequence-level divergence—answer divergence proves critical for effective distillation: training on *delta prompts* with non-zero Δ continues to improve the student, while *zero-delta prompts* saturate rapidly regardless of data scale. Importantly, our analysis reveals that existing datasets fall short precisely on answer divergence—**up to 69% of the prompts in standard chart / document reasoning datasets are effectively zero-delta**, *i.e.,* the student and teacher produce the exact same answer distribution. The remaining 31% exhibit significantly lower diversity than the original pool, making naive filtering an unreliable solution.

Motivated by these observations, we repurpose existing datasets not as a direct training source, but as seeds for synthesizing better prompts. Our key insight is that unlike fixed data curation, synthesis allows us to actively search for the properties that matter in distillation—**non-zero divergence and broad skill coverage**. We materialize this idea into a staged pipeline that (1) expands the prompt pool via seed-guided synthesis, and (2) augments specific reasoning skills that student fails to exhibit, all with the teacher model itself as the data generator. The resulting prompts exhibit both higher answer divergence and greater diversity than the existing datasets they originated from, even after aggressive filtering for quality.

We apply our pipeline to chart & document reasoning and high-resolution, perception-centric reasoning—domains where high-quality reasoning problems are particularly sparse—and produce **DELTAPROMPTS**, a dataset of 200k synthetic, divergence-inducing prompts. With on-policy distillation, DELTAPROMPTS yields up to 15% relative improvement across benchmarks, even on top of an already highly-optimized reasoning model (*e.g.,* Qwen3-VL-8B-Thinking [4]). These gains transfer to an entirely different model family (*e.g.,* GLM-4 [54]) without regenerating the data, and extend to out-of-domain tasks such as STEM reasoning. DELTAPROMPTS also serves as an effective off-policy SFT resource, substantially outperforming off-the-shelf data mixtures. Taken together, our results suggest that the central bottleneck in multimodal distillation is not in aggregating more prompts or trajectories, but in constructing prompts where the teacher and student meaningfully disagree.

2 What prompts are fit for distillation?

2.1 Formulating Distillation Objective

Given a teacher model π_T and a student model π_θ, knowledge distillation optimizes the student to minimize the divergence between the two models' output distributions conditioned on a prompt x:

$$\mathcal{L}_{\text{distill}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{D}\left(\pi_T(\cdot|x) \,\|\, \pi_\theta(\cdot|x)\right)\right] \tag{1}$$

where $\mathcal{D}$ is the prompt distribution and $\mathbb{D}$ is a divergence measure. The choice of divergence $\mathbb{D}$ and the sampling strategy for outputs give rise to different instantiations of this objective—for example, in supervised knowledge distillation [49, 16], outputs are sampled from the teacher π_T and the student is trained to minimize the forward KL, either by matching the token-level probabilities or by maximizing likelihood of teacher outputs [28, 24]. Conversely, on-policy distillation (OPD) [2] samples from the student's own distribution and minimizes the reverse KL:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{\widehat{y}\sim\pi_\theta(\cdot|x)}\left[\log\frac{\pi_\theta(\widehat{y}|x)}{\pi_T(\widehat{y}|x)}\right]\right] \tag{2}$$

Beyond its original formulation, several additional improvements have been proposed to augment this objective, such as harnessing the student model itself as a teacher with privileged information [74, 69], better decoding mechanism [55], and auxiliary training signals [22, 65]. We discuss further related works in §D.

Setting aside the algorithmic specifics, we observe that the learning signal in distillation is fundamentally driven by the distributional gap between teacher and student—*i.e.,* how the two models diverge in their outputs given an input x. From the perspective of dataset $\mathcal{D}$, the value of each prompt $x \in \mathcal{D}$ ultimately reduces to how much capability gap it induces between the two models. In principle, one could quantify this per-prompt gap by directly measuring the sequence-level divergence $\mathbb{D}(\pi_T(\cdot|x)||\pi_\theta(\cdot|x))$. But in practice, this is both computationally intractable and highly noisy: the distributions are defined over variable-length sequences $\hat{y}$, where a trivial phrasing difference or an early token deviation can dominate the estimate of divergence.

2.2 Answer Divergence Δ

Instead of comparing full output sequences, we propose answer divergence Δ over the discrete space of final answers. This reduces the support from variable-length token sequences to a much smaller set of equivalence classes over final answers, making the computation tractable while focusing on the functional disagreement between models—whether they arrive at different conclusions, not whether they phrase the same conclusion differently.

Defining answer divergence Let $\mathcal{Y}$ denote the space of all possible variable-length sequences, and $\mathcal{A}$ denote the discrete space of possible final answers (*e.g.,* multiple-choice options, extracted scalar values, or open-ended answers). We define an extraction function $\mathcal{E} : \mathcal{Y} \to \mathcal{A}$ that maps a full sequence y to its final answer a. We then naturally define the induced answer distribution as:

$$P_\pi(a|x) = \sum_{y\in\mathcal{Y}:\,\mathcal{E}(y)=a} \pi(y|x) = \mathbb{E}_{y\sim\pi(\cdot|x)}\left[\mathbb{I}(\mathcal{E}(y)=a)\right] \tag{3}$$

The answer divergence $\Delta(x)$ can now be formulated by applying our chosen divergence measure $\mathbb{D}$ (*e.g.,* reverse KL) over the induced answer distributions of the two models:

$$\Delta(x) = \mathbb{D}_{\text{KL}}\left(P_\theta(\cdot|x)||P_T(\cdot|x)\right) \tag{4}$$

In practice, the exact marginalization over the sequence space in Eq. 3 is computationally intractable. We therefore approximate the true answer distribution by sampling $K = 16$ trajectories from both the teacher and student for a given prompt x. To robustly handle surface-level variations in the answer space, we employ an LLM judge (Qwen3-32B) to evaluate the semantic equivalence of final answers across the K samples. Grouping equivalent answers, we construct the empirical answer distributions, $\widehat{P}_T(a|x)$ and $\widehat{P}_\theta(a|x)$, from which we compute the empirical answer divergence $\widehat{\Delta}(x)$. In addition to handling surface-level variations, our LLM-judge based approach generalizes to open-ended or long-form answers, where exact string matching fails to capture semantic equivalence.

2.3 Impact of Answer Divergence on Distillation

Our next step is to validate whether the answer divergence of the prompts actually leads to systematic differences in model performance. To this end, we run a series of controlled training runs using standard multimodal datasets for chart & document reasoning.

We first aggregate the train splits of off-the-shelf datasets—ChartQA, SciVQA, InfoVQA, arXivQA [45, 5, 46, 35]—to create a mixture of 139k unique reasoning problems and measure their answer

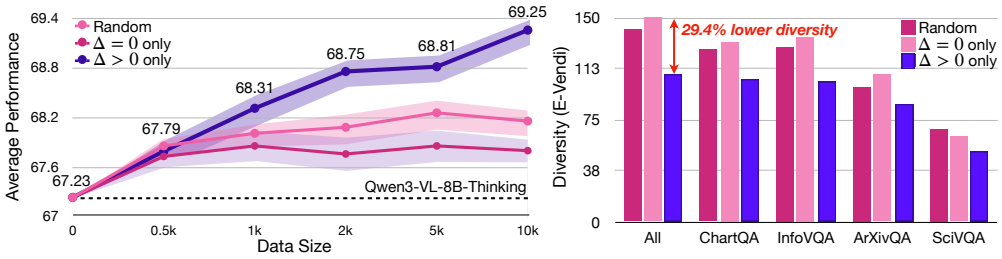


Figure 2: (Left) **Divergence is critical for effective OPD**. When controlling for diversity, delta prompts ($\Delta > 0$) improves with data size while random or zero-delta subsets meet early saturation. (Right) **Existing datasets trade off divergence against diversity**—filtering existing mixtures for delta prompts ($\Delta > 0$) reduces diversity by up to 29.4% as measured by E-Vendi.

divergence. While controlling for the data scale ($N = 0.5k, \cdots, 10k$), we sample 3 different types of subsets with unique divergence characteristics: (1) *delta prompts* ($\Delta > 0$ only), (2) *zero-delta prompts*: ($\Delta = 0$ only), and (3) a random subset. During this process, we find that data diversity confounds the effect of answer divergence—*e.g.,* naive sampling of prompts with non-zero Δ significantly degrades the diversity of the subset, thus obscuring the effect of higher divergence. We therefore employ a balanced-sampling algorithm introduced by Jung et al. [23] to match the diversity of all subsets in the embedding space[1]. Finally, we run on-policy distillation on each subset using Qwen3-VL-235B-Thinking as the teacher and Qwen3-VL-8B-Thinking as the student, and average the performance across 4 standard benchmarks—CharXiv, InfoVQA, ChartQAPro, SEEDBench2-Plus [57, 46, 44, 33]. For further experimental details, see §A.

Answer divergence is critical for effective scaling. The results are shown in Fig. 2 (Left). We first observe that even with just 500 prompts from any subset, models quickly improve by $\sim 1\%$ in average performance—we posit that this phenomenon is likely a baseline improvement, enabled by the exposure to dense token feedback from a strong teacher regardless of the nature of prompts. But after this point, the three subsets show strikingly different scaling—*zero-delta prompts* ($\Delta = 0$) quickly saturate, with stagnating performance even when using 20 times more prompts than the baseline point. With *delta prompts* ($\Delta > 0$), on the contrary, performance scales log-linearly with dataset size, showing consistent improvement with more prompts. Overall, these results demonstrate that answer divergence is critical for reliable scaling of distillation: *delta prompts* drive continued learning, while *zero-delta prompts* exhaust their value almost immediately.

2.4 Do off-the-shelf datasets provide enough divergence?

Majority of prompts in existing datasets are zero-delta. We additionally investigate whether existing datasets offer enough divergence for distillation. Using the same datasets and teacher-student pair as in §2.3, we plot the answer divergence of each prompt in a quantized distribution in Fig. 3 and §C.1. Here, we unexpectedly find that the majority of problems in existing datasets are effectively zero-delta—the two models produce homogeneous answer distribution across 16 rollouts, for up to 69% of the problems. The result explains the random subset's scaling behavior in Fig. 2 (Left): as majority of samples in existing datasets are indeed zero delta, a random subset will highly likely inherit this property, leading to saturated performance similar to zero-delta subset.

Divergence is often at odds with diversity. The above observation motivates us to ask a natural question—can't we just use the remaining 31% of the prompts for distillation? Here, we find that naively filtering zero-delta prompts is suboptimal, as the process leads to significant degradation in prompt diversity. In Fig. 2 (Right), we compare the diversity of the distinct subsets with E-Vendi [23], a diversity metric known to better correlate with the trained model's generalization[2]. Across all datasets, we find that delta prompts exhibit significantly lower E-Vendi than random or zero-delta subsets, implying that the divergence in these datasets stems from a narrower set of systematic failure modes rather than a broad range of capability gaps.

[1]We stop at $N = 10k$ primarily due to data availability: after filtering the existing mixture for delta prompts (which removes $\sim$70% of the pool) and then matching diversity with balanced-sampling, the largest subset we can sample from the 139k pool is on the order of 10k.

[2]Following Jung et al. [23], we use gte-Qwen-7B-Instruct [36] as the embedding model for E-Vendi.

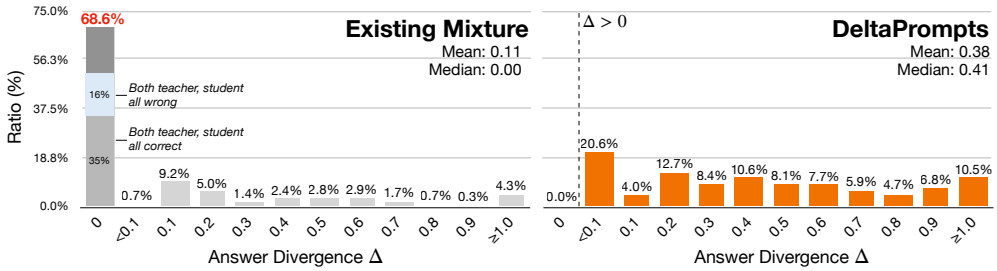


Figure 3: (Left) Divergence distribution of an off-the-shelf chart / document reasoning data mixture. **69% of the existing prompts are zero-delta—*i.e.,* the teacher and student generate the exact same answer distribution—producing minimal learning value.** (Right) Divergence distribution of our synthesized dataset, DELTAPROMPTS. **Our pipeline not only removes all zero-delta prompts, but also produces a well-dispersed distribution of non-zero divergence.**

3 Synthesizing DELTAPROMPTS

Our analyses in §2 motivate an alternative use of existing data—rather than training on the prompts directly, we repurpose them as seeds to synthesize better ones. In principle, one might hope to sidestep this issue by simply scaling the seed pool: if 31% of prompts are delta prompts, a larger dataset would yield a larger filtered subset. In practice, however, high-quality multimodal reasoning data remains scarce [73, 50], and more fundamentally, filtering alone cannot resolve the diversity-divergence tension we identify in §2.4—the filtered subset is always a narrower slice of the original distribution. Below, we introduce two scalable methods that repurpose existing data as seeds, both leveraging the teacher model itself to generate divergence-inducing prompts at scale.

3.1 Synthesis Pipeline

Seed-guided Generation As the most straightforward approach, we generate novel problems by prompting the teacher with existing prompts as few-shot examples. We first sample a set of N image-prompt pairs $(x_i, I_i)_{i=1}^{N}$ from a seed dataset. Given a new image I' (either drawn from the same dataset or a separate image source), we prompt the model to create a novel problem x' for I':

$$x' \sim \pi_T\left(\cdot|(x_1, I_1), \cdots, (x_N, I_N), I'\right) \tag{5}$$

To ensure the generated problems are well-defined, we additionally apply consistency-based filtering: for each generated x', we sample K independent responses from the teacher and retain only those where the teacher produces a consistent answer across majority of responses. This allows us to remove ill-posed or ambiguous problems from our pool that cannot even be reliably solved by the data generator. Seed-guided generation on its own does not explicitly target higher divergence; we subsequently employ rejection stage below to filter out zero-delta candidates.

Skill-guided Generation Despite the scalability of seed-guided generation, it does not explicitly search for the types of problems that teacher and student diverge. Inspired by earlier works on skill-guided data curation [9, 26, 42], we introduce a complementary skill-guided generation that strategically samples more from the student's failure modes.

The process proceeds in two steps. We first perform skill extraction: for each prompt x with non-zero Δ in our candidate pool, we sample a corresponding student trajectory τ_θ and a teacher trajectory τ_T such that the two trajectories disagree in final answers, *i.e.,* $\mathcal{E}(\tau_\theta) \neq \mathcal{E}(\tau_T)$. Skills are then extracted by contrasting these two trajectories—we prompt the teacher to identify the reasoning skill present in τ_T but absent in τ_θ, in a form of short natural language phrase (*e.g., interpolating logarithmic scales in a chart*). Iterating the extraction over all prompts x in our pool, we obtain a set of discriminative skills that were missing in the student but were exhibited by the teacher. Finally, we utilize this skill pool by further conditioning the data generator on K sampled skills:

$$x' \sim \pi_T\left(\cdot|(x_1, I_1), \cdots, (x_N, I_N), s_1, \cdots, s_K, I'\right) \tag{6}$$

The same consistency-based filtering is applied to ensure solvability of the generated prompts. For few-shot demonstrations, we use the synthesized prompts from the seed-guided generation—naturally forming a two-stage pipeline—and analyze the impact of this choice in §4.1.2.

Rejection for Answer Divergence Finally, we measure the answer divergence $\Delta(x)$ for each generated prompt and reject all zero-delta candidates. Viewed broadly, the role of synthesis in our pipeline is to produce an abundance of diverse candidates, so that we can afford to be selective—retaining only the prompts with non-zero divergence. In §4.1, we show that this framework not only yields a substantially higher fraction of delta prompts than filtering existing datasets (57% vs. 31%), but also retains higher diversity after filtering.

3.2 DELTAPROMPTS

Using our pipeline, we generate DELTAPROMPTS, a dataset of 200k divergence-inducing prompts for chart & document reasoning and high-resolution, perception-centric reasoning. We leverage state-of-the-art VLMs—Qwen3-VL-8B-Thinking and Qwen3-VL-235B-Thinking—as the target model pair for distillation.

We set the number of few-shot examples $N = 1$ and number of sampled skills $K = 10$, based on our preliminary observation that the number of few-shot examples has minimal impact on generation quality, while more sampled skills improve prompt diversity. We utilize publicly available data sources as both our seed prompts and image sources, and run both seed-guided and skill-guided generation, producing 110k and 90k prompts respectively in our two-stage process. We provide further details on our synthesis pipeline including the specific prompts and data sources in §A.2.

4 Experimental Results

4.1 Data Quality Analysis

4.1.1 Does DELTAPROMPTS improve beyond the seed data it originated from?

Does our synthesis pipeline actually improve the data quality beyond the seed datasets it starts from? In Figure 3 and Table 1, we validate this by comparing DELTAPROMPTS against existing datasets, in terms of their answer divergence and diversity. Compared to the existing datasets in Figure 5, DELTAPROMPTS substantially improves answer divergence—our dataset is not only clear of any zero-delta prompts (as expected from the rejection protocol), but its distribution is also far more dispersed across the non-negative range. In addition, we observe consistent improvements in diversity as measured by E-Vendi, both before and after filtering for non-zero answer divergence (Table 1). Overall, the result confirms that our synthesis approach improves divergence and diversity beyond the seed data, rather than producing a homogeneous augmentation to the existing prompts.

Dataset	Divergence		E-Vendi	
	Mean (Median)	Var.	Pre-filter	Post-filter
Existing Mixture	0.11 (0.00)	0.063	141.67	108.19
DELTAPROMPTS	0.38 (0.41)	0.095	162.46	136.38

Table 1: Comparison of data quality between existing mixture and DELTAPROMPTS. **DELTAPROMPTS shows significantly higher divergence and diversity than the existing mixture it originated from.**

4.1.2 Understanding Seed-guided vs. Skill-guided Generation

We also compare the effectiveness of the two stages in our pipeline—seed-guided generation and skill-guided generation. We specifically measure two metrics: the yield of delta prompts (*i.e.,* the fraction of prompts retained after rejection) and the diversity of those filtered prompts.

Synthesis Pipeline	Yield	E-Vendi
Sampling from Existing Mixture	0.31	108.19
Seed-guided Generation	0.44	128.64
Skill-guided Generation w/ seed-guided prompts	0.57	142.47
Skill-guided Generation w/ off-the-shelf prompts	0.46	127.33

Table 2: Comparison of synthesis pipelines. **Skill-guided generation performs best**, with seed-guided outputs as its most effective starting point.

In Table 2, we find that skill-guided generation leads to better yield and higher diversity compared to seed-guided generation, aligning with our intuition that the extracted skills can better capture the capability gap between the two models. However, when we directly run the skill-guided generation with off-the-shelf prompts as seeds (*i.e.,* bypassing Stage 1), both metrics fall back to the level of seed-guided generation. This suggests that the superiority of our second stage builds on the

Model	$\text{CharXiv}_{\text{RQ}}$	$\text{InfoVQA}_{\text{val}}$	$\text{ChartQAPro}_{\text{clean}}$	SEEDBench2-Plus	Average
Qwen3-VL-8B-Thinking [4]	53.14 (0.07)	85.33 (0.12)	63.33 (0.11)	69.87 (0.10)	67.92 (0.04)
GLM-4.1V-9B-Thinking [54]	53.76 (0.06)	85.04 (0.09)	66.04 (0.13)	73.03 (0.10)	69.47 (0.04)
Bee-8B-RL [73]	57.00 (0.08)	69.26 (0.10)	65.96 (0.11)	68.55 (0.11)	65.19 (0.06)
REVisual-R1 [7]	41.50 (0.07)	80.01 (0.13)	59.93 (0.12)	70.14 (0.12)	62.90 (0.07)
Vision-R1-7B [18]	37.28 (0.07)	78.79 (0.09)	44.14 (0.10)	69.57 (0.14)	57.45 (0.05)
ARES-RL-7B [6]	43.22 (0.08)	81.29 (0.07)	63.93 (0.15)	68.77 (0.12)	64.30 (0.06)
Qwen-8B-DeltaThinker	**60.52 (0.09)**	**86.53 (0.11)**	71.09 (0.09)	73.43 (0.11)	72.89 (0.05)
GLM-9B-DeltaThinker	59.10 (0.10)	85.78 (0.09)	**72.55 (0.12)**	**74.81 (0.13)**	**73.06 (0.07)**

Model	RealWorldQA	$\text{HRBench}_{\text{4K}}$	V*	VisualProbe			Average
				Easy	Medium	Hard	
Qwen3-VL-8B-Thinking	73.07 (0.08)	72.22 (0.05)	76.96 (0.16)	34.75 (0.23)	13.61 (0.17)	10.86 (0.44)	46.92 (0.15)
GLM-4.1V-9B-Thinking	72.19 (0.08)	73.75 (0.05)	79.58 (0.19)	39.01 (0.27)	17.91 (0.15)	16.04 (0.39)	49.75 (0.18)
Bee-8B-RL	72.03 (0.10)	60.15 (0.09)	58.12 (0.16)	26.95 (0.23)	14.93 (0.15)	10.40 (0.48)	40.43 (0.16)
REVisual-R1	64.36 (0.12)	71.88 (0.05)	69.11 (0.16)	43.97 (0.23)	20.15 (0.17)	18.87 (0.44)	48.06 (0.19)
Vision-R1-7B	67.58 (0.10)	75.38 (0.05)	81.15 (0.16)	36.03 (0.23)	16.40 (0.15)	15.47 (0.39)	48.67 (0.13)
ARES-RL-7B	66.67 (0.08)	72.13 (0.05)	71.20 (0.19)	38.94 (0.27)	18.50 (0.15)	11.24 (0.44)	46.45 (0.17)
Qwen-8B-DeltaThinker	75.82 (0.08)	77.25 (0.09)	82.73 (0.19)	**51.77 (0.23)**	20.15 (0.15)	16.98 (0.44)	54.12 (0.15)
GLM-9B-DeltaThinker	**77.04 (0.10)**	**80.25 (0.09)**	**84.25 (0.16)**	50.04 (0.27)	**21.55 (0.15)**	**20.04 (0.39)**	**55.53 (0.13)**

Table 3: On-policy distillation result on chart & document reasoning (Top) and real-world perception-centric reasoning (Bottom). **DELTAPROMPTS enables up to 15% relative improvement on average across benchmarks, for both the target teacher-student pair (Qwen-8B-DeltaThinker) and an entirely new model family (GLM-9B-DeltaThinker).** We perform 5 independent runs across all baselines and report the standard error.

expanded data pool from seed-guided generation, and therefore, the two stages complement each other rather than one method strictly subsuming another.

4.2 On-Policy Distillation with DELTAPROMPTS

4.2.1 Experimental Setup

Next, we validate DELTAPROMPTS by running on-policy distillation with our target teacher-student pair (Qwen3-VL-235B-Thinking and Qwen3-VL-8B-Thinking). We implement the algorithm using verl [53], and use vllm [31] to run an external server that returns teacher model's log probability given a sequence. We train the student for 1 epoch on DELTAPROMPTS, and learn a single unified model for both chart & document reasoning and high-resolution perception-centric reasoning. We refer to this model as **Qwen3-8B-DeltaThinker**.

For evaluation, we use 10 widely-used benchmarks across the two domains, as shown in Table 3. We compare against both off-the-shelf reasoning models [4, 54] and the strongest open-recipe models from the community [6, 7, 18, 73]. Notably, we find that reported results from the baselines often depend on differing inference protocols (*e.g.,* the choice of LLM judge for answer verification [12]) and incompatible versions of evaluation frameworks, making direct comparison with published numbers unreliable. For fair comparison, we rerun all models across all benchmarks for 5 independent runs and report the standard error. We refer readers to §A for further experimental details.

4.2.2 Main Result

The results in Table 3 show that **Qwen3-8B-DeltaThinker significantly improves over Qwen3-VL-8B-Thinking—7% relative improvement in chart & document reasoning and 15% in real-world perception-centric reasoning, averaged across benchmarks**. These improvements are particularly notable given that our base model, Qwen3-VL-8B-Thinking, is itself a highly optimized reasoning model that already outperforms all open-recipe baselines—yet DeltaThinker still yields substantial gains on the hardest benchmarks such as CharXiv and VisualProbe. Compared to the open-recipe models, which employ staged training pipelines and order-of-magnitude more training samples (*e.g.,* Bee-8B-RL aggregates 15M samples across 150 sources with two-stage SFT followed by RL), our approach achieves stronger results with significantly less data, demonstrating the sample efficiency of targeted prompt synthesis. These results suggest that distillation on a high-quality set of prompts targeted for the student could outweigh heuristic aggregation of existing datasets.

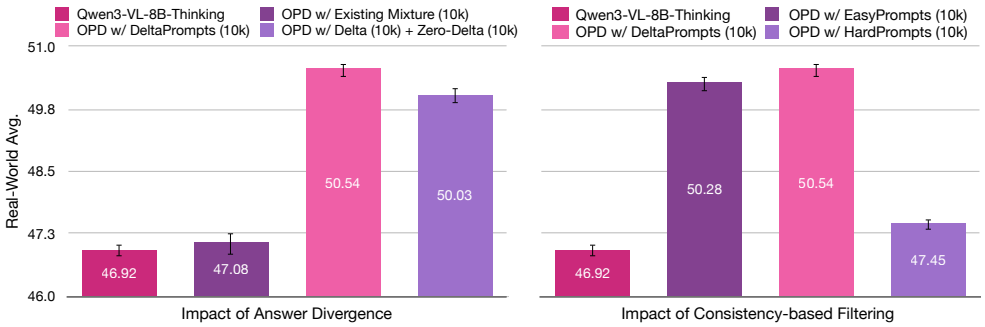


Figure 4: (Left) Ablation on answer divergence. **DELTAPROMPTS leads to the best results even under a data size-controlled setup, while adding additional zero-delta data does not help.** (Right) Ablation on consistency-based filtering. **Further optimizing for teacher consistency (EasyPrompts) yields no gain, while training on prompts the teacher cannot reliably solve (HardPrompts) degrades performance.**

4.2.3 DELTAPROMPTS Generalizes Better to Out-of-Domain Tasks

Motivated by prior reports that OPD mitigates catastrophic forgetting in supervised fine-tuning [52], we examine whether DeltaThinker maintains the strong performance of the base model on out-of-domain tasks such as math, puzzle and STEM reasoning.

The results on MathVista, LogicVista and MMMU-Pro [43, 62, 70] in Table 5 present an unexpected finding: **our model not only maintains, but *improves* over the base model across all three benchmarks.** Importantly, we find that such gains are not universal to all OPD runs—when running OPD with 10k subset of existing data, the model sees early degradation on math and puzzle, whereas DELTAPROMPTS, on the same scale, either maintains or improves performance. This demonstrates that out-of-domain generalization depends not only on the choice of learning algorithm, but equally on the divergence and diversity of prompts it trains on.

4.2.4 Does Answer Divergence Really Matter?

Despite the compelling performance of DELTAPROMPTS, it remains a question whether the gains are truly driven by improving divergence in the training prompts, or simply by distilling a stronger teacher on more data. We analyze this via a series of controlled training runs, where we run distillation with (1) a random-sampled 10k subset of existing mixture, (2) a subset of DELTAPROMPTS with the same size, and (3) a 20k subset comprising 10k samples from DELTAPROMPTS and 10k zero-delta prompts rejected in our pipeline. The result in Figure 4 (Left) shows that even in this strictly controlled setup, OPD with DELTAPROMPTS shows clear improvement over both the base model and direct OPD on the existing mixture. Furthermore, we find that augmenting DELTAPROMPTS with zero-delta prompts does not help, but degrade the performance—further confirming that zero-delta prompts act as noise rather than useful signal, even when they double the training set size.

4.2.5 Does Consistency-based Filtering Matter?

During the synthesis of DELTAPROMPTS, we improve the solvability of generated prompts by sampling 16 teacher responses per prompt and retaining those with teacher avg@16 ≥ 0.5, *i.e.,* the teacher produces a consistent answer across majority of responses. We ablate the impact of this consistency-based filtering by defining two subsets of real-world perception-centric prompts: (1) *EasyPrompts*—prompts with teacher avg@16 $= 1$, *i.e.,* teacher produces a consistent answer across all 16 responses, and (2) *HardPrompts*—prompts with teacher avg@16 < 0.5, *i.e.,* even the teacher fails to produce consistent answer for majority of responses. As with DELTAPROMPTS, we ensure that all subsets have no zero-delta prompts, strictly isolating the effect of teacher consistency. Controlling for data scale (10k), we run distillation on each subset and compare average performance.

In Fig. 4 (Right), we find that further optimizing for teacher consistency (avg@16 $= 1$) does not help, with *EasyPrompts* marginally underperforming DELTAPROMPTS. The result validates our choice of a permissive consistency threshold, which increases the yield of the pipeline without sacrificing

Model	Chart & Document Avg.	Real-World Avg.	MathVista$_{testmini}$	LogicVista	MMLU-Pro$_{vision}$
Qwen3-VL-8B-Instruct	66.85 (0.05)	47.82 (0.13)	77.24 (0.25)	57.21 (0.29)	42.14 (0.23)
+ SFT on Existing Mixture	67.52 (0.04)	49.50 (0.18)	74.50 (0.20)	55.05 (0.23)	41.10 (0.20)
+ **SFT on DELTAPROMPTS**	**71.57 (0.05)**	55.04 (0.17)	75.20 (0.25)	57.27 (0.23)	**47.74 (0.16)**
+ **OPD on DELTAPROMPTS**	69.32 (0.06)	**55.17 (0.15)**	**77.70 (0.24)**	**58.87 (0.19)**	46.34 (0.23)

Table 4: Distilling a non-reasoning model (Qwen3-VL-8B-Instruct) using DELTAPROMPTS. **SFT on DELTAPROMPTS forgets less than the existing mixture on out-of-domain tasks, while staying competitive to on-policy distillation on in-domain tasks.** We generate 4 teacher trajectories per prompt without additional filtering.

downstream performance. In contrast, training on *HardPrompts* degrades performance—aligning with our intuition that teacher feedback becomes noisier with overly hard or ill-defined prompts.

4.3 Generalization to a Novel Teacher-Student Pair

Our synthesis pipeline aims to expose the gap between a specific pair of teacher and student, raising a natural question: can we reuse DELTAPROMPTS for a new pair of models without re-running the full pipeline? We test this by distilling with an entirely different model family—GLM-4.5V as teacher and GLM-4.1V-9B-Thinking as student. Instead of regenerating prompts from scratch, we directly reuse DELTAPROMPTS, simply filtering for non-zero answer divergence under the new pair. This lightweight rejection step retains 154k of the original 200k prompts.

The results are presented in the bottom rows of Table 3. Consistent with our findings on Qwen3-VL, DELTAPROMPTS yields coherent improvements across all benchmarks for the new pair of models. Overall, the results imply that (1) our prompts capture generalizable capability gaps that transfer across model families, rather than overfitting to the idiosyncrasies of a specific teacher-student pair, and (2) the rejection step—filtering for non-zero divergence—is sufficient to adapt the dataset, making our pipeline practical beyond the original models it was designed for.

4.4 Reusing DELTAPROMPTS for Off-Policy Fine-Tuning

Beyond on-policy distillation, we investigate whether DELTAPROMPTS can serve as an off-policy resource for standard supervised fine-tuning (SFT). We generate 4 trajectories per prompt using Qwen3-VL-235B-Thinking and train a non-reasoning model, Qwen3-VL-8B-Instruct on this data via SFT. We did not perform any additional curation such as trajectory-level filtering, in order to simulate the most basic usage of this dataset focusing on prompt quality. We also run OPD with the same student model for direct comparison.

The results in Table 4 show that for in-domain tasks (Chart & Document Avg. and Real-World Avg.), SFT on DELTAPROMPTS substantially outperforms both the base model and SFT on existing mixture, matching or exceeding direct OPD on the non-reasoning model. On out-of-domain benchmarks, SFT on DELTAPROMPTS does show degraded performance compared to OPD, consistent with prior observations that SFT is prone to forgetting [8, 40]. However, when compared to SFT on existing mixture, DELTAPROMPTS notably generalizes better to out-of-domain tasks while still delivering stronger in-domain performance—suggesting that the quality of the training prompts partially compensates for the algorithmic limitations of SFT.

5 Conclusion

We identify a critical data bottleneck in multimodal distillation: the prevalence of zero-delta prompts in standard datasets that impede effective learning of the student model. We establish *answer divergence* as a principled metric for quantifying the capability gap between the teacher and student. To systematically source high-divergence prompts, we propose a two-stage synthesis pipeline that performs targeted rejection sampling for the student's failure modes. Our resulting dataset, DELTAPROMPTS, yields substantial and highly transferable gains across diverse reasoning benchmarks, and generalizes better to out-of-domain tasks, even when distilling into highly optimized student models.

References

[1] D. Acuna, C.-H. H. Yang, Y. Deng, J. Jung, X. Lu, P. Ammanabrolu, H. Kim, Y.-H. Liao, and Y. Choi. Long grounded thoughts: Synthesizing visual problems and reasoning chains at scale, 2026.

[2] R. Agarwal, N. Vieillard, Y. Zhou, P. Stanczyk, S. Ramos, M. Geist, and O. Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2024.

[3] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[4] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, W. Ge, Z. Guo, Q. Huang, J. Huang, F. Huang, B. Hui, S. Jiang, Z. Li, M. Li, M. Li, K. Li, Z. Lin, J. Lin, X. Liu, J. Liu, C. Liu, Y. Liu, D. Liu, S. Liu, D. Lu, R. Luo, C. Lv, R. Men, L. Meng, X. Ren, X. Ren, S. Song, Y. Sun, J. Tang, J. Tu, J. Wan, P. Wang, P. Wang, Q. Wang, Y. Wang, T. Xie, Y. Xu, H. Xu, J. Xu, Z. Yang, M. Yang, J. Yang, A. Yang, B. Yu, F. Zhang, H. Zhang, X. Zhang, B. Zheng, H. Zhong, J. Zhou, F. Zhou, J. Zhou, Y. Zhu, and K. Zhu. Qwen3-vl technical report, 2025.

[5] E. Borisova, N. Rauscher, and G. Rehm. SciVQA 2025: Overview of the first scientific visual question answering shared task. In T. Ghosal, P. Mayr, A. Singh, A. Naik, G. Rehm, D. Freitag, D. Li, S. Schimmler, and A. De Waard, editors, *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 182–210, Vienna, Austria, July 2025. Association for Computational Linguistics.

[6] J.-S. Byun, J. Chun, J. Kil, and A. Perrault. Ares: Alternating reinforcement learning and supervised fine-tuning for enhanced multi-modal chain-of-thought reasoning through diverse ai feedback, 2024.

[7] S. Chen, Y. Guo, Z. Su, Y. Li, Y. Wu, J. Chen, J. Chen, W. Wang, X. Qu, and Y. Cheng. Advancing multimodal reasoning: From optimized cold start to staged reinforcement learning, 2026.

[8] T. Chu, Y. Zhai, J. Yang, S. Tong, S. Xie, D. Schuurmans, Q. V. Le, S. Levine, and Y. Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training, 2025.

[9] A. Didolkar, A. Goyal, N. R. Ke, S. Guo, M. Valko, T. Lillicrap, D. Rezende, Y. Bengio, M. Mozer, and S. Arora. Metacognitive capabilities of llms: An exploration in mathematical problem solving, 2024.

[10] Y. Ding, S. Luo, H. Chung, and S. C. Han. Pdf-vqa: A new dataset for real-world vqa on pdf documents. In G. De Francisci Morales, C. Perlich, N. Ruchansky, N. Kourtellis, E. Baralis, and F. Bonchi, editors, *Machine Learning and Knowledge Discovery in Databases: Applied Data Science and Demo Track*, pages 585–601, Cham, 2023. Springer Nature Switzerland.

[11] H. Duan. RealWorldQA, What's New? https://huggingface.co/blog/KennyUTC/realworldqa, 2024.

[12] H. Duan, J. Yang, Y. Qiao, X. Fang, L. Chen, Y. Liu, X. Dong, Y. Zang, P. Zhang, J. Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11198–11201, 2024.

[13] Y. Gu, L. Dong, F. Wei, and M. Huang. Minillm: On-policy distillation of large language models, 2026.

[14] Y. Hao, Z. Li, L. Sun, W. Wang, N. Yi, S. Song, C. Qin, M. Zhou, Y. Zhan, and X. Lang. Driveaction: A benchmark for exploring human-like driving decisions in vla models, 2025.

[15] C. He, Y. Chen, C. Xiao, X. Han, and L. Wen. Student-in-the-loop chain-of-thought distillation via generation-time selection, 2026.

[16] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015.

[17] H. Huang, J. Song, Y. Zhang, and P. Ren. Selectkd: Selective token-weighted knowledge distillation for llms, 2025.

[18] W. Huang, B. Jia, Z. Zhai, S. Cao, Z. Ye, F. Zhao, Z. Xu, X. Tang, Y. Hu, and S. Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models, 2026.

[19] J. Hübotter, F. Lübeck, L. Behric, A. Baumann, M. Bagatella, D. Marta, I. Hakimi, I. Shenfeld, T. K. Buening, C. Guestrin, and A. Krause. Reinforcement learning via self-distillation, 2026.

[20] T. Jain, C. Lennan, Z. John, and D. Tran. Imagededup. https://github.com/idealo/imagededup, 2019.

[21] Y. Jiang, C. Chan, M. Chen, and W. Wang. Lion: Adversarial distillation of proprietary large language models, 2023.

[22] W. Jin, T. Min, Y. Yang, S. R. Kadhe, Y. Zhou, D. Wei, N. Baracaldo, and K. Lee. Entropy-aware on-policy distillation of language models, 2026.

[23] J. Jung, S. Han, X. Lu, S. Hallinan, D. Acuna, S. Prabhumoye, M. Patwary, M. Shoeybi, B. Catanzaro, and Y. Choi. Prismatic synthesis: Gradient-based data diversification boosts generalization in LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[24] J. Jung, P. West, L. Jiang, F. Brahman, X. Lu, J. Fisher, T. Sorensen, and Y. Choi. Impossible distillation: from low-quality model to high-quality dataset & model for summarization and paraphrasing, 2024.

[25] S. Jung, S. Yoon, D. Kim, and H. Lee. Todi: Token-wise distillation via fine-grained divergence control, 2025.

[26] S. Kaur, S. Park, A. Goyal, and S. Arora. Instruct-skillmix: A powerful pipeline for llm instruction tuning, 2024.

[27] J. Kim, X. Luo, M. Kim, S. Lee, D. Kim, J. Jeon, D. Li, and Y. Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms?, 2026.

[28] Y. Kim and A. M. Rush. Sequence-level knowledge distillation. In J. Su, K. Duh, and X. Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[29] J. Ko, T. Chen, S. Kim, T. Ding, L. Liang, I. Zharkov, and S.-Y. Yun. Distillm-2: A contrastive approach boosts the distillation of llms, 2025.

[30] J. Ko, S. Kim, T. Chen, and S.-Y. Yun. Distillm: Towards streamlined distillation for large language models, 2024.

[31] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[32] X. Lai, J. Li, W. Li, T. Liu, T. Li, and H. Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search, 2025.

[33] B. Li, Y. Ge, Y. Chen, Y. Ge, R. Zhang, and Y. Shan. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension, 2024.

[34] L. Li, Y. Lin, S. Ren, P. Li, J. Zhou, and X. Sun. Dynamic knowledge distillation for pre-trained language models, 2021.

[35] L. Li, Y. Wang, R. Xu, P. Wang, X. Feng, L. Kong, and Q. Liu. Multimodal ArXiv: A dataset for improving scientific comprehension of large vision-language models. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14369–14387, Bangkok, Thailand, aug 2024. Association for Computational Linguistics.

[36] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang. Towards general text embeddings with multi-stage contrastive learning, 2023.

[37] H. Lin, Z. Liu, Y. Zhu, C. Qin, J. Lin, X. Shang, C. He, W. Zhang, and L. Wu. Mmfinereason: Closing the multimodal reasoning gap via open data-centric methods, 2026.

[38] J. Liu, J. Wu, X. Pan, G. Cheung, S. Ma, and C. Tao. Air: Post-training data selection for reasoning via attention head influence, 2025.

[39] L. Liu and M. Zhang. Less is more: Selective reflection for compatible and efficient knowledge distillation in large language models, 2025.

[40] A. Lu, T. Feng, H. Yuan, W. Li, and Y. Sun. Why does rl generalize better than sft? a data-centric perspective on vlm post-training, 2026.

[41] K. Lu and T. M. Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. https://thinkingmachines.ai/blog/on-policy-distillation.

[42] K. Lu, H. Yuan, Z. Yuan, R. Lin, J. Lin, C. Tan, C. Zhou, and J. Zhou. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models, 2023.

[43] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts, 2024.

[44] A. Masry, M. S. Islam, M. Ahmed, A. Bajaj, F. Kabir, A. Kartha, M. T. R. Laskar, M. Rahman, S. Rahman, M. Shahmohammadi, M. Thakkar, M. R. Parvez, E. Hoque, and S. Joty. ChartQAPro: A more diverse and challenging benchmark for chart question answering. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19123–19151, Vienna, Austria, July 2025. Association for Computational Linguistics.

[45] A. Masry, D. X. Long, J. Q. Tan, S. Joty, and E. Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning, 2022.

[46] M. Mathew, V. Bagal, R. P. Tito, D. Karatzas, E. Valveny, and C. V. Jawahar. Infographicvqa, 2021.

[47] OpenAI. Gpt-4o mini: advancing cost-efficient intelligence, 2024.

[48] SamaAI. Huggingface Hub: SamaAI/sama-drives-california. https://huggingface.co/datasets/SamaAI/sama-drives-california, 2023.

[49] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[50] G. Sarch, L. Cai, Q. Wang, H. Wu, D. Chen, and Z. Liu. Vero: An open rl recipe for general visual reasoning, 2026.

[51] W. Shen, J. Pei, Y. Peng, X. Song, Y. Liu, J. Peng, H. Sun, Y. Hao, P. Wang, J. Zhang, and Y. Zhou. Skywork-r1v3 technical report, 2025.

[52] I. Shenfeld, M. Damani, J. Hübotter, and P. Agrawal. Self-distillation enables continual learning, 2026.

[53] G. Sheng, C. Zhang, Z. Ye, X. Wu, W. Zhang, R. Zhang, Y. Peng, H. Lin, and C. Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[54] V. Team, W. Hong, W. Yu, X. Gu, G. Wang, G. Gan, H. Tang, J. Cheng, J. Qi, J. Ji, L. Pan, S. Duan, W. Wang, Y. Wang, Y. Cheng, Z. He, Z. Su, Z. Yang, Z. Pan, A. Zeng, B. Wang, B. Chen, B. Shi, C. Pang, C. Zhang, D. Yin, F. Yang, G. Chen, H. Li, J. Zhu, J. Chen, J. Xu, J. Xu, J. Chen, J. Lin, J. Chen, J. Wang, J. Chen, L. Lei, L. Gong, L. Pan, M. Liu, M. Xu, M. Zhang, Q. Zheng, R. Lyu, S. Tu, S. Yang, S. Meng, S. Zhong, S. Huang, S. Zhao, S. Xue, T. Zhang, T. Luo, T. Hao, T. Tong, W. Jia, W. Li, X. Liu, X. Zhang, X. Lyu, X. Zhang, X. Fan,

X. Huang, Y. Xue, Y. Wang, Y. Wang, Y. Wang, Y. An, Y. Du, Y. Huang, Y. Niu, Y. Shi, Y. Wang, Y. Wang, Y. Yue, Y. Li, Y. Liu, Y. Zhang, Y. Wang, Y. Zhang, Z. Xue, Z. Du, Z. Hou, Z. Wang, P. Zhang, D. Liu, B. Xu, J. Li, M. Huang, Y. Dong, and J. Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2026.

[55] J. Wang, E. Briakou, H. Dadkhahi, R. Agarwal, C. Cherry, and T. Cohn. Don't throw away data: Improving sequence knowledge distillation with minimum bayes risk decoding. In *Scaling Self-Improving Foundation Models without Human Supervision*, 2025.

[56] W. Wang, L. Ding, M. Zeng, X. Zhou, L. Shen, Y. Luo, and D. Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models, 2024.

[57] Z. Wang, M. Xia, L. He, H. Chen, Y. Liu, R. Zhu, K. Liang, X. Wu, H. Liu, S. Malladi, A. Chevalier, S. Arora, and D. Chen. Charxiv: Charting gaps in realistic chart understanding in multimodal llms, 2024.

[58] Y. Wen, Z. Li, W. Du, and L. Mou. f-divergence minimization for sequence-level knowledge distillation, 2023.

[59] L. Wiedmann, O. Zohar, A. Mahla, X. Wang, R. Li, T. Frere, L. von Werra, A. R. Gosthipaty, and A. Marafioti. Finevision: Open data is all you need, 2025.

[60] P. Wu and S. Xie. V*: Guided visual search as a core mechanism in multimodal llms, 2023.

[61] T. Wu, C. Tao, J. Wang, R. Yang, Z. Zhao, and N. Wong. Rethinking Kullback-Leibler divergence in knowledge distillation for large language models. In O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, and S. Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5737–5755, Abu Dhabi, UAE, Jan. 2025. Association for Computational Linguistics.

[62] Y. Xiao, E. Sun, T. Liu, and W. Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts, 2024.

[63] Y. Xu, H. Sang, Z. Zhou, R. He, Z. Wang, and A. Geramifard. Tip: Token importance in on-policy distillation, 2026.

[64] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu. Qwen3 technical report, 2025.

[65] C. Yang, C. Qin, Q. Si, M. Chen, N. Gu, D. Yao, Z. Lin, W. Wang, J. Wang, and N. Duan. Self-distilled rlvr, 2026.

[66] Y. Yang, X. He, H. Pan, X. Jiang, Y. Deng, X. Yang, H. Lu, D. Yin, F. Rao, M. Zhu, B. Zhang, and W. Chen. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization, 2025.

[67] Y. Yang, M. Lai, W. Zhao, X. Fan, Z. Xi, M. Wu, C. Huang, J. Zhao, H. Lv, J. Tong, Y. Zhou, Y. Zou, Q. Guo, T. Gui, Q. Zhang, and X. Huang. Which reasoning trajectories teach students to reason better? a simple metric of informative alignment, 2026.

[68] T. Ye, L. Dong, Z. Chi, X. Wu, S. Huang, and F. Wei. Black-box on-policy distillation of large language models, 2026.

[69] T. Ye, L. Dong, X. Wu, S. Huang, and F. Wei. On-policy context distillation for language models, 2026.

[70] X. Yue, T. Zheng, Y. Ni, Y. Wang, K. Zhang, S. Tong, Y. Sun, B. Yu, G. Zhang, H. Sun, Y. Su, W. Chen, and G. Neubig. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark, 2025.

[71] D. Zhang, J. Li, S. Wang, W. Wang, G. Chen, H. Zhang, S. Diao, M. Liu, X. Lu, J. Jung, J. Hu, K. Sapra, W. Ouyang, A. Tao, Y. Choi, J. Kautz, G. Liu, Y. Dong, and Z. Yu. Tinyeye: Sharpening visual reasoning of tiny models with offline policy optimization, 2026.

[72] R. Zhang, R. M. S. Khan, Z. Tan, D. Li, S. Wang, and T. Chen. The quest for efficient reasoning: A data-centric benchmark to cot distillation, 2026.

[73] Y. Zhang, B. Ni, X.-S. Chen, H.-R. Zhang, Y. Rao, H. Peng, Q. Lu, H. Hu, M.-H. Guo, and S.-M. Hu. Bee: A high-quality corpus and full-stack suite to unlock advanced fully open mllms, 2026.

[74] S. Zhao, Z. Xie, M. Liu, J. Huang, G. Pang, F. Chen, and A. Grover. Self-distilled reasoner: On-policy self-distillation for large language models, 2026.

[75] Z. Zheng, M. Yang, J. Hong, C. Zhao, G. Xu, L. Yang, C. Shen, and X. Yu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning, 2026.

[76] J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, H. Tian, Y. Duan, W. Su, J. Shao, Z. Gao, E. Cui, X. Wang, Y. Cao, Y. Liu, X. Wei, H. Zhang, H. Wang, W. Xu, H. Li, J. Wang, N. Deng, S. Li, Y. He, T. Jiang, J. Luo, Y. Wang, C. He, B. Shi, X. Zhang, W. Shao, J. He, Y. Xiong, W. Qu, P. Sun, P. Jiao, H. Lv, L. Wu, K. Zhang, H. Deng, J. Ge, K. Chen, L. Wang, M. Dou, L. Lu, X. Zhu, T. Lu, D. Lin, Y. Qiao, J. Dai, and W. Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025.

A Experimental Details

A.1 Answer Divergence

In our main experiments, we compute answer divergence using Qwen3-VL-235B-Thinking as the teacher and Qwen3-VL-8B-Thinking as the student. For each prompt, we sample 16 responses from both models, instructing each to end with `Final Answer: [ANSWER]` for easy rule-based extraction. We then use Qwen3-32B [64] as an LLM judge to group semantically equivalent answers, yielding empirical answer distributions $\widehat{P}_T(a|x)$ and $\widehat{P}_\theta(a|x)$. The answer divergence $\widehat{\Delta}$ is computed as the reverse KL between these distributions. When the supports do not fully overlap (*e.g.,* the student never produces an answer that the teacher generates), we apply Laplace smoothing—adding a pseudocount of 1 to all answer categories before normalization—to ensure no answer receives zero probability.

To analyze the impact of answer divergence (§2.3), we create 3 distinct subsets—*delta prompts*, *zero-delta prompts*, and a random subset—from 4 existing datasets: ChartQA, SciVQA, InfoVQA, and arXivQA. We control for prompt diversity using the balanced-sampling approach of Jung et al. [23] (presented in Algorithm 1 for completeness). Specifically, we encode each prompt using an embedding model [36], perform K-means clustering over the representations, then sample in a balanced manner—up-sampling from sparse clusters and down-sampling from dense ones. We tune the number of clusters to match the diversity of the most diverse subset (*zero-delta prompts*). With this process, we reduce the diversity gap between subsets to within 1%, compared to the 29.4% degradation observed under naive sampling (Fig. 2). For all diversity comparisons (Fig. 2, Table 1 and Table 2), we measure the embedding entropy of 10k subset random-sampled from each dataset to remove the effect of the dataset size.

A.2 Generating DELTAPROMPTS

We use our teacher model (Qwen3-VL-235B-Thinking) as the data generator to produce DELTAPROMPTS. The specific prompts used for data generation and answer aggregation are provided in §B. For chart & document reasoning, we use ChartQA [45], InfoVQA [46], arXivQA [35], SciVQA [5] as seed datasets, and for real-world perception-centric reasoning, we use DeepEyes47k [75], DriveAction [14] and VisualProbe$_{\text{train}}$ [32]. We source images for the new prompts from arXivQA, SciVQA, PDF-VQA [10] (chart & document reasoning), and DeepEyes47k, VisualProbe$_{\text{train}}$, Div2k & Flickr2k [3], and sama-drives-california [48] (real-world perception-centric reasoning). We find that these image sources often contain near-duplicate images with minimal differences between frames; we deduplicate each source using perceptual hashing from ImageDedup [20].

To preempt data leakage, we additionally run the deduplication logic against all test images from our evaluation benchmarks (Table 3), filtering out any matches from our collected image pool. We define a contaminated image as one with a hamming distance of less than 10 (after hashing) from any test set image, following the default setup in ImageDedup [20]. Through this process, 3.1% of collected images are flagged—all chart images from arXivQA or SciVQA—some of which are false positives detected due to visual similarity despite presenting different underlying data. We conservatively remove all flagged images from our synthesis pipeline before producing DELTAPROMPTS.

A.3 Training with DELTAPROMPTS

We implement on-policy distillation using verl [53], an open-source framework for online RL with LLMs. We find that a naive implementation that colocates the student and teacher in a sequential training loop significantly degrades the throughput due to the scale of our teacher model (235B parameters). To address this, we run a separate vllm [31] server for the teacher's log-probability computation, enabling $\sim$512 parallel rollouts on a single GB200 node and hence accelerated training. We use 4 GB200 nodes for our training runs.

One practical consideration is that the log probabilities from the same model on the same sequence can differ between HuggingFace Transformers and vllm due to implementation differences. While this is a known issue in the community and the gap is small for our base models (*e.g.,* the average log-probability difference across 1000 test sequences was 0.04 for Qwen3-VL-8B-Thinking), it directly affects the training signal and future implementations should account for this discrepancy.

Algorithm 1 *Higher Diversity Sampling*

Input: Data representation $D \in \mathbb{R}^{|\mathcal{D}| \times d}$, number of clusters k and target subset size N_{target}
Output: Indices of selected subset $S \subseteq \{1, \cdots, |\mathcal{D}|\}$

```
S ← ∅                                          ▷ Initialize the subset.
{c_1, ··· , c_k} = K-means(D)                  ▷ Cluster data. c_i is a set of indices corresponding to cluster i.
while |S| < N_target do
    c ← random-sample({c_1, ··· , c_k}, 1)     ▷ Randomly pick a sampling cluster.
    S_new ← random-sample(c, ⌈N_target / k⌉)   ▷ Sample new samples from the chosen cluster.
    S ← S ∪ S_new                              ▷ Add new samples to the subset.
return S                                       ▷ Return the sampled subset.
```

Throughout our OPD experiments, we set learning rate = 5e-7, batch size = 128, and generate 4 rollouts per prompt. We train for 1 epoch with a maximum response length of 16384. In our SFT experiment, we use learning rate = 5e-5, batch size = 512, and train for 5 epochs.

A.4 Evaluation

We evaluate on 10 widely-used benchmarks across both domains: CharXiv [45], InfoVQA [46], ChartQAPro [44], SEEDBench2-Plus [33], RealWorldQA [11], HRBench [56], V* [60], and VisualProbe Easy / Medium / Hard [32]. We use VLMEvalKit [12] as our evaluation framework, and implement separate evaluation scripts for ChartQAPro and VisualProbe, which are not included in VLMEvalKit.

Throughout the evaluation, we follow VLMEvalKit's protocol: (1) generate outputs from the evaluated model using vllm, (2) verify model-generated answers against the ground truth via an LLM judge, and (3) aggregate accuracy. We use GPT-4o-mini [47] as the judge for all evaluations, following VLMEvalKit's official recommendation. We make two dataset-specific modifications. For InfoVQA, the default metric in VLMEvalKit relies on naive string-overlap scoring, which leads to high false-positive and false-negative rates. Following Zhang et al. [73], we replace the default metric with judge-based accuracy. For ChartQAPro, we find that a small portion of problems are structured as lists of multiple questions, making answer parsing in VLMEvalKit protocol unreliable. We evaluate only single-question problems, referring to this subset as $\text{ChartQAPro}_{\text{clean}}$.

Across all baselines, we use the recommended decoding parameters for each model, and default to Qwen3-VL's setup if not available. For Qwen-8B-DeltaThinker and GLM-9B-DeltaThinker, we use temperature = 0.7, top_p = 0.8, top_k = 20, and max output length = 32768.

B Prompts

Prompt for Seed-guided Generation (Chart & Document Reasoning)

A problem for the first image:
{reference_question}

Above are two images. For the first image, I have given you a reasoning problem that asks about the image. Your task is to create a harder problem for the second image.
- Do not just copy the problem verbatim from the example, and try to be creative.
- The new problem should be about the second image, and it should be similar in complexity or harder than the given problem.
- Make sure your problem leads to an objective answer without any subjectivity.
- It would be great if you could create a more natural problem that incorporates various reasoning skills in a meaningful way, rather than artificially asking for them.
- Try to solve your problem to make sure that it is not ill-defined, ambiguous, or unsolvable.
- Do not specifically refer to the second image as *second image* in your question—the students will only be given one image. Just call it an *image*.
- Do not leak the problem-relevant information in your question statement, when it can be inferred from the image.
- Answer in English.

Only output the final hard problem of yours and nothing else. Your output should be formatted as: *Final Problem: your hard problem.*

Prompt for Seed-guided Generation (Real-world Perception-centric Reasoning)

A problem for the first image:
{reference_question}

Above are two images. For the first image, I have given you a reasoning problem that asks about the image. Your task is to create a harder problem for the second image.
- Do not just copy the problem verbatim from the example, and try to be creative.
- The new problem should be about the second image, and it should be similar in complexity or harder than the given problem.
- Make sure your problem leads to an objective answer without any subjectivity.
- It would be great if you could create a more natural problem that incorporates various reasoning skills in a meaningful way, rather than artificially asking for them. You are encouraged to focus on hard-to-find small details in the second image when creating the problem.
- Try to solve your problem to make sure that it is not ill-defined, ambiguous, or unsolvable.
- Do not specifically refer to the second image as *second image* in your question—the students will only be given one image. Just call it an *image*.
- Do not leak the problem-relevant information in your question statement, when it can be inferred from the image.
- Answer in English.

Only output the final hard problem of yours and nothing else. Your output should be formatted as: *Final Problem: your hard problem.*

Prompt for Skill Extraction

You are an expert AI researcher specializing in error analysis for Multimodal Large Language Models. Your task is to analyze specific failure modes in visual reasoning tasks by comparing *teacher trajectories* (reasoning that leads to the correct answer) against *student trajectories* (reasoning that leads to an incorrect answer).

Analysis Framework
For the provided input, perform the following discriminative analysis:

1. *Compare Visual Attention*: Where did the teacher look versus the student? Did the student miss a region entirely, or did they misinterpret what they saw?
2. *Compare Logic/Semantics*: Did the student fail in reasoning despite seeing correctly? Check for arithmetic errors (*e.g.,* adding numbers wrong), misunderstanding definitions (*e.g.,* what counts as a "car"), wrong logical deductions (*e.g.,* false implication, contradictions, or invalid spatial reasoning), or other logical fallacies.
3. *Compare Root Cause*: Identify the core difficulty (*e.g.,* occlusion, visual camouflage, saliency bias, arithmetic failure, logical fallacy).

And then, finally,
4. *Extract Generalizable Skill*: Generate a single phrase describing the abstract skill required based on the root cause and the core difficulty factor (*e.g.,* changing "missed the car behind the van" to "instance segmentation failure due to occlusion").

Final Answer
Please output the generalizable skill in a single phrase. Do not include any other text or comments.

Input Data

Question:
{question}

Teacher Trajectory (Correct):
{teacher_trajectory (τ_T)}

Student Trajectory (Incorrect):
{student_trajectory (τ_θ)}

Prompt for Skill-guided Generation (Both domains)

A problem for the first image:
{reference_question}

Above are two images. For the first image, I have given you a reasoning problem that asks about the image. Your task is to create a harder problem for the second image.

Target Reasoning Skills
Your problem should require one or more of the following reasoning skills:
- {skill 1}
- {skill 2}
...
- {skill 10}

Requirements
- Do not just copy the problem verbatim from the example, and try to be creative.
- The new problem should be about the second image, and it should be similar in complexity or harder than the given problem.
- Make sure your problem leads to an objective answer without any subjectivity.
- It would be great if you could create a more natural problem that incorporates various reasoning skills in a meaningful way, rather than artificially asking for them.
- Try to solve your problem to make sure that it is not ill-defined, ambiguous, or unsolvable.
- Do not specifically refer to the second image as *second image* in your question—the students will only be given one image. Just call it an *image*.
- Do not leak the problem-relevant information in your question statement, when it can be inferred from the image.
- Answer in English.

Only output the final hard problem of yours and nothing else. Your output should be formatted as: *Final Problem: your hard problem*.

LLM Judge-based Answer Grouping

Given a question and multiple student's answers to the question, group them into the groups of equivalent answers. The answers might be different in their surface forms but semantically equivalent. Do not attempt to solve the question or verify the correctness of the answers, just group them based on semantic equivalence. Your response should be in a json format where key is the answer (string) and the value is the list of students (string), formatted as:
{{
    "answer 1": [student generation indices corresponding to answer 1],
    "answer 2": [student generation indices corresponding to answer 2],
    ...
}}

Example
Question:
On the lecture slide, first count the number of lines of text below the lecture title. Then, count the number of words in the course name "Mathematics for Computer Science". Next, count the number of letters in the instructor's name. Finally, count the number of distinct colors in the gradient bar. Multiply these four values together. What is the result?

Student Answers:
Student 1: 512
Student 2: 484
Student 3: The final result is 484.
Student 4: 512
Student 5: None
Student 6: 29
Student 7: **Final Answer** 512.
Student 8: 484

Grouped Answers:
{{
    "512": [1, 4, 7],
    "484": [2, 5, 8],
    "29": [6],
    "None": [5],
}}

Question:
{question}

Student Answers:
{answers_str}

Grouped Answers:

C Additional Results

C.1 Answer Divergence Distribution

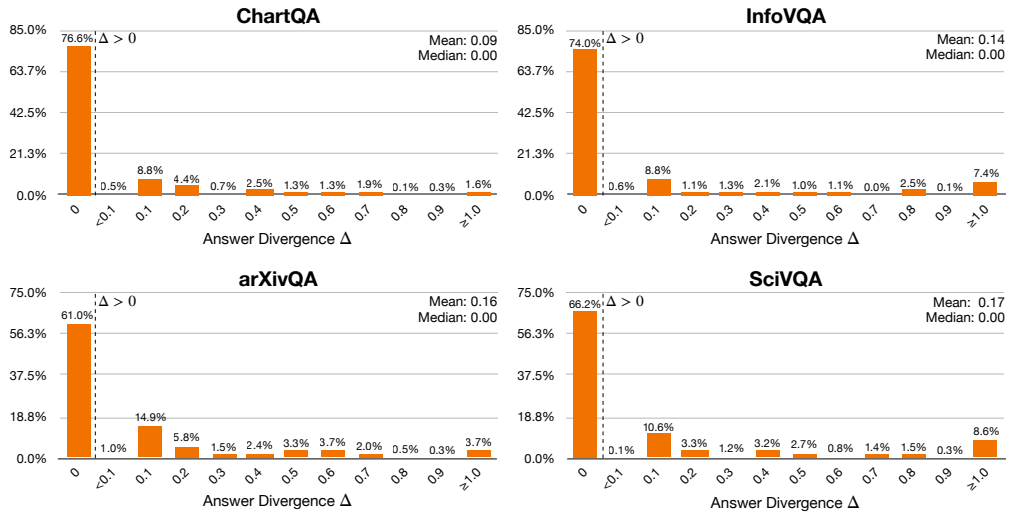


Figure 5: Distribution of answer divergence of the four datasets used in §2.4, plotted individually. We use the same set of teacher-student models as in §2.4. In all datasets, student and teacher tend to generate homogeneous output distributions, with zero-delta prompts comprising from 61% up to 77% of the entire distribution.

C.2 Out-of-Domain Generalization

Model	$\text{MathVista}_{\text{mini}}$	LogicVista	$\text{MMMU-Pro}_{\text{Vision}}$
Qwen3-VL-8B-Thinking	81.12	56.64	53.41
OPD w/ Seed Data (10k)	79.05	56.24	53.64
OPD w/ DELTAPROMPTS (10k)	81.05	58.04	53.75
Qwen-8B-DeltaThinker	**81.74**	**61.97**	**56.30**

Table 5: Evaluation on Out-of-Domain Tasks. **Qwen-8B-DeltaThinker, despite specializing on chart / document / perception-centric reasoning, shows on-par or better performance than Qwen3-VL-8B-Thinking across math, puzzle and STEM reasoning.**

D Related Works

Algorithmic Advances in Distillation The algorithmic foundations of knowledge distillation have been extensively refined, beginning with classical formulations that match teacher output distributions via forward KL [16, 49]. Subsequent work has proposed alternative divergence objectives to improve training stability and mitigate mode-seeking pathologies [13, 30, 29, 61, 25, 58], and has addressed the distribution shift between teacher and student through on-policy distillation [2, 74, 41, 69]. More recent variants extend this framework through self-distillation [74, 19], black-box settings where only teacher outputs are accessible [21, 68], and auxiliary training signals [22, 65]. Notably, Kim et al. [27] observe that naive distillation can bias the student toward narrow patterns when the training data lacks coverage—similar in spirit to our finding that distilling on zero-delta prompts reinforces stagnation rather than driving learning. While these works have made substantial progress on the algorithmic front, they largely treat the training prompts as a given.

Data-Centric Perspectives on Distillation Our work sits within a smaller but growing line of research that views distillation through a data-centric lens. Early evidence from Li et al. [34] showed that adaptive data selection can match full-data performance with a fraction of the samples, and subsequent methods have introduced principled selection criteria based on student-teacher compatibility [39], informative trajectory ranking [67], attention-head influence [38], or token-level acceptance rates [17, 15]. Most closely related to our work, Xu et al. [63] studies data importance in OPD, but focuses on token-level data selection for a fixed set of prompts. Complementing these efforts, DC-CoT [72] provides a systematic benchmark of data-centric knobs for CoT distillation, while Jung et al. [23] proposes gradient-based diversification for synthetic data generation. We differ in two key ways: our selection criterion is answer divergence—a principled teacher-student gap measure—and our pipeline actively synthesizes prompts targeted at this signal rather than filtering from a fixed pool.

Open Recipes for Multimodal Reasoning In parallel, a rapidly growing body of work has released open recipes for post-training multimodal reasoning models [76, 7, 18, 73, 50, 37, 51, 66]. These efforts typically aggregate existing datasets with heuristic quality filters and apply staged SFT or RL pipelines over teacher-generated trajectories. Lu et al. [40] offers a direct data-centric analysis of this trend, arguing that RL's advantage over SFT stems from implicit filtering toward medium-difficulty samples, with Chu et al. [8] reaching a similar conclusion through a broader comparison. Most relevant to our setting, Long Grounded Thoughts[1] synthesizes visual problems at scale—though the work focuses on scaling reasoning-heavy trajectories rather than teacher-student divergence. Our work is complementary: we isolate a principled quantity (answer divergence) as the guiding criterion, and show that prompt synthesis targeted at this signal yields substantial improvements across both on-policy and off-policy distillation regimes.

E Limitations and Future Work

While DELTAPROMPTS establishes a robust foundation for divergence-driven distillation, our synthesis pipeline measures answer divergence against the teacher-student pair at the beginning of training, and uses the resulting dataset throughout. As the student evolves during distillation, the capability gap may shift—prompts that were once divergence-inducing may become zero-delta. We mitigate this in practice by training for a single epoch, so each prompt is seen only once, limiting the impact of drift. Still, a more active approach could iteratively re-measure divergence against the student's current state and regenerate targeted prompts. Additionally, our pipeline generates novel text prompts but still sources images from existing datasets rather than synthesizing them; a natural extension is to close this generative loop through diffusion-based generation or structured editing.

Beyond these methodological extensions, our empirical validation focuses on visual domains where images are abundant but high-quality reasoning data is sparse—chart, document, and real-world perception-centric reasoning. While these provide a clear testbed, the core principles of answer divergence are broadly applicable, and a particularly promising direction is extending this framework to long-horizon multimodal tasks such as autonomous web navigation, embodied agent control, or long-form video understanding. We view DELTAPROMPTS as a first step toward a more principled, data-centric approach to distillation.

F Broader Impacts

This work demonstrates that carefully constructed prompt distributions can substantially improve the efficiency of knowledge distillation in vision-language models. On the positive side, our findings suggest a path toward training capable compact models with significantly fewer resources, potentially democratizing access to strong multimodal reasoning capabilities. The principled framework we introduce—measuring and optimizing for answer divergence—may also generalize beyond distillation to other settings where data quality matters more than quantity. On the other hand, more efficient distillation techniques could accelerate the proliferation of capable models with limited oversight, and the synthesis pipeline itself could be repurposed to generate misleading visual reasoning problems. We encourage practitioners to apply these methods responsibly and in accordance with established ethical guidelines for model development and deployment.

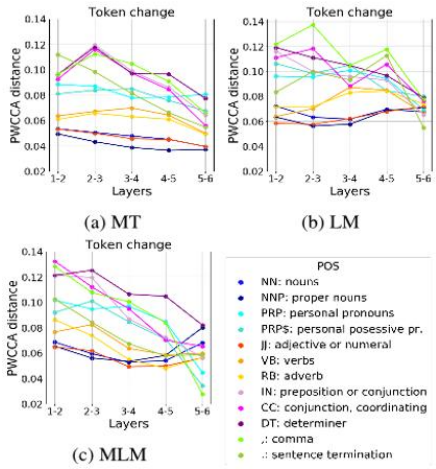
G Qualitative Examples

G.1 Skill Extraction

In Examples 1 to 4 below, we present qualitative examples of extracted skills from our synthesis pipeline, using the input problems generated via seed-guided generation. We include the key snippet from answer extraction trace generated by the teacher during skill extraction. Overall, the pipeline is able to discover precisely where the student trajectory diverged from the teacher's, and output these into a concise skill description—providing targeted signals for the subsequent generation stage to focus on the student's actual weaknesses.

Skill Extraction Example 1

Input Problem



How many part-of-speech tags in the second image exhibit a peak PMWCA distance within the 2-3 layer interval in all three subplots (a), (b) and (c), and additionally show a consistent decline in PMWCA distance from the 4-5 to 5-6 layer interval across the same subplots?

Skill Extraction Trace

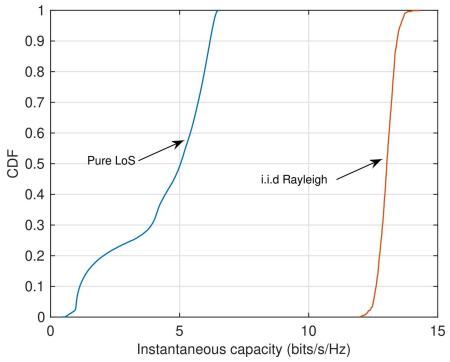
... The student incorrectly claimed all 11 POS tags meet these conditions. The student appears to have misinterpreted the plots, likely by assuming all lines peak at 2-3 and decline from 4-5 to 5-6 `...[omitted]`
The generalizable skill needed is the ability to validate patterns systematically across multiple data representations without overgeneralizing.

Extracted Skill

Cross-referential pattern validation across multiple data representations

Skill Extraction Example 2

Input Problem



For the 80% probability mass range (between the 10th and 90th percentiles) of each distribution, calculate the width in bits/s/Hz. How many times wider is the Pure LoS distribution's range compared to the i.i.d Rayleigh distribution's range? Round your answer to the nearest whole number.

Skill Extraction Trace

... Student misread the percentiles, interpreting the 10th percentile of Pure LoS as 2.5 but placing the 90th percentile at 10.5 (should be 6.5). Similarly, they misread the i.i.d Rayleigh range as 8.5–13.5 (should be 12.2–13.8) . . . `[omitted]`

Extracted Skill

Spatial mapping of CDF values to x-axis quantiles

Skill Extraction Example 3

Input Problem



What color is the text printed on the red shopping cart?

Skill Extraction Trace

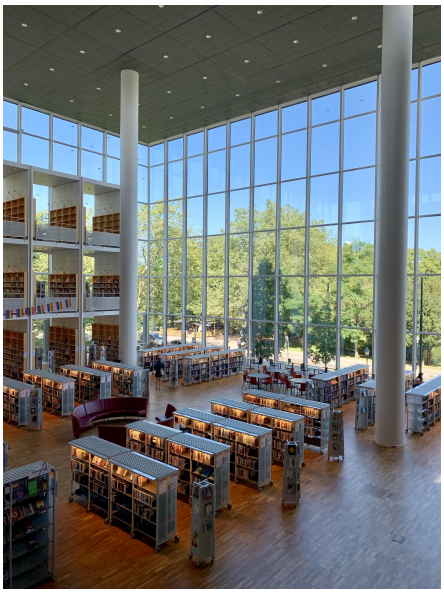
... Student thought "Walmart carts have black text", but the image might not be a Walmart cart, or the text color is different here. The student relied on a faulty assumption (saliency bias towards known brands) instead of visual evidence. . . . `[omitted]`

Extracted Skill

Avoiding saliency bias from prior knowledge when interpreting visual evidence

Skill Extraction Example 4

Input Problem



How many rainbow flags are visible on the upper-level balcony in the image?
A: 4
B: 5
C: 6
D: 7

Skill Extraction Trace

... The student's logic was to count the flags one by one, but they ended at 5. `...[omitted]`

Extracted Skill

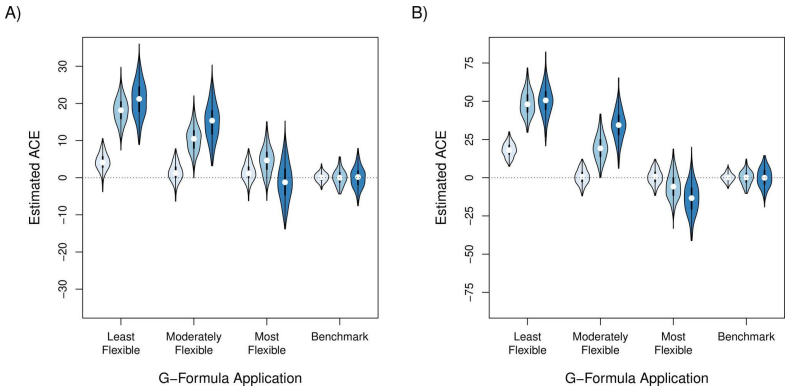
Precise enumeration of discrete objects in a sequence

G.2 DELTAPROMPTS

We also present qualitative examples of synthesized prompts in DELTAPROMPTS, focusing on skill-guided generation. Each example shows the new image passed to the data generator, the sampled skills used for conditioning, and the resulting prompt.

DELTAPROMPTS Example 1

New Image



Sampled Skills

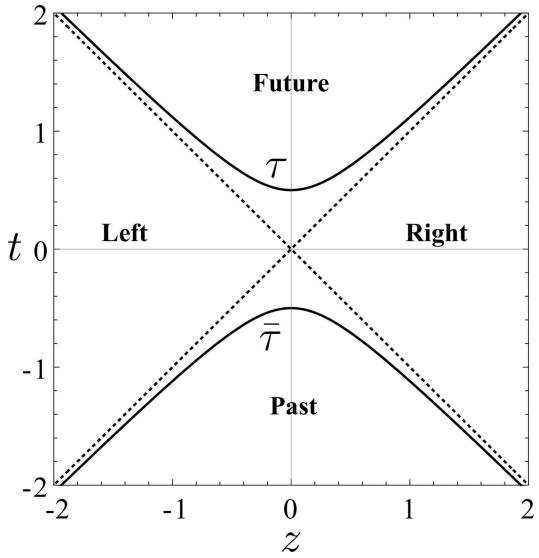
Correctly interpreting positions in violin plots, Multi-condition cross-validation across visual subplots, `...[omitted]`

Generated Prompt

For Panel A, count the number of G–Formula Application categories where at least one violin plot has an Estimated ACE value exceeding 20. For Panel B, count the number of categories where at least one violin plot has an Estimated ACE value exceeding 50. What is the product of these two counts?

DELTAPROMPTS Example 2

New Image



Sampled Skills

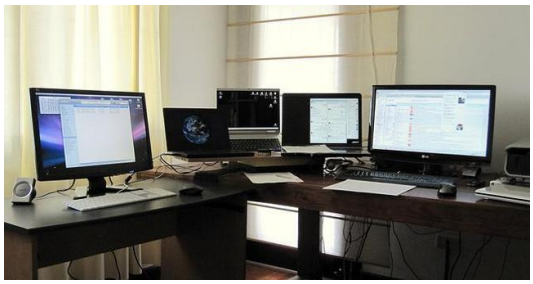
Visual Coordinate Extraction and Function Evaluation, Parameter Instantiation from Discrete Data Points, `...[omitted]`

Generated Prompt

Given that the solid curve labeled τ in the Future region of the spacetime diagram represents a hyperbola defined by $t^2 - z^2 = c$ for some constant c, first determine c using the curve's intersection with the t-axis, then calculate the slope of the curve at that intersection point.

DELTAPROMPTS Example 3

New Image



Sampled Skills

Coordinate system interpretation with strict inequality constraints, Accurate spatial relationship interpretation in lateral vs. depth axes, . . . [omitted]

Generated Prompt

How many monitors are positioned such that their horizontal center lies within the rightmost 30% of the image's width?
A) 1
B) 2
C) 3
D) 4

DELTAPROMPTS Example 4

New Image



Sampled Skills

Color Perception Failure due to Saliency Bias, Distinguishing between object-relative and viewer-relative spatial reference frames, . . . [omitted]

Generated Prompt

A photographer captured this scene from a position directly behind the bench. If the individual wearing the white helmet wishes to face the bicycle located on the right side of the image, in which direction must they rotate their head relative to their own body orientation?

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction precisely reflect the scope of our contributions: (1) introducing answer divergence as a principled metric for prompt selection in distillation, (2) proposing a two-stage synthesis pipeline that targets this metric, and (3) producing DeltaPrompts and validating it across on-policy distillation, cross-family transfer, and off-policy SFT settings, as detailed in §4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the presented work in §E, along with future directions to overcome them.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: Our work is empirical in nature and does not introduce theoretical results requiring formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides comprehensive experimental details sufficient for reproduction: §4.2.1 and §A describe the training setup (learning rate, batch size, rollouts per prompt, max response length, framework), evaluation protocol (benchmarks, LLM judge, decoding parameters), and synthesis pipeline configuration (teacher/student models, seed datasets, image sources, deduplication). §B provides the exact prompts used for data generation and LLM-judge-based answer aggregation.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We attach artifacts to our supplementary materials, including the data (subsampled to match the file size constraint) and the code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: §4.2.1 and §A specifies the training and evaluation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the standard error of the mean as a statistical significance metric, measured across 5 independent runs in all our main experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: §A includes details about the type and amount of compute used for our training run.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impact of our work in §F.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our domain of analyses focuses on tasks in academic settings, such as question-answering on charts & documents or spatial reasoning on static images. We did not scrape any Internet data, and our models are trained on these specific tasks. Thus we did identify a need for specialized safeguards beyond standard ethical considerations.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all existing assets used by either their URL or the corresponding paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The synthesized dataset (DeltaPrompts) is provided in the supplementary materials, along with documentation both in main section and appendix describing its structure, generation pipeline, and intended usage.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are central to our methodology: Qwen3-VL-235B-Thinking serves as the teacher and data generator throughout our synthesis pipeline (§3), and Qwen3-32B is used as an LLM judge for answer aggregation (§2.2). Their specific roles and prompts used are fully documented in §3, §4 and the appendix.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.