DOES SELF-TRAINING ON PRIVILEGED REPAIRS CLOSE REPRESENTATION GAPS? A DIAGNOSTIC EVALUATION IN FOUR DOMAINS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models often answer a question correctly from one representation of an object and incorrectly from an informationally equivalent one. We evaluate a strategy that turns such disagreements into supervision: the model, shown both representations and both of its traces, rewrites its failed trace, and a LoRA student that sees only the weak representation is trained to match the frozen model's token distribution over the rewrite, alongside replay of questions it already answers correctly. The privileged context thus decides what the student is trained toward. Using Qwen3-VL-8B-Instruct across four domains (synthetic social graphs, Lichess positions, WikiTableQuestions and MolLangBench), we measure what the strategy changes relative to the frozen model and where those changes come from. It narrows the representation gap sharply on chess and graphs, where the weak representation is the image (the chess image arm gains 17 points), gives small gains on tables, and widens the gap on molecules, where the weakest representation barely moves; the representations the model already reads well change little. The gains lean toward questions that point to a specific part of the image, such as one node's arrows or one row of the board, over questions that require searching or comparing across the whole picture, with exceptions in both directions. Against answer-only fine-tuning on the same data the results are mixed; it sometimes transfers better. Rerun from its own trained model, it improves once more and then stops. The strategy cannot supervise a representation its teacher cannot reason in, needs two additions to train stably, and can leave correct answers resting on reasoning that contradicts the data.

1 INTRODUCTION

A vision-language model shown a picture of a small directed social graph answers questions about who follows whom far less reliably than when it is shown the very same graph as a list of edges. On chess positions the pattern repeats: the same model reads a piece list almost perfectly and a rendered board far worse, and a FEN string, which is also text, sits in between. The information is identical; only the surface form differs. We call the spread a *representation gap*.

Every question on the wrong side of the gap looks like free supervision. The model's correct trace from the strong representation is a demonstration, its failed trace from the weak one localises the error, and the gold answer certifies the endpoint, though not the steps. A natural strategy follows: let the model, shown both representations, repair its own failed trace, and train it to produce the repair from the weak representation alone. Variants of this idea, in which a model conditioned on privileged context teaches itself to behave the same way without it, are increasingly common (Askell et al., 2021; Snell et al., 2022; Zhao et al., 2026; Ye et al., 2026). They are rarely taken apart, however, with diagnostics that separate what such training teaches from what it merely changes about response length or answer priors. Nor, to our knowledge, has self-distillation, with one set of weights as both teacher and student, been studied with another representation of the same object as the privileged context, across modalities, such as a picture of a graph paired with its list of edges.

We provide such an evaluation for one concrete instance: self-training on repairs the model writes with privileged context. One set of frozen weights plays both roles. The *teacher*, the base model

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

shown both representations and both of its traces, rewrites the failed trace so that it reaches the strong representation's answer while citing only the weak one. The *student*, the same weights under a LoRA adapter (Hu et al., 2022), sees only the weak-representation prompt and is trained to match the frozen model's token distribution over that rewrite (§3.3), alongside replay of questions it already answers correctly. The privileged context thus *generates* the targets. We evaluate it against the frozen base model around six questions.

- **RQ1, efficacy** (§5). Does the strategy raise weak-representation accuracy without degrading the strong representations? It narrows the gap sharply on chess and graphs, gives small gains on tables and widens the gap on molecules; no strong representation falls by more than 1.22 points.
- **RQ2, design choices** (§6). What does it need to train stably? Concise rephrasing of the teacher's traces, and anchor-balanced batches.
- **RQ3, generalisation** (§7). Do the gains transfer? Mostly not: the one robust transfer is to graph layouts never used in training. Against answer-only SFT, which is stronger in domain everywhere but tables, the results are mixed (§8).
- **RQ4, diagnostics** (§9). Where do the gains come from? From questions that name where their answer lies, from responses that no longer truncate, and on one benchmark from a shifted response prior.
- **RQ5, limits** (§10). What can it not reach? A representation the teacher cannot reason in (SMILES), and faithful reasoning: correct answers can still rest on traces that contradict the data.
- **RQ6, iteration** (§11). Does rerunning the strategy from its own trained model keep improving it? A second round adds a further gain on the image arm and a third adds none; transfer survives but does not grow.

Contributions. **(1) Method-agnostic diagnostics** for any method that claims to close a representation gap: headroom-normalised per-category gains, the share of gains from responses that no longer truncate, and the shift in response prior (§9). **(2) An open four-representation dataset.** We release every question with its four informationally equivalent representations, for the community to run further studies of this kind: graphs and chess are generated by us (chess positions drawn from the public Lichess database), and tables and molecules are assembled from the open WikiTableQuestions and MolLangBench releases, with the other representations derived by us (§4, Appendix A). **(3) A case study** applying both to self-training on privileged repairs in four domains (RQ1–RQ6). **(4) Conditions for stable training**: two failure modes, style contagion and off-target drift, each exposed in a development run and removed by one addition (§3.4, §6).

2 Related Work

Privileged information, context distillation and self-distillation. Our teacher is a privileged-information learner (Vapnik & Vashist, 2009; Lopez-Paz et al., 2016): it conditions on side information present at training time and absent at test time, here another view the same model already reads better, and teacher and student share every weight. The closest prior mechanism is context distillation (Askell et al., 2021; Snell et al., 2022), in which a model conditioned on a context teaches itself to behave the same way without it; our context holds a second representation and two traces. Recent on-policy self-distillation conditions the same model on privileged solutions or context (Zhao et al., 2026; Ye et al., 2026). Distillation from a larger teacher (Hinton et al., 2015; Hsieh et al., 2023) needs a stronger model; rejection-sampling fine-tuning (Yuan et al., 2023) filters a model's own samples by correctness, and SDFT (Yang et al., 2024) rewrites targets into the model's own distribution. Our training (§3) resembles rejection-sampling fine-tuning and STaR, which fine-tune on answer-verified self-generated traces, and differs in where the targets come from and in fitting the frozen model's token distribution rather than the tokens alone; its teacher repairing its own failed trace is close to self-refinement (Madaan et al., 2023), here with privileged information. We evaluate what such training changes rather than propose a new variant.

Reasoning traces and faithfulness. Step-level supervision improves reasoning over outcome-only supervision but needs step labels (Uesato et al., 2022; Lightman et al., 2024); STaR's rationalization instead produces traces by conditioning generation on the answer and keeps those that reach it (Zelikman et al., 2022). Our traces are verified only at the answer, which is weaker than step-level

supervision, since a trace can reach the right answer through wrong steps. A stated chain of thought also need not reflect the computation that produced the answer (Turpin et al., 2023; Lanham et al., 2023), and a plausible rationale is not necessarily a faithful one (Jacovi & Goldberg, 2020). Our teacher template adds a specific risk: it tells the teacher never to hint at the other representation, so a corrected trace can state an observation it took from the strong representation as if it had read it from the weak one.

Representation sensitivity. Encoding choice measurably changes graph-reasoning accuracy in text-only LLMs (Fatemi et al., 2024; Wang et al., 2023), and vision-language models fail on elementary visual tasks that are trivial in symbolic form (Rahmanzadehgervi et al., 2024; Tong et al., 2024). GITA (Wei et al., 2024) and MMTab (Zheng et al., 2024) construct paired visual and textual views of the same graph or table, but use the pairing to *measure*, not as a self-supervision signal. We say *representation gap* rather than "modality gap" because the weak view is often text, and "modality gap" already names a property of embedding geometry (Liang et al., 2022).

Off-policy distillation and forgetting. GKD (Agarwal et al., 2024) and MiniLLM (Gu et al., 2024) sample from the student; we sample from the teacher once, offline, so exposure bias is not ruled out. Against forgetting, RLHF and EWC penalise divergence (Ouyang et al., 2022; Kirkpatrick et al., 2017) and Learning without Forgetting (Li & Hoiem, 2018) distils the old model's outputs; anchor rows are a data-level form of that self-replay (§3.4).

3 THE STRATEGY UNDER EVALUATION

The strategy has two stages that use the frozen model in different ways, and privileged information enters only in the first. **(i) Target construction** (offline, once): for each pair row, the frozen model is shown both representations and both of its own traces and writes a corrected trace y, kept only if it reaches the gold answer; the same model then compresses y into $\tilde{y}$ without changing its answer. **(ii) Training**: a LoRA student that sees only the weak prompt is trained toward $\tilde{y}$ with a token-level KL to the frozen model, alongside anchor rows that replay questions it already answers correctly (§3.3). In short, the privileged context decides *what* the student is trained toward; it never enters the student's input.

3.1 SETUP

A chess position can be shown as a picture of a board, a FEN string or a list of pieces; a graph as a drawing or a list of edges. Every view carries the same information, so a question has one correct answer whichever view the model sees. When the model succeeds from a *strong* view and fails from a *weak* one on the same question, we have the gold answer, a correct trace to learn from and a failed trace to repair.

Representations. Formally, a *datapoint* x (a graph, a table, a chess position, a molecule) admits $K = 4$ informationally equivalent *representations* $\rho_1(x), \ldots, \rho_K(x)$: one is a rendered image, the other three are textual serializations. A *question* q about x has a verified gold answer $a^\star$. Because $a^\star$ is a property of x and not of any one representation, it is the same for all K. A template π_k turns $(\rho_k(x), q)$ into the student prompt p_k; for the image representation the pixels are attached as a separate content part. The four prompts are otherwise identical. Each carries the same closed-world clause (answer only from what is shown) and the same answer contract (free-form reasoning ending in a single `<answer>` tag), so the arms differ only in the representation.

Mining training rows. Let π_{θ_0} be the frozen base model and $g(\hat{a}, a^\star) \in \{0, 1\}$ the grader. We run the base model greedily on every prompt p_k and record its answer $\hat{a}_k$ and reasoning trace τ_k. **Pair rows** are the disagreements: a strong representation s the model got right, $g(\hat{a}_s, a^\star) = 1$, and a weak representation w it got wrong, $g(\hat{a}_w, a^\star) = 0$, on the same question; when several strong representations exist for one failure we keep one, by a fixed per-domain preference order. **Anchor rows** are the successes: any (q, k) with $g(\hat{a}_k, a^\star) = 1$, stored with the base model's response. Pair rows are what the teacher (§3.2) repairs; anchor rows keep the student close to the base model's existing correct behaviour (§3.4).

3.2 TARGET CONSTRUCTION: WHERE THE PRIVILEGED CONTEXT IS USED

The teacher context of a pair row, $c = \mathcal{T}(q,\ \rho_s(x), \tau_s, \hat{a}_s,\ \rho_w(x), \tau_w, \hat{a}_w)$, is a single self-contained prompt that frames the model as an analyst comparing two traces of one question and instructs it to rewrite the failed trace so that it reaches the strong representation's answer $\hat{a}_s$, changing only the steps that caused the error, citing only facts visible in $\rho_w(x)$, and emitting no meta-commentary and no reference to the other representation. The output must read as the model's own direct answer. The teacher is *privileged only by conditioning*: it is π_{θ_0} with the LoRA adapter disabled, and no second model is ever loaded. One completion per row is sampled offline, $y \sim \pi_{\theta_0}(\cdot \mid c)$ at T=0.4, top-p 0.9, top-k 20, and kept only if it is untruncated and grades correct against $a^\star$. The graph run used an earlier wording of one template rule, asking for a clean first-person trace rather than one in the failed trace's own voice (Appendix D).

The rewrite. The corrected trace y still narrates its own repair, so each kept y is rewritten once more by the same frozen model (§3.4 explains why). The *rewrite prompt* $\mathcal{R}(y)$ contains y and, for an image row, the row's image; it never contains the strong representation. It asks for the same reasoning as a concise, first-attempt trace, sampled at the settings above, and the rewrite must end in the same `<answer>` tag as y. Its output, $\tilde{y}$, is the completion the student is trained on; a row whose rewrite is truncated, changes the answer or leaks the framing keeps y and c instead. Anchor targets, the base model's own correct responses, go through the same rewrite.

3.3 TRAINING OBJECTIVE

The student is trained on two kinds of rows. A *pair row* pairs the weak prompt p_w with the rewritten target $\tilde{y}$ of §3.2; an *anchor row* pairs a prompt p_k the base model already answers correctly with that model's own response, rewritten the same way. Pair rows teach the correction; anchor rows replay the base model's correct behaviour, a form of self-replay close to Learning without Forgetting (Li & Hoiem, 2018), so that the correction does not erode what already works. Only the LoRA adapter is trained.

Training minimises a token-level KL divergence to the frozen model π_{θ_0} (adapter disabled). Each row supplies a student prompt p, a target $\tilde{y}$, and a reference context $c^\dagger$ under which the frozen model scores $\tilde{y}$: the rewrite prompt $\mathcal{R}(y)$ for a pair row (c if its rewrite failed validation), and the student prompt p_k itself for an anchor row. At each position t of $\tilde{y}$, with f denoting next-token logits and $T = 0.4$ the rollout temperature, the student and reference next-token distributions are

$$P_t^{\text{stu}} = \text{softmax}\big(\tfrac{1}{T} f_\theta(p, \tilde{y}_{<t})\big), \qquad P_t^{\text{ref}} = \text{softmax}\big(\tfrac{1}{T} f_{\theta_0}(c^\dagger, \tilde{y}_{<t})\big).$$

The loss is the KL divergence from the reference to the student, averaged over the tokens of $\tilde{y}$ and restricted at each position to $\mathcal{V}_t$, the reference's k=100 most probable tokens:

$$\mathcal{L}(\theta) = \mathbb{E}\left[\frac{1}{|\tilde{y}|}\sum_{t=1}^{|\tilde{y}|}\sum_{v\in\mathcal{V}_t} P_t^{\text{ref}}(v)\ \log\frac{P_t^{\text{ref}}(v)}{P_t^{\text{stu}}(v)}\right]. \tag{1}$$

Gradients flow only through the student. For an anchor row both inputs are identical at $\theta = \theta_0$, so its loss and gradient are exactly zero at initialization.

3.4 TWO ADDITIONS FOR STABLE TRAINING

Trained naively, Eq. 1 fails in two separate ways, and each is removed by one addition (§6 gives the evidence).

Style contagion and concise rephrasing. The teacher's corrected trace is long and narrates its own repair. Trained on directly, the student acquires the narration: completion length blows up, and responses enter recount loops and hit the token cap. The rewrite of §3.2 removes this: $\mathcal{R}(y)$ asks the model to keep every identifier, number and list order, reproduce the final `<answer>` tag character for character, and drop backtracking and re-verification, while keeping the trace's own voice and layout (Appendix D).

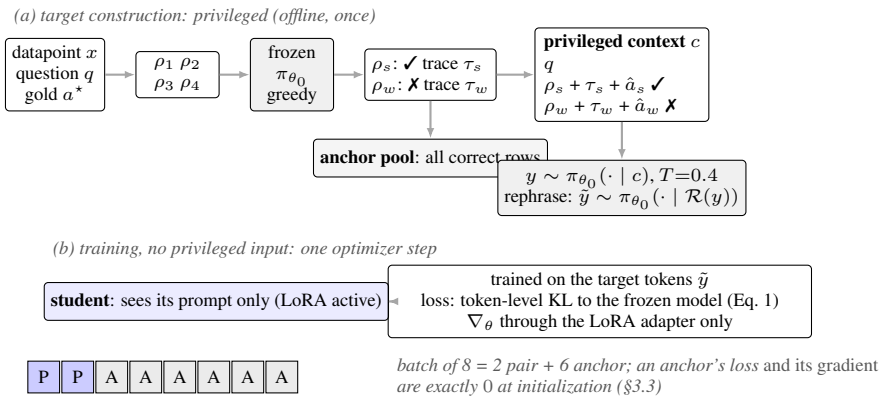

Figure 1: The strategy. **(a)** A question the frozen model gets right from ρ_s and wrong from ρ_w becomes a *pair row*, and every correct attempt enters the *anchor pool*. From the privileged context c, which holds both representations and both traces, the same model writes a corrected trace y and rephrases it into a concise $\tilde{y}$. **(b)** Training fits $\tilde{y}$ with Eq. 1 in batches of 2 pair and 6 anchor rows; the student never sees the privileged context (§3.3).



Figure 2: The image representation of one datapoint in each domain: a synthetic social graph, a Lichess position, a WikiTableQuestions table rendered as a photographed document (Appendix A) and a MolLangBench molecule with its atom indices. Every datapoint also has three textual representations (Table 1).

Off-target drift and anchor-balanced batching. Mined pairs are one-sided by construction: in three of the four domains most fail on the image side, and every pair row is a failure case. An objective built only from failures has no term that preserves what already works. An anchor row (§3.3) contributes no learning signal at initialization, only a restoring force that grows as the student deviates from base behaviour on that row. **Anchor-balanced batching** makes preservation outweigh correction in every step: each optimizer step holds **2 pair rows and 6 anchor rows** (a ratio chosen on graphs), the anchors drawn from the other three representations and, wherever possible, from the *same question* as the pair rows, so that in one step the student is told to change how it reads q from the weak representation and to keep how it reads q from every other one. Anchor rows are self-replay: they add no information the base model lacks.

4 Evaluation Protocol

Data. Table 1 lists the four domains and Figure 2 shows each one's image; tables are WikiTableQuestions, given as a photographed-document image, a column-wise CSV, a JSON list of rows and prose (Appendix A). Every question is asked of all four representations, so arms are compared question-for-question, and splits are held out by object (graph, position, table, molecule).

Training and scoring. Qwen3-VL-8B-Instruct (Bai et al., 2025), served with vLLM (Kwon et al., 2023) and trained with TRL (von Werra et al., 2020): LoRA (r=16, α=32) on the language model only, learning rate 10^{-5} with a 200-step warmup and cosine decay, 8 rows per step, 2 epochs, one H200. Decoding is greedy with a 1,024 to 4,096-token answer cap, each domain uses its own grader, and every checkpoint is scored against a frozen base run whose model, decoding settings and image bytes hash identically.

Table 1: The four domains, with frozen-base accuracy per representation on the held-out test split, ordered weakest (**bold**) to strongest. Graphs (2,640 synthetic social graphs) and chess (2,650 Lichess positions (Lichess, 2024)) are ours; tables are WikiTableQuestions (Pasupat & Liang, 2015) and molecules the MolLangBench *recognition* subset (Cai et al., 2026), both with the release's own questions. Global: all four arms pooled; gap: strongest minus weakest.

		Base accuracy by arm (%)					
Domain	**Test items**	weakest	2nd	3rd	strongest	**Global**	**Gap**
Graphs	3,600	**63.3** img	81.4 adj	86.3 edge	86.9 prose	79.5	23.6
Chess	4,000	**41.5** img	55.0 fen	90.8 prose	95.0 plist	70.6	53.5
Tables	12,260	**70.0** csv	71.4 img	76.5 prose	78.5 json	74.1	8.5
Molecules	13,600	**13.5** smi	31.5 img	47.1 table	50.3 prose	35.6	36.8

Table 2: Held-out test results, base → trained; arms ordered weakest (first row, shaded) to strongest.

Domain	**Arm**	**base**	**trained**	**Δ**
Graphs n=3,600	image	63.33	71.22	**+7.89**
	adj. list	81.44	80.22	−1.22
	edge list	86.33	86.22	−0.11
	prose	86.89	87.67	+0.78
	global	**79.50**	**81.33**	**+1.83**
Chess n=4,000	image	41.50	58.50	**+17.00**
	FEN	55.00	58.10	+3.10
	prose	90.80	92.70	+1.90
	piece list	95.00	95.80	+0.80
	global	**70.58**	**76.28**	**+5.70**
Tables n=12,260	col. CSV	70.02	72.53	**+2.51**
	image	71.35	72.37	+1.01
	prose	76.54	79.15	+2.61
	JSON	78.47	80.16	+1.70
	global	**74.09**	**76.05**	**+1.96**
Molecules n=13,600	SMILES	13.47	14.18	+0.71
	image	31.53	38.29	**+6.76**
	mol. table	47.09	50.03	+2.94
	prose	50.26	54.44	+4.18
	global	**35.59**	**39.24**	**+3.65**

5 RQ1: EFFICACY

Training narrows the gap where the weak representation is an image, gains little where it is text, and preserves the strong representations (Table 2).

(a) Image arms gain most. On chess and graphs the image arm gains far more than any other arm: the chess image arm gains 17 points and the chess gap narrows from 53.5 to 37.3. Part of this may be supervision share rather than modality, since chess FEN is also weak but gains only 3 points, and nearly three-quarters of chess pairs are image rows.

(b) Weak text arms gain little. On tables the weakest arm, a column-wise CSV, gains no more than the others, so the gap barely moves. On molecules the weakest arm, SMILES, barely moves while the others rise, so the gap widens, from 36.8 to 40.3 points (§10).

(c) Strong representations are preserved. No strong representation falls by more than 1.22 points, and on chess, tables and molecules every one rises.

6 RQ2: DESIGN CHOICES

Both additions of §3.4 were adopted after single-seed graph runs, so this evidence is anecdotal.

Table 3: Transfer: the Table 2 checkpoints with no further training, scored with each benchmark's own prompts and grader (accuracy in %). GVLQA-AUGLY redraws GVLQA-BASE's graphs with five unseen layout engines; MolLangBench is five task types held out of training (Appendix C). [a] The default style asks the model to "analyze step by step"; the alternative drops that sentence. [b] -2.48 on the 524 items both models answer.

Benchmark	Arm	base	trained	Δ
GVLQA-BASE	image	49.74	52.68	**+2.94**
n=8,776	text	53.30	53.54	+0.24
GVLQA-AUGLY				
n=43,880	image	51.07	53.75	**+2.68**
ChessQA	default style[a]	5.20	5.67	+0.47
n=3,000	alt. style[a]	4.37	5.83	**+1.47**
MMTab TQA	image	64.22	65.36	**+1.14**
	text grid	61.50	62.07	+0.57
n=21,090	**micro**	**62.86**	**63.71**	**+0.85**
MolLangBench	image[b]	44.50	46.50	+2.00
n=1,000	SMILES	13.70	13.40	−0.30

(a) Rephrasing removes style contagion. Trained on the teacher's raw traces, the student learned to ruminate: accuracy fell, and most of the loss came from responses cut off at the length cap. With rephrased targets, the next run was the first to beat the base model on accuracy and truncation at once.

(b) Anchors remove off-target drift. Trained only on corrections, the student improves the weak representation but loses accuracy on representations it already answered correctly. Mixing the model's own correct answers back into every batch prevents that regression.

(c) The anchor loss is not a proxy for retention. A layout whose anchors share the pair row's question beat one with fewer same-question anchors on test, although the latter looked better on every training scalar. The difference is within the spread across our graph runs, so it is a single observation, but it cautions against choosing a layout by its training loss.

7 RQ3: Generalisation

Each domain's checkpoint is also scored on an *external* benchmark with no further training (Table 3).

(a) Graphs transfer. The GVLQA image arm gains $+2.94$ while the text arm barely moves, so the representation gap narrows on a benchmark we never trained on, and the gain holds on graph layouts never seen in training, though part of it is a shift in how often the model answers "Yes" (§9).

(b) The other domains barely do. Every other gain is small and, on inspection, only partly reading: on MMTab it nearly vanishes once answers are normalised, ChessQA starts from near floor, and on the held-out molecule tasks the gain comes from responses that no longer truncate (Appendix C).

8 Answer-only SFT: a baseline in all four domains

We compare against the simplest alternative: in each domain, a second model trained on the same rows with plain cross-entropy on the bare gold answer and no reasoning (Table 4).

(a) In domain, answer-only SFT wins where one reading step answers the question. It is 10 to 14 points ahead on graphs, chess and molecules and nearly closes the image gap in all three, partly because it always answers while ours and the base often run out of tokens; on the molecule rows both answer, the two tie. On tables, whose questions often need arithmetic over rows, it stops computing and falls 8 points below the base model, losing mostly on numeric answers.

(b) Outside the training distribution the results are mixed. Ours transfers better on GVLQA, ChessQA and MMTab, where answer-only SFT widens the GVLQA gap or falls below the base

Table 4: Our training against answer-only SFT (*SFT*) on the same rows, order, adapter and schedule (accuracy in %). In domain is the held-out test split (for graphs in the augmented training rendering, so base and our numbers differ from Table 2). Img. gap: best text arm minus image, which is Table 1's gap only where the image is weakest. GVLQA-AUGLY: five unseen layouts. [a]44% of MolLangBench golds are "none"; answering "none" every time scores 44.0. [b]Items whose substructure is present.

		base	ours	SFT
Graphs				
in domain	global	77.22	78.94	**88.72**
	image	55.00	62.89	**83.22**
	img. gap	31.44	23.78	**7.67**
GVLQA-BASE	image	49.74	**52.68**	50.75
	text	53.30	53.54	**56.02**
	img. gap	3.57	**0.87**	5.26
GVLQA-AUGLY	image	51.07	**53.75**	52.03
Chess				
in domain	global	70.58	76.28	**89.10**
	image	41.50	58.50	**89.80**
	img. gap	53.50	37.30	**7.50**
ChessQA	text	4.37	**5.83**	2.10

		base	ours	SFT
Tables				
in domain	global	74.09	**76.05**	65.69
	image	71.35	**72.37**	61.31
	img. gap	7.11	**7.80**	8.87
MMTab TQA	micro	62.86	**63.71**	60.91
Molecules				
in domain	global	35.59	39.24	**53.37**
	image	31.53	38.29	**60.50**
	img. gap	18.73	16.15	**1.03**
MolLangBench	image[a]	44.50	46.50	**57.70**
	all[b]	14.8	14.9	14.6

model. On the held-out molecule tasks SFT scores far higher, mostly by answering "none"; where the substructure is present, all three models tie.

9 RQ4: Where the gains come from

Three diagnostics account for much of the measured gain.

(a) Reading a named region. Gains are largest on questions that name where in the picture the answer lies (a node, a square, a rank or a file) and smallest on questions whose answer can be anywhere, so that the model must search or compare across the whole picture. Normalised by headroom, local questions on the graph and chess image arms recover 42% of what they could gain and global ones 14% (Table 5): `followers_count` ("How many people follow X?") rises from 54% to 79%, while `most_followed` barely moves. The split is not an artefact of starting accuracy, since global questions start lower yet gain half as much in absolute terms, though it has exceptions in both directions. It points to better reading of a named region, not better aggregation.

(b) Finishing instead of looping. A response is truncated when it hits the answer-length cap without an answer, typically because the model keeps re-checking its work or repeats itself. Wherever truncation was a common failure, part of the gain is responses that now finish. Truncation falls on chess, tables and molecules, and the effect is largest outside the training domains: 57% of the ChessQA items the checkpoint fixes had been truncated by the base model, and on the held-out molecule tasks almost all of them had (Appendix C).

(c) A shifted response prior. On GVLQA's main yes/no task (`connectivity`, 56% of the benchmark) the base model answers "Yes" far too often from the picture. Training brings that rate down to the text arm's, and image accuracy rises from 57% to 62%. The shift is uniform rather than item by item: on a task where the model already said "Yes" too rarely, it costs accuracy (Appendix C).

10 RQ5: Limits

The strategy has two limits that accuracy alone does not show.

(a) It cannot teach a representation its teacher cannot reason in. SMILES is the weakest molecule arm by 18 points and barely moves ($+0.71$). The limit is upstream of training: even with its own successful trace from another arm in context, the teacher fails on 65% of SMILES rows, so few SMILES targets exist to train on (Appendix B).

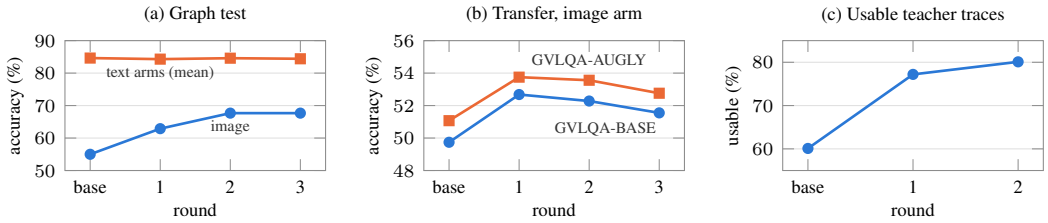

Figure 3: Rerunning the strategy on graphs, each round started from the previous round's model; "base" is the frozen model and 1–3 the model after each round. **(a)** Held-out graph test (augmented renders, as in Table 4): the image arm and the mean of the three text arms. **(b)** Image arm on GVLQA-BASE and on GVLQA-AUGLY's five unseen layouts. **(c)** Share of the model's corrected traces that are usable (untruncated, correct, no leaked framing) when it serves as the teacher; the round-3 model was not used as one.

(b) Correct answers can rest on borrowed or false steps. Reading the traces shows three patterns. Corrections are often relabelled observations: asked where White's knight stands, the base model says f4, and the teacher's corrected trace is the same trace with f4 replaced by d4. The corrected fact can come from the other representation, which the teacher saw but may not mention, so the target states it as read from the image; training on such targets rewards "I see X" when X came from elsewhere, a risk we show but do not measure. And right answers can rest on wrong steps or on the benchmark's conventions: in one molecule the student reaches the gold count while leaving a carbonyl oxygen out of the molecule's oxygens, having learned to count a bridging anhydride as two esters.

11 RQ6: Iterating the procedure

Because the trained model reads the image better than the base model, it can serve as base, teacher and student of a second round, and that round's model of a third. On graphs we ran both with the same recipe and training questions (Figure 3). **The teacher improves each round** (c): more of its corrected traces are usable, mostly because fewer run out of tokens. **The second round helps and the third does not** (a): the second lifts the image arm by a further 4.8 points without moving the text arms, narrowing the gap from 23.8 to 19.7 points, and the third leaves the image arm unchanged, so on this training set two rounds is the useful depth. **Transfer survives but does not grow** (b): later rounds stay about two points above the base model but slightly below the first round, almost entirely because they increasingly answer one too low on a single counting task (matching).

12 Threats to validity

Statistical and data. There is one model and one seed per domain, and outside graphs one configuration; checkpoints were partly selected on test, and the graph checkpoint changed two variables at once (augmented data and a larger image budget). External benchmarks are floor-limited.

13 Conclusion

Self-training on privileged repairs narrows the representation gap where the weak representation is an image, transfers little, gives mixed results against answer-only SFT, and improves once more when rerun from its own model. Much of its gain comes from reading named regions, finishing instead of looping and a shifted response prior; whether any part is better reasoning, rather than better reading stated as reasoning, remains open.

Reproducibility statement

The full pipeline is five scripts run in order, each consuming the previous one's output, and every stage is chunk-checkpointed so re-running the same command resumes it. Section 3 gives the objective and batch construction, and §4 the training hyperparameters. Appendix A documents dataset construction, including dropped items and the reason for each. Test comparisons run against a *fingerprinted* frozen

baseline: model id, decoding parameters, context length and the resized image bytes are hashed. This is what makes a configuration change detectable rather than silently mismatched. External benchmarks are scored with their own prompts and grading rules, and each is run through a pre-flight that grades the release's *own* gold answers with our grader before any GPU boots. Source, prompts, grading code and the four-representation dataset (every question with all four of its representations, for all four domains) are included in the supplementary material.

ETHICS STATEMENT

This work uses no human subjects and no personal data. The chess positions come from the public Lichess open database (Lichess, 2024) and are used only as board states; no player identity is retained. The table domain is built on the public WikiTableQuestions release (Pasupat & Liang, 2015) and the molecule domain on the public MolLangBench release; the graph domain is synthetic, and the person-names in its questions are drawn from a fixed generic list and refer to no real individual.

AI USE STATEMENT

We used generative AI tools as coding assistants for implementation, refactoring and documentation of the evaluation and training pipeline, and for editing assistance on this manuscript. All AI-assisted code was reviewed and tested by the authors, and every number reported here was produced by executing that code, not by a language model. One dataset component is itself model-generated and is disclosed as such in the main text and in Appendix A: the table domain's prose representation is generated by an external LLM from the table's JSON serialization under three stated prohibitions and mechanically checked for cell recoverability. No research ideas, experimental designs, results or citations in this paper were generated by AI without author verification. We take responsibility for the final content of this work.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations (ICLR)*, 2024.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

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

Feiyang Cai, Jiahui Bai, Tao Tang, Guijuan He, Joshua Luo, Tianyu Zhu, Srikanth Pilla, Gang Li, Ling Liu, and Feng Luo. MolLangBench: A comprehensive benchmark for language-prompted molecular structure recognition, editing, and generation. In *International Conference on Learning Representations (ICLR)*, 2026. Dataset: `https://huggingface.co/datasets/ChemFM/MolLangBench`.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software: Practice and Experience*, 30(11):1203–1233, 2000.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics (ACL)*, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4198–4205, 2020.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *ACM Symposium on Operating Systems Principles (SOSP)*, 2023.

Greg Landrum. RDKit: Open-source cheminformatics, 2024. `https://www.rdkit.org`.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y. Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Lichess. The lichess open database, 2024. `https://database.lichess.org`.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. *arXiv preprint arXiv:2407.06581*, 2024.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, 2023.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5–6):544–557, 2009.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformer reinforcement learning, 2020. `https://github.com/huggingface/trl`.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Yanbin Wei, Shuai Fu, Weisen Jiang, Zejian Zhang, Zhixiong Zeng, Qi Wu, James T. Kwok, and Yu Zhang. GITA: Graph to visual and textual integration for vision-language graph reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. Self-distillation bridges distribution gap in language model fine-tuning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. Multimodal table understanding. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

A DATASET CONSTRUCTION

Graphs (ours). A directed "follows" social graph, 5–10 people and 5–15 edges, rendered four ways: a Graphviz (Gansner & North, 2000) PNG, an adjacency list, an edge list, and one prose sentence per edge. Ten questions per graph, one per category, each with a verified gold and a fixed response schema: `edge_exists`, `followees_count`, `followers_count`, `followees_list`, `followers_list`, `mutual_following`, `common_follower`, `most_followed`, `least_followed`, `triangle`. Splits 2,500 / 50 / 90 graphs; the 90-graph test split never changed across the project, so every graph number in this paper is on the same 3,600 items. The final run trains on an augmented corpus: each graph is re-rendered with node shape (19 options), node style (5), palette (8), font (10), edge geometry and background sampled per graph, and the prose independently re-worded; the graph and the answer are never changed. The augmentation hard-codes the `dot` layout engine, which is what makes GVLQA-AUGLY's five other engines a genuinely held-out axis.

One category is dead for both models. `triangle` hits the token cap without emitting an answer on $\sim$90% of responses (324/360 for the base), contributing a near-constant $\sim$9% of dead rows to every global number. It affects both arms identically so deltas are unbiased.

Chess (ours). 2,650 positions replayed out of the Lichess open database (Lichess, 2024): both players rated $\geq$1800, one ply drawn uniformly from that game's own $[30\%, 90\%]$ window clamped to plies 20–120 (sampling a *fraction* rather than an absolute ply avoids selecting for long grinding draws; the 90% cap stops short of mating nets and resignations), stratified into three equal piece-count bands (12–19, 20–26, 27–32). Representations: a rendered board at 555 px with rank/file labels on all four edges and White always at the bottom; the FEN string; a nested colour $\rightarrow$ piece $\rightarrow$ squares mapping; and English sentences, one per piece. Two rendering choices are load-bearing. *Edge labels*: the unlabelled board was tried first and went too far: the model scored 0% on `most_occupied_file`, 2% on `rank_occupants` and 3% on `rank_count` where FEN scored 9%, 82% and 91%; adding labels moved image accuracy $25.9\% \rightarrow 39.5\%$ and still left a 56-point gap to the piece list. *Fixed orientation*: orientation as a hidden variable would confound every square-location question. Nothing that highlights a square is ever drawn, since a check marker would answer a question straight from the pixels.

Three design rules govern the ten questions (`pawn_count`, `rank_count`, `file_count`, `total_comparison`, `rank_occupants`, `most_occupied_file`, `square_colour`, `same_colour_pair`, `has_piece_type`, `piece_square`): **(1) on-board placement only**: nothing asks about side to move, castling, en passant or check, because *only FEN carries any of that*, and such a question would measure what a representation *knows* rather than how well the model can *read* it; **(2) no chess ability is tested**: no legality, no tactics, no evaluation; **(3) a closed-world clause in every prompt**, since absences are stated in no representation. Probe selection is engineered rather than uniform so that no answer is guessable from the prior: square-based probes are drawn from *occupied* squares (a uniformly drawn square is empty 65% of the time, which would turn three questions into occupancy questions), and `has_piece_type` picks an absent (colour, piece type) pair with probability 0.857 to land the yes/no split at 50/50. The most frequent answer per template runs 6–50%.

Tables (WikiTableQuestions). The questions are the release's own, verbatim: 1,930 tables over four splits-by-table with zero table overlap, verified on the table id in both directions. All four representations are derived from one parse of the table, never from the release's own files, because both of its text formats would put an *information* gap where the experiment reads a representation gap: its CSV allows a real newline inside a cell, so 18.2% of training tables span more physical lines than they have rows (worst case: 8 data rows over 275 lines); its HTML carries styling on 100% of tables, links on 93.2%, images on 35.3% and `colspan`/`rowspan` on 29.1%, and on 13.4% a `<tr>` count that disagrees with the CSV's row count; it is a different, richer table. We emit instead: a transposed CSV (one line per column), a JSON *list of row objects* (a list, never a column-keyed dict, because row order is semantic in WTQ; values stay strings so this is not the one arm that pre-classifies its cells), our own PIL render in a photographed-document style (size is an output of a font ladder, never below 12 px; no row-index column, which would be a free handle the text arms lack), and English prose generated by an external LLM from the JSON under three prohibitions (no

aggregates, no row numbers, nothing dropped), with cell recoverability checked mechanically and a failing table regenerated once. Grading uses the release's own `check_denotation`, comparing typed denotations rather than strings, which ∼6% of questions require. 178 tables (8.4%) are dropped whole and recorded: 163 unrenderable (116 that fit at no font/ceiling setting, median 100 rows; 47 containing a script no installed font covers, the dangerous case, since PIL draws a tofu box silently and the image arm would show squares where the CSV shows a name) and 15 whose prompt exceeds the budget. The dropped tables skew huge, so this tenant measures tables that fit on a page. Mined pairs have the weak side spread across all four arms (image 35%, CSV 32%, prose 19%, JSON 15%).

Molecules (MolLangBench *recognition*). A datapoint is one row, not one molecule: each of the release's tasks samples its own candidates independently, so molecules almost never recur across tasks (train: 29,138 unique SMILES over 29,200 rows). Representations: the release's own 1024×1024 PNGs passed through byte-for-byte (they draw the atom index beside every atom on most tasks, which is what makes the picture answerable in the same index space as everything else); the release's SMILES string; an atom/bond/stereochemistry table; and the same content as deterministic English prose. Both of ours are generated by RDKit (Landrum, 2024) alone. *The index contract is the whole point*: almost every answer *is* a list of atom indices, so RDKit's SMILES parse order is the shared coordinate system that makes four arms comparable; it was verified against the release's own golds on the three tasks that can be recomputed from scratch, 6,000 of 6,000 rows matching. A MOL block was rejected as the fourth representation on two measurements: it numbers atoms $1..N$ while image, SMILES and prose all agree on 0-based (an arithmetic tax indistinguishable from a representation gap), and its stereochemistry is inseparable from its coordinates: zeroing them destroyed the stereo descriptors of 1,260 of 4,000 sampled molecules. One release defect is excluded: the `ketone` task's 200 train and 200 test rows are the same molecules in the same order with identical golds, differing only in pixels, so `ketone` is dropped from test (`aldehyde` has zero overlap, so the defect is task-specific). Five further tasks are held out of training entirely as a zero-shot task-transfer set.

B WHY SMILES IS THE ARM TRAINING CANNOT REACH

SMILES is the only one of the four molecule representations that does not hand the model atom indices. The image labels them, the table and the prose list them, and SMILES makes the model derive them by scanning the string. When it chooses to derive rather than guess it frequently never finishes: the responses are degenerate repetition loops, most often on ring-closure digits. We spent eleven stage-1 generation runs on this, differing only in the prompt and (twice) the temperature, and the result is worth reporting because it is negative in an informative way.

No prompt change ever reduced SMILES truncation; every one raised it. The series, all greedy: no step-by-step line at all 38.6% → add "reason step by step" 71.5% → add anti-restart answer rules 74.1% → add a token-scan procedure 75.0% → add hard constraints 94.9%. The more rigid the scaffold, the earlier the response dies; in the token-scan run, 76% of truncated responses never finished enumerating the string. The best SMILES arm is the one with the *least* reasoning instruction, and it is unusable, because it buys lower truncation with worse reasoning on every arm (accuracy-given-non-truncated falls to 34.2% for SMILES and 40–44% for the others, against 55.5% and 79–80% with the step-by-step line), and because the teacher's privileged context *is* a reasoning trace, so a non-reasoning student cannot be trained from it.

More tokens make it worse, not better. A paired run at 2,048 vs. 4,096 max tokens with identical prompts moved truncation *up*, $60.3\% \rightarrow 72.4\%$: only 6.7% of stuck cases were rescued and 292 of 1,800 newly broke. Truncation here is non-convergent reasoning, not budget exhaustion, the same conclusion a head-to-head 2048-vs-4096 control reached on GVLQA.

The one thing that worked was decoding, not wording. With the prompt byte-identical, $T{=}0.3$ against greedy cut SMILES truncation by 4.6 pt and raised SMILES accuracy by 2.8 pt, improving all three other arms too, a larger move than any of the eight prompt edits. A greedy decoder has no escape from an argmax loop; sampling gives every step a chance to leave. We did *not* adopt it, because the greedy setting is the single configuration the correct/wrong split is mined under and the frozen test baseline fingerprints, so moving it invalidates every delta measured against that split. The configuration this study points at therefore remains untested.

The consequence for the method is the one stated in §12: the teacher, handed its own successful trace from a strong arm, still fails on 65% of SMILES rows (38.6% wrong or missing answer, 25.4% truncated) against 33–36% on the other three arms, so only 820 of 6,538 pair rows supervise SMILES.

C ADDITIONAL RESULTS

Transfer, benchmark by benchmark. Contamination is measured, not argued: MMTab's WikiTableQuestions split is excluded because 4,065 of its 4,333 questions are in our splits, including *all* of our val and test; the four scored subsets and ChessQA's 1,100 placements overlap ours in zero.

Only on GVLQA does the gap itself narrow on foreign data, and it narrows by the image arm rising while the text arm barely moves, not by the text arm falling. On the five re-rendered layouts the checkpoint's gap goes *negative* (-0.0021 pooled, negative on three of five individually): the image arm has caught the text arm. That the gain survives five graph-drawing engines the training never used is the strongest anti-memorisation evidence here, because our in-domain augmentation randomises some twenty drawing parameters but hard-codes the `dot` engine, making layout the one visual axis held out by construction.

On ChessQA and MMTab the transfer is a level gain, not a gap closure, and neither benchmark has a representation set matching ours. ChessQA ships no image arm; across its three *text* renderings the FEN − piece-list gap *widens* by 0.021 $[0.002, 0.041]$, because the FEN arm is pinned at 0.1–0.3% by truncation and has no headroom. On MMTab the image arm is the stronger one ($+2.7$ pt over the text grid for the base), because their renders are clean and their grid is degraded as text, so "image gains twice what text does" is correct and "the gap narrowed" is not.

Gains are not uniform across subsets. On MMTab, TAT-QA $+3.37$, AIT-QA $+2.05$ and TABMWP $+0.92$ improve while HiTab regresses -1.08, a surface-form artefact that disappears under light normalisation (-0.13; MMTab sensitivity analyses, below). Our pre-registered subset ordering by similarity to our training came out nearly reversed, a failed prediction.

GVLQA: a prior shift, not a per-item recalibration. On GVLQA's `cycle` task the trained model applies a roughly constant -0.06 shift to $P(\text{yes})$ on every layout. Where the base over-answered "Yes" accuracy rises; on the two layouts where it already *under*-answered ($P(\text{yes}) = 0.28, 0.29$ against a gold rate of 0.49) accuracy falls by 4.0 and 5.4 points. The correction does not depend on the direction of the error.

Table 5: Image-arm accuracy per question category, base → trained model, and the share of headroom recovered, $\Delta/(100 - \text{base})$. ^{G}Global: the answer can lie anywhere in the picture, so the model must search or compare across it; every other category is local, naming the node, pair of nodes, square, rank or file that holds the answer. Pooled over both domains and excluding `triangle`, local categories recover 42% of their headroom and global ones 14% (§9).

Graphs (n=90 per category)					**Chess** (n=100 per category)				
category	base	ckpt	Δ	rec.	category	base	ckpt	Δ	rec.
`followers_count`	54.4	78.9	**+24.4**	54%	`rank_occupants`	26	74	**+48**	65%
`followees_list`	61.1	76.7	**+15.6**	40%	`rank_count`	36	62	**+26**	41%
`followees_count`	71.1	85.6	**+14.4**	50%	`piece_square`^{G}	40	65	**+25**	42%
`followers_list`	50.0	61.1	**+11.1**	22%	`same_colour_pair`	80	97	**+17**	85%
`edge_exists`	90.0	96.7	+6.7	67%	`file_count`	9	24	**+15**	16%
`mutual_following`	94.4	98.9	+4.4	80%	`square_colour`	80	94	**+14**	70%
`most_followed`^{G}	54.4	57.8	+3.3	7%	`total_comparison`^{G}	38	47	+9	15%
`least_followed`^{G}	57.8	61.1	+3.3	8%	`has_piece_type`^{G}	74	81	+7	27%
`common_follower`	75.6	76.7	+1.1	5%	`pawn_count`^{G}	23	28	+5	6%
`triangle`‡	24.4	18.9	−5.6	–	`most_occupied_file`^{G}	9	13	+4	4%

‡∼90% of `triangle` responses hit the token cap without emitting an answer, for both models.

The chess gain does not depend on clutter. The chess image gain is flat across material ($+17.1$, $+16.4$, $+17.6$ over sparse, mid and dense piece-count bands) even though base accuracy falls

monotonically with piece count, so it is not a trick that only works on uncluttered boards. The two chess tasks that need a scan down a column stay far below the piece-list arm ($9\% \rightarrow 13\%$ and $9\% \rightarrow 24\%$): vertical scanning is the failure this training does not fix.

Table 6: GVLQA-AUGLY: the same 8,776 questions over the same graphs, re-rendered by five layout engines the training never used (`dot`, the sixth, is the in-distribution control). Accuracy is the image arm; the gap joins GVLQA-BASE's text arm by item id. Bootstrap CIs over items, 2,000 draws.

layout	base	ckpt	Δ	base gap	ckpt gap	gap Δ [95% CI]
`circo`	0.4994	0.5258	**+0.0263**	+0.0336	+0.0097	−0.0239 [−0.0288, −0.0190]
`fdp`	0.4901	0.5215	**+0.0314**	+0.0430	+0.0139	−0.0291 [−0.0346, −0.0238]
`neato`	0.5309	0.5562	**+0.0253**	+0.0022	**−0.0207**	−0.0229 [−0.0285, −0.0173]
`sfdp`	0.5141	0.5397	**+0.0255**	+0.0189	**−0.0042**	−0.0231 [−0.0285, −0.0179]
`twopi`	0.5188	0.5444	**+0.0256**	+0.0142	**−0.0090**	−0.0232 [−0.0283, −0.0183]
pooled	**0.5107**	**0.5375**	**+0.0268**	**+0.0224**	**−0.0021**	−0.0245 [−0.0268, −0.0222]
`dot` (control)	*0.4933*	*0.5267*	*+0.0334*	+0.0398	+0.0088	−0.0310

Micro and macro disagree on GVLQA, and both are honest. `connectivity` is 56% of the split, so any micro-average is largely a report on it, and it carries essentially the whole image-arm effect: pooled over the five unseen layouts the image arm gains $+5.19$ points there while five of the remaining six tasks are inside noise and `cycle` is negative (-1.45). Macro over the seven tasks is $+0.0049$ against micro's $+0.0268$. Four of seven tasks score in the single digits to low tens for *both* models under the benchmark's no-reasoning-allowed format suffix, so their deltas carry almost no information. We quote micro with its fix/break counts and put macro beside it; neither should be quoted alone.

GVLQA's pooled gap understates the phenomenon. One task (`cycle`) has a degenerate text arm: both models answer "Yes" to 99.8% of its text-arm questions, giving accuracy 0.497 (exactly the base rate) against a near-calibrated image arm at 0.76. It therefore shows a -26-point inverted gap that drags the pooled figure. Dropping its 1,230 items, the base representation gap is $+0.0847$ rather than $+0.0357$ and the narrowing is -0.0322 $[-0.0379, -0.0258]$. This is a finding about the benchmark, not about our checkpoint.

MMTab sensitivity analyses. MMTab's own metric applies no normalisation whatsoever, which costs *both* models $\sim$12 points of absolute accuracy. Under light normalisation (lowercase; strip `$`, `%`, commas and a trailing period; canonicalise numbers) the base goes $62.86\% \rightarrow 74.72\%$ and the checkpoint $63.71\% \rightarrow 75.35\%$; the training delta shrinks to $+0.63$ points and HiTab's regression disappears. One stratum is broken by design: TABMWP's multiple-choice prompt lists options as `(A) ... (B) ...` and never says how to answer, the gold is the option *text*, and the model answers with the *letter* on 3,463 of 3,984 rows. As scored it is 12.5%, and mapping letters back to option text puts it at 85.4% with the checkpoint's apparent $+1.18$-point gain vanishing to $+0.03$. With both repairs, pooled accuracy is $89.81\% \rightarrow 90.20\%$ ($+0.39$) and the entire transfer effect is TAT-QA ($+4.15$) and AIT-QA ($+1.08$), the two financial-table subsets, both of which need numeric reasoning over a dense layout. Two-thirds of MMTab TQA's measured error for this model is answer formatting, not table understanding. We quote the as-scored numbers as the headline because they are MMTab's own metric and the only ones comparable to anyone else's MMTab result.

ChessQA is prompt-style dependent, and both runs are reported. The benchmark's `format_task_prompt` takes a `prompt_style`; both values used here are upstream's own, differing only in whether one sentence ("Analyze step by step and explain your reasoning.") is present. Under the default style, 85.0% of base and 82.5% of checkpoint responses hit the 2,048-token cap (42% of them ending with their last 200 characters already repeated earlier in the response), and the checkpoint gives $+0.47$ points. Under the alternative style, which drops that sentence, it gives $+1.47$ points. The alternative also costs the *base* model accuracy on the items it completes (51.7% $\rightarrow$ 28.7% conditional on completion), so it buys completions and pays in reasoning. The defensible framing is that the effect is real under a regime that lets the model finish and absent under one where both models ruminate past the cap; the truncation improvement itself is present under *both* styles (-2.45 and -3.15 points).

Molecules: five MolLangBench task types held out of training. MolLangBench's recognition test split has five task types (phthalimide, steroid, alpha amino acid, penam and glucose-like pyranose; 200 molecules each) that contributed no stage-1 generation, pair row, rollout or anchor row, so the set is new on the task axis. It is new on the molecule axis too: none of its 1,000 molecules matches any of the 32,600 rows we trained or scored on, by exact string, canonical SMILES with or without stereochemistry, or InChIKey. It is not a separate benchmark (the tasks ship in the release the domain was built from), and the prompts are ours, because the release ships none for these tasks; ours copy the shipped substructure-counting prompts element for element and differ only in the chemical definition and the atom count. Both arms the release ships are scored, n=1,000 each, with the Table 3 checkpoint. The image arm goes $44.50\% \rightarrow 46.50\%$ (102 fixed, 82 broken) and SMILES $13.70\% \rightarrow 13.40\%$. Truncation on the image arm falls $0.434 \rightarrow 0.382$ and is lower on all five tasks, while SMILES stays at 83%. On the 524 rows both models answer gradably the checkpoint is 2.48 points *worse* (11 fixed, 24 broken), and 91 of the image arm's 102 fixes are base truncations: the checkpoint converts truncations into answers, and the recovered answers are right slightly more often than wrong. The defensible claim is that the termination behaviour transfers to unseen tasks and the accuracy gain is not established.

Table 7 breaks this down by task: the image arm moves by more than two points only on `penam`, while its truncation falls on all five. Two further splits sharpen the reading. First, 44% of the items have a gold count of zero, answerable by reporting that the molecule contains none of the group. Nearly all of the SMILES arm's correct answers are of this kind: on molecules that do contain the group it answers 11 of 560 correctly before training and 8 after. The image arm does localise the group, but gets only about a fifth of those rows right ($19.8\% \rightarrow 20.9\%$). Second, on these tasks the image is the strong arm, and the image–SMILES gap widens slightly, from 30.8 to 33.1 points, because only the image arm moves; the gap widens in the same direction on the trained molecule tasks (§5).

Table 7: The five MolLangBench task types held out of training, 200 molecules each: accuracy per arm and the image arm's truncation rate, base → trained model (%).

Task	Image accuracy	Image truncation	SMILES accuracy
`phthalimide`	43.0 → 45.0	48.5 → 41.5	8.5 → 7.5
`steroid`	45.5 → 45.5	46.0 → 43.0	14.5 → 13.5
`alpha_amino_acid`	54.5 → 53.5	36.5 → 32.5	14.5 → 14.5
`penam`	32.5 → 39.5	51.5 → 44.5	17.0 → 17.0
`glucose_like_pyranose`	47.0 → 49.0	34.5 → 29.5	14.0 → 14.5
all five	44.5 → 46.5	43.4 → 38.2	13.7 → 13.4

D PROMPTS

Teacher (stage 3). The teacher prompt is self-contained (it is not the student's prompt with a hint appended) and contains, in order: the shared question; the representation that led to the correct answer, its reasoning trace delimited by `<correct_trace>` tags, and its answer marked correct; the representation that led to the wrong answer, its trace in `<failed_trace>` tags, and its answer marked wrong; and a final line restating the strong representation's answer. The task block instructs the model to rephrase *only* the failed trace so that following it step by step reaches that answer, to change only the steps that caused the error, to stay grounded in the wrong representation (the correct one is provided "ONLY as a reference for what sound reasoning looks like"), to keep the failed trace's own narrative voice, to add no meta-commentary and never to hint that it was shown the other representation or the answer, and to end with exactly one `<answer>` tag. When either side of the pair is the image arm, the datapoint's pixels are attached to the teacher request. The rule against hinting at the other representation hides where the trace's claims came from, so a corrected trace can state an observation taken from the strong representation as if read from the weak one (§10).

Two design decisions were measured rather than argued. *The teacher gets no notation key.* Showing each representation under a bare type label, without the student's per-representation "meaning" bullets, was compared against including them over 18,857 rollouts: usable rollouts were flat (56.97% with, 57.14% without, inside a 37% per-pair churn from resampling at T=0.4) while kept completions

grew from a median 597 to 734 characters, which the rephrasing stage then has to undo. The teacher never lacked the notation: all 5,151 of its wrong answers were well-formed values of the correct type, so it reads every representation correctly and then miscounts. *The output is not wrapped in tags*, because the completion is trained as the student's own response and must read as a direct answer.

Rephrasing (stage 4). The rewrite prompt asks for one confident line of reasoning: keep only what leads to the answer, remove all second-guessing and re-verification and state the corrected reasoning as if it were right the first time, aim for 30–60% of the original length, add no new steps or facts, keep every identifier, name and number exactly and lists in the same order, and reproduce the final `<answer>` tag character for character as the very last thing in the output. Rule 7 is the one worth naming: it *preserves* the trace's existing opening, closing, voice and layout rather than prescribing a house style. An earlier version mandated a fixed opener and closer measured off one domain's traces (65% opened "To…", 64% closed "Therefore"); the same measurement over 2,847 chess traces gives 15% and 35%, with no dominant opener to mandate instead. Since the rewrite *is* the training target, it must stay inside the student's own style distribution; an off-style target would train a style shift on top of the de-rumination.

Leakage filter. Both stages reject a completion containing any of a fixed list of substrings that betray the privileged framing (`rephras`, `privileged`, `failed trace`, `other representation`, `verified answer`, and similar), matched case-insensitively and chosen to be specific enough that legitimate reasoning never matches. A rejected rollout is dropped in stage 3; in stage 4 the row falls back to its unrephrased completion. Leakage accounts for 0.0–1.1% of rollouts depending on domain and arm.

Teacher brevity: a pilot that argued against the obvious fix. Chess teacher traces are long (mean 250 tokens against the graph domain's 110), so a 760-row pilot compared the teacher template with and without a harsh brevity rule before committing to a multi-hour rollout run. The rule traded truncation for answer errors at *identical* yield: truncation 21.8% $\rightarrow$ 13.9%, wrong or missing answer 24.5% $\rightarrow$ 31.4%, kept 53.4% vs. 54.3%. On the 602 prompts both pilots share the paired picture is worse than the totals suggest: the rule rescues 43 truncated rollouts into usable ones but breaks 64 that were already fine, net -21. Shorter traces are not free: the rollouts that stop in time often stop by dropping the reasoning that got the answer right. The no-rule template was used, and brevity was left to the rephrasing stage, where it is applied to a trace that has *already* been graded correct.