Structural Rationale Distillation via Reasoning Space Compression

Jialin Yang[1,3]* **Jiankun Wang**[2]* **Jiajun Wu**[1,3]
Henry Leung[1] **Jiayu Zhou**[2] **Steve Drew**[1,3]†
[1]University of Calgary [2]University of Michigan [3]TAIH

Abstract

When distilling reasoning from large language models (LLMs) into smaller ones, teacher rationales for similar problems often vary wildly in structure and strategy. Like a chef who makes the same dish differently each time, this inconsistency burdens the student with noisy supervision that is hard to internalize. We propose Distillation through Reasoning Path Compression (D-RPC), which constrains the teacher to follow a compact, dynamically maintained bank of reusable high-level reasoning paths. For each training question, D-RPC retrieves the most relevant path and conditions the teacher to follow it, producing rationales that are consistent across similar problems yet diverse enough to cover different problem types. A PAC-Bayes analysis formalizes a trade-off between bank size and coverage, with smaller banks limiting path diversity at the risk of coverage gaps. Our ablations favor an intermediate bank size, consistent with this trade-off. Across five math and commonsense reasoning benchmarks with two student models, D-RPC achieves the highest average accuracy among the evaluated distillation methods, while using fewer tokens than template-heavy alternatives.

1 Introduction

Knowledge distillation trains a small language model (SLM) to reason by imitating a large language model (LLM) teacher [Shridhar et al., 2023, Magister et al., 2023]. However, the teacher's rationales for similar problems often vary substantially in structure and strategy. An apprentice chef shown a different recipe for the same dish each day learns more slowly, even when every recipe is valid. Distillation faces the same dynamic. Asked the same type of math problem repeatedly, the teacher may generate a "compute rate, then combine quantities" solution once and an "estimate total, then subtract difference" solution the next [Madaan et al., 2023, Arcuschin et al., 2025]. This *rationale divergence* has three costs. First, the student sees many one-off patterns instead of reusable strategies, making internalization harder [Chen et al., 2025b]. Second, covering all valid solution paths requires many teacher generations, inflating cost [Zelikman et al., 2022]. Third, highly variable supervision can amplify hallucination during fine-tuning [Shridhar et al., 2023].

Prior work has tried to mitigate this variance through structured reasoning. Approaches like Buffer-of-Thought [Yang et al., 2024] and ReasonFlux [Yang et al., 2025a] reuse hierarchical thought templates to improve LLM reasoning at inference time. However, these verbose templates impose heavy context loads that SLMs, with their limited capacity and context windows, struggle to exploit [Magister et al., 2023, Zou et al., 2025b]. The core question remains open: *can we compress the teacher's reasoning space into a compact, consistency-preserving structure before distilling it to a student?*

We propose Distillation through Reasoning Path Compression (D-RPC), which guides teacher rationale generation with a compact bank of canonical, high-level reasoning paths.[3] Just as a chef

*Equal contribution.

†Corresponding author: steve.drew@ucalgary.ca

[3]Code and data are available at https://github.com/denoslab/D-RPC.

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

who consults a cookbook produces more consistent dishes than one who improvises freely, a teacher conditioned on a shared set of reasoning paths produces more consistent supervision. D-RPC works in two stages. First, it builds the cookbook: a seed set of teacher-solved problems is clustered by intent to identify canonical solution strategies, forming the *reasoning path bank*. Second, it teaches from the cookbook: for each new training question, the most relevant paths are retrieved from the bank, and the teacher generates a detailed rationale conditioned on the retrieved path. If the teacher discovers a novel valid solution not in the bank, it is buffered and periodically merged back so that the cookbook grows over time. The student SLM is then fine-tuned via LoRA on the resulting consistent supervision.

Our main contributions are as follows.

1. We propose D-RPC, which compresses the teacher's reasoning space into a compact, dynamically maintained reasoning path bank. By routing similar questions to shared canonical paths, D-RPC reduces rationale divergence and provides more consistent supervision for SLMs.
2. We formalize a trade-off between bank size and coverage via a PAC-Bayes generalization bound with an explicit $\log K_{\mathrm{bank}}$ term. Smaller banks tighten the path-entropy bound but can increase coverage slack, a trade-off reflected in the bank-size ablation.
3. Across five benchmarks, D-RPC achieves the highest average accuracy for both Llama 3.1 8B Instruct and Qwen 3 1.7B among the evaluated methods. It also uses fewer tokens than template-heavy alternatives.

2 Related Work

Distilling reasoning from teachers to students. UNICOTT iteratively constructs structured explanations to reduce supervision noise [Zhuang et al., 2025]. COTCD introduces difficulty-aware curricula for stable student learning [Yu et al., 2025], and RLKD captures implicit multi-branch structures instead of mimicking a single trajectory [Xu et al., 2025]. ReasonFlux-PRM scores intermediate steps to filter distillation data and provide dense rewards [Zou et al., 2025a]. MCC-KD generates multiple rationales per question and minimizes a bidirectional KL divergence across their answer distributions to enforce *intra-question* consistency [Chen et al., 2023], and MIND addresses misalignment between a teacher's "optimal" rationale and the student's evolving capacity by synthesizing capability-adaptive multi-perspective supervision through a teaching-assistant network [Cui et al., 2026]. D-RPC focuses on structural reuse across related questions. Its compact path bank guides the teacher before rationale generation. This approach can complement methods that adapt supervision to the student's current abilities.

Reusing reasoning processes at inference time. A parallel line of work treats reasoning as a reusable process rather than a one-off generation. Buffer of Thought and ReasonFlux reuse thought templates [Yang et al., 2024, 2025a]. RoT retrieves thought steps from similar problems [Ahmed et al., 2025], and Thought Propagation reuses solutions through analogical propagation [Yu et al., 2024]. In the agent setting, ReasoningBank [Ouyang et al., 2026] accumulates a memory of natural-language strategies extracted from past successful and failed trajectories and merges them into the agent's prompt, with no weight updates. All of these methods improve LLM or agent performance at inference time by strengthening prompts. However, they do not address the variance of the supervision signal used to train smaller students. D-RPC instead operates at *distillation time*, compressing the teacher's supervision distribution so that consistent reasoning structure is internalized into the student's weights and no retrieval is required at deployment.

Reasoning-path supervision for SFT. Skip-Thinking refines CoTs through block training, allowing non-inference blocks to be skipped [Chen et al., 2025a]. EDIT identifies key decision-making steps by comparing paired CoTs and upweights them [Dai et al., 2025]. QR-Distill improves multi-path refinement through quality filtering and peer model teaching [Lei et al., 2025]. Reasoning Scaffolding abstracts teacher rationales into sequences of discrete semantic signals and trains the student to predict them alongside each step [Wen et al., 2025], enforcing structure within each trace rather than across related questions. D-RPC uses a shared bank to encourage reusable structures during teacher generation. This design can be combined with rationale filtering, step reweighting, or scaffolding during student training.

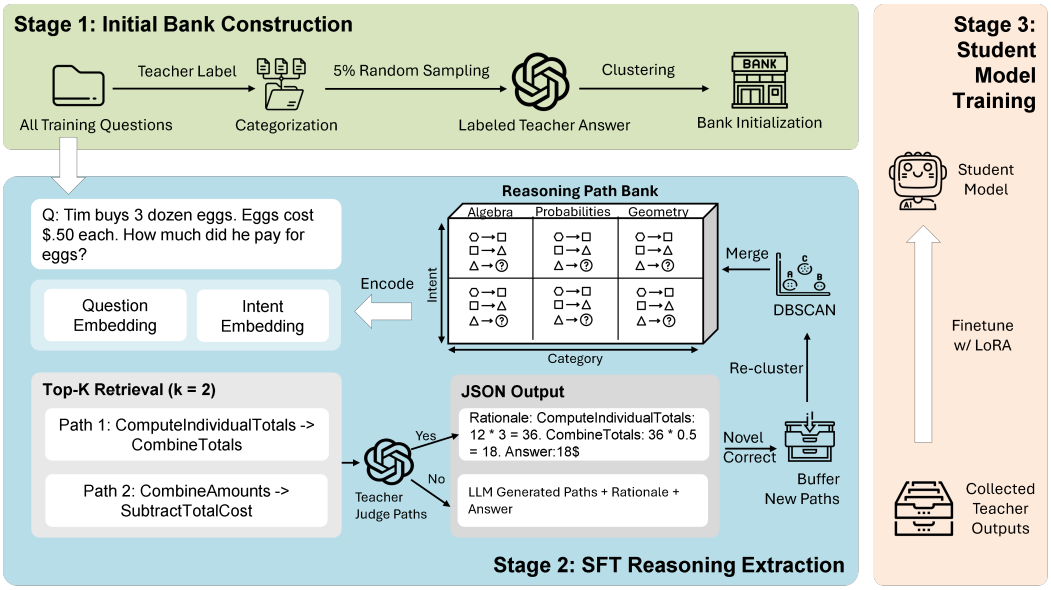

Figure 1: D-RPC pipeline. Stage 1, Reasoning bank initialization. A seed subset of questions is solved by the teacher; the resulting reasoning paths are clustered by intent to form the initial reasoning path bank. Stage 2, Bank-guided supervision. For each training question, retrieve the top-K_{ret} candidate paths, condition the teacher to follow one, and collect the supervision tuple. Novel correct paths are buffered and periodically merged back into the bank via re-clustering. Stage 3, Student model training. The student SLM is fine-tuned with LoRA on the collected supervision tuples.

3 Methodology

Returning to our cooking analogy, D-RPC builds a cookbook, teaches from it, and trains the student. In the first stage, it assembles a compact bank of canonical, high-level reasoning paths by having the teacher solve a small seed set and clustering the resulting solution strategies. In the second stage, it retrieves the best-matching path for each new training question, conditions the teacher on that path, and collects the resulting supervision tuples. In the third stage, the student SLM is fine-tuned on the collected supervision via LoRA. Figure 1 illustrates the full pipeline; Algorithm 1 gives the pseudocode. In short, D-RPC takes a teacher LLM and a question set and produces two outputs: a reasoning path bank and a LoRA-trained student. Everything in between is bank construction (Stage 1), bank-guided supervision (Stage 2), and student training (Stage 3).

3.1 Notation and setup

Let $\{X_i\}_{i=1}^{N}$ be a dataset of training questions. The teacher LLM labels each question with a *category* C_i such as "Arithmetic" and a finer-grained *intent* T_i such as "compute unit rate". For distillation, the teacher produces a supervision tuple $Y_i = (\Pi_i, R_i, A_i)$ where Π_i is a *reasoning path*, meaning a short ordered sequence of abstract steps such as "compute the unit rate, then combine quantities"; R_i is a *detailed rationale* that instantiates the path with concrete computations; and A_i is the final answer. We write $\mathcal{P}$ for the universe of high-level reasoning paths (ordered sequences of abstract steps) and $\mathcal{C}$ for the set of all categories. The student is trained on (X_i, Y_i) only; category and intent labels are used only to build and query the bank.

3.2 Reasoning bank initialization

Categorize all questions. The teacher assigns a category–intent pair to every question: $(C_i, T_i) = f_{\text{cat}}(X_i)$. We then draw a category-balanced seed set $\mathcal{S}_{\text{seed}}$ (5% of training data in our experiments), sampling proportional to $\hat{p}(C)$ so that every problem type is represented.

Elicit reasoning paths from the seed set. For each seed question $X_i \in \mathcal{S}_{\text{seed}}$, the teacher generates a high-level reasoning path and answer: $(\Pi_i, A_i) = f_{\text{path}}(X_i, C_i, T_i)$.

Cluster intents to form the bank. Within each category c, we embed the teacher-generated intent strings via $e(t) = \text{Embed}(t)$ and cluster them with DBSCAN [Ester et al., 1996]. Each cluster yields a *canonical intent* $\tilde{t}$—a representative label for a group of semantically similar intents; for example,

"unit-rate computation," "compute unit rate," and "rate-times-quantity" may collapse into the single canonical intent "unit-rate computation." We denote the set of canonical intents for category c by $\tilde{\mathcal{T}}(c)$. The resulting *reasoning path bank* $\mathcal{B}$ maps each (category, canonical intent) pair to its set of observed reasoning paths: $\mathcal{B}(c, \tilde{t}) \subseteq \mathcal{P}$. The total number of distinct paths stored across all bank entries is $K_{\text{bank}} := \big|\bigcup_{c \in \mathcal{C}} \bigcup_{\tilde{t} \in \tilde{\mathcal{T}}(c)} \mathcal{B}(c, \tilde{t})\big|$.

3.3 Bank-guided supervision

Routing and retrieval. For a new training question X_j, the teacher predicts $(C_j, T_j) = f_{\text{cat}}(X_j)$. Because the predicted intent string T_j can be noisy [Madaan et al., 2023], we match X_j to a canonical intent via the question embedding rather than the surface form of T_j: $\tilde{T}_j = \arg\max_{\tilde{t} \in \tilde{\mathcal{T}}(C_j)} \cos\big(e(X_j), e(\tilde{t})\big)$. We then retrieve the top-K_{ret} candidate paths from the matched bank entry: $\mathcal{P}^{(K_{\text{ret}})}(X_j) = \text{TopK}_{\Pi \in \mathcal{B}(C_j, \tilde{T}_j)} \cos\big(e(X_j), e(\Pi)\big)$.

Path-guided rationale generation. The teacher receives the question and the retrieved candidate paths, picks the best-fitting one (or conservatively refines one if none fits directly), and generates a detailed rationale following it: $(\Pi_j, R_j, A_j) = f_{\text{teach}}\big(X_j, \mathcal{P}^{(K_{\text{ret}})}(X_j)\big)$, where Π_j is the path actually used. Details of the teacher prompt are in Appendix B.1. Every resulting tuple $(X_j, (\Pi_j, R_j, A_j))$ is added to the supervision dataset $\mathcal{D}_{\text{SFT}}$ regardless of answer correctness.

Dynamic bank refinement. If the teacher produces a path $\Pi_j \notin \bigcup_{c,\tilde{t}} \mathcal{B}(c, \tilde{t})$ and the answer is correct, we buffer the record (C_j, T_j, Π_j) into a set $\mathcal{M}$. Once $|\mathcal{M}| \geq \tau_{\text{buf}}$, we re-cluster: the raw intents T_j are merged into the canonical intent clusters $\tilde{\mathcal{T}}(c)$, and the new paths are added to the corresponding bank entries $\mathcal{B}(c, \tilde{t})$. This allows the bank to grow beyond its initial coverage.

3.4 Student fine-tuning

With $\mathcal{D}_{\text{SFT}}$ collected, we fine-tune the student SLM using LoRA [Hu et al., 2021]: the backbone is frozen and only low-rank adapters θ are trained. The objective is the standard supervised NLL:

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{(X,Y) \sim \mathcal{D}_{\text{SFT}}}[-\log p_\theta(Y \mid X)], \qquad Y = (\Pi, R, A).$$

Because related questions share reasoning paths, the supervision signal has lower variance than unconstrained teacher generation, making it easier for the student to internalize reusable solution strategies. The full D-RPC algorithm is given in Algorithm 1.

4 Theoretical Analysis

Why does a moderate-sized bank work best? In learning terms, an overly small bank induces high approximation bias because many examples are assigned to mismatched path supports, whereas an overly large bank increases supervision entropy and reduces parameter sharing across examples. The best regime balances coverage against compression. This section formalizes that intuition. We first state the main result, then unpack the machinery behind it.

Scope. The analysis below applies specifically to the LoRA-based SFT instantiation described in Section 3. Proposition 1, which controls entropy via banking, is general, but the KL-control argument in Propositions 2–3 relies on the LoRA parameterization. Convergence and information-theoretic guarantees for low-rank distillation more broadly have been studied in [Soarez et al., 2026]; our result is complementary in that the supervision-entropy floor is controlled explicitly through the bank-size term $\log K_{\text{bank}}$, which their analysis does not consider. Extending to other fine-tuning regimes would require a different complexity argument.

4.1 Main result: the trade-off between bank size and coverage

Theorem 1 (Banked LoRA distillation bound)**.** *Under Assumptions 1–4 (stated below), let $\hat{\theta}_S$ be the regularized LoRA minimizer and define a Gaussian posterior $\mathsf{Q}_S = \mathcal{N}(\hat{\theta}_S, \sigma^2_{\text{post}} I_m)$ with prior $\mathsf{P} = \mathcal{N}(0, \sigma_0^2 I_m)$. Then for any $\delta \in (0,1)$, with probability at least $1 - \delta$ over the training set S,*

$$L(\mathsf{Q}_S) \;\leq\; \underbrace{\widehat{L}_S(\mathsf{Q}_S)}_{\textit{training loss}} + \underbrace{\sqrt{\frac{2\tau \cdot M \cdot N}{n}}}_{\textit{generalization gap}} + \underbrace{\frac{c\tau \cdot N}{n}}_{\textit{lower-order}}, \tag{1}$$

Algorithm 1 Distillation through Reasoning Path Compression (D-RPC)

Require: Questions $\{X_i\}_{i=1}^N$; teacher function $f_{\text{cat}}, f_{\text{path}}, f_{\text{teach}}$; encoder $e(\cdot)$; retrieval size K_{ret}; buffer size τ_{buf}.
Ensure: Bank $\mathcal{B}$; supervision $\mathcal{D}_{\text{SFT}}$; LoRA params θ.

for $i = 1$ **to** N **do**
  $(C_i, T_i) \leftarrow f_{\text{cat}}(X_i)$
end for
$\mathcal{S}_{\text{seed}} \leftarrow \textsc{CategoryBalancedSample}(\{X_i\}, \{C_i\})$
for $X_i \in \mathcal{S}_{\text{seed}}$ **do**
  $(\Pi_i, A_i) \leftarrow f_{\text{path}}(X_i, C_i, T_i)$
end for
$(\mathcal{B}, \{\tilde{\mathcal{T}}(c)\}) \leftarrow \textsc{InitBank}(\mathcal{S}_{\text{seed}}, \{(C_i, T_i, \Pi_i)\})$
$\mathcal{D}_{\text{SFT}} \leftarrow \emptyset$; $\mathcal{M} \leftarrow \emptyset$
for $j = 1$ **to** N **do**
  $(C_j, T_j) \leftarrow f_{\text{cat}}(X_j)$
  $\tilde{T}_j \leftarrow \arg\max_{\tilde{t} \in \tilde{\mathcal{T}}(C_j)} \cos(e(X_j), e(\tilde{t}))$
  $\mathcal{P}^{(K_{\text{ret}})} \leftarrow \text{TopK}_{\Pi \in \mathcal{B}(C_j, \tilde{T}_j)} \cos(e(X_j), e(\Pi))$
  $(\Pi_j, R_j, A_j) \leftarrow f_{\text{teach}}(X_j, \mathcal{P}^{(K_{\text{ret}})})$
  $\mathcal{D}_{\text{SFT}} \leftarrow \mathcal{D}_{\text{SFT}} \cup \{(X_j, (\Pi_j, R_j, A_j))\}$
  if $\textsc{IsCorrect}(A_j)$ **and** $\Pi_j \notin \bigcup_{c,\tilde{t}} \mathcal{B}(c, \tilde{t})$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{(C_j, T_j, \Pi_j)\}$
  end if
  if $|\mathcal{M}| \geq \tau_{\text{buf}}$ **then**
    $\mathcal{B} \leftarrow \textsc{Recluster}(\mathcal{B}, \mathcal{M})$; $\mathcal{M} \leftarrow \emptyset$
  end if
end for
$\theta \leftarrow \arg\min_\theta \ \mathbb{E}_{(X,Y)\sim\mathcal{D}_{\text{SFT}}}\big[-\log p_\theta(Y \mid X)\big]$
return $\mathcal{B}, \theta$

where $c > 0$ is a universal constant, τ bounds the per-example loss, and:

$$M := \log K_{\text{bank}} + \mathbb{E}[\text{H}(R \mid X, \Pi)] + \mathbb{E}[\text{H}(A \mid X, \Pi, R)] + \varepsilon, \tag{2}$$

$$N := \underbrace{\tfrac{1}{\sigma_0^2 \lambda} \widehat{L}_S(0)}_{\textit{frozen-model fit}} + \tfrac{m}{2}(\rho - 1 - \ln \rho) + \ln \tfrac{1}{\delta}. \tag{3}$$

The generalization gap scales with $\sqrt{M \cdot N}$. The bank size K_{bank} enters through M via $\log K_{\text{bank}}$, creating two opposing forces. Making the bank smaller reduces $\log K_{\text{bank}}$, so supervision is more consistent, but increases the coverage slack ε because some problems have no good path, raising M from the other side. Conversely, making the bank larger reduces ε by improving coverage, but increases $\log K_{\text{bank}}$ by weakening consistency, also raising M. An intermediate bank size can balance reuse and coverage by lowering $M(K_{\text{bank}})$. The accuracy pattern in Table 4 is consistent with this trade-off. Banking can also reduce N by lowering the frozen-model loss $\widehat{L}_S(0)$, since banked rationales are more predictable to the pretrained student.

4.2 Assumptions and supporting results

We now state the assumptions and intermediate results that compose into Theorem 1. Full proofs are deferred to the appendix.

Setup. The student is an autoregressive model $p_\theta(y \mid x)$ with frozen backbone; only LoRA weights $\theta \in \mathbb{R}^m$ are trained. The loss is the NLL: $\ell(\theta; z) = -\log p_\theta(y \mid x)$. Population and empirical risks are $L(\theta) = \mathbb{E}[\ell(\theta; Z)]$ and $\widehat{L}_S(\theta) = \frac{1}{n}\sum_i \ell(\theta; Z_i)$. For a distribution Q over θ, the Gibbs risk is $L(\mathsf{Q}) = \mathbb{E}_{\theta\sim\mathsf{Q}}[L(\theta)]$.

Assumption 1 (Bounded NLL)**.** *$0 \leq \ell(\theta; Z) \leq \tau$ a.s. for all θ in the posterior support (justified by length normalization or probability lower bounds).*

Part I: Banking controls supervision entropy (general).

Assumption 2 (Path-bank support)**.** *Under the banked regime, $\Pi \in \mathcal{B}_{\text{path}}$ a.s. given X, where $\mathcal{B}_{\text{path}} := \bigcup_{c,\tilde{t}} \mathcal{B}(c, \tilde{t})$ is the global path set and $|\mathcal{B}_{\text{path}}| = K_{\text{bank}}$.*

Proposition 1 (Banking reduces path uncertainty)**.** *Under Assumption 2, the conditional entropy of the supervision satisfies*

$$\mathbb{E}[\mathrm{H}(Y \mid X)] \;\le\; \log K_{\mathrm{bank}} + \mathbb{E}[\mathrm{H}(R \mid X, \Pi)] + \mathbb{E}[\mathrm{H}(A \mid X, \Pi, R)]. \tag{4}$$

This follows from the chain rule: $\mathrm{H}(Y \mid X) = \mathrm{H}(\Pi \mid X) + \mathrm{H}(R \mid X, \Pi) + \mathrm{H}(A \mid X, \Pi, R)$, with $\mathrm{H}(\Pi \mid X) \le \log K_{\mathrm{bank}}$ by finite support (proof in Appendix E).

Assumption 3 (Near-floor empirical fit)**.** *There exists a posterior* Q *such that*

$$\widehat{L}_S(\mathsf{Q}) \le \mathbb{E}[\mathrm{H}(Y \mid X)] + \varepsilon, \qquad \varepsilon \ge 0. \tag{5}$$

This is an optimization-dependent condition: it says the trained model nearly matches the information-theoretic floor. We verify empirically that this holds for our AQUA experiments in Section 5.7 (Table 6).

Combining Proposition 1 with Assumption 3 and a standard Bernstein-type PAC-Bayes bound, proved in Appendix F, yields a generalization bound where K_{bank} explicitly controls the deviation term through M as defined in Eq. 2.

Part II: Banking can reduce the complexity term (LoRA-specific).

Assumption 4 (Regularized LoRA ERM)**.** *Training returns a minimizer of the regularized objective:* $\hat{\theta}_S \in \arg\min_\theta [\widehat{L}_S(\theta) + \frac{\lambda}{2}\|\theta\|^2]$, $\lambda > 0$.

Proposition 2 (Norm control)**.** *Under Assumption 4:*

$$\|\hat{\theta}_S\|^2 \le \frac{2}{\lambda}\widehat{L}_S(0). \tag{6}$$

Proof. Optimality of $\hat{\theta}_S$ gives $\widehat{L}_S(\hat{\theta}_S) + \frac{\lambda}{2}\|\hat{\theta}_S\|^2 \le \widehat{L}_S(0)$; since $\widehat{L}_S(\hat{\theta}_S) \ge 0$, the result follows. □

Proposition 3 (KL bound)**.** *For Gaussian prior* $\mathsf{P} = \mathcal{N}(0, \sigma_0^2 I)$ *and posterior* $\mathsf{Q}_S = \mathcal{N}(\hat{\theta}_S, \sigma_{\mathrm{post}}^2 I)$ *with* $\rho = \sigma_{\mathrm{post}}^2/\sigma_0^2$*:*

$$\mathrm{KL}(\mathsf{Q}_S \| \mathsf{P}) \le \frac{1}{\sigma_0^2 \lambda}\widehat{L}_S(0) + \frac{m}{2}(\rho - 1 - \ln \rho). \tag{7}$$

Proof in Appendix G.

The KL term depends on the frozen-model loss $\widehat{L}_S(0)$. Banking can reduce $\widehat{L}_S(0)$ and hence the complexity penalty N in Theorem 1.

Proposition 4 (Sufficient conditions for frozen-model alignment)**.** *Under any of the following conditions, banking reduces the frozen-model population cross-entropy* ($\mathcal{L}_0^{(\star)} := \mathbb{E}_X \mathbb{E}_{Y \sim p^\star(\cdot|X)}[-\log p_0(Y \mid X)]$ *for regime* $\star \in \{\mathrm{bank}, \mathrm{free}\}$*):*

- ***D1.*** Bank as projection: *the banked target conditional minimizes the frozen-model cross-entropy over all distributions supported on* K_{bank} *paths.*
- ***D2.*** Canonicalization improves frozen likelihood pointwise: *there exists a map* c *on targets such that* $-\log p_0(c(y) \mid x) \le -\log p_0(y \mid x)$ *for all* (x, y)*, and the banked regime restricts support to such canonical forms.*
- ***D3.*** Entropy drop dominates mismatch increase: *the reduction in* $\mathbb{E}[\mathrm{H}(Y \mid X)]$ *from banking exceeds any increase in* $\mathbb{E}[\mathrm{KL}(p^\star(\cdot \mid X) \| p_0(\cdot \mid X))]$.

Formal statements and proofs are in Appendix H. The conclusion of this proposition is directly testable; Section 5.7 confirms that $\widehat{L}_S(0)$ *is indeed lower under banking than under free generation.*

Proof of Theorem 1. Substitute the KL bound (Proposition 3) into the K_{bank}-explicit PAC-Bayes bound and collect terms into M and N. Full derivation in Appendix I. □

5 Experiments

5.1 Datasets

We evaluate on five benchmarks (see details in Appendix D.1): GSM8K [Cobbe et al., 2021] for grade-school arithmetic, AQUA [Ling et al., 2017] for multiple-choice math, StrategyQA [Geva et al., 2021] for multi-hop commonsense, AI2ARC [Clark et al., 2018] for science reasoning, and MATH [Hendrycks et al., 2021] for competition-level mathematics. These range from relatively simple, such as GSM8K, to challenging, such as MATH, allowing us to test whether reasoning-path compression scales across difficulty levels.

Table 1: Post-distillation accuracy (%) across five benchmarks and two student models. Values are mean ± std over 10 repeated executions with seed 42. **Bold** marks the best result per column. Shading marks D-RPC. Dashes mark inapplicable settings.

	Student: Llama 3.1 8B Instruct ($r = 64$)					
Strategy	GSM8K	AQUA	StrategyQA	AI2ARC	MATH	Avg.
CoT	83.96 (±0.33)	64.02 (±0.77)	73.32 (±1.31)	87.92 (±0.29)	41.57 (±0.35)	70.16
Freeform	81.63 (±0.72)	60.39 (±1.61)	72.27 (±1.27)	92.41 (±0.34)	41.68 (±0.18)	69.68
SuperCorrect	82.42 (±0.46)	59.45 (±2.59)	–	–	36.62 (±0.46)	–
DCoT	81.86 (±0.64)	61.81 (±2.43)	72.53 (±1.81)	80.78 (±12.38)	45.23 (±0.53)	68.44
SGFT	77.94 (±0.69)	60.16 (±2.67)	70.31 (±1.29)	86.71 (±0.76)	36.15 (±0.43)	66.25
D-RPC (Ours)	**85.41** (±0.49)	**67.52** (±1.59)	**74.15** (±1.49)	**92.92** (±0.34)	**48.76** (±0.45)	**73.75**

	Student: Qwen 3 1.7B ($r = 64$)					
Strategy	GSM8K	AQUA	StrategyQA	AI2ARC	MATH	Avg.
CoT	77.92 (±0.29)	59.92 (±0.89)	61.92 (±0.92)	88.78 (±0.40)	49.21 (±0.40)	67.55
Freeform	73.38 (±0.68)	62.99 (±0.83)	61.75 (±2.29)	88.70 (±0.30)	48.09 (±0.26)	66.98
SuperCorrect	76.44 (±0.58)	63.78 (±2.59)	–	–	42.82 (±0.36)	–
DCoT	73.99 (±0.69)	59.65 (±1.37)	**66.16** (±1.39)	85.74 (±2.90)	54.91 (±0.34)	68.09
SGFT	65.44 (±0.59)	49.06 (±10.60)	60.26 (±2.78)	75.99 (±0.18)	29.44 (±0.29)	56.04
D-RPC (Ours)	**78.29** (±0.95)	**74.76** (±1.84)	64.59 (±2.46)	**88.82** (±0.54)	**59.72** (±0.39)	**73.24**

5.2 Implementation Details

Clustering and embedding. Intent strings are embedded with `all-MiniLM-L6-v2` [Reimers and Gurevych, 2019] and clustered via DBSCAN ($\texttt{eps} = 0.25$, $\texttt{min_samples} = 2$). No extensive sweep over clustering or embedding choices was performed; we discuss this limitation in Section 7.

Evaluation protocol. In the main comparisons, all methods use the final checkpoint (epoch 2.0, no validation-based selection), identical LoRA configuration (r=64, α=128), and a maximum output length of 2048 tokens. Unless otherwise specified, we report means and standard deviations over 10 repeated executions with the random seed fixed to 42. The standard deviations capture variability from nondeterministic GPU execution. More details are in Appendix D.2.

Metrics. We report *accuracy* (Acc) (exact match against gold; unparseable outputs count as incorrect) and *format validity* (FV) (accuracy among outputs that parse into valid JSON).

5.3 Baselines

We compare D-RPC against five distillation strategies, all sharing the same teacher (GPT-5.1) and SFT configuration, evaluated on two student models, i.e., Llama 3.1 8B Instruct and Qwen 3 1.7B. As an un-distilled reference, **Direct Prompting** evaluates the raw student without fine-tuning. Among unstructured approaches, **Zero-Shot CoT** [Kojima et al., 2022] distills chain-of-thought rationales and **Freeform** lets the teacher freely determine step count and granularity without predefined paths. Among structured approaches, **DCoT** [Do et al., 2025] directly distills chain-of-thought outputs, **SGFT** [Bi et al., 2024] fine-tunes on structured rationales without retrieval, and **SuperCorrect** [Yang et al., 2025b] uses hierarchical XML thought templates (math-only; not applied to StrategyQA or AI2ARC). Training details and prompts are in Appendices D.2 and B.1.

5.4 Main Results

Table 1 reports absolute post-distillation accuracy across all five benchmarks for both student models. We report absolute accuracy rather than gains over each method's own Direct Prompting baseline, because Direct Prompting baselines differ across methods due to different prompting formats and relative gains can be misleading. The impact of LoRA rank and α is discussed in Appendix D.3.

Llama 3.1 8B Instruct. D-RPC achieves the highest accuracy on all five benchmarks: 85.41% on GSM8K, 67.52% on AQUA, 74.15% on StrategyQA, 92.92% on AI2ARC, and 48.76% on MATH. D-RPC outperforms the next-best method by +3.53 points on MATH and +3.50 on AQUA. D-RPC also surpasses SuperCorrect [Yang et al., 2025b] at substantially lower token cost, as detailed in Appendix D.3.

Qwen 3 1.7B. The same pattern holds with a smaller, different-family student: D-RPC leads on GSM8K, AQUA, AI2ARC, and MATH, and is competitive on StrategyQA, where DCoT is slightly higher. The gains on MATH are especially large at $+10.51$ over CoT, demonstrating that path compression scales to challenging benchmarks and smaller models.

5.5 Analysis

Why D-RPC helps. The gains on MATH and AQUA are consistent with our hypothesis that structural compression improves supervision across related questions. The bank encourages reuse of high-level reasoning paths, while semantic retrieval selects suitable paths for each question. This matching step provides an additional refinement whose benefit varies across models and tasks. D-RPC also outperforms SuperCorrect with fewer tokens in the reported comparisons (Appendix D.3).

Path controls and answer filtering. Table 2 compares D-RPC with controls that preserve its prompt, output schema, and generation budget, using the conditions defined in Appendix D.4. On AQUA, all three path controls lower mean accuracy for both students. On MATH, Empty and Random are slightly higher for both students, while Random-Global is lower for Llama and slightly higher for Qwen. These small MATH gains over D-RPC are below the reported standard deviations across fixed-seed executions. Filtered-SFT increases mean accuracy on MATH and decreases it on AQUA, with teacher answer-error rates reported in Appendix D.5.

Table 2: Path controls and filtered-SFT accuracy (%). Means and sample standard deviations use ten executions with seed 42, evaluated on all 254 AQUA and 5000 MATH test questions. Llama and Qwen denote the students in Table 1.

	AQUA		MATH	
Condition	Llama	Qwen	Llama	Qwen
D-RPC	$67.52_{\pm 1.59}$	$74.76_{\pm 1.84}$	$48.76_{\pm 0.45}$	$59.72_{\pm 0.39}$
Empty	$66.54_{\pm 1.16}$	$72.24_{\pm 1.02}$	$48.89_{\pm 0.42}$	$60.07_{\pm 0.39}$
Random	$65.75_{\pm 2.32}$	$73.90_{\pm 1.80}$	$48.89_{\pm 0.36}$	$60.05_{\pm 0.49}$
Random-Global	$65.43_{\pm 2.44}$	$73.94_{\pm 2.49}$	$48.34_{\pm 0.34}$	$59.83_{\pm 0.41}$
Filtered-SFT	$65.75_{\pm 2.12}$	$74.25_{\pm 1.30}$	$48.84_{\pm 0.45}$	$60.42_{\pm 0.44}$

Structural reuse in teacher supervision. Table 3 shows increasing reuse and decreasing path entropy from Empty through the random controls to D-RPC on both datasets. The student results motivate extending the current embedding similarity retrieval with rationale quality and student learning needs, as discussed in Section 6.

Table 3: Structure of teacher rationales. Each path is the ordered sequence of rationale step names, with empty paths excluded. N counts nonempty rationales and K_{used} counts distinct paths. Reuse is $100(1 - K_{\text{used}}/N)$, and H_{path} is the Shannon entropy of path frequencies within each dataset and condition, in bits.

	AQUA				MATH			
Condition	N	K_{used}	Reuse (%)	H_{path}	N	K_{used}	Reuse (%)	H_{path}
D-RPC	9922	435	95.6	7.16	9562	868	90.9	7.85
Empty	9609	4495	53.2	10.90	9473	5210	45.0	11.66
Random	9875	1678	83.0	9.43	9599	2541	73.5	9.74
Random-Global	9876	1724	82.5	9.48	9581	2565	73.2	9.77
Filtered-SFT	8827	410	95.4	7.13	7482	700	90.6	7.74

5.6 Ablation Study

We conduct three ablations to examine bank construction and guidance. All use identical SFT configurations. Results are averaged over 10 repeated executions with seed 42, following Section 5.2.

Heterogeneous vs. homogeneous initialization. Table 5 compares two seed sets: a **Heterogeneous** set sampled evenly across intent clusters to promote diversity and a **Homogeneous** set drawn from the largest clusters to limit diversity. Heterogeneous initialization yields higher mean accuracy at 1k and 2k examples, while homogeneous initialization performs better from 3k to 5k.

Table 4: Effect of bank size on GSM8K Acc and format validity.

Size	Paths	Acc. (%)	FV (%)
50	49	$83.19_{\pm 0.8}$	$83.32_{\pm 0.8}$
75	75	$\mathbf{84.34_{\pm 0.8}}$	$\mathbf{84.61_{\pm 0.8}}$
100	97	$83.28_{\pm 0.6}$	$83.41_{\pm 0.7}$
125	119	$82.90_{\pm 0.7}$	$83.08_{\pm 0.7}$
150	145	$82.91_{\pm 0.5}$	$82.97_{\pm 0.5}$

Table 5: Heterogeneous vs. homogeneous bank initialization on GSM8K across training set sizes.

Bank	Paths		Acc (%)		FV (%)	
Size	hetero	homo	hetero	homo	hetero	homo
1000	50	57	$\mathbf{83.03_{\pm 0.7}}$	$82.30_{\pm 0.6}$	$\mathbf{83.23_{\pm 0.7}}$	$82.37_{\pm 0.6}$
2000	97	93	$\mathbf{83.24_{\pm 0.8}}$	$83.12_{\pm 0.3}$	$\mathbf{83.38_{\pm 0.8}}$	$83.21_{\pm 0.4}$
3000	151	133	$84.18_{\pm 0.5}$	$\mathbf{84.36_{\pm 0.4}}$	$84.27_{\pm 0.5}$	$\mathbf{84.46_{\pm 0.4}}$
4000	188	186	$83.83_{\pm 0.7}$	$\mathbf{84.87_{\pm 0.6}}$	$83.95_{\pm 0.7}$	$\mathbf{84.95_{\pm 0.6}}$
5000	243	252	$84.58_{\pm 0.7}$	$\mathbf{84.94_{\pm 0.6}}$	$84.69_{\pm 0.7}$	$\mathbf{85.18_{\pm 0.7}}$

Table 6: Empirical bound components on AQUA (per-token NLL). $K_{\text{bank}} = 42$ canonical paths for D-RPC. Test NLL uses teacher-generated targets.

Model	Method	$\widehat{L}_S(0)$	$\widehat{L}_S(\hat{\theta}_S)$	$L_{\text{test}}(\hat{\theta}_S)$	Gap	$\|\hat{\theta}_S\|^2$	Acc. (%)
Qwen 3 1.7B ($r = 64$)	D-RPC	**2.68**	0.30	0.42	**0.11**	2792	**74.0**
	CoT	2.70	0.35	0.55	0.20	2760	61.4
	Free	3.29	**0.27**	**0.38**	0.11	**2732**	63.0
Llama 3.1 8B ($r = 128$)	D-RPC	**2.01**	**0.28**	0.47	0.19	6318	**63.4**
	CoT	2.02	0.38	0.67	0.29	6264	61.8
	Free	2.72	0.29	**0.45**	**0.17**	**6216**	59.4

Bank guidance vs. no bank. Table 1 compares full D-RPC with Freeform generation under the shared teacher and training settings. D-RPC achieves higher accuracy across the reported benchmarks. This comparison evaluates the combined effect of path guidance and output schema.

Bank size: the coverage–compression trade-off. Table 4 varies the number of seed questions from 50 to 150 while fixing the SFT training set at 2k. Mean accuracy peaks at 84.34% with 75 paths from 75 seed questions, and is lower with 49 paths or 119 to 145 paths. This pattern is consistent with the theoretical trade-off between bank size and coverage.

5.7 Empirical Validation of the Bound

Table 6 reports the key quantities from Theorem 1 measured on AQUA with both student models. All NLL values are per-token (length-normalized), consistent with the bounded-loss Assumption 1 and the training objective. Test NLL measures agreement with teacher-generated targets. Task accuracy measures agreement with gold answers. We approximate the Gibbs risk $\widehat{L}_S(\mathsf{Q}_S)$ by the point estimate $\widehat{L}_S(\hat{\theta}_S)$, standard for tight posteriors [Dziugaite and Roy, 2017].

Banking reduces frozen-model NLL. $\widehat{L}_S(0)$ is lowest for D-RPC on both models (Qwen: 2.68 vs. 2.70/3.29; Llama: 2.01 vs. 2.02/2.72), showing better fit to banked targets for the two tested students. This is consistent with the frozen-model alignment established in Proposition 4 and shows that banking can reduce N in Theorem 1.

Generalization gap and accuracy. D-RPC achieves the highest test accuracy on both models (Qwen: 74.0%; Llama: 63.4%). CoT consistently has the largest generalization gap (Qwen: 0.20; Llama: 0.29), while D-RPC and Freeform are comparable, consistent with Theorem 1 predicting a tighter bound under moderate bank size.

Adapter norms. The adapter norms $\|\hat{\theta}_S\|^2$ are comparable across strategies within each model, with Freeform slightly lower than D-RPC on both. For fixed prior and regularization parameters, lower frozen-model NLL reduces N in Eq. 3. The bank-size contribution appears separately in M (Eq. 2).

6 Limitations

Evaluation scope. All experiments use a single teacher, GPT-5.1, and two student architectures, Llama 3.1 8B Instruct and Qwen 3 1.7B, across five reasoning benchmarks. While D-RPC's gains are consistent across both student families and all task types tested, we have not verified whether the improvements transfer to other teacher models, larger or smaller student scales, or domains beyond math and commonsense reasoning such as code generation and scientific QA.

Computational overhead. D-RPC's pipeline incurs about $2\times$ the teacher-query cost of standard CoT, across three components: a categorization pass over all training questions, a seed-set pass on 5% of the training data to build the initial bank, and a rationale-generation pass conditioned on retrieved paths. DCoT generates three CoT rationales per question, with a teacher-query cost of roughly $3\times$ that of standard CoT. The categorization pass supports seed sampling and intent clustering, making it a possible target for reducing teacher-query overhead. This cost is paid offline and does not affect student inference, though it may limit applicability where teacher API costs dominate.

Statistical scope. The main comparisons use repeated executions with seed 42, so their uncertainty estimates are conditional on this seed. Appendix D.4 separately reports AQUA results across ten training seeds.

Path selection and supervision quality. Our retrieval procedure ranks candidate reasoning paths by embedding similarity after category and intent routing. A natural extension is to combine this matching criterion with rationale quality and the student's learning needs. Token-level uncertainty could guide supervision weighting, while quality estimates could inform path selection and bank refinement [Kim and Baek, 2026, Yan et al., 2025]. Evaluating these extensions on MATH could clarify when greater structural reuse translates into higher student accuracy.

Theoretical generality. The PAC-Bayes generalization analysis in Section 4 is derived under LoRA-based SFT with Gaussian weight priors. Extending the bound to full fine-tuning, alternative parameter-efficient methods such as adapters or prefix tuning, or non-Gaussian prior families remains open and would broaden the theoretical applicability of the framework.

7 Conclusion

We proposed D-RPC, which compresses the teacher's reasoning space into a compact, reusable bank of canonical paths before distillation. By routing semantically similar questions to shared paths and dynamically absorbing novel solutions, the bank provides consistent yet diverse supervision without verbose templates. A PAC-Bayes generalization bound formalizes the trade-off between bank size and coverage through an explicit $\log K_{\mathrm{bank}}$ term. The bank-size ablation favors an intermediate size, while the frozen-model NLL diagnostic shows better fit to banked targets. Across five reasoning benchmarks, D-RPC achieves the highest average accuracy among the evaluated methods for both student models. More broadly, our results suggest that structuring the supervision distribution before training is an effective and lightweight strategy for improving SLM reasoning.

Acknowledgements and Disclosures

Funding. This material is based in part upon work supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program (RGPIN-2024-03954), NSERC Alliance - Alberta Innovates Advance Program ALLRP 602246-24, and Digital Research Alliance of Canada (alliancecan.ca).

Competing interests. The authors declare no competing interests.

References

Ammar Ahmed, Azal Ahmad Khan, Ayaan Ahmad, Sheng Di, Zirui Liu, and Ali Anwar. Retrieval-of-thought: Efficient reasoning via reusing thoughts, 2025. URL https://arxiv.org/abs/2509.21743.

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. Chain-of-thought reasoning in the wild is not always faithful, 2025. URL https://arxiv.org/abs/2503.08679.

Jing Bi, Yuting Wu, Weiwei Xing, and Zhenjie Wei. Enhancing the reasoning capabilities of small language models via solution guidance fine-tuning, 2024. URL https://arxiv.org/abs/2412.09906.

Paul C. Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which llm reasoning steps matter?, 2025. URL https://arxiv.org/abs/2506.19143.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Hongzhan Chen, Siyue Wu, Xiaojun Quan, Rui Wang, Ming Yan, and Ji Zhang. Mcc-kd: Multi-cot consistent knowledge distillation, 2023. URL https://arxiv.org/abs/2310.14747.

Xiao Chen, Sihang Zhou, Ke Liang, Xiaoyu Sun, and Xinwang Liu. Skip-thinking: Chunk-wise chain-of-thought distillation enable smaller language models to reason better and faster, 2025a. URL https://arxiv.org/abs/2505.18642.

Xinghao Chen, Zhijing Sun, Wenjin Guo, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, and Xiaoyu Shen. Unveiling the key factors for distilling chain-of-thought reasoning, 2025b. URL https://arxiv.org/abs/2502.18001.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. In *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning, 2022. URL https://arxiv.org/abs/2205.09712.

Jin Cui, Jiaqi Guo, Jiepeng Zhou, Ruixuan Yang, Jiayi Lu, Jiajun Xu, Jiangcheng Song, Boran Zhao, and Pengju Ren. Mind: From passive mimicry to active reasoning through capability-aware multi-perspective cot distillation, 2026. URL https://arxiv.org/abs/2601.03717.

Chengwei Dai, Kun Li, Wei Zhou, and Songlin Hu. Capture the key in reasoning to enhance CoT distillation generalization. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 441–465, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.21. URL https://aclanthology.org/2025.acl-long.21/.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. Active prompting with chain-of-thought for large language models, 2024. URL https://arxiv.org/abs/2302.12246.

Cong-Thanh Do, Rama Doddipatla, and Kate Knill. Effectiveness of chain-of-thought in distilling reasoning capability from large language models, 2025. URL https://arxiv.org/abs/2511.05184.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics (TACL)*, 2021.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS*, 2021.

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

Minsang Kim and Seung Jun Baek. Explain in your own words: Improving reasoning via token-selective dual knowledge distillation. In *International Conference on Learning Representations*, 2026. URL https://singkru.github.io/publication/pdfs/paper-iclr.pdf.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Zhenyu Lei, Zhen Tan, Song Wang, Yaochen Zhu, Zihan Chen, Yushun Dong, and Jundong Li. Learning from diverse reasoning paths with routing and collaboration, 2025. URL https://arxiv.org/abs/2508.16861.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step, 2024. URL https://arxiv.org/abs/2306.14050.

Ziang Li, Manasi Ganti, Zixian Ma, Helena Vasconcelos, Qijia He, and Ranjay Krishna. Rethinking human preference evaluation of llm rationales, 2025. URL https://arxiv.org/abs/2509.11026.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*, 2017.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback, 2023.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason, 2023. URL https://arxiv.org/abs/2212.08410.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. Reasoningbank: Scaling agent self-evolving with reasoning memory, 2026. URL https://arxiv.org/abs/2509.25140.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning, 2024. URL https://arxiv.org/abs/2402.13950.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models, 2023. URL https://arxiv.org/abs/2210.03350.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning, 2019. URL https://arxiv.org/abs/1906.02361.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL https://arxiv.org/abs/1908.10084.

Devendra Singh Sachan, Mike Lewis, Dani Yogatama, Luke Zettlemoyer, Joelle Pineau, and Manzil Zaheer. Questions are all you need to train a dense passage retriever, 2023. URL https://arxiv.org/abs/2206.10658.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling reasoning capabilities into smaller language models, 2023. URL https://arxiv.org/abs/2212.00193.

Alberlucia Rafael Soarez, Daniel Kim, Mariana Costa, and Alejandro Torre. Demystifying low-rank knowledge distillation in large language models: Convergence, generalization, and information-theoretic guarantees, 2026. URL https://arxiv.org/abs/2603.22355.

Manveer Singh Tamber, Forrest Sheng Bao, Chenyu Xu, Ge Luo, Suleman Kazi, Minseok Bae, Miaoran Li, Ofer Mendelevitch, Renyi Qu, and Jimmy Lin. Benchmarking LLM faithfulness in RAG with evolving leaderboards. In Saloni Potdar, Lina Rojas-Barahona, and Sebastien Montella, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 799–811, Suzhou (China), November 2025. Association for Computational Linguistics. ISBN 979-8-89176-333-3. doi: 10.18653/v1/2025.emnlp-industry.54. URL https://aclanthology.org/2025.emnlp-industry.54/.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. Scott: Self-consistent chain-of-thought distillation, 2023a. URL https://arxiv.org/abs/2305.01879.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023b. URL https://arxiv.org/abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Xiangyu Wen, Junhua Huang, Zeju Li, Min Li, Jianyuan Zhong, Zhijian Xu, Mingxuan Yuan, Yongxiang Huang, and Qiang Xu. Reasoning scaffolding: Distilling the flow of thought from llms, 2025. URL https://arxiv.org/abs/2509.23619.

Sarah Wiegreffe, Jack Hessel, Swabha Swayamdipta, Mark Riedl, and Yejin Choi. Reframing human-ai collaboration for generating free-text explanations, 2022. URL https://arxiv.org/abs/2112.08674.

Shicheng Xu, Liang Pang, Yunchang Zhu, Jia Gu, Zihao Wei, Jingcheng Deng, Feiyang Pan, Huawei Shen, and Xueqi Cheng. Distilling the implicit multi-branch structure in llms' reasoning via reinforcement learning, 2025. URL https://arxiv.org/abs/2505.16142.

Jianzhi Yan, Le Liu, Youcheng Pan, Shiwei Chen, Yang Xiang, and Buzhou Tang. Towards efficient cot distillation: Self-guided rationale selector for better performance with fewer rationales, 2025. URL https://arxiv.org/abs/2509.23574.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E. Gonzalez, and Bin Cui. Buffer of thoughts: Thought-augmented reasoning with large language models, 2024. URL https://arxiv.org/abs/2406.04271.

Ling Yang, Zhaochen Yu, Bin Cui, and Mengdi Wang. Reasonflux: Hierarchical llm reasoning via scaling thought templates. *arXiv preprint arXiv:2502.06772*, 2025a.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Minkai Xu, Joseph E. Gonzalez, Bin Cui, and Shuicheng Yan. Supercorrect: Advancing small llm reasoning with thought template distillation and self-correction, 2025b. URL https://arxiv.org/abs/2410.09008.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023a. URL https://arxiv.org/abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023b. URL https://arxiv.org/abs/2210.03629.

Junchi Yu, Ran He, and Rex Ying. Thought propagation: An analogical approach to complex reasoning with large language models, 2024. URL https://arxiv.org/abs/2310.03965.

Xiang Yu, Yujia Huo, Liqiong Cai, and Xuewei Luo. Chain-of-thought curriculum distillation: Teaching smaller models to reason step-by-step. In *Proceedings of the 2025 2nd International Symposium on Artificial Intelligence for Education*, ISAIE '25, page 812–819, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715792. doi: 10.1145/3775073.3775200. URL https://doi.org/10.1145/3775073.3775200.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022. URL https://arxiv.org/abs/2203.14465.

Xianwei Zhuang, Zhihong Zhu, Zhichang Wang, Xuxin Cheng, and Yuexian Zou. Unicott: A unified framework for structural chain-of-thought distillation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. Reasonflux-prm: Trajectory-aware prms for long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2506.18896*, 2025a.

Tao Zou, Xinghua Zhang, Haiyang Yu, Minzheng Wang, Fei Huang, and Yongbin Li. Eifbench: Extremely complex instruction following benchmark for large language models. *arXiv preprint arXiv:2506.08375*, 2025b.

A Additional Related Work

Reasoning prompting and structured inference. Prompting methods such as few-shot prompting [Brown et al., 2020] and Chain-of-Thought (CoT) [Wei et al., 2023] improve LLM reasoning by eliciting intermediate steps. Subsequent work explored richer structured rationales during inference, including self-consistency [Wang et al., 2023b], self-questioning [Press et al., 2023], ReAct [Yao et al., 2023b], and Tree-of-Thoughts [Yao et al., 2023a]. A recurring limitation is that these methods often induce miscellaneous, divergent, and sometimes unfaithful rationales [Tamber et al., 2025, Paul et al., 2024, Wang et al., 2023a]. These issues are particularly pronounced for SLMs. The limited capacity and context windows make them more sensitive to verbosely structured prompting schemes, where the model must explicitly construct and traverse the data structure during generation.

Rationale used for supervised fine-tuning. Model-generated rationales are widely used as an intermediate supervision signal in supervised fine-tuning (SFT), as they can improve both task performance and interpretability [Rajani et al., 2019, Wiegreffe et al., 2022]. To enhance rationale quality, MoRSD studies how to select rich, high-quality, informative rationales [Yan et al., 2025]. Thought Anchors explores identifying key rationales that dominate the decision of final results [Bogdan et al., 2025]. Earlier work on Symbolic CoT Distillation showed that even sub-1B-parameter students can acquire step-by-step reasoning when trained on diverse rationales sampled from a much larger teacher, and that diversity in the sampled chains, not their individual likelihood, is the dominant factor in distillation success [Li et al., 2024]. However, teacher rationales can vary substantially in clarity, step granularity, and correctness across questions and difficulty levels [Diao et al., 2024, Creswell et al., 2022], leading to a decrease in distillation performance. Existing studies suggest that such rationale divergence has been linked to reduced faithfulness and training instability [Sachan et al., 2023, Arcuschin et al., 2025, Li et al., 2025].

B Additional Prompting Details

B.1 Prompt Templates

We apply four prompting strategies to the teacher model GPT-5.1 to generate rationales.

Chain-of-Thought Prompt.

```
Think step by step in a few precise steps (no more than six sentences) to solve the
↪  problem.

Then output ONLY a compact JSON object of the form:
{
  "rationale": "<explanation>",
  "ans": <numeric_answer>
}

Rules:
- "ans" must be a number, not a string.
- No additional text before or after the JSON.
```

Stage 1 Initial Bank Construction Prompt(Ours).

```
You are a rigorous but concise math tutor.
You solve math problems carefully and explain your reasoning briefly and clearly.
Avoid unnecessary prose; show only the key steps needed to reach the answer.

You are solving a math or word problem.
- Decompose the problem into abstract reasoning steps.
- For each step, use a general action name (no question-specific words).
- Each step must show actual computation.

Question: <QUESTION_TEXT>
Category: <ROUTE_CATEGORY>
Intent: <INTENT_TEXT>

Output ONLY valid JSON in the following format, and AVOID question-specific wording
↪  in reasoning_path keys:
{
```

```
  "route": {
    "difficulty": <1|2|3>,
    "budget": <Follow Budget Contract>,
    "reasoning_path": ["<DescriptiveStepName1>", "<DescriptiveStepName2>", ...]
  },
  "rationale": {
    "<DescriptiveStepName1>": "Step1: <...> Step2: <...> Step3: <...>",
    "<DescriptiveStepName2>": "Step1: <...> Step2: <...>",
    ...
  },
  "ans": <numeric>
}

Hard rules:
- Output must match the required JSON schema exactly.
- "route" must be a dictionary containing exactly: difficulty, budget, and
↪  reasoning_path.
- "reasoning_path" must be a list of strings representing ordered steps.
- The keys in "rationale" must match the strings in "reasoning_path" exactly.
- Output JSON only; no extra text.

Budget contract:
- Difficulty 1 implies Budget 2.
- Difficulty 2 or 3 implies Budget 3.
- len(reasoning_path) must be <= budget.

Reasoning key naming policy:
1) Keys must be TitleCase letters only (A--Z, a--z). No spaces, underscores, or
↪  digits.
2) Keys must be high-level descriptive summaries of the action taken; avoid
↪  question-specific wording.
3) Do NOT use generic sequential names (e.g., StepOne, CalculationOne).

Rationale requirements:
- For each reasoning_path entry, write a detailed multi-step explanation in one
↪  string.
- Use Step1:, Step2:, Step3:, ... labels; use as many substeps as needed (substeps
↪  do NOT count toward budget).
- Each substep should be concise and computational.
```

Stage 2 SFT Reasoning Extraction Prompt(Ours).

```
You are a structured reasoning tutor.
You will be given:
- A math question
- One or more routing plans including:
  - Category
  - Intent
  - Budget
- ReasoningPathOptions (candidate paths)

Pick the best routing plan from the options and follow it if adequate; otherwise,
refine only the reasoning_path conservatively.

Question:
<QUESTION_TEXT>

Route:
{
  "category": "<CATEGORY>",
  "intent": ["<INTENT>"],
  "difficulty": <1|2|3>,
  "budget": <BUDGET>,
  "reasoning_path_options": [
    ["<PathOption1>"],
    ["<PathOption2>", "<PathOption3>"]
```

```
  ]
}

Task Instructions:

1) Assess reasoning_path suitability:
- If reasoning_path is missing and reasoning_path_options is provided, select the
↪  best option.
- If the selected reasoning_path can solve the problem, keep it unchanged.
- If not, create a revised reasoning_path conservatively:
  * Use Title-Case letters only (A--Z, a--z)
  * No digits, underscores, or question-specific wording
  * Length must be <= budget
- Keep category, intent, difficulty, and budget unchanged.

2) Generate detailed rationale:
- For each entry in the chosen reasoning_path, write a detailed explanation.
- Label sub-steps as Step1:, Step2:, Step3:, ...
- Use as many sub-steps as needed (sub-steps do NOT count toward budget).
- Each sub-step must include explicit computation or derivation when applicable.

3) Final answer:
- Compute the numeric answer and place it in "ans".

Output ONLY valid JSON (no markdown, no extra text):
{
  "route": {
    "category": "<CATEGORY>",
    "intent": ["<INTENT>"],
    "difficulty": <1|2|3>,
    "budget": <BUDGET>,
    "reasoning_path": ["<FinalPath1>", "<FinalPath2_optional>"]
  },
  "rationale": {
    "<DescriptiveStepName1>": "Step1: <...> Step2: <...> Step3: <...>",
    "<DescriptiveStepName2>": "Step1: <...> Step2: <...>",
    ...
  },
  "ans": <numeric>
}

Requirements:
- Each reasoning_path entry must appear exactly once as a key in rationale.
- Keys in rationale must match reasoning_path entries exactly.
- Do not exceed the budget in number of reasoning_path steps.
- Rationale values must be single strings with Step1:, Step2:, ... labels.
```

Freeform Prompt.

```
You are a rigorous but concise math tutor.
You solve math problems carefully and explain your reasoning briefly and clearly.
Avoid unnecessary prose; show only the key steps needed to reach the answer.

You are solving a math or word problem.
- Decompose the problem into abstract reasoning steps.
- For each step, use a general action name (no question-specific words).
- Each step must show actual computation.

Question:
<QUESTION_TEXT>

Your task:
Solve the question and return ONLY a valid JSON object.

Output ONLY valid JSON with freely chosen reasoning paths:
{
```

```
  "rationale": {
    "<DescriptiveStepName1>": "<concise derivation>",
    "<DescriptiveStepName2>": "<concise derivation>",
    ...
  },
  "ans": <numeric>
}

Rules:
- Freely choose up to three reasoning paths with high-level, TitleCase names
↪  (letters only; no spaces, digits, or underscores).
- Keys must be human-readable action labels (e.g., PlanComputation, CombineTotals);
↪  avoid generic names like ReasoningPath1 and avoid question-specific wording.
- Keep each rationale concise and focused on the computation or logic.
- If a second path is unnecessary, omit it.
- "ans" must be numeric (no strings, units, or words).
- Do not include any markdown or extra text; return JSON only.
```

SuperCorrect XML Prompt.

```
You are a rigorous but concise math tutor.
You solve math problems carefully and explain your reasoning briefly and clearly.
Avoid unnecessary prose; show only the key steps needed to reach the answer.

Transform the solution of the following math problem into a step-by-step XML format.
Each step should be enclosed within tags like <Step1> </Step1>.
For each step, determine if the step is challenging or tricky; if so, add a
↪  detailed explanation
enclosed within <Key> </Key> as annotations to help the student understand the step
↪  correctly.
After all steps, summarize the common solution pattern to help generalize to
↪  similar problems
within <Generalized> </Generalized>. Finally, present the final answer enclosed
↪  within
<Answer> </Answer>.

Problem:
<QUESTION_TEXT>
```

Table 7: Three-slot excerpt from the reasoning-path bank $\mathcal{B}$. Each slot maps a (category, canonical intent) pair to one or more canonical paths. This is an illustrative fragment; the full bank contains 50–150 paths.

Category	Canonical Intent	Paths in $\mathcal{B}(c, \tilde{t})$
Arithmetic	unit-rate	Π_1: `[ComputeUnitRate, MultiplyByCount]`
		Π_2: `[IdentifyRatio, ScaleToTarget]`
Arithmetic	total-cost	Π_3: `[ComputeItemCost, SumComponents]`
Commonsense	temporal feasibility	Π_4: `[RecallKeyFacts, CheckTimeline, DeduceAnswer]`

C Qualitative Examples of the D-RPC Pipeline

This section provides simplified, illustrative examples that trace each stage of D-RPC on concrete problem instances. The JSON structures and path names follow the actual format produced by our pipeline (see prompt templates in Appendix B.1); specific values are abbreviated for clarity. We use GSM8K arithmetic questions as the primary running example and include a StrategyQA instance to illustrate cross-domain applicability.

Concrete example. For GSM8K, the bank might contain the entry:

Category: Arithmetic **Intent:** unit-rate computation
Reasoning Paths: {[compute rate, combine quantities], [find unit rate, multiply by count]}

Two canonical strategies for the same intent, enough for coverage yet few enough for consistency.

C.1 Reasoning-Path Bank Structure

Table 7 shows a three-slot excerpt from the reasoning-path bank $\mathcal{B}$ (defined in Section 3.2) to illustrate its structure. Each row corresponds to a (category, canonical intent) pair and lists the reasoning paths $\Pi \in \mathcal{B}(c, \tilde{t})$ associated with that slot. Paths are ordered lists of abstract, reusable step names following the TitleCase naming policy enforced by the Stage 1 prompt in Appendix B.1. The full bank in our GSM8K experiments contains 50–150 paths depending on configuration (see the ablation in Table 4).

C.2 Question Categorization

Before any reasoning takes place, the teacher's categorization function f_{cat} from Section 3.2, Step 1, assigns a category–intent pair (C_i, T_i) to every training question. The examples below show this output for three questions spanning two domains.

```
Q1: "A baker sells cupcakes for $3 each. If
  she sold 48 on Monday and 32 on Tuesday,
  how much did she earn in total?"
  -> Category: Arithmetic
     Intent:   total-cost aggregation

Q2: "Tom drives 60 miles in 1.5 hours. How
  far can he drive in 4 hours at the same
  speed?"
  -> Category: Arithmetic
     Intent:   unit-rate computation

Q3: "Did Albert Einstein ever visit the
  Moon?"
  -> Category: Commonsense
     Intent:   temporal feasibility check
```

These labels serve only as keys for bank lookup and path retrieval; they are not included in the student's training data. Q1 and Q2 share a category but differ in intent, so they are routed to different

bank slots. Q3 illustrates that the same categorization mechanism applies to non-mathematical domains.

C.3 Retrieved Paths vs. Freeform-Generated Paths

A central claim of this work is that unconstrained teacher generation introduces unnecessary supervision variance: structurally similar questions receive structurally different rationales. To illustrate this contrast, we show the reasoning paths produced by D-RPC and by the Freeform baseline for two unit-rate questions that share the same underlying solution strategy.

Q2: "Tom drives 60 miles in 1.5 hours. How far can he drive in 4 hours?"

D-RPC: `[ComputeUnitRate, MultiplyByCount]`
Freeform: `[FindSpeed, MultiplyDistance]`

Q4: "A factory makes 150 widgets in 5 hours. How many in 8 hours?"

D-RPC: `[ComputeUnitRate, MultiplyByCount]`
Freeform: `[SetUpProportion, CrossMultiply]`

Under D-RPC, both questions match the bank slot `Arithmetic/unit-rate` and retrieve the same canonical path Π_1, so the student observes a single consistent template. Under Freeform, the teacher improvises a fresh structure for each question, choosing `FindSpeed` for Q2 and `SetUpProportion` for Q4, producing valid but structurally incompatible rationales. When the student trains on both examples via $\mathcal{L}_{\mathrm{SFT}} = -\log p_\theta(Y \mid X)$, the conflicting output structures act as label noise: the model must allocate capacity to reconcile multiple solution formats for the same problem type, rather than reinforcing a single reusable pattern. This is precisely the supervision-variance reduction that the entropy-floor analysis in Proposition 1 formalizes.

C.4 Bank-Guided Rationale Generation

Once a path is retrieved, the teacher receives the question and the candidate paths via the Stage 2 prompt f_{teach} described in Section 3.3, with the full prompt in Appendix B.1, and generates the supervision tuple $Y = (\Pi, R, A)$. Below is the complete input–output pair for Q2.

```
Input to teacher (Stage 2):
  Question: "Tom drives 60 miles in 1.5 hours.
    How far can he drive in 4 hours at the
    same speed?"
  Route:
  {
    "category": "Arithmetic",
    "intent": ["unit-rate computation"],
    "difficulty": 1,
    "budget": 2,
    "reasoning_path_options": [
      ["ComputeUnitRate", "MultiplyByCount"],
      ["IdentifyRatio", "ScaleToTarget"]
    ]
  }

Teacher output (Y):
  {
    "route": {
      "category": "Arithmetic",
      "intent": ["unit-rate computation"],
      "difficulty": 1,
      "budget": 2,
      "reasoning_path":
        ["ComputeUnitRate", "MultiplyByCount"]
    },
    "rationale": {
      "ComputeUnitRate":
        "Step1: Distance = 60 miles,
```

```
        Time = 1.5 hours.
        Step2: Speed = 60 / 1.5 = 40 mph.",
    "MultiplyByCount":
      "Step1: Target time = 4 hours.
       Step2: Distance = 40 * 4 = 160 miles."
  },
  "ans": 160
}
```

The teacher selects path option Π_1 = `[ComputeUnitRate, MultiplyByCount]` from the two candidates provided, then instantiates each abstract step with concrete sub-step computations to form the rationale R. The student's training example is (X, Y) where $Y = (\Pi, R, A)$: the reasoning path, the detailed rationale, and the final answer.

Contrast: Freeform generation on the same question. Under the Freeform prompt, f_{teach} receives no candidate paths and must generate both the solution structure and content from scratch:

```
Freeform teacher output (Q2):
{
  "rationale": {
    "FindSpeed":
      "Step1: Speed = 60 / 1.5 = 40 mph.",
    "MultiplyDistance":
      "Step1: Distance = 40 * 4 = 160 miles."
  },
  "ans": 160
}
```

The final answer is identical, but the output lacks the `route` field entirely and uses the ad-hoc path name `FindSpeed` rather than the canonical `ComputeUnitRate`. When a different unit-rate question is posed, the Freeform teacher may invent yet another structure, for instance `SetUpProportion` → `CrossMultiply` for Q4 above, whereas the bank-guided teacher consistently selects from the same small set of canonical paths. Over the full training set, this structural consistency reduces the variance of the supervision signal that the student must fit, which is the mechanism underlying D-RPC's empirical gains.

D Supplementary Setting

D.1 Detailed Datasets

Table 8: Datasets used for reasoning transfer training and evaluation. Task type, answer format, and the number of training and evaluation samples for each dataset are reported.

Dataset	Task	Answer	#Train	#Eval
GSM8K	Math	Numeric	7,473	1,319
AQUA	Math	Letter	10,000	254
StrategyQA	Commonsense	T/F	2,061	229
AI2ARC	Science	Letter	6,230	1,557
MATH	Comp. math	Numeric	10,000	5,000

GSM8K [Cobbe et al., 2021] is a grade-school mathematics dataset consisting of arithmetic word problems that require multi-step numerical reasoning.

AQUA [Ling et al., 2017] is a multiple-choice mathematics reasoning dataset that requires selecting the correct option based on symbolic and logical reasoning.

StrategyQA [Geva et al., 2021] is a commonsense reasoning dataset in which each question requires implicit multi-hop reasoning over factual knowledge, with Boolean (True/False) answers.

D.2 Training Details

The main supervised fine-tuning (SFT) comparisons are conducted using an identical training configuration to ensure fair and controlled comparisons across different reasoning supervision strategies. We fine-tune `Llama-3.1-8B-Instruct` as the student model using the AdamW optimizer with a

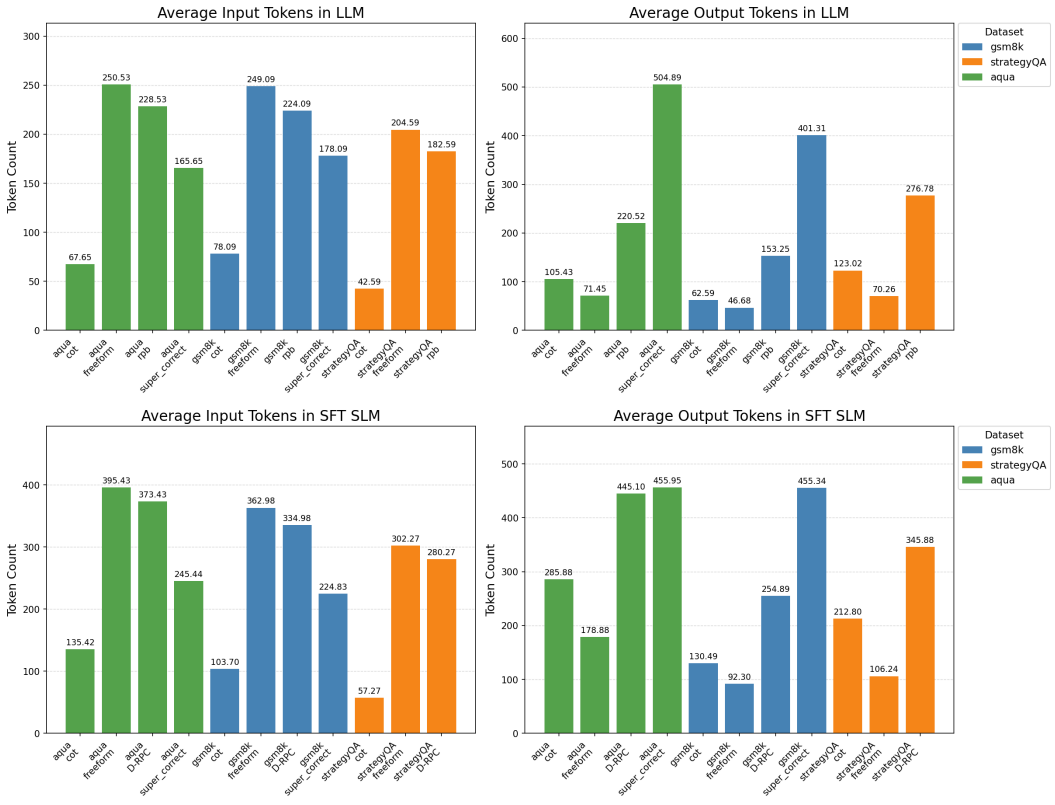

Figure 2: Token usage analysis under different reasoning supervision strategies. **Top:** Average input and output token usage of the teacher LLM during rationale generation. **Bottom:** Average input and output token usage of the SFT student model at evaluation time.

learning rate of 1×10^{-4} and a weight decay of 0.01. The maximum sequence length is set to 2048 tokens.

Training is performed for two epochs with a per-device batch size of 2 and gradient accumulation over 8 steps, resulting in an effective batch size of 16. We employ a linear learning rate scheduler with a warmup ratio of 3%, and apply gradient clipping with a maximum norm of 1.0 to stabilize training.

To improve parameter efficiency, we adopt LoRA-based fine-tuning. LoRA adapters are applied to the query, key, value, and output projection layers with default rank $r = 64$, scaling factor $\alpha = 128$, and a dropout rate of 0.05. All experiments are conducted using bfloat16 precision. The Llama 3.1 8B checkpoints in Table 6 use $r = 128$.

Model checkpoints are saved every 250 training steps, and training logs are recorded every 25 steps. Each repeated execution uses seed 42, except for the training seeds specified in Appendix D.4. The evaluation protocol in Section 5.2 describes the source of variation. Unless otherwise specified, no limit is imposed on the number of training samples. We use the same hyperparameter configuration for Qwen 3 1.7B.

D.3 Supplementary Experiments

Token cost: D-RPC improves efficiency relative to template-heavy supervision. Figure 2 compares average input/output token usage for both the teacher LLM during rationale generation and the SFT student at evaluation time. While D-RPC includes retrieved reasoning paths and thus uses substantially more teacher input tokens than CoT and a comparable amount to Freeform, it substantially reduces teacher output length relative to SuperCorrect, yielding an overall reduction of roughly 33–35% in total teacher tokens per example across the math datasets. On the student side, D-RPC-trained models produce more explicit rationales than CoT and Freeform on every dataset, while staying shorter on

Table 9: AQUA performance across training fractions under different LoRA ranks (r=128 vs. r=64). Bold denotes the higher accuracy between the two ranks for each method and training fraction, FV denotes Format Validity.

Train (%) = 10					Train (%) = 60				
Strategy	r128 Acc	r128 FV	r64 Acc	r64 FV	Strategy	r128 Acc	r128 FV	r64 Acc	r64 FV
cot	**64.96**	67.35	62.20	64.75	cot	60.24	61.45	**62.20**	63.45
freeform	61.02	61.75	**62.20**	67.23	freeform	**59.45**	60.89	57.48	59.35
D-RPC(ours)	61.42	63.93	**63.78**	66.80	D-RPC(ours)	**64.17**	64.94	63.78	65.85
super_correct	59.06	61.70	**59.84**	62.98	super_correct	**58.66**	62.45	57.87	61.44
Train (%) = 20					Train (%) = 70				
cot	60.24	60.96	**64.96**	67.90	cot	**66.93**	69.11	61.42	61.51
freeform	57.87	59.27	**58.27**	60.41	freeform	59.06	59.52	**63.78**	65.06
D-RPC(ours)	**63.78**	68.22	62.20	63.31	D-RPC(ours)	65.75	66.80	**68.50**	70.45
super_correct	**57.87**	61.11	56.30	58.90	super_correct	**58.66**	63.36	57.87	61.25
Train (%) = 30					Train (%) = 80				
cot	**64.96**	66.80	62.20	63.45	cot	**65.35**	65.61	61.81	62.06
freeform	**59.06**	60.73	57.09	59.67	freeform	**62.99**	63.75	59.45	61.13
D-RPC(ours)	**66.93**	68.00	62.60	65.16	D-RPC(ours)	**66.14**	68.57	62.99	64.00
super_correct	**61.02**	64.44	54.72	57.69	super_correct	**64.96**	66.39	58.27	61.18
Train (%) = 40					Train (%) = 90				
cot	**66.54**	68.15	63.78	64.29	cot	**65.35**	65.61	64.17	64.68
freeform	**60.63**	62.35	58.27	58.96	freeform	**58.27**	60.16	57.87	58.57
D-RPC(ours)	66.14	67.74	**67.72**	69.80	D-RPC(ours)	67.72	68.80	**68.11**	70.33
super_correct	**61.02**	63.45	59.45	61.70	super_correct	**62.99**	65.38	60.24	63.09
Train (%) = 50					Train (%) = 100				
cot	62.60	63.10	**66.14**	66.67	cot	**67.72**	67.98	65.75	66.01
freeform	57.48	58.87	**57.87**	58.80	freeform	**64.57**	65.08	56.30	57.20
D-RPC(ours)	**64.17**	65.46	61.42	64.46	D-RPC(ours)	**69.69**	70.68	66.14	67.20
super_correct	**57.87**	61.44	54.33	56.54	super_correct	**60.63**	62.86	56.30	59.17

average than SuperCorrect supervision (with the gap largest on GSM8K), avoiding the output-length explosion that increases inference latency.

LoRA Setting: How LoRA Rank and Alpha impact the results. Table 9 compares the rankings of LoRA at the same training ratios of r=64 and r=128. The impact of ranking depends on the training data volume. At low data ratios (10-20%), r=64 outperforms r=128, consistent with the phenomenon that the low-ranked adapter acts as an implicit regularizer to resist fitting noise. As the training ratio increases, the trend shifts to capacity-constrained adaptation, where r=128 achieves higher accuracy at 80-100% data ratios. This suggests that r=64 underfits once sufficient supervision is provided. At all data ratios, D-RPC remains the strongest policy, achieving the best accuracy across most fractions and peaking at r=128 (69.69% accuracy) while maintaining good format validity. This indicates that its structured guidance is robust to ranking selection and can benefit from additional adapter capacity as the amount of data increases.

Implications for reasoning space compression. The bank-size ablation favors an intermediate size, while the effect of initialization diversity varies with the training set size. These results support managing the balance between path reuse and coverage when constructing the bank.

D.4 AQUA results across training seeds

Table 10 reports accuracy on all 254 AQUA test questions using training seeds 43 to 52 and the LoRA configuration in Section 5.2. Empty omits candidate paths, while Random and Random-Global use independent draws sampled uniformly from the reasoning path bank. Filtered-SFT retains only D-RPC supervision tuples with correct final answers. The relative ordering of Empty and Random differs between the two students.

Table 10: AQUA test accuracy (%) across ten training seeds. Entries report means and sample standard deviations.

Condition	Llama 3.1 8B Instruct	Qwen 3 1.7B
Empty	$66.93_{\pm 1.41}$	$72.83_{\pm 2.83}$
Random	$66.34_{\pm 2.40}$	$74.21_{\pm 1.95}$
Random-Global	$66.30_{\pm 2.15}$	$73.43_{\pm 1.37}$
Filtered-SFT	$68.35_{\pm 2.18}$	$75.16_{\pm 2.10}$

D.5 Teacher answer errors and filtering

Table 11 reports final-answer errors in the D-RPC supervision collected for each dataset, and Table 12 compares teacher accuracy across path controls. Filtered-SFT removes the 1173 incorrect-answer tuples from AQUA and 2518 from MATH before student training, with student results in Table 2.

Table 11: Teacher final-answer errors in collected D-RPC supervision.

Dataset	Tuples	Errors	Error (%)
GSM8K	7473	370	5.0
AQUA	10000	1173	11.7
StrategyQA	2061	397	19.3
AI2ARC	6230	97	1.6
MATH	10000	2518	25.2

Table 12: Teacher final-answer accuracy (%) under the path controls.

Condition	AQUA	MATH
D-RPC	88.3	74.8
Empty	85.7	72.3
Random	88.2	73.7
Random-Global	88.1	73.6

D.6 Additional Random-Global executions

Table 13 reports an additional ten executions of Random-Global on the Fir cluster using seed 42 and all 5000 MATH test questions. These executions use the training configuration and greedy decoding protocol of Table 2, with a 2048-token output limit and generation batch size one.

Table 13: Additional Random-Global accuracy on MATH (%), reported as mean and sample standard deviation.

Llama 3.1 8B Instruct	Qwen 3 1.7B
$48.06_{\pm 0.38}$	$60.22_{\pm 0.39}$

E Proof of Proposition 1

Proof. By the chain rule for conditional entropy,

$$\mathrm{H}(Y \mid X) = \mathrm{H}(\Pi, R, A \mid X) = \mathrm{H}(\Pi \mid X) + \mathrm{H}(R \mid X, \Pi) + \mathrm{H}(A \mid X, \Pi, R).$$

Under Assumption 2, for each x, the conditional distribution $\Pi \mid X = x$ is supported on at most K_{bank} values, hence $\mathrm{H}(\Pi \mid X = x) \le \log K_{\mathrm{bank}}$. Taking expectation over X yields $\mathbb{E}[\mathrm{H}(\Pi \mid X)] \le \log K_{\mathrm{bank}}$. Substituting into the chain rule gives Eq. (4). □

F Second-order PAC-Bayes: proof sketches for the bounded-loss route

For completeness, we state the intermediate results referenced in Section 4 and used in the derivation of Theorem 1.

Proposition 5 (Bernstein-type PAC-Bayes for bounded losses)**.** *Under Assumption 1, for any prior* P *independent of* S *and any* $\delta \in (0,1)$*, with probability at least* $1-\delta$ *over* $S \sim \mathcal{D}^n$*, simultaneously for all posteriors* Q*,*

$$L(\mathsf{Q}) \le \widehat{L}_S(\mathsf{Q}) + \sqrt{\frac{2\,\widehat{V}_S(\mathsf{Q})\big(\mathrm{KL}(\mathsf{Q}\|\mathsf{P}) + \ln\frac{1}{\delta}\big)}{n}} + \frac{c\,\tau\big(\mathrm{KL}(\mathsf{Q}\|\mathsf{P}) + \ln\frac{1}{\delta}\big)}{n}. \tag{8}$$

where $c > 0$ *is a universal numerical constant and* $\widehat{V}_S(\mathsf{Q}) := \mathbb{E}_{\theta\sim\mathsf{Q}}[\frac{1}{n}\sum_i(\ell(\theta; Z_i) - \widehat{L}_S(\theta))^2]$.

Lemma 1 (Mean–variance domination for bounded losses)**.** *Under Assumption 1,* $\widehat{V}_S(\mathsf{Q}) \le \tau\,\widehat{L}_S(\mathsf{Q})$.

Corollary 1 (Mean-dependent second-order PAC-Bayes)**.** *Substituting Lemma 1 into Proposition 5: with probability at least* $1-\delta$*,*

$$L(\mathsf{Q}) \le \widehat{L}_S(\mathsf{Q}) + \sqrt{\frac{2\,\tau\,\widehat{L}_S(\mathsf{Q})\big(\mathrm{KL}(\mathsf{Q}\|\mathsf{P}) + \ln\frac{1}{\delta}\big)}{n}} + \frac{c\,\tau\big(\mathrm{KL}(\mathsf{Q}\|\mathsf{P}) + \ln\frac{1}{\delta}\big)}{n}. \tag{9}$$

Corollary 2 (Explicit K_{bank}-dependence)**.** *Under Assumptions 1–3, the same bound holds with* $\widehat{L}_S(\mathsf{Q})$ *replaced by* M *(Eq. 2) in the square-root term.*

F.1 Proof sketch of Proposition 5

Proposition 5 is a standard Bernstein-type PAC-Bayes inequality for bounded losses. A typical proof follows these steps:

1. **Bernstein mgf bound for bounded variables.** For a bounded random variable $U \in [0,\tau]$, one upper bounds $\log \mathbb{E}[\exp(\lambda(\mathbb{E}[U] - U))]$ in terms of λ, τ, and $\mathrm{Var}(U)$, yielding a Bernstein-style concentration inequality.
2. **Change of measure (Donsker–Varadhan).** For any measurable f,
$$\mathbb{E}_{\theta\sim\mathsf{Q}}[f(\theta)] \le \mathrm{KL}(\mathsf{Q}\|\mathsf{P}) + \log \mathbb{E}_{\theta\sim\mathsf{P}}[\exp(f(\theta))].$$
3. **High-probability conversion.** Apply Markov's inequality and a standard argument to obtain a statement that holds with probability at least $1-\delta$, introducing $\ln(1/\delta)$, and obtain a bound simultaneously for all Q.
4. **Variance-sensitive form.** Collect terms to obtain a deviation term scaling with $\sqrt{\widehat{V}_S(\mathsf{Q})}$ plus a lower-order $O((\mathrm{KL} + \ln(1/\delta))/n)$ term, where boundedness introduces the factor τ in the latter.

F.2 Proof of Lemma 1

Proof. Fix θ and define U as a random variable taking values $\ell(\theta; Z_i)$ under the empirical distribution over indices $i \sim \mathrm{Unif}\{1,\dots,n\}$. Under Assumption 1, $U \in [0,\tau]$, hence $U^2 \le \tau U$. Therefore,

$$\mathrm{Var}(U) = \mathbb{E}[U^2] - \mathbb{E}[U]^2 \le \tau\mathbb{E}[U] - \mathbb{E}[U]^2 \le \tau\mathbb{E}[U].$$

Noting that $\mathbb{E}[U] = \widehat{L}_S(\theta)$, we obtain

$$\frac{1}{n}\sum_{i=1}^{n}\big(\ell(\theta; Z_i) - \widehat{L}_S(\theta)\big)^2 = \mathrm{Var}(U) \le \tau\,\widehat{L}_S(\theta).$$

Finally, take expectation over $\theta \sim \mathsf{Q}$ to get $\widehat{V}_S(\mathsf{Q}) \le \tau\,\widehat{L}_S(\mathsf{Q})$. □

F.3 Derivation of Corollary 1

Corollary 1 follows immediately by substituting Lemma 1 (Eq. (1)) into Proposition 5 (Eq. (8)).

F.4 Derivation of Corollary 2

Starting from Corollary 1 (Eq. (9)), apply Assumption 3 (Eq. (5)) and Proposition 1 (Eq. (4)) to upper bound $\widehat{L}_S(\mathsf{Q})$ by $\log K_{\text{bank}} + \mathbb{E}[\mathrm{H}(R \mid X, \Pi)] + \mathbb{E}[\mathrm{H}(A \mid X, \Pi, R)] + \varepsilon$. Substituting this into Eq. (9) yields Eq. (2).

G Norm control and Gaussian KL: proofs

G.1 Proof of Proposition 2

Proof. By optimality of $\hat{\theta}_S$,

$$F_S(\hat{\theta}_S) \le F_S(0) \quad \Longrightarrow \quad \widehat{L}_S(\hat{\theta}_S) + \frac{\lambda}{2}\|\hat{\theta}_S\|_2^2 \le \widehat{L}_S(0).$$

Since $\widehat{L}_S(\hat{\theta}_S) \ge 0$, we conclude $\frac{\lambda}{2}\|\hat{\theta}_S\|_2^2 \le \widehat{L}_S(0)$, which implies Eq. (6). □

G.2 Proof of Proposition 3

Proof. For isotropic Gaussians $\mathcal{N}(\mu, \sigma_{\text{post}}^2 I_m)$ and $\mathcal{N}(0, \sigma_0^2 I_m)$, the KL divergence has the closed form

$$\mathrm{KL} = \frac{\|\mu\|_2^2}{2\sigma_0^2} + \frac{m}{2}\Big(\rho - 1 - \ln\rho\Big), \qquad \rho = \frac{\sigma_{\text{post}}^2}{\sigma_0^2}.$$

Setting $\mu = \hat{\theta}_S$ yields the equality in Eq. (7). The inequality follows by substituting Proposition 2 into the mean-shift term $\|\hat{\theta}_S\|_2^2/(2\sigma_0^2)$. □

H Formal justification of frozen-model alignment under banking

This appendix provides the formal proofs behind Proposition 4 (stated in Section 4). The goal is to justify the following condition by a cross-entropy decomposition and designable sufficient conditions.

Assumption 5 (Frozen-model alignment under banking (restated))**.** $\mathbb{E}_{S\sim\mathcal{D}_{\text{bank}}^n}[\widehat{L}_S(0)] \le \mathbb{E}_{S\sim\mathcal{D}_{\text{free}}^n}[\widehat{L}_S(0)]$.

H.1 Population form and cross-entropy decomposition

Define the frozen-model population cross-entropy in regime $\star \in \{\text{free}, \text{bank}\}$ by

$$\mathcal{L}_0^{(\star)} := \mathbb{E}_X \mathbb{E}_{Y\sim p^{(\star)}(\cdot|X)}\big[-\log p_0(Y \mid X)\big],\ Y = (\Pi, R, A).$$

Assuming integrability so that $\mathbb{E}_{S\sim(\mathcal{D}_\star)^n}[\widehat{L}_S(0)] = \mathcal{L}_0^{(\star)}$, Assumption 5 follows from

$$\mathcal{L}_0^{(\text{bank})} \le \mathcal{L}_0^{(\text{free})}. \tag{10}$$

For any target conditional $p^\star(\cdot \mid x)$,

$$\mathbb{E}_{Y\sim p^\star(\cdot|x)}[-\log p_0(Y|x)] = \mathrm{H}\big(p^\star(\cdot\,|x)\big) + \mathrm{KL}\big(p^\star(\cdot\mid x)\|p_0(\cdot\,|x)\big),$$

and averaging over X yields

$$\mathcal{L}_0 = \mathbb{E}[\mathrm{H}(Y \mid X)] + \mathbb{E}\big[\mathrm{KL}\big(p^\star(\cdot \mid X)\|p_0(\cdot \mid X)\big)\big]. \tag{11}$$

Thus, Eq. (10) is equivalent to

$$\Delta\mathrm{H} + \Delta\mathrm{KL} \le 0, \tag{12}$$

where $\Delta\mathrm{H}$ and $\Delta\mathrm{KL}$ denote the bank-minus-free differences of the two terms in Eq. (11). Proposition 1 implies the Π-banking mechanism reduces (or upper-bounds) the path-uncertainty component through $\mathrm{H}(\Pi \mid X) \le \log K_{\text{bank}}$, while Assumption 5 additionally requires that any increase in mismatch to p_0 does not offset the entropy reduction.

H.2 Sufficient designable conditions

Condition D1 (Bank as projection under the frozen model). Let $\mathfrak{P}_{\mathrm{bank}}(K_{\mathrm{bank}})$ denote the family of conditional distributions supported on a K_{bank}-template path bank:

$$\mathfrak{P}_{\mathrm{bank}}(K_{\mathrm{bank}}) := \Big\{ p(\cdot \mid x) : \mathrm{supp}(\Pi \mid X = x) \subseteq \mathcal{B}, |\mathcal{B}| = K_{\mathrm{bank}} \Big\},$$

with no restriction on R, A given (X, Π). Define $p^{\star}_{\mathrm{bank}}$ as

$$p^{\star}_{\mathrm{bank}} \in \operatorname*{arg\,min}_{p \in \mathfrak{P}_{\mathrm{bank}}(K_{\mathrm{bank}})} \mathbb{E}_X \mathbb{E}_{Y \sim p(\cdot|X)}[-\log p_0(Y \mid X)]. \tag{13}$$

Then $\mathcal{L}_0^{(\mathrm{bank})} \le \mathcal{L}_0^{(\mathrm{free})}$ holds by construction.

Condition D2 (Canonicalization improves frozen likelihood pointwise). If there exists a map c on targets such that for all (x, y),

$$-\log p_0(c(y) \mid x) \le -\log p_0(y \mid x), \tag{14}$$

and the banked regime replaces Y by $c(Y)$ (or restricts support to such canonical forms), then Eq. (10) follows immediately by taking expectations.

Condition D3 (Entropy drop dominates mismatch increase). From Eq. (12), it suffices that

$$\mathbb{E}[\mathrm{H}_{\mathrm{free}}(Y \mid X)] - \mathbb{E}[\mathrm{H}_{\mathrm{bank}}(Y \mid X)] \;\ge\; \mathbb{E}[\mathrm{KL}_{\mathrm{bank}} - \mathrm{KL}_{\mathrm{free}}],$$

i.e., any increase in mismatch to p_0 is smaller than the entropy reduction due to banking.

Under any of Conditions D1–D3, Assumption 5 is justified, establishing Proposition 4.

I Proof sketch of Theorem 1

Proof sketch. Fix $\delta \in (0, 1)$. On the event where Corollary 2 holds (probability at least $1 - \delta$), set $\mathsf{Q} = \mathsf{Q}_S$ and apply Proposition 3 to upper bound $\mathrm{KL}(\mathsf{Q}_S \| \mathsf{P})$ by

$$\frac{1}{\sigma_0^2 \lambda} \widehat{L}_S(0) + \frac{m}{2}\big(\rho - 1 - \ln \rho\big).$$

Substitute this into the deviation terms in Eq. (2). Then use the definitions of M (Eq. (2)) and N (Eq. (3)) to obtain Eq. (1). □

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 present D-RPC, the bank-size/coverage trade-off, and evaluation on five benchmarks with two students. Matched retrieval is presented as an additional refinement.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: Section 6 discusses the single-teacher evaluation, teacher-query overhead, fixed-seed uncertainty, path selection, and the scope of the LoRA-based analysis.
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: [Yes]
   Justification: Section 4 states the assumptions, with proofs and derivations in Appendices E, F, G, H, and I.
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: Algorithm 1, Section 5, and Appendices B.1 and D.2 specify the method, prompts, data, training, and evaluation settings.
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: The repository provides code, data, reproduction instructions, and a comparison of released predictions with Table 1.
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, etc.) necessary to understand the results?
   Answer: [Yes]
   Justification: Section 5 and Appendix D.2 give dataset splits, training configurations, checkpoint selection, and decoding, with additional AQUA seeds in Appendix D.4.
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: Main-table standard deviations describe ten executions with seed 42 and nondeterministic GPU execution. Table 10 separately reports means and sample standard deviations across training seeds 43 to 52.
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: [No]
   Justification: GPU execution times and the total compute budget are not fully documented. Section 6 and Appendix D.3 report teacher-query overhead and token costs.
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: The work uses public benchmarks and a commercial teacher API under its terms, without recruiting human participants. Funding and competing interests are disclosed after the conclusion.

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: D-RPC can make reasoning capabilities more accessible through smaller student models. These models can also inherit teacher biases and produce incorrect rationales, which may mislead users who rely on their explanations.

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., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The work studies distillation for established math and commonsense benchmarks and does not introduce a new general-purpose pretrained model or scraped dataset.

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: [No]

    Justification: The original dataset and model sources are credited in Sections 3 and 5. Individual asset licenses and API terms are not explicitly listed.

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 released assets include installation and usage instructions, data schemas, prediction manifests, and supplementary experiment documentation.

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: All supervision is teacher-generated on public benchmarks, with no crowdsourcing or human participants.

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: The work does not involve research with human participants.

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 work? If so, does the paper include any information regarding which LLMs were used and for what purpose?

    Answer: [Yes]

    Justification: GPT-5.1 performs categorization, reasoning-path elicitation, and rationale generation, while Llama 3.1 8B Instruct and Qwen 3 1.7B are the students. Their roles and prompts are described in Sections 3 and 5 and Appendix B.1.