DISTILL WHAT YOU TRUST: RELIABILITY-AWARE MULTI-TEACHER ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-teacher on-policy distillation allows a student to learn from complementary specialists on its own trajectories. Domain-routed approaches, however, select one teacher per example and keep it fixed throughout the response. This design both depends on domain labels that mixed training corpora often lack and cannot adapt teacher selection when the expertise required changes within a trajectory. We observe that each specialist deviates more from a shared reference on in-domain prompts than on out-of-domain prompts, on average. Based on this observation, we propose **TrustMOPD**, which replaces example-level teacher selection with label-free, token-level supervision allocation. At each student-generated prefix, TrustMOPD measures this displacement in next-token preferences, calibrates its magnitude across teachers, and uses the resulting scores as proxies for local reliability to weight teacher-specific distillation losses. Evaluated across mathematics, code, and instruction following, TrustMOPD closes 91.5% and 98.0% of the overall-score gap between the initial student and oracle-routed teachers when trained on SINGLECAP and MULTICAP, respectively, compared with 54.4% and 54.5% for the strongest label-free baseline in each setting. On SINGLECAP, it approaches label-based MOPD without using domain labels. Our code is available at `https://anonymous.4open.science/r/TrustMOPD-F624`.

1 INTRODUCTION

Integrating the complementary capabilities of multiple specialized models into a single student is a fundamental challenge in building broadly capable language models (Wan et al., 2024; Tian et al., 2025). On-policy distillation (OPD) queries a teacher along trajectories generated by the student, providing supervision at states the student actually visits (Lu, 2025; Agarwal et al., 2024; Gu et al., 2024; Ko et al., 2024). Multi-teacher OPD (MOPD) extends this framework to complementary specialists, enabling a student to acquire expertise in mathematical reasoning, code generation, and instruction following within a unified training process (Ma et al., 2026; Gao et al., 2026). This raises a central question: how should supervision be allocated among teachers at each generation state?

Domain-routed MOPD approaches address this question at the example level (Xiaomi LLM-Core Team et al., 2026). As illustrated in Figure 1(a), they select one teacher based on the example's domain label and keep that choice fixed throughout the response (Ma et al., 2026; Gao et al., 2026). This design assumes that an available domain label identifies a suitable teacher and that the selected teacher remains appropriate throughout the response. These assumptions may fail when domain labels are unavailable or supervision needs change within a response. For instance, stronger mathematical reasoning does not necessarily imply better adherence to instruction constraints (Fu et al., 2026a). The illustrative prompt in Figure 1(a) requires mathematical derivation, code-based verification, and JSON formatting. A mathematics specialist may provide useful guidance during derivation without being equally suited to code verification or output formatting. Assigning one teacher to the full response cannot accommodate such changes in supervision needs.

More fundamentally, a teacher's global domain expertise does not guarantee local supervision reliability at the current generation state (He et al., 2026a; Wang et al., 2026b). Global expertise describes what a teacher is good at, whereas local reliability concerns whether its learning signal at the current state is suitable and worth transferring. In OPD, local reliability must be assessed at student-generated prefixes rather than fixed reference prefixes (Fu et al., 2026b). Teacher-student

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

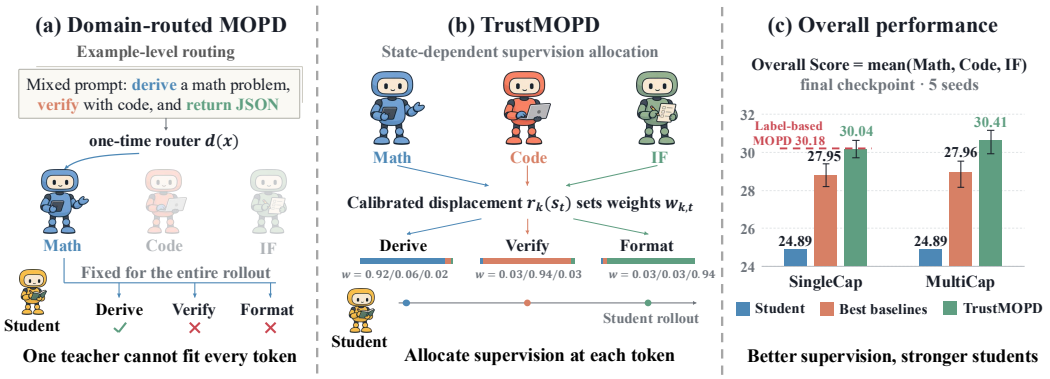

Figure 1: **(a)** Domain-routed MOPD assigns one teacher per example for the full response, even when a mixed prompt requires different expertise for derivation, code verification, and output formatting. **(b)** TrustMOPD uses calibrated teacher-reference displacement as a reliability proxy and converts the scores into token-level weights without domain labels. **(c)** TrustMOPD outperforms the strongest label-free baselines when trained on SINGLECAP and MULTICAP, while approaching label-based MOPD on SINGLECAP. Overall is the mean of mathematics, code, and instruction-following scores. Results are means over five seeds; error bars denote standard deviations.

disagreement is also insufficient as a reliability measure: a large discrepancy may reflect useful expertise, but it may instead arise from stylistic preferences or other idiosyncratic behavior that the student need not learn (Wang et al., 2026c; Li et al., 2026b). Allocating supervision across specialists therefore requires a state-dependent reliability estimate that is comparable across teachers and does not rely on domain labels.

To meet these requirements, we propose **TrustMOPD**, which allocates supervision using each teacher's displacement in next-token preferences relative to a shared reference, as illustrated in Figure 1(b). Our empirical analysis in Figure 3(a) shows that each teacher's mean displacement is largest on prompts from its own domain. However, raw magnitudes differ across teachers: on mathematics prompts, the code teacher's displacement exceeds that of the mathematics teacher. We therefore divide each teacher's displacement by its fixed offline mean to calibrate the scores across teachers. The resulting scores favor the domain-matched teacher on average, motivating their use as proxies for local supervision reliability. At each student-generated prefix, TrustMOPD converts the calibrated scores into supervision weights, assigning greater weight to teachers with higher scores. These weights combine teacher-specific distillation losses into a single training objective, allowing supervision allocation to adapt along the trajectory without domain labels or a learned router. Our theoretical analysis relates teacher-reference displacement to changes in next-token log odds and provides a continuation-utility interpretation under an idealized KL-regularized optimum.

We evaluate TrustMOPD with three specialists in mathematics, code, and instruction following using two training sets: SINGLECAP, containing single-capability prompts, and MULTICAP, containing both single- and multi-capability prompts. Figure 1(c) summarizes the overall performance. On SINGLECAP, TrustMOPD closes 91.5% of the overall-score gap between the initial student and oracle-routed teachers, versus 54.4% for the strongest label-free baseline, approaching label-based MOPD without domain labels. On MULTICAP, where most prompts combine two capabilities and lack a unique domain label, TrustMOPD increases the recovery ratio from 54.5% for the strongest label-free baseline to 98.0%. Further ablations indicate that calibration and token-level allocation are particularly beneficial when training prompts combine multiple capabilities. Together, these results support label-free, state-dependent supervision allocation as an effective approach to integrating complementary expertise from multiple teachers.

2 PRELIMINARIES

On-policy distillation (OPD) trains a student on its own rollouts by querying the teacher at student-generated prefixes (Lu, 2025; Agarwal et al., 2024; Gu et al., 2024; Ko et al., 2024). Let $\mathcal{D}$ denote

the input distribution, π_θ the student model, and $\{\pi_k\}_{k=1}^K$ a collection of K teachers. Given an input $x \sim \mathcal{D}$, the student samples a variable-length response $y \sim \pi_\theta(\cdot \mid x)$, whose length is denoted by $|y|$. At position $t \in \{1, \ldots, |y|\}$, the student-generated state is $s_t = (x, y_{<t})$, where $y_{<t} = (y_1, \ldots, y_{t-1})$ is the generated prefix. We write $p_{\theta,t} = \pi_\theta(\cdot \mid s_t)$ and $q_{k,t} = \pi_k(\cdot \mid s_t)$ for the next-token distributions of the student and teacher k, respectively.

Domain-routed multi-teacher OPD uses each training input's domain label to select a teacher (Ma et al., 2026). Indexing each domain by its corresponding teacher, let $d(x) \in \{1, \ldots, K\}$ denote the domain label of input x. The student is trained to minimize

$$\mathcal{L}_{\mathrm{MOPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\frac{1}{|y|} \sum_{t=1}^{|y|} D_{\mathrm{KL}} \left(p_{\theta,t} \,\|\, q_{d(x),t} \right) \right]. \tag{1}$$

Thus, all tokens in a response are distilled from the same domain-specific teacher, using reverse KL at student-generated prefixes (Gu et al., 2024; Gao et al., 2026).

3 TrustMOPD: Reliability-Aware Supervision Allocation

TrustMOPD computes and calibrates a reference-relative displacement score for each teacher at every student-generated state, then uses these scores as proxies for local reliability to allocate supervision. The resulting weights determine each teacher's contribution to student training, enabling adaptive supervision without domain labels or a learned router.

3.1 Reference-Relative Displacement as a Proxy for Local Reliability

We use domain specialists obtained by applying domain-specific RL to a shared initial policy π_{ref} as teachers, with π_{ref} serving as the reference for measuring their changes in next-token preferences. Figure 3(a) shows that each teacher's mean displacement is largest on its own domain, supporting a connection between reference-relative policy change and specialization. We therefore use displacement as the basis of a local reliability proxy, with cross-teacher calibration introduced in §3.2. Unlike teacher–student divergence, which can reflect both expertise gaps and stylistic differences, teacher–reference displacement measures the changes induced by specialist training.

Let $\mathcal{V}$ denote the vocabulary and $q_{\mathrm{ref},t} = \pi_{\mathrm{ref}}(\cdot \mid s_t)$ the reference's next-token distribution at state s_t. We define the centered reference-relative displacement $d_{k,t} \in \mathbb{R}^{|\mathcal{V}|}$ and its magnitude $\rho_{k,t}$ by

$$d_{k,t}(v) := \log \frac{q_{k,t}(v)}{q_{\mathrm{ref},t}(v)} - \frac{1}{|\mathcal{V}|} \sum_{v' \in \mathcal{V}} \log \frac{q_{k,t}(v')}{q_{\mathrm{ref},t}(v')}, \tag{2}$$

$$\rho_{k,t} := \|d_{k,t}\|_2.$$

The mean is taken over the vocabulary at the same state. Centering removes the token-independent component of the log-policy ratio, which carries no information about relative next-token preferences. The Euclidean norm then discards the direction of the signed displacement while retaining its magnitude, yielding one nonnegative score per teacher. Because both distributions are conditioned on the student's current prefix, $\rho_{k,t}$ can vary across positions within a response. Further analysis in §3.4 provides an exact log-odds interpretation independent of the teacher's training objective and a utility-contrast interpretation under an idealized KL-regularized optimum.

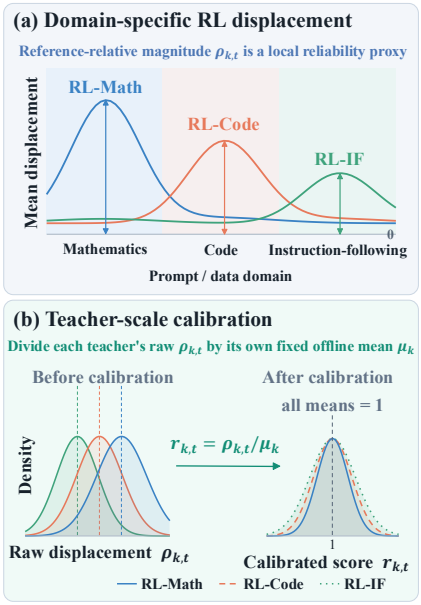

Figure 2: **(a)** Schematic of domain-dependent teacher-reference displacement, with different scales across teachers. **(b)** Dividing each teacher's displacement by its fixed calibration mean μ_k gives each teacher's scores a token-weighted mean of one on the calibration distribution.

3.2 From Displacement to Supervision Weights

Although $\rho_{k,t}$ measures how strongly teacher k departs from the reference at state s_t, its absolute scale is not directly comparable across teachers. Independent RL runs may differ in reward scale, regularization strength, and update magnitude, leading to systematic differences in displacement scale across teachers. For example, Figure 3(a) reports mean raw displacements of 6.33 for the code teacher and 4.08 for the domain-matched mathematics teacher on mathematics prompts. As illustrated in Figure 2(b), we normalize each displacement by the teacher's token-weighted mean magnitude on a fixed calibration distribution. Let π_{θ_0} denote the student before distillation, $\mathcal{D}_{\text{cal}}$ an unlabeled calibration sample, and $\mathbb{E}_{\text{cal}}$ the expectation over $x \sim \mathcal{D}_{\text{cal}}$ and $y \sim \pi_{\theta_0}(\cdot \mid x)$. We define

$$\mu_k := \frac{\mathbb{E}_{\text{cal}}\left[\sum_{t=1}^{|y|} \rho_{k,t}\right]}{\mathbb{E}_{\text{cal}}\left[|y|\right]}, \qquad r_{k,t} := \frac{\rho_{k,t}}{\mu_k}. \tag{3}$$

The calibration constants $\{\mu_k\}_{k=1}^{K}$ are estimated once and fixed throughout distillation. Each teacher therefore has unit token-weighted mean score on the calibration distribution, with $r_{k,t} > 1$ indicating above-baseline displacement. This normalization cancels any teacher-specific multiplicative factor shared by the calibration and current states. Cross-teacher comparison is relative to the chosen calibration distribution and requires no domain labels.

TrustMOPD converts the calibrated scores into supervision weights through power normalization:

$$w_{k,t} := \frac{r_{k,t}^{\gamma}}{\sum_{j=1}^{K} r_{j,t}^{\gamma}}, \qquad \sum_{k=1}^{K} w_{k,t} = 1, \tag{4}$$

where $\gamma > 0$ controls how strongly differences in calibrated scores affect supervision allocation. We use $\gamma > 1$ to amplify relative differences in calibrated scores and concentrate supervision on higher-scoring teachers. Larger values of γ produce more concentrated allocations.

3.3 Reliability-Weighted Multi-Teacher Distillation

We extend equation 1 to a weighted sum of teacher-specific distillation losses at each student-generated state:

$$\mathcal{L}_{\text{MOPD}}(\theta; w) := \mathbb{E}_{\substack{x \sim \mathcal{D} \\ y \sim \pi_\theta(\cdot|x)}} \left[\frac{1}{|y|} \sum_{t=1}^{|y|} \sum_{k=1}^{K} w_{k,t}\, D_{\text{KL}}\left(p_{\theta,t} \,\|\, q_{k,t}\right) \right], \qquad w_{k,t} \geq 0, \quad \sum_{k=1}^{K} w_{k,t} = 1. \tag{5}$$

Setting $w_{k,t} = \mathbb{I}\{k = d(x)\}$ at every position recovers the domain-routed MOPD objective in equation 1. TrustMOPD instead replaces these one-hot weights with the reliability weights defined in Section 3.2:

$$\underbrace{w_{k,t}^{\text{MOPD}} = \mathbb{I}\{k = d(x)\}}_{\text{label-based}} \quad \longrightarrow \quad \underbrace{w_{k,t}^{\text{Trust}} = \frac{(\rho_{k,t}/\mu_k)^{\gamma}}{\sum_{j=1}^{K} (\rho_{j,t}/\mu_j)^{\gamma}}}_{\text{label-free}}. \tag{6}$$

Substituting w^{Trust} into equation 5 defines $\mathcal{L}_{\text{TrustMOPD}}(\theta)$. The weights adapt to each student-generated state while keeping the total teacher weight at one. Their computation requires no domain labels or a learned router. The weights are computed on each sampled rollout and treated as constants during the corresponding update via stop-gradient.

3.4 Properties of the TrustMOPD Allocation

For positive teacher and reference distributions, $d_{k,t}(v) - d_{k,t}(u)$ exactly equals the teacher–reference change in log odds between tokens v and u, independently of the training objective. The following results give a utility interpretation; proofs and further discussion appear in Appendix A.

Theorem 1 (Utility-contrast recovery)**.** *Assume each teacher π_k is Bellman-optimal at all calibration and distillation states for a finite-horizon problem with finite rewards and a per-state KL penalty relative to the full-support reference π_{ref}, with fixed coefficient $\beta_k > 0$. Let $Q_k^\star(s_t, v)$ denote the*

current token reward plus optimal KL-regularized future value, $\bar{Q}^{\star}_{k,t}$ its vocabulary mean, and $\mathbf{1}$ the all-ones vector. Then

$$d_{k,t}(v) = \frac{Q^{\star}_k(s_t, v) - \bar{Q}^{\star}_{k,t}}{\beta_k}, \qquad \rho_{k,t} = \frac{\|Q^{\star}_k(s_t, \cdot) - \bar{Q}^{\star}_{k,t}\mathbf{1}\|_2}{\beta_k}. \tag{7}$$

Theorem 1 therefore interprets $\rho_{k,t}$ as measuring how strongly continuation utility varies across next-token choices for a given teacher.

Corollary 1 (Relative-utility allocation)**.** *Under the assumptions of Theorem 1, let $\sigma_{k,t} := \|Q^{\star}_k(s_t, \cdot) - \bar{Q}^{\star}_{k,t}\mathbf{1}\|_2$ and let $\bar{\sigma}_k$ be its token-weighted calibration mean, defined as in equation 3. If $\bar{\sigma}_k > 0$ for every teacher, then at any state where $\sum_j \sigma_{j,t} > 0$,*

$$r_{k,t} = \frac{\sigma_{k,t}}{\bar{\sigma}_k}, \qquad w^{\mathrm{Trust}}_{k,t} = \frac{(\sigma_{k,t}/\bar{\sigma}_k)^{\gamma}}{\sum_{j=1}^{K}(\sigma_{j,t}/\bar{\sigma}_j)^{\gamma}}. \tag{8}$$

Thus, calibration cancels the explicit $1/\beta_k$ factor, and supervision follows utility-contrast magnitudes relative to each teacher's calibration baseline.

4 Experiments

4.1 Setup

Models. By default, we use the SmolLM3-3B checkpoints released by OpenMOPD (Bakouch et al., 2025; Gao et al., 2026). The mixed-domain SFT checkpoint serves as both the student initialization π_{θ_0} and the shared reference π_{ref}. The $K = 3$ teachers are domain-specific RL fine-tunes of this checkpoint for mathematics, code, and instruction following. Appendix B.1 lists all model checkpoints and links.

Training data. We construct two training sets of $7{,}458$ prompts each, excluding evaluation data from both. SingleCap, sampled from OpenMOPD (Gao et al., 2026), contains $1{,}540$ mathematics, $1{,}870$ code, and $4{,}048$ instruction-following prompts, each with a unique domain label. Inspired by MathIF (Fu et al., 2026a), we construct MultiCap with both single-capability prompts and prompts combining two capabilities: mathematics with instruction-following constraints, code with instruction-following constraints, and mathematics–code compositions. For these mixed-capability prompts, no single domain label fully captures the expertise required for teacher selection. We compare methods only within each training set. Appendix B.2 details data construction and filtering.

Baselines. All distillation methods share the same training protocol and differ only in teacher supervision allocation. We compare TrustMOPD with Uniform weighting ($w_{k,t} = 1/K$), Random weighting sampled independently at each token from the uniform distribution over the teacher simplex, and three single-teacher variants that use the mathematics, code, or instruction-following teacher throughout training. The Dirichlet distribution is uniform over nonnegative teacher-weight vectors that sum to one. On SingleCap, we additionally evaluate MOPD (Ma et al., 2026), which selects one teacher per response using the ground-truth domain label. We omit this baseline on MultiCap, where mixed-capability prompts lack a unique domain label. We also report three training-free references: the initial student, the arithmetic parameter average of the three teachers (Wortsman et al., 2022), and Routed Teachers, which directly uses the corresponding RL teacher to generate responses for each evaluation domain. Baseline details appear in Appendix B.3.

Evaluation. We evaluate mathematics on AIME25 (Mathematical Association of America, 2025) and AIME26 (Mathematical Association of America, 2026), code on LiveCodeBench v5/v6 (Jain et al., 2025), and instruction following on IFEval (Zhou et al., 2023) and IFBench (Pyatkin et al., 2025). For each example, we sample 16 responses for mathematics, 5 for code, and 1 for instruction following, all at a temperature of 0.6. Scores are averaged over the sampled responses. Each domain score is the mean of its two benchmark scores; *overall* is the unweighted mean across domains. For trained methods, we evaluate the final checkpoint from each of five training seeds and report the mean $\pm$ standard deviation across seeds. No evaluation set is used for checkpoint selection. Additional experimental details, hyperparameters, algorithm pseudocode, and computing cost appear in Appendix B.4–B.6, D.

Table 1: For trained methods, results are mean ± standard deviation over five seeds, using the final checkpoint from each run. **Bold** and underlined values mark the best and second-best label-free methods within each block; green rows use ground-truth domain labels and are excluded from ranking. Δ_{Init} is the overall-score change from the initial student. The "teacher alone" rows evaluate each specialist directly; their domain-matched scores form the ROUTED TEACHERS row.

Data	Method / allocation rule	Math	Code	IF	Overall	Δ_{Init}
		AIME 25/26	LCB v5/v6	IFEval/IFBench		
No-distillation references						
	Student initialization	17.02	15.97	41.66	24.89	0.00
	Math teacher alone π_{math}	22.50	17.26	41.16	26.97	+2.08
	Code teacher alone π_{code}	20.62	21.34	41.55	27.84	+2.95
	IF teacher alone π_{IF}	15.83	14.92	47.73	26.16	+1.27
	Oracle-routed teachers	22.50	21.34	47.73	30.52	+5.63
	Parameter-averaged teacher	19.29	16.84	43.86	26.66	+1.77
SINGLECAP	MOPD[1] $\mathbf{e}_{z(x)}$	22.58 ±0.77	20.48 ±0.39	47.49 ±1.37	30.18 ±0.25	+5.29
SINGLECAP	Uniform $\frac{1}{K}\mathbf{1}$	20.14 ±0.23	18.67 ±0.38	45.04 ±1.23	27.95 ±0.59	+3.06
SINGLECAP	Random $\text{Dirichlet}(1,\ldots,1)$	20.22 ±0.95	18.31 ±0.59	44.62 ±0.99	27.72 ±0.72	+2.83
SINGLECAP	Single teacher $\mathbf{e}_{\text{math}}$	21.80 ±0.61	17.01 ±0.45	40.80 ±0.73	26.54 ±0.29	+1.65
SINGLECAP	Single teacher $\mathbf{e}_{\text{code}}$	20.44 ±1.48	**20.77** ±0.56	42.17 ±0.71	27.79 ±0.55	+2.90
SINGLECAP	Single teacher $\mathbf{e}_{\text{IF}}$	16.41 ±0.50	15.60 ±0.51	**48.33** ±0.53	26.78 ±0.13	+1.89
SINGLECAP	TrustMOPD $\propto \mathbf{r}_t^{\gamma}$	**21.91** ±0.61	20.67 ±0.75	47.56 ±0.48	**30.04** ±0.47	+5.15
MULTICAP	Uniform $\frac{1}{K}\mathbf{1}$	20.20 ±0.43	18.87 ±0.52	43.88 ±1.42	27.65 ±0.53	+2.76
MULTICAP	Random $\text{Dirichlet}(1,\ldots,1)$	20.24 ±1.13	18.95 ±0.66	44.69 ±1.20	27.96 ±0.49	+3.07
MULTICAP	Single teacher $\mathbf{e}_{\text{math}}$	21.88 ±0.66	17.20 ±0.70	41.80 ±1.28	26.96 ±0.55	+2.07
MULTICAP	Single teacher $\mathbf{e}_{\text{code}}$	20.18 ±0.94	**21.25** ±0.37	41.83 ±1.71	27.75 ±0.69	+2.86
MULTICAP	Single teacher $\mathbf{e}_{\text{IF}}$	15.52 ±0.87	14.70 ±0.80	46.98 ±0.80	25.73 ±0.38	+0.84
MULTICAP	TrustMOPD $\propto \mathbf{r}_t^{\gamma}$	**22.92** ±1.40	20.69 ±0.50	**47.62** ±1.41	**30.41** ±0.60	+5.52

4.2 OVERALL EFFECTIVENESS

Table 1 evaluates two settings: SINGLECAP tests whether TrustMOPD can approach label-based routing without using domain labels, while MULTICAP tests its effectiveness when training prompts combine capabilities. The "teacher alone" rows show that no specialist's score on any other evaluated domain falls more than 1.3 points below the initial student's, indicating limited cross-domain degradation in this setup.

• **Obs. 1: TrustMOPD approaches label-based routing on SINGLECAP without domain labels.** TrustMOPD achieves 30.04 ± 0.47 overall, 0.14 points below label-based MOPD and 2.09 points above the strongest label-free baseline, UNIFORM. This closes 91.5% of the overall-score gap between the initial student and ROUTED TEACHERS, compared with 54.4% for UNIFORM. TrustMOPD improves scores in all three domains over the initial student.

• **Obs. 2: TrustMOPD remains effective with mixed-capability training data.** On MULTICAP, TrustMOPD reaches 30.41 ± 0.60 overall, exceeding the strongest label-free baseline, RANDOM, by 2.45 points. This closes 98.0% of the overall-score gap between the initial student and ROUTED TEACHERS, compared with 54.5% for RANDOM. TrustMOPD outperforms the initial student across all domains and achieves the highest label-free scores in mathematics and instruction following.

4.3 ANALYZING STATE-DEPENDENT SUPERVISION ALLOCATION

We examine domain alignment in raw displacement, calibrated scores, and supervision weights, and illustrate how allocation varies within individual responses. For rollout i with response y_i, we compute teacher k's mean weight as $\bar{w}_{i,k} = |y_i|^{-1} \sum_{t=1}^{|y_i|} w_{i,k,t}$ and group rollouts by domain or capability composition. Labels are used only for analysis, never for allocation.

• **Obs. 3: Displacement and allocation exhibit domain alignment.** We first analyze SINGLECAP, where each prompt targets a single capability and has a unique domain label. Figure 3(a) shows that each teacher's mean raw displacement is highest on its own domain, although the scales differ

[1] Our reported MOPD results use the OpenMOPD (Gao et al., 2026) implementation.

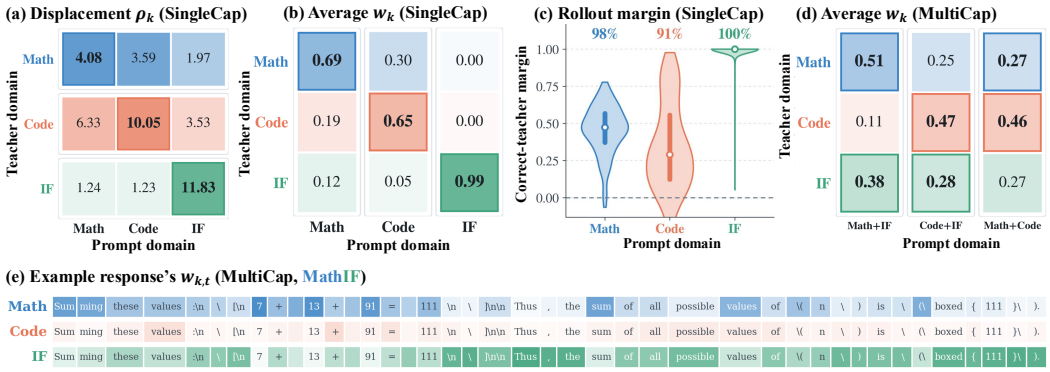

Figure 3: **State-dependent supervision allocation by TrustMOPD.** (a) Mean raw displacement $\rho_{k,t}$ from the shared reference. Each teacher's mean displacement is largest on its matching domain, but raw magnitudes differ in scale across teachers. (b) Mean rollout-level teacher weights on SINGLECAP, grouped by prompt domain. (c) Distribution of the domain-matched teacher's margin $m_i = \bar{w}_{i,d_i} - \max_{k \neq d_i} \bar{w}_{i,k}$, where $\bar{w}_{i,k}$ is teacher k's mean weight over tokens in rollout i, and $d_i = d(x_i)$ is the prompt's ground-truth domain index. Percentages report the fraction of rollouts with $m_i > 0$. (d) Mean rollout-level teacher weights on MULTICAP, grouped by capability composition; outlined cells mark relevant teachers. (e) Token-level teacher weights for an example mathematics-and-instruction-following response. Rows correspond to teachers; darker shading indicates higher weights.

across teachers. Table 6 shows that after calibration, the domain-matched teacher has the highest mean score in each prompt domain. Figure 3(b) shows corresponding alignment in mean teacher weights: the matched teachers receive 0.69, 0.65, and 0.99 for mathematics, code, and instruction following, respectively. Figure 3(c) further shows that the domain-matched teacher receives the largest rollout-averaged weight on 98%, 91%, and 100% of rollouts, respectively.

• **Obs. 4: Allocation favors relevant teachers on average and varies within a response.** Figure 3(d) shows that on MULTICAP, relevant teachers jointly receive 0.89, 0.75, and 0.73 of total weight for mathematics plus instruction following, code plus instruction following, and mathematics plus code, respectively, all exceeding the $2/3$ share under uniform weighting. For mathematics-and-instruction-following prompts, the respective teacher weights are 0.51 and 0.38, compared with 0.11 for the code teacher. In the example response shown in Figure 3(e), the mathematics teacher receives its highest weights around the operands "7", "13", and "91" and their arithmetic composition. After the result "111" is obtained, the instruction-following teacher dominates the concluding explanation and the formatting command "`\boxed{111}`", while the code teacher remains weak throughout. Additional teacher-weight plots and token-level visualizations covering all three capability compositions appear in Appendix C.3, with further training-time allocation examples in Appendix E.

4.4 ABLATIONS, SENSITIVITY, AND GENERALIZATION

Ablation settings. Figure 4 compares ablations of three design choices. First, we replace the reference-relative displacement proxy $\rho_{k,t}$ with teacher likelihood, negative teacher entropy, or teacher–student KL, calibrating each alternative before applying the same power normalization. Second, "Without μ_k" retains the displacement proxy but disables calibration by setting $\mu_k = 1$. Third, "Response-level" applies each teacher's rollout-averaged weight at every token, preserving its total weight within the rollout while removing within-response variation in allocation. All variants share the student, teacher pool, and training protocol of full TrustMOPD. Detailed definitions and formulas for all variants appear in Appendix C.4.

• **Obs. 5: Displacement performs best among tested proxies, while calibration and token-level allocation contribute more on MULTICAP.** Relative to full TrustMOPD, alternative proxies reduce overall by 1.63–2.08 points on SINGLECAP and 1.48–3.06 on MULTICAP. Calibration and allocation granularity have larger effects on MULTICAP: removing μ_k reduces overall by 1.28 points, compared with 0.31 on SINGLECAP; response-level allocation incurs a 0.92-point drop, compared

with 0.05 on SINGLECAP. These differences suggest that calibrating teacher scores and adapting weights within a response are more beneficial when training prompts combine multiple capabilities. Table 9 provides absolute scores and per-domain results.

Sensitivity settings. We vary the sharpness exponent $\gamma \in \{3, 5, 7, 9\}$ with other settings fixed and include UNIFORM as a reference. We report overall gains over the initial student. Unless explicitly varied, all sensitivity and transfer experiments use $\gamma = 7$ and the default training hyperparameters. For both analyses, we evaluate the final checkpoint from each of five training seeds.

• **Obs. 6: Performance remains stable on SINGLECAP but peaks at $\gamma = 7$ on MULTICAP.** Figure 5(a) shows that TrustMOPD outperforms matched UNIFORM controls at every tested γ. On SINGLECAP, gains remain between 5.11 and 5.40 points as the mean maximum teacher weight increases from 0.66 to 0.86 in panel (b). On MULTICAP, gains rise from 4.42 at $\gamma = 3$ to 5.52 at $\gamma = 7$, then decline to 4.87 at $\gamma = 9$ despite further concentration. These results suggest that sharpening allocation benefits mixed-capability training up to a point, but further concentration need not improve performance. Table 10 provides absolute scores.

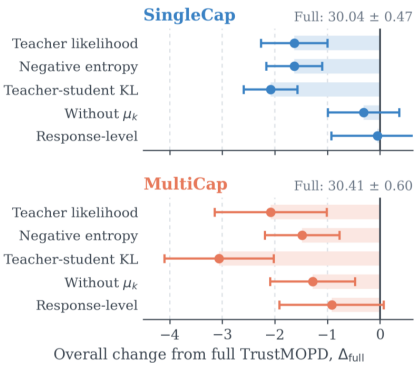

Figure 4: **Core component ablations on SINGLECAP and MULTICAP.** Each row changes one component of TrustMOPD while holding other settings fixed. Points show mean overall-score differences between each variant and full TrustMOPD; the vertical line at zero marks the reference. Error bars denote standard deviations of paired score differences across five seeds. Further right is better.

Transfer settings. For backbone transfer, we replace the default SmolLM3-3B setup with DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025) as the student initialization and shared reference, together with three public RL-finetuned teachers derived from this checkpoint. For data-source transfer, we rebuild both training sets using DeepMath-103K (He et al., 2026b), KodCode-V1 (Xu et al., 2025), and the Tülu 3 persona instruction-following dataset (Lambert et al., 2025), matching the original dataset sizes and capability compositions while retaining the default student and teachers. Evaluation data are excluded from both training sets. We report overall gains over UNIFORM under the same model and data settings. Model checkpoints and data construction details appear in Appendices B.1 and B.2, respectively.

• **Obs. 7: TrustMOPD remains effective across model backbones and training-data sources.** Figure 5(c) shows that on DeepSeek-R1-Distill-Qwen-7B, TrustMOPD exceeds matched UNIFORM controls in overall score by 1.40 points on SINGLECAP and 1.15 on MULTICAP. Figure 5(d) shows that TrustMOPD remains effective with independently sourced prompts, outperforming matched UNIFORM controls by 2.22 points on SINGLECAP and 1.95 on MULTICAP. Its overall scores are slightly lower than those obtained with the corresponding default training sets, by 0.60 and 0.79 points, respectively. These results support the transferability of TrustMOPD's advantage over uniform weighting. Table 11 provides absolute scores.

5 RELATED WORK

On-policy distillation and adaptive supervision. OPD queries teachers at student-generated prefixes (Agarwal et al., 2024; Gu et al., 2024; Lu, 2025); Zan et al. (2026) review the broader literature. Related Knowledge Distillation (KD) methods refine objectives (Wu et al., 2025; Ko et al., 2024; 2025; Jin et al., 2026b) or construct supervision targets (Jang et al., 2026; Zhang et al., 2026c), while diagnostic studies examine teacher–student compatibility and OPD failures (Li et al., 2026b; Fu et al., 2026b; Wang et al., 2026b). BRTS selects teacher rollouts by correctness and student alignment (Zhang et al., 2026b); RG-OPD and TGOPD gate distillation using verifier feedback (Akhondzadeh et al., 2026; Zhang et al., 2026d), while ReOrder-OPD orders prompts by a reliability proxy (Zhu et al., 2026). Finer-grained methods select informative or teachable tokens (Huang et al., 2026; Xu et al., 2026; Wang et al., 2026c), apply position-dependent weights (Liu et al., 2026b), restrict distillation to reasoning prefixes (Zhang et al., 2026a), or combine trajectory

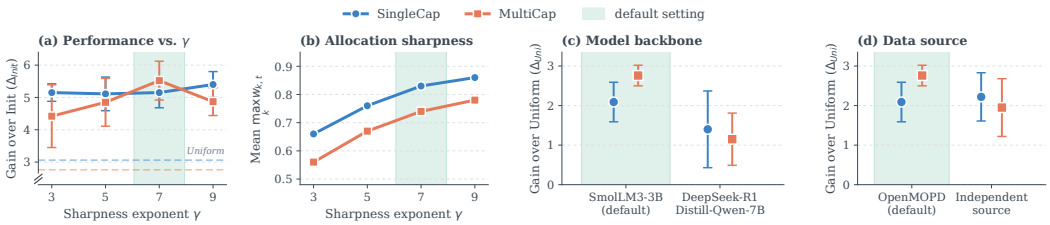

Figure 5: **Sensitivity and transfer of TrustMOPD.** (a) Overall gain over the initial student, Δ_{Init}, as the sharpness exponent γ varies. Dashed lines indicate the corresponding UNIFORM baselines. (b) Allocation sharpness, measured by $\mathbb{E}_{s_t}[\max_k w_{k,t}]$, as γ varies. (c) Transfer to a different backbone and teacher pool. (d) Transfer across training-data sources with the models fixed. In (c,d), Δ_{Uni} is the overall gain over the matched UNIFORM baseline, computed by subtracting scores within each seed. Results in (a,c,d) are means over five seeds; error bars denote standard deviations of these gains. The y-axis in (a) is truncated.

filtering with token reweighting (Li et al., 2026c). IDA-OPD attenuates updates predicted to contract entropy (Yang et al., 2026). For vision-language models, VA-OPD reweights rollouts by visual advantage and averages token-level KL within high- and low-advantage groups (Liu et al., 2026a).

Multi-teacher distillation and capability integration. Multi-teacher KD uses multiple teachers' outputs or representations (You et al., 2017; Fukuda et al., 2017), with adaptive variants learning instance-level teacher weights (Yuan et al., 2021) or sampling teachers per step (Ding et al., 2024). LLM integration transfers rationales (Tian et al., 2025; Jin et al., 2026a), matches teacher distributions (Wan et al., 2024), or distills interaction graphs (Chen et al., 2024). Parameter merging combines weights or parameter updates (Wortsman et al., 2022; Ilharco et al., 2023; Yadav et al., 2023); FuseChat combines distribution fusion with parameter merging (Wan et al., 2025). Under offline supervision, EWAD weights teachers by entropy and interpolates between teacher KD and gold supervision using inter-teacher agreement (Sumit et al., 2026). In OPD, MOPD selects one teacher per response using domain labels (Ma et al., 2026), while Open-MOPD balances optimization budgets and refreshes rewards under oracle routing (Gao et al., 2026). DF-OPD generates questions with teachers and derives routing labels from teacher identity (Li et al., 2026a). Other methods adapt domain sampling (Sun et al., 2026), verify teacher eligibility from answers (He et al., 2026a), or weight contributions by post-debate confidence (Wang et al., 2026a). Uncertainty-calibrated MOPD retains prompt-level routing, combining dual-temperature sampling, positive-advantage-density trajectory filtering, and entropy-calibrated token-update gating (Liu et al., 2026c). VG-OPD verifies criterion-specific expert gains and localizes supervision using teacher–student disagreement (Xu & Zhang, 2026). For vision-language models, Video-MOPD retains domain routing and selects examples by verified teacher reliability and teacher–student performance gaps (Qin et al., 2026); H-OPD weights vision-language and text-only teachers at each token using prediction entropy (Yin et al., 2026). TrustMOPD instead derives token-level teacher weights from calibrated displacement relative to a shared reference, without domain labels, outcome verifiers, learned routers, or teacher debate.

6 CONCLUSION

We introduce TrustMOPD, a label-free method that allocates token-level supervision using calibrated teacher displacement from a shared reference. Across both training corpora, it achieves the highest overall score among the compared label-free distillation methods and approaches label-based MOPD on SINGLECAP. Analyses show domain alignment in displacement and supervision weights and illustrate within-response changes in allocation. Transfer experiments further show gains over matched uniform-weighting baselines on an additional backbone and independently sourced training data. These results support calibrated reference-relative displacement as a useful signal for integrating complementary teacher capabilities. The current formulation assumes a shared reference. Future work includes extending TrustMOPD to teachers without a common initialization and evaluating it on larger models and a broader range of datasets and domains.

AI USE STATEMENT

We used generative AI tools as assistants for writing, literature discovery, and research discussions. Specifically, these tools helped refine phrasing, grammar, concision, and readability; locate potentially relevant prior work for the authors to examine; and discuss author-developed research ideas and check the clarity and logical consistency of methodological descriptions. The research questions, core methodology, experimental design, and interpretation of results were developed and determined by the authors. The authors reviewed and verified all incorporated AI-assisted content and take full responsibility for the final manuscript, references, and reported results.

ETHICS STATEMENT

This work distills publicly released language-model checkpoints into a single student model using publicly available prompt datasets. It involves no human-subject research and does not collect private or sensitive data. As with knowledge distillation more broadly, the student may inherit biases, unsafe behaviors, and other limitations from its teachers. TrustMOPD's supervision allocation mechanism is designed to integrate teacher capabilities and is not intended as a safety mechanism. Consolidating multiple specialized teachers into a single student may reduce deployment costs, supporting beneficial applications while also lowering barriers to misuse. Downstream deployments should therefore include safety and bias evaluations tailored to their intended applications and incorporate appropriate safeguards.

REPRODUCIBILITY STATEMENT

Method and implementation. §3 specifies TrustMOPD's displacement score, calibration procedure, supervision weights, and distillation objective. Our implementation builds on the public MOPD codebase and is available at `https://anonymous.4open.science/r/TrustMOPD-F624`. It includes displacement scoring, calibration, teacher weighting, prompt-set construction, and the evaluation harness.

Experimental protocols. Appendix B documents the models, prompt-set compositions, baselines, and training and calibration protocols. Table 2 lists the public model repositories and initialization checkpoints, including the provenance of the shared pre-RL reference. For the data-source transfer experiment in Figure 5(d), §B.2.1 identifies the upstream dataset repositories and per-class sample counts. §B.2.2 specifies the decontamination rule, thresholds, and comparison pool, which includes the teachers' RL training prompts. The sharpness exponent γ is defined in §3.2; its default value of 7 is specified in §4.4, and Appendix C.5 provides sensitivity results. §4.1 describes the evaluation benchmarks, generation counts, sampling temperatures, and score aggregation. For trained distillation methods, we evaluate the final checkpoint from each of five training seeds and report means and standard deviations, without using evaluation sets for checkpoint selection. Table 12 reports total GPU-hours and peak GPU memory for each allocation rule, as well as the one-off calibration cost.

Theoretical derivations. Appendix A provides the assumptions and proofs underlying the theoretical analysis. §A.2 derives the utility identity used to interpret teacher displacement, and §A.5 discusses how this interpretation applies to practical RL teachers. §A.3 proves the centering identity and characterizes the effect of batch-local normalization on between-batch mean scores. §A.4 derives the geometric log-target aggregation identity and states the conditions under which it holds.

Diagnostics without distillation training. Two empirical diagnostics can be reproduced without distillation training: the calibrated-score matrix in Table 6 and the calibration constants used in our experiments, highlighted in the two tinted rows of Table 7. Reproducing these diagnostics requires only the publicly released teacher and reference checkpoints, the disclosed prompt sets, and the specified rollout and calibration protocols. Because the shared reference also serves as the initial student, neither diagnostic requires a distilled student checkpoint.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria, 2024.

Mohammad Sadegh Akhondzadeh, Vijay Lingam, Atula Tejaswi, Chanakya Ekbote, Sujay Sanghavi, and Aleksandar Bojchevski. Reward-gated on-policy distillation. *arXiv preprint*, 2026.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner, 2025. URL `https://huggingface.co/blog/smollm3`.

Justin Chen, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. MAGDi: Structured distillation of multi-agent interaction graphs improves reasoning in smaller language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, volume 235, pp. 7220–7235, Vienna, Austria, 2024.

Yang Chen, Zihan Liu, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Training math reasoning model with reinforcement learning, 2025a. URL `https://research.nvidia.com/labs/adlr/acemath_rl/`.

Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. AceReason-Nemotron: Advancing math and code reasoning through reinforcement learning. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, San Diego, CA, USA and Mexico City, Mexico, 2025b.

Zixiang Ding, Guoqing Jiang, Shuai Zhang, Lin Guo, and Wei Lin. How to trade off the quantity and capacity of teacher ensemble: Learning categorical distribution to stochastically employ a teacher for distillation. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI 2024)*, pp. 17915–17923, Vancouver, Canada, 2024.

Tingchen Fu, Yafu Li, Jiawei Gu, Xiaoye Qu, and Yu Cheng. Scaling reasoning, losing control: Evaluating instruction following in large reasoning models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL 2026)*, pp. 40445–40463, San Diego, CA, USA, 2026a.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint*, 2026b.

Takashi Fukuda, Masayuki Suzuki, Gakuto Kurata, Samuel Thomas, Jia Cui, and Bhuvana Ramabhadran. Efficient knowledge distillation from an ensemble of teachers. In *Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH 2017)*, pp. 3697–3701, Stockholm, Sweden, 2017.

Huan-ang Gao, Haohan Chi, Yong Yan, Shiyuan Feng, Hanlin Wu, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Open-MOPD: Diagnosing and fixing capability imbalance in multi-teacher on-policy distillation. *arXiv preprint*, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li,

H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Xixiang He, Xingming Li, Baiqi Wu, Qiyao Sun, Xuanyu Ji, Ao Cheng, and Qingyong Hu. Learn from whoever is right: Answer-verified multi-teacher distillation for multi-domain LLMs. *arXiv preprint*, 2026a.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. DeepMath-103K: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *Proceedings of the 14th International Conference on Learning Representations (ICLR 2026)*, Rio de Janeiro, Brazil, 2026b.

Haiduo Huang, Jiangcheng Song, Yadong Zhang, and Pengju Ren. SelecTKD: Selective token-weighted knowledge distillation for LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2026)*, pp. 19287–19297, Denver, CO, USA, 2026.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*, Kigali, Rwanda, 2023.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *Proceedings of the 13th International Conference on Learning Representations (ICLR 2025)*, Singapore, 2025.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42217–42227, San Diego, CA, USA, 2026.

Ruihan Jin, Pengpeng Shao, Zhengqi Wen, Jinyang Wu, Mingkuan Feng, Shuo Yang, Chu Yuan Zhang, and Jianhua Tao. Exploring knowledge purification in multi-teacher knowledge distillation for LLMs. In *Proceedings of the 14th International Conference on Learning Representations (ICLR 2026)*, Rio de Janeiro, Brazil, 2026a.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Proceedings of the 43rd International Conference on Machine Learning (ICML 2026)*, Seoul, South Korea, 2026b.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, volume 235, pp. 24872–24895, Vienna, Austria, 2024.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, volume 267, pp. 31044–31062, Vancouver, Canada, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Proceedings of the Second Conference on Language Modeling (COLM 2025)*, Montreal, Canada, 2025.

Gengsheng Li, Mao Zheng, Mingyang Song, Jie Sun, Zeyuan Liu, Ruiqi Liu, Qiyong Zhong, Haiyun Guo, Junfeng Fang, and Jinqiao Wang. Data-free on-policy distillation. *arXiv preprint*, 2026a.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint*, 2026b.

Yuying Li, Leqi Zheng, Yongzi Yu, Wenrui Zhou, Xuchang Zhong, Xing Hu, Jing Jin, Hangjie Yuan, and Tao Feng. Filter, then reweight: Rethinking optimization granularity in on-policy distillation. *arXiv preprint*, 2026c.

Ruiqi Liu, Xiaolei Lv, Gengsheng Li, Ximo Zhu, Zhiheng Wang, Zhengbo Zhang, Junkai Chen, Zhiheng Li, Bo Li, Jun Gao, and Shu Wu. Visual-advantage on-policy distillation for vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS 2026)*, Sydney, Australia; Atlanta, GA, USA; Paris, France, 2026a.

Xiaogeng Liu, Xinyan Wang, Yingzi Ma, Yechao Zhang, and Chaowei Xiao. When are teacher tokens reliable? position-weighted on-policy self-distillation for reasoning. In *Advances in Neural Information Processing Systems (NeurIPS 2026)*, Sydney, Australia; Atlanta, GA, USA; Paris, France, 2026b.

Ziyuan Liu, Jiao Ou, Jian Liang, Ruiming Tang, and Cheng Luo. Preserving general capabilities during domain specialization with uncertainty-calibrated MOPD. *arXiv preprint*, 2026c.

Kevin Lu. On-policy distillation, 2025. URL `https://thinkingmachines.ai/blog/on-policy-distillation/`.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. MOPD: Multi-teacher on-policy distillation for capability integration in LLM post-training. *arXiv preprint*, 2026.

Mathematical Association of America. 2025 american invitational mathematics examination (AIME), 2025. URL `https://maa.org/maa-invitational-competitions/`.

Mathematical Association of America. 2026 american invitational mathematics examination (AIME), 2026. URL `https://maa.org/maa-invitational-competitions/`.

Hao Peng, Yunjia Qi, Xiaozhi Wang, Bin Xu, Lei Hou, and Juanzi Li. VerIF: Verification engineering for reinforcement learning in instruction following. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP 2025)*, Suzhou, China, 2025.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hanna Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, volume 38, San Diego, CA, USA and Mexico City, Mexico, 2025.

Zhenxin Qin, Peng Shi, Cong Han, Yinlong Qian, Zequn Jie, and Lin Ma. Video-MOPD: Multi-teacher on-policy distillation for video understanding. *arXiv preprint*, 2026.

Dipto Sumit, Ankan Kumar Roy, Sadia Khair Rodela, Atia Haque Asha, Mourchona Afrin, Niloy Farhan, and Farig Yousuf Sadeque. Reliability gated multi-teacher distillation for low resource abstractive summarization. *arXiv preprint*, 2026.

Zechen Sun, Zhiwei Zhang, Fei Zhao, Juntao Li, Mu Chuan, Huayu Deng, Guojian Zhan, Wenliang Chen, Yao Hu, and Min Zhang. D^3-MOPD: Dynamic domain ScheDuling for efficient multi-teacher distillation. *arXiv preprint*, 2026.

Yijun Tian, Yikun Han, Xiusi Chen, Wei Wang, and Nitesh V. Chawla. Beyond answers: Transferring reasoning capabilities to smaller LLMs using multi-teacher knowledge distillation. In *Proceedings of the 18th ACM International Conference on Web Search and Data Mining (WSDM 2025)*, Hannover, Germany, 2025.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei BI, and Shuming Shi. Knowledge fusion of large language models. In *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria, 2024.

Fanqi Wan, Longguang Zhong, Ziyi Yang, Ruijun Chen, and Xiaojun Quan. FuseChat: Knowledge fusion of chat models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP 2025)*, pp. 21618–21642, Suzhou, China, 2025.

Jianze Wang, Ying Liu, Jinlong Chen, Xuchun Hu, Qilong Zhang, Yu Cao, Jun Wang, Hua Yang, Yong Xie, and Qianglong Chen. MAD-OPD: Breaking the ceiling in on-policy distillation via multi-agent debate. *arXiv preprint*, 2026a.

Rui Wang, Hongru Wang, Yi Chen, Boyang Xue, Tianqing Fang, Wenhao Yu, and Kam-Fai Wong. Demystifying on-policy distillation: Roles, pathologies, and regulations. *arXiv preprint*, 2026b.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint*, 2026c.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*, volume 162, pp. 23965–23998, Baltimore, MD, USA, 2022.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking Kullback-Leibler divergence in knowledge distillation for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025)*, pp. 5737–5755, Abu Dhabi, United Arab Emirates, 2025.

Xiaomi LLM-Core Team, Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, Gang Xie, Hailin Zhang, Hanglong Lv, Hanyu Li, Heyu Chen, Hongshen Xu, Houbin Zhang, Huaqiu Liu, Jiangshan Duo, Jianyu Wei, Jiebao Xiao, Jinhao Dong, Jun Shi, Junhao Hu, Kainan Bao, Kang Zhou, Lei Li, Liang Zhao, Linghao Zhang, Peidian Li, Qianli Chen, Shaohui Liu, Shihua Yu, Shijie Cao, Shimao Chen, Shouqiu Yu, Shuo Liu, Tianling Zhou, Weijiang Su, Weikun Wang, Wenhan Ma, Xiangwei Deng, Bohan Mao, Bowen Ye, Can Cai, Chenghua Wang, Chengxuan Zhu, Chong Ma, Chun Chen, Chunan Li, Dawei Zhu, Deshan Xiao, Dong Zhang, Duo Zhang, Fangyue Liu, Feiyu Yang, Fengyuan Shi, Guoan Wang, Hao Tian, Hao Wu, Heng Qu, Hongfei Yi, Hongxu An, Hongyi Guan, Xing Zhang, Yifan Song, Yihan Yan, Yihao Zhao, Yingchun Lai, Yizhao Gao, Yu Cheng, Yuanyuan Tian, Yudong Wang, Zhen Tang, Zhengju Tang, Zhengtao Wen, Zhichao Song, Zhixian Zheng, Zihan Jiang, Jian Wen, Jiarui Sun, Jiawei Li, Jinlong Xue, Jun Xia, Kai Fang, Menghang Zhu, Nuo Chen, Qian Tu, Qihao Zhang, Qiying Wang, Rang Li, Rui Ma, Shaolei Zhang, Shengfan Wang, Shicheng Li, Shuhao Gu, Shuhuai Ren, Sirui Deng, Tao Guo, Tianyang Lu, Weiji Zhuang, Weikang Zhang, Weimin Xiong, Wenshan Huang, Wenyu Yang, Xin Zhang, Xing Yong, Xu Wang, Xueyang Xie, Yilin Jiang, Yixin Yang, Yongzhe He, Yu Tu, Yuanliang Dong, Yuchen Liu, Yue Ma, Yue Yu, Yuxing Xiang, Zhaojun Huang, Zhenru Lin, Zhipeng Xu, Zhiyang Chen, Zhonghua Deng, Zihan Zhang, and Zihao Yue. MiMo-V2-Flash technical report. *arXiv preprint*, 2026.

Xun Xu and Zaixi Zhang. Who teaches which token? verifier-gated multi-expert on-policy distillation for scientific reasoning. *arXiv preprint*, 2026.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token importance in on-policy distillation. *arXiv preprint*, 2026.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. KodCode: A diverse, challenging, and verifiable synthetic dataset for coding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6980–7008, Vienna, Austria, 2025.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, pp. 7093–7115, New Orleans, LA, USA, 2023.

Run Yang, Runpeng Dai, Jie Sun, Jielei Zhang, Fan Zhou, Hongtu Zhu, Peiyi Li, and Longwen Gao. Influence-directed distillation: Solving the diversity bottleneck in sampled-token on-policy distillation. *arXiv preprint*, 2026.

Qixiang Yin, Huanjin Yao, Yuchen Cai, Jianghao Chen, Ziyi Wang, Min Yang, Fei Su, and Zhicheng Zhao. H-OPD: Confidence aware heterogeneous multi-teacher multimodal on-policy distillation. *arXiv preprint*, 2026.

Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2017)*, Halifax, Nova Scotia, Canada, 2017.

Fei Yuan, Linjun Shou, Jian Pei, Wutao Lin, Ming Gong, Yan Fu, and Daxin Jiang. Reinforced multi-teacher selection for knowledge distillation. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021)*, pp. 14284–14291, Online, 2021.

Zhaochen Zan, Runze Liu, Tao Ye, Dandan Tu, and Haoyu Song. Echoes of the oracle: A survey of on-policy distillation in the era of large language models, 2026. URL `https://doi.org/10.13140/RG.2.2.18644.77441`.

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler, II, Qian Qian, Gregory D Lyng, Sanjit Singh Batra, and Robert E. Tillman. Fast and effective on-policy distillation from reasoning prefixes. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25553–25569, San Diego, CA, USA, 2026a.

Ke Zhang, Yunjie Tian, Dongdi Zhao, Yijiang Li, Yuanye Liu, Vishal M Patel, and Di Fu. On-policy distillation with best-of-N teacher rollout selection. *arXiv preprint*, 2026b.

Xinsen Zhang, Zhenkai Ding, Tianjun Pan, Run Yang, Chun Kang, Xue Xiong, and Jingnan Gu. OPSDL: On-policy self-distillation for long-context language models. *arXiv preprint*, 2026c.

Zhiwei Zhang, Zechen Sun, Fei Zhao, Kang Peng, Bin Liang, Huayu Deng, Yao Hu, Kam-Fai Wong, and Mu Chuan. Verify before you distill: Prompt-level teacher gating for on-policy distillation. *arXiv preprint*, 2026d.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint*, 2023.

Ximo Zhu, Ruiqi Liu, Rong Wang, Ping Wu, Xiang Zheng, Wenzhuo Xu, Xubin Yao, Zhiyuan Yan, Bo Li, Jun Gao, and Xiaolei Lv. ReOrder-OPD:reliability-aware prompt ordering for on-policy distillation. *arXiv preprint*, 2026.

A Theoretical Analysis and Structural Properties

This appendix develops the utility interpretation and structural properties of TrustMOPD. §A.1 defines the sequential RL setting, and §A.2 proves the utility-contrast and calibration results. §A.3 explains centering and the frozen calibration divisor, while §A.4 derives the mixed log-target implementation. §A.5 connects these results to practical teacher distributions.

A.1 SEQUENTIAL RL SETTING AND OPTIMAL-TEACHER ASSUMPTION

We model autoregressive generation as a finite-horizon Markov decision process. At step t, the state $s_t = (x, y_{<t})$ consists of a prompt x and the generated prefix $y_{<t}$, and the action $v_t \in \mathcal{V}$ is the next token. The transition appends the selected token to the prefix until termination. For teacher k, let $r_k(s_t, v_t)$ denote the reward contribution associated with its specialization. This formulation includes terminal-only rewards as a special case.

Let π_{ref} be a fixed reference policy with full support over the token vocabulary at every nonterminal state, and let $\beta_k > 0$ be a teacher-specific regularization coefficient. We consider the KL-regularized objective

$$J_k(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=1}^{|y|} r_k(s_t, v_t) - \beta_k \sum_{t=1}^{|y|} \log \frac{\pi(v_t \mid s_t)}{\pi_{\text{ref}}(v_t \mid s_t)}\right], \tag{9}$$

where τ denotes a generation trajectory and $|y|$ is its response length. The regularization term is equivalently β_k times the expected sum of $D_{\text{KL}}(\pi(\cdot \mid s_t) \,\|\, \pi_{\text{ref}}(\cdot \mid s_t))$ over states visited by π.

The utility interpretation in §A.2 uses the following assumption.

Assumption 1 (Optimal teacher)**.** *For every teacher k, π_k coincides with the Bellman-optimal policy $\pi_k^\star$ for the objective in equation 9 at every state considered in the analysis, including calibration states and states visited by the student.*

Let $V_k^\star(s)$ denote the optimal KL-regularized return from state s, with $V_k^\star(s) = 0$ at terminal states. Define the optimal continuation utility as

$$Q_k^\star(s, v) = r_k(s, v) + \mathbb{E}_{s'|s,v}\big[V_k^\star(s')\big]. \tag{10}$$

Thus, $Q_k^\star(s, v)$ combines the immediate reward with the optimal regularized return after taking action v.

A.2 PROOFS OF THEOREM 1 AND COROLLARY 1

We first derive the closed-form optimal policy for the sequential objective in equation 9. Under Assumption 1, taking its log-ratio relative to the reference and centering over the vocabulary yields Theorem 1. Normalizing the resulting displacement magnitude by its calibration mean then gives Corollary 1.

Closed-form optimal policy. At every nonterminal state s, Bellman optimality gives

$$V_k^\star(s) = \max_{q\in\Delta(\mathcal{V})}\left\{\sum_{v\in\mathcal{V}} q(v) Q_k^\star(s, v) - \beta_k \,\text{KL}\big(q \,\|\, \pi_{\text{ref}}(\cdot \mid s)\big)\right\}, \tag{11}$$

where $\Delta(\mathcal{V})$ is the probability simplex over the token vocabulary. The continuation utility $Q_k^\star$ incorporates the immediate reward and all subsequent rewards and KL penalties through $V_k^\star$. Thus, the state-wise optimization in equation 11 accounts for the full remaining generation trajectory.

Lemma 1 (Reference-regularized optimal policy)**.** *The maximizer of equation 11 is unique and satisfies*

$$\pi_k^\star(v \mid s) = \pi_{\text{ref}}(v \mid s) \exp\left(\frac{Q_k^\star(s, v) - V_k^\star(s)}{\beta_k}\right), \tag{12}$$

where

$$V_k^\star(s) = \beta_k \log \sum_{v'\in\mathcal{V}} \pi_{\text{ref}}(v' \mid s) \exp\left(\frac{Q_k^\star(s, v')}{\beta_k}\right). \tag{13}$$

Proof. Fix s and abbreviate $Q(v) = Q_k^\star(s, v)$, $q_{\text{ref}}(v) = \pi_{\text{ref}}(v \mid s)$, and $\beta = \beta_k$. Define the normalized distribution $\widehat{q}(v) = q_{\text{ref}}(v) \exp(Q(v)/\beta - c_s)$, where

$$c_s = \log \sum_{v'} q_{\text{ref}}(v') \exp\left(\frac{Q(v')}{\beta}\right), \qquad \log \frac{\widehat{q}(v)}{q_{\text{ref}}(v)} = \frac{Q(v)}{\beta} - c_s. \tag{14}$$

Because q_{ref} has full support, so does $\widehat{q}$. For any $q \in \Delta(\mathcal{V})$, substitution gives

$$\sum_v q(v)Q(v) - \beta\,\mathrm{KL}(q \,\|\, q_{\mathrm{ref}}) = \beta c_s - \beta\,\mathrm{KL}(q \,\|\, \widehat{q}).$$

Since $\beta > 0$ and KL divergence is nonnegative, the objective is uniquely maximized at $q = \widehat{q}$, with optimal value βc_s. Consequently, $V_k^\star(s) = \beta c_s$, which gives equation 13. Substituting this equality into the definition of $\widehat{q}$ gives equation 12. □

Proof of Theorem 1. Fix a student-generated state s_t. Under Assumption 1, write $q_{k,t} = \pi_k(\cdot \mid s_t) = \pi_k^\star(\cdot \mid s_t)$ and $q_{\mathrm{ref},t} = \pi_{\mathrm{ref}}(\cdot \mid s_t)$. Taking logarithms in equation 12 yields

$$\log \frac{q_{k,t}(v)}{q_{\mathrm{ref},t}(v)} = \frac{Q_k^\star(s_t, v) - V_k^\star(s_t)}{\beta_k}. \tag{15}$$

Define the uniform vocabulary mean $\bar{Q}_{k,t}^\star = |\mathcal{V}|^{-1} \sum_{v'} Q_k^\star(s_t, v')$. Averaging equation 15 over the vocabulary gives $(\bar{Q}_{k,t}^\star - V_k^\star(s_t))/\beta_k$. Subtracting this mean, as prescribed by equation 2, cancels the state-value term and gives

$$d_{k,t}(v) = \frac{Q_k^\star(s_t, v) - \bar{Q}_{k,t}^\star}{\beta_k}, \qquad \rho_{k,t} = \frac{\left\| Q_k^\star(s_t, \cdot) - \bar{Q}_{k,t}^\star \mathbf{1} \right\|_2}{\beta_k}. \tag{16}$$

This establishes equation 7 and proves Theorem 1. The same derivation applies to a selected candidate set $\mathcal{A}_t$ by taking the centering mean and displacement norm over $\mathcal{A}_t$. Renormalizing the teacher and reference distributions on this set contributes only token-independent offsets, which centering removes. For this version, the calibration argument below uses the corresponding candidate-restricted utility contrasts and the same candidate-selection rule used to compute displacement.

Centering removes the common utility offset while preserving scaled differences between actions. §A.3 establishes the corresponding distribution-level invariance without requiring teacher optimality.

Proof of Corollary 1. Let

$$\sigma_{k,t} := \left\| Q_k^\star(s_t, \cdot) - \bar{Q}_{k,t}^\star \mathbf{1} \right\|_2, \qquad \bar{\sigma}_k := \frac{\mathbb{E}_{\mathrm{cal}}\left[\sum_{t=1}^{|y|} \sigma_{k,t}\right]}{\mathbb{E}_{\mathrm{cal}}[|y|]}. \tag{17}$$

Here, $\bar{\sigma}_k$ is the token-weighted mean utility-contrast magnitude under the calibration distribution. Assume $\bar{\sigma}_k > 0$ for every teacher. Because β_k is constant across states for teacher k, substituting equation 16 into equation 3 gives

$$\mu_k = \frac{\bar{\sigma}_k}{\beta_k}, \qquad r_{k,t} = \frac{\rho_{k,t}}{\mu_k} = \frac{\sigma_{k,t}}{\bar{\sigma}_k}. \tag{18}$$

At any state where $\sum_j \sigma_{j,t} > 0$, at least one calibrated score is positive, so the normalization denominator in equation 4 is nonzero. Since $z \mapsto z^\gamma$ is strictly increasing on $[0, \infty)$ for $\gamma > 0$, exponentiation and normalization preserve the ranking of the calibrated scores:

$$\arg\max_k w_{k,t}^{\mathrm{Trust}} = \arg\max_k r_{k,t} = \arg\max_k \frac{\sigma_{k,t}}{\bar{\sigma}_k}. \tag{19}$$

Equations equation 18–equation 19 establish equation 8 and prove Corollary 1.

Calibration therefore cancels the explicit $1/\beta_k$ factor and expresses each teacher's current utility-contrast magnitude relative to its own calibration mean. TrustMOPD assigns the greatest supervision weight to the teacher with the largest relative utility contrast at the current student-generated state.

A.3 Centering and the frozen calibration divisor

This subsection explains two design choices in TrustMOPD: centering the teacher-reference log-ratio and normalizing its magnitude by a frozen calibration divisor. The resulting properties follow from the construction and do not require Assumption 1.

Centering on a candidate set. Fix a state and a candidate set $\mathcal{A}$ with $|\mathcal{A}| = n \geq 2$. Let q_k and q_{ref} be the teacher and reference distributions, both positive on $\mathcal{A}$. Define the restricted log-ratio vector and its centered version as

$$g_k(v) = \log \frac{q_k(v)}{q_{\text{ref}}(v)}, \qquad \bar{g}_k = \frac{1}{n} \sum_{v \in \mathcal{A}} g_k(v), \qquad d_k = g_k - \bar{g}_k \mathbf{1}.$$

For a vector g on $\mathcal{A}$, the corresponding reference-weighted distribution is

$$p_g(v) = \frac{q_{\text{ref}}(v) \exp(g(v))}{\sum_{u \in \mathcal{A}} q_{\text{ref}}(u) \exp(g(u))}.$$

In particular, p_{g_k} equals the teacher distribution renormalized on $\mathcal{A}$.

Proposition 1 (Canonical centered representative)**.** *Adding a constant to every coordinate of g_k leaves p_{g_k} unchanged. Among all vectors $g_k + \lambda \mathbf{1}$ with $\lambda \in \mathbb{R}$, d_k is the unique minimum-norm representative and the orthogonal projection of g_k onto $\mathbf{1}^\perp$. Moreover,*

$$\|g_k\|_2^2 = \|d_k\|_2^2 + n\bar{g}_k^2. \tag{20}$$

Proof. A common shift by λ multiplies the numerator and denominator of p_{g_k} by the same factor $\exp(\lambda)$, leaving the distribution unchanged. The squared norm of the shifted vector is

$$\|g_k + \lambda \mathbf{1}\|_2^2 = \|g_k\|_2^2 + 2\lambda \langle g_k, \mathbf{1} \rangle + n\lambda^2. \tag{21}$$

This strictly convex quadratic is uniquely minimized at $\lambda = -\bar{g}_k$, yielding d_k. Since $\langle d_k, \mathbf{1} \rangle = 0$, the decomposition $g_k = d_k + \bar{g}_k \mathbf{1}$ is orthogonal, establishing the projection property and equation 20. □

Centering also preserves every pairwise change in log odds:

$$d_k(v) - d_k(v') = \log \frac{q_k(v)}{q_k(v')} - \log \frac{q_{\text{ref}}(v)}{q_{\text{ref}}(v')}, \qquad v, v' \in \mathcal{A}. \tag{22}$$

Thus, the centered vector retains how the teacher changes the relative preference between candidate tokens, while its norm excludes the common-offset component in equation 20. Renormalizing either distribution on $\mathcal{A}$ changes its log-probabilities only by a constant and therefore leaves d_k unchanged. These identities apply exactly on the selected candidate set, over which the displacement norm is computed.

A fixed baseline across training batches. TrustMOPD uses a frozen calibration divisor $\mu_k > 0$ while recomputing the displacement magnitude $\rho_{k,t}$ on the current student's states and candidate sets.

To see the distinction from batch-local normalization, fix teacher k and a batch B containing $m \geq 1$ token positions. Let $\hat{\mu}_k(B) = m^{-1} \sum_{t \in B} \rho_{k,t}$ be the current batch mean. When $\hat{\mu}_k(B) > 0$, normalizing by this mean gives

$$\frac{1}{m} \sum_{t \in B} \frac{\rho_{k,t}}{\hat{\mu}_k(B)} = 1. \tag{23}$$

Batch-local normalization therefore fixes each teacher's mean normalized score at 1, regardless of changes in its mean displacement across batches. With the frozen divisor, the corresponding mean is instead

$$\frac{1}{m} \sum_{t \in B} r_{k,t} = \frac{1}{m} \sum_{t \in B} \frac{\rho_{k,t}}{\mu_k} = \frac{\hat{\mu}_k(B)}{\mu_k}. \tag{24}$$

A value above or below 1 indicates that the current batch's mean displacement is respectively above or below the teacher's initial calibration mean. The frozen divisor thus preserves changes in mean displacement relative to a fixed baseline, while the numerator adapts to the states visited by the evolving student.

A.4 EXACT AGGREGATION INTO A MIXED LOG-TARGET

The weighted distillation objective in equation 5 can be evaluated using a single mixed log-target. This follows from the affine dependence of reverse KL on teacher log-probabilities and does not require Assumption 1. We work at a fixed student-generated state and suppress the token index. All distributions below are normalized on the same support $\mathcal{A}$, which is the selected candidate set when candidate-based distillation is used. Teacher probabilities are positive on this support, and $\ell_k = \log q_k$ denotes teacher k's log-probability vector.

Aggregation of affine losses.

Proposition 2 (Exact aggregation for affine token losses)**.** *Suppose the token loss has the form*

$$\ell_{\mathrm{distill}}(q_k, p_\theta) = \langle a(p_\theta), \ell_k \rangle + b(p_\theta), \tag{25}$$

where $a(p_\theta)$ and $b(p_\theta)$ do not depend on teacher k. For weights $w_k \geq 0$ satisfying $\sum_{k=1}^K w_k = 1$, define $\ell_{\mathrm{mix}} = \sum_{k=1}^K w_k \ell_k$. Then

$$\sum_{k=1}^K w_k \ell_{\mathrm{distill}}(q_k, p_\theta) = \langle a(p_\theta), \ell_{\mathrm{mix}} \rangle + b(p_\theta). \tag{26}$$

Proof. Expanding the weighted sum gives

$$\sum_{k=1}^K w_k \big(\langle a(p_\theta), \ell_k \rangle + b(p_\theta) \big) = \left\langle a(p_\theta), \sum_{k=1}^K w_k \ell_k \right\rangle + b(p_\theta) \sum_{k=1}^K w_k.$$

Substituting the definition of ℓ_{mix} and $\sum_{k=1}^K w_k = 1$ yields equation 26. □

Reverse KL and the geometric-mean target. Reverse KL satisfies the proposition because

$$D_{\mathrm{KL}}(p_\theta \| q_k) = \langle p_\theta, \log p_\theta \rangle - \langle p_\theta, \ell_k \rangle. \tag{27}$$

Consequently, its weighted sum is exactly

$$\sum_{k=1}^K w_k D_{\mathrm{KL}}(p_\theta \| q_k) = \langle p_\theta, \log p_\theta \rangle - \langle p_\theta, \ell_{\mathrm{mix}} \rangle.$$

The mixed vector ℓ_{mix} is the logarithm of the unnormalized weighted geometric mean $\prod_{k=1}^K q_k(v)^{w_k}$. Its normalized distribution is

$$\widetilde{q}_{\mathrm{mix}}(v) = \frac{\exp(\ell_{\mathrm{mix}}(v))}{Z_{\mathrm{mix}}}, \qquad Z_{\mathrm{mix}} = \sum_{v \in \mathcal{A}} \exp(\ell_{\mathrm{mix}}(v)). \tag{28}$$

Since $\log \widetilde{q}_{\mathrm{mix}}(v) = \ell_{\mathrm{mix}}(v) - \log Z_{\mathrm{mix}}$, we obtain

$$\sum_{k=1}^K w_k D_{\mathrm{KL}}(p_\theta \| q_k) = D_{\mathrm{KL}}(p_\theta \| \widetilde{q}_{\mathrm{mix}}) - \log Z_{\mathrm{mix}}. \tag{29}$$

Implementation consequence. Within each distillation update, the sampled states, selected support, teacher scores, and allocation weights are held fixed for differentiation. Using ℓ_{mix} directly in the affine expression preserves the scalar loss exactly. Our implementation normalizes the mixed log-target and evaluates a single reverse-KL loss $D_{\mathrm{KL}}(p_\theta \| \widetilde{q}_{\mathrm{mix}})$. By equation 29, this loss has the same student gradient as the original weighted objective because Z_{mix} is constant with respect to the student parameters during the update.

A.5 SCOPE OF THE THEORETICAL INTERPRETATION

Utility interpretation. Under Assumption 1, §A.2 establishes an exact connection between centered teacher-reference displacement and utility contrast. The derivation considers a fixed full-support reference and a KL-regularized sequential objective with a positive teacher-specific coefficient β_k that is constant across states. This setting provides a theoretical interpretation of the displacement signal and its calibration.

Table 2: **Model checkpoints used in our experiments.** Each block contains a shared pre-RL checkpoint and three distinct RL-finetuned teachers used for mathematics, code, and instruction following, respectively. The shaded row serves as both the student initialization π_{θ_0} and the reference π_{ref} for computing teacher displacement. The default setting uses the Open-MOPD SmolLM3-3B checkpoints. The backbone transfer setting uses three publicly released teachers sharing DeepSeek-R1-Distill-Qwen-7B as their pre-RL initialization. All identifiers refer to Hugging Face repositories.

Role	HuggingFace identifier	Reference
Default setting: SmolLM3-3B (Table 1, and blocks (a,b,d) of Figure 5)		
π_{θ_0} and π_{ref}	`BytedTsinghua-SIA/Open-MOPD-SmolLM3-3B-MixSFT`	Bakouch et al. (2025)
Teacher, math	`BytedTsinghua-SIA/Open-MOPD-SmolLM3-3B-RL-Math`	Gao et al. (2026)
Teacher, code	`BytedTsinghua-SIA/Open-MOPD-SmolLM3-3B-RL-Code`	Gao et al. (2026)
Teacher, IF	`BytedTsinghua-SIA/Open-MOPD-SmolLM3-3B-RL-IF`	Gao et al. (2026)
Backbone transfer: DeepSeek-R1-Distill-Qwen-7B (block (c) of Figure 5; teachers not trained by us)		
π_{θ_0} and π_{ref}	`deepseek-ai/DeepSeek-R1-Distill-Qwen-7B`	Guo et al. (2025)
Teacher, math	`nvidia/AceMath-RL-Nemotron-7B`	Chen et al. (2025a)
Teacher, code	`nvidia/AceReason-Nemotron-7B`	Chen et al. (2025b)
Teacher, IF	`THU-KEG/R1-Distill-Qwen-7B-VerIF`	Peng et al. (2025)

Licenses differ across rows and none of them is unrestricted: the Open-MOPD models and the VerIF teacher are Apache-2.0, DeepSeek-R1-Distill-Qwen-7B is MIT, and the two NVIDIA teachers are released under the NVIDIA Open Model License.

Applicability beyond KL-regularized training. TrustMOPD operates directly on the observed teacher and reference distributions and does not require an explicit KL penalty during teacher training. Even for teachers trained without KL regularization, centered displacement preserves the teacher-reference changes in pairwise log odds in equation 22. The centering property in Proposition 1, the frozen-divisor analysis in §A.3, and the aggregation identity in Proposition 2 likewise hold under their stated support and normalization conditions. These distribution-level properties provide the basis for applying TrustMOPD across teacher-training configurations, while Assumption 1 supplies the additional utility interpretation.

B Experimental Details

This appendix documents the experimental setup. §B.1 lists the model checkpoints, and §B.2 describes training-data construction and decontamination, including the settings used for backbone and data-source transfer, respectively. §B.3 defines the baselines and reference methods. §B.4 describes the training and calibration protocol, §B.5 lists the hyperparameters, and §B.6 provides the calibration and training algorithms.

B.1 Models

Table 2 lists the model repositories and checkpoints used in our experiments. Each model group consists of a shared pre-RL reference and three publicly released teachers obtained by RL fine-tuning from that reference. This relationship allows equation 2 to measure each teacher's distributional change from its pre-RL initialization.

Default model setup. Our default experiments use the SmolLM3-3B checkpoints released by Open-MOPD. The mixed-domain SFT checkpoint serves as both the student initialization π_{θ_0} and the shared reference π_{ref}. The former gives all compared distillation methods the same starting weights; the latter provides the common baseline for measuring teacher displacement. Using the same checkpoint for both roles is an experimental choice, as the allocation rule does not require the student to be initialized from the reference.

The $K = 3$ teachers specialize in mathematics, code, and instruction following, respectively, and were trained using reinforcement learning with verifiable rewards. Their released checkpoints correspond to different training durations: a few hundred steps for the mathematics and code teachers and several thousand steps for the instruction-following teacher. The instruction-following teacher also

uses a shorter response-length limit. These differences in training configuration motivate measuring each teacher's displacement relative to its own calibration baseline through equation 3. §C.2 reports the resulting calibration constants.

Backbone transfer setup. For the backbone transfer experiment in Figure 5(c), we use DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025) as both the student initialization and the shared reference. We use three distinct publicly released teacher checkpoints: AceMath-RL-Nemotron-7B as the mathematics teacher (Chen et al., 2025a), AceReason-Nemotron-7B as the code teacher (Chen et al., 2025b), and R1-Distill-Qwen-7B-VerIF as the instruction-following teacher (Peng et al., 2025). All three were obtained by RL fine-tuning from DeepSeek-R1-Distill-Qwen-7B, allowing teacher displacement to be computed relative to their shared pre-RL initialization.

We compute a separate calibration constant for each teacher using the same procedure as in the default setup. This experiment replaces the student, reference, and all three teacher checkpoints while retaining the TrustMOPD allocation procedure. Table 2 provides the exact repository identifiers.

B.2 Training data

The two default training sets, SingleCap and MultiCap, each contain 7458 prompts drawn from the same pool of source data. For each set, we follow the corresponding Open-MOPD data composition, preserving its per-category sample counts and proportions (Gao et al., 2026).

Single-capability training set. SingleCap contains 1540 mathematics, 1870 code, and 4048 instruction-following prompts. Each prompt is assigned to one of these three domains, providing the domain label $d(x)$ used by the MOPD routing rule in equation 1.

Mixed-capability training set. MultiCap contains six prompt categories: 256 mathematics, 256 code, 1579 instruction-following, 2131 mathematics–instruction-following, 2980 code–instruction-following, and 256 mathematics–code prompts. The three mixed-capability categories account for 5367 prompts (72.0%), while the remaining 2091 prompts (28.0%) belong to a single domain.

For a mixed-capability prompt, the composition identifies two relevant specialist domains but does not by itself specify which single teacher should supervise the entire response. Applying a single-teacher routing rule to these prompts would therefore require an additional assignment rule.

Comparison protocol. Matching the total prompt counts keeps the training-set size fixed across the two settings. Within each setting, the compared distillation methods use the same training prompts. The two settings address complementary questions: SingleCap measures the effectiveness of allocation without domain labels against a label-based routing reference, while MultiCap evaluates allocation when individual prompts combine capabilities.

B.2.1 Corpus for data-source transfer

For the data-source transfer experiment in Figure 5(d), we replace the training prompts while retaining the default student initialization, reference, and teacher checkpoints. Table 3 lists the replacement data sources. The source datasets are released under MIT (DeepMath-103K), CC-BY-NC-4.0 (KodCode-V1), and ODC-BY (Tülu 3 persona instruction following). All KodCode-based categories use problems from the `Algorithm` and `Data_Structure` subsets of KodCode-V1. We construct separate SingleCap and MultiCap training sets, each containing 7458 prompts and matching the per-category counts of its default counterpart. The overlap-screening procedure is described in §B.2.2.

Mathematics and code with instruction-following constraints. For the mathematics–instruction-following and code–instruction-following categories, we pair each replacement problem with an instruction-following constraint block from a corresponding default MultiCap prompt. We replace the original problem statement and copy the constraint block verbatim, preserving its natural-language description and associated checker identifiers and arguments. This construction preserves the number of occurrences of each checker identifier exactly. It therefore changes the source problems while retaining the instruction-following constraints attached to them.

Table 3: **Sources and composition of the data-source transfer corpus.** The two training sets are used in Figure 5(d). Each column matches the category counts and proportions of its corresponding default training set, following the Open-MOPD composition. Mixed-capability prompts combine a source problem with an instruction-following (IF) constraint block or a mathematical derivation block. The mathematics–code category uses problems from KodCode-V1. Construction details appear in §B.2.1.

Prompt category	Source and construction	SINGLECAP	MULTICAP
Mathematics	`zwhe99/DeepMath-103K` (He et al., 2026b)	1540	256
Code	`KodCode/KodCode-V1` (Xu et al., 2025)	1870	256
Instruction following	`allenai/tulu-3-sft-personas-instruction-following` (Lambert et al., 2025)	4048	1579
Mathematics–IF	DeepMath-103K + IF constraint block	—	2131
Code–IF	KodCode-V1 + IF constraint block	—	2980
Mathematics–code	KodCode-V1 + mathematical derivation block	—	256
Total		7458	7458

Table 4: **Decontamination statistics for the data-source transfer corpus.** The four removal columns report within-pool duplicates and matches against the comparison corpora using the rules in §B.2.2. All three candidate pools are screened against the same 113,742 comparison texts, covering evaluation datasets and the default teachers' RL training prompts. *Exact* denotes normalized exact matching; *5-gram* denotes word-level 5-gram Jaccard similarity of at least 0.8. *Kept* reports the candidate pool remaining before quota-based sampling of the training sets in Table 3.

		Within-pool		Cross-corpus		
Pool	Candidates	Exact	5-gram	Exact	5-gram	Kept
DeepMath-103K	103,022	2,945	1,010	2,441	194	96,432
KodCode-V1	64,249	25,718	14	0	0	38,517
Tülu 3 personas IF	29,980	18	0	0	0	29,962

Mathematics–code composition. We construct mathematics–code prompts from the replacement code pool by appending the same fixed instruction block requesting a mathematical derivation. Each prompt therefore combines an explicit programming task with a derivation requirement, while the underlying code problem varies across examples.

Matching criteria. In addition to per-category sample counts, we match the proportions of standalone instruction-following prompts containing one, two, or three constraints and the prompt-length buckets of the corresponding default sets. Difficulty, mathematical topic, and code-task format are determined by the replacement sources and are not explicitly matched.

Use in distillation. The reconstructed sets supply prompts for student rollouts, with supervision provided by the fixed teachers through equation 5. The Tülu rule checkers and KodCode unit tests are not used to compute training rewards.

B.2.2 DECONTAMINATION

We screen all upstream candidates for overlap with the comparison corpora and with previously accepted candidates before sampling the final training sets. Table 4 reports the numbers removed by each rule and the sizes of the retained candidate pools.

Matching rules. A candidate is excluded if either of two criteria identifies a match. The first is exact equality after Unicode NFKC normalization, case folding, and removal of non-alphanumeric characters. The second is a word-level 5-gram Jaccard similarity of at least 0.8, computed exactly. For code candidates, we additionally apply both criteria to the upstream parent instruction. This step

Example prompts from the data-source transfer corpus

(a) SINGLECAP: instruction following

Source: `allenai/tulu-3-sft-personas-instruction-following`

Write a technical report on a new statistical method you have researched. Divide the report into three sections. In the first section, provide a detailed explanation of the theoretical foundation of the method. [. . .] End the report with the exact sentence: "This method bridges the gap between theory and practice."

▷ The source prompt is used unchanged, with no appended instructions. Its constraint annotations (`specific ending`, `format:number of sections`) are retained as metadata.

(b) MULTICAP: mathematics–instruction following

Source: `zwhe99/DeepMath-103K`

On the unit sphere defined by $x^2 + y^2 + z^2 = 1$, consider n points $A_1, A_2, \ldots, A_n$ where $n \geq 2$. Determine the maximum possible value of the sum $\sum_{1 \leq i < j \leq n} |A_i A_j|^2$.

There should be 2 paragraphs. Paragraphs are separated with the markdown divider: ***

▷ The source mathematics problem is followed by an instruction-following constraint copied verbatim from a matched default MULTICAP prompt. The associated checker identifier `paragraphs:paragraphs` and arguments {} are retained as metadata. The original problem from the matched prompt is replaced.

(c) MULTICAP: mathematics–code

Source: `KodCode/KodCode-V1`

Binary Search Tree (BST) Deletion — Implement the delete operation in a Binary Search Tree and ensure the tree retains its properties after deletion. Write a function `delete_node` that deletes a node with a given value from a BST. [. . .]

Alongside your implementation, derive the answer mathematically: state the closed-form expression or recurrence your algorithm is computing, justify it, and give its complexity. Put the final expression in `\boxed{}`.

▷ The source code problem is followed by a fixed mathematical derivation request. The appended block is identical across all 256 prompts in this category.

Figure 6: **Examples of the data construction described in §B.2.1.** The displayed text preserves the prompt wording, with omissions in problem statements marked by [. . .]. Colored separators and explanatory annotations are added for presentation and are not part of the prompts. Code–instruction-following prompts follow the construction in (b), using a code problem in place of the mathematics problem. Each example is the first prompt in its category in stored order. Category sizes are reported in Table 3.

accounts for KodCode's construction of multiple examples from a shared seed problem and screens for overlap at the parent-instruction level as well as the rendered-prompt level.

Comparison corpora. The comparison pool contains 113,742 texts. It covers the six benchmarks in §4.1, nine additional evaluation datasets spanning mathematics, code, and instruction following, and the 104,848-prompt RL training pool used by the default teachers.

Screening and sampling order. A candidate is accepted only if it passes screening against both the comparison pool and the previously accepted candidates. We then sample from the retained pools using the category and prompt-length quotas in §B.2.1. The *Kept* column in Table 4 reports these pools before sampling, rather than the final training sets.

B.2.3 EXAMPLE PROMPTS

Figure 6 illustrates three construction patterns in the data-source transfer corpus: an unchanged source prompt, a source problem paired with an instruction-following constraint, and a code problem paired with a mathematical derivation request.

B.3 BASELINES AND REFERENCES

The trained allocation baselines follow the common protocol in §B.4. Within each experimental setting, they use the same student initialization and training prompts, with teacher supervision determined by the allocation rules below.

Fixed teacher weighting. UNIFORM assigns equal weight to all teachers, $w_{k,t} = 1/K$, at every token, providing a baseline for evaluating adaptive allocation. The three single-teacher baselines each select one fixed specialist k_0, with $w_{k,t} = \mathbb{I}\{k = k_0\}$ throughout training. They perform on-policy distillation from the mathematics, code, or instruction-following teacher, respectively, measuring the performance obtained from each specialist individually.

Random token-level weighting. RANDOM independently samples a weight vector at each token: $w_{\cdot,t} \sim \mathrm{Dirichlet}(1, \ldots, 1)$. This distribution is uniform over nonnegative teacher-weight vectors that sum to one. The weights vary across tokens but are independent of the student-generated state s_t. This baseline tests whether non-uniform token-level weighting alone can account for the gains from adaptive allocation.

Label-based routing. MOPD uses the supplied domain label $d(x)$ to select one teacher for the entire response, as in equation 1: $w_{k,t} = \mathbb{I}\{k = d(x)\}$ for every token t. It serves as a label-based routing reference on SINGLECAP, where every prompt has one assigned domain. We omit it on MULTICAP, where 72.0% of prompts combine two capabilities and lack a unique domain label for single-teacher routing.

References without distillation. We report three references that do not use the distillation training prompts and are therefore shared across SINGLECAP and MULTICAP for a fixed model setup. The initial student establishes the performance before distillation. The parameter-averaged teacher uses the arithmetic mean of the three specialists' parameters, $\theta_{\mathrm{avg}} = K^{-1} \sum_{k=1}^{K} \theta_k$, providing a model-merging reference without additional training. ROUTED TEACHERS uses the evaluation-domain label to select the corresponding specialist to generate each response. It therefore retains three teacher models and requires domain labels at inference time.

B.4 TRAINING AND CALIBRATION PROTOCOL

Shared training settings. All distillation methods use the training settings of the underlying MOPD implementation, without method-specific hyperparameter tuning. Each run consists of 116 training steps with a batch size of 64. Prompts are truncated to at most 4096 tokens, and the maximum response length is 8192 tokens.

Candidate-token scoring. At each student-generated state, we select the student's top $C = 128$ candidate tokens. Teacher and reference probabilities are renormalized over this shared candidate set before computing the centered displacements in §3.1. The student distribution is likewise renormalized over the candidate set when forming the distillation objective. The teacher log-probabilities are combined using the allocation weights, and the resulting mixed log-target is normalized over the candidate set to form the reverse-KL distillation target. During each update, sampled states and candidate sets are held fixed, and teacher scores and allocation weights are detached.

Calibration. Before distillation training, we estimate the teacher-specific constants μ_k in equation 3 using eight batches of rollouts generated by the initial student. Calibration uses the same candidate-selection rule and response-length limit as training. The resulting constants are frozen throughout the training run, while displacement scores are recomputed on the evolving student's states.

We calibrate separately for each model setup and training prompt set. This accounts for the dependence of the calibration scale on the teacher, prompt distribution, and response-length window; §C.2 reports the corresponding constants. Calibration uses no domain labels. For variants using calibrated scores, we follow the same protocol and re-estimate the constants for each scoring rule.

Table 5: **Training, calibration, and evaluation settings.** *Shared* denotes settings common to the compared distillation methods. *Calibrated* denotes settings used by TrustMOPD and the proxy ablations that consume calibrated scores in Table 9. The default allocation exponent is $\gamma = 7$; its sensitivity analysis uses $\{3, 5, 7, 9\}$. Evaluation sampling settings are specified separately from training rollouts.

Hyperparameter	Value	Applies to
Optimization		
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$)	Shared
Peak learning rate	1×10^{-6}	Shared
LR schedule	Constant after linear warmup	Shared
Warmup ratio	0.03	Shared
Weight decay	0.0	Shared
Gradient clipping	1.0	Shared
Training steps	116	Shared
Batch size (prompts per step)	64	Shared
Micro-batch size per device	4	Shared
Precision	bfloat16	Shared
Rollout generation		
Rollout temperature	0.6	Shared
Rollout top-p	1.0	Shared
Rollouts per prompt	1	Shared
Max prompt length	4096	Shared
Max response length	8192	Shared
Distillation objective		
Divergence	Reverse KL, $D_{\mathrm{KL}}(p_{\theta,t} \| q_{k,t})$	Shared
Candidate-set size C	128 (student top-C)	Shared
Student renormalization	On the candidate set	Shared
Teacher-pool size K	3 (math, code, IF)	Shared
Loss aggregation	Mean of per-response token means	Shared
Allocation		
Sharpness exponent γ	7; sweep: $\{3, 5, 7, 9\}$	TrustMOPD
Offline calibration of μ_k (Algorithm 1)		
Calibration batches	8, generated by π_{θ_0}	Calibrated
Domain labels used	None	Calibrated
Calibration configuration	Matched model pool, prompt set, response-length limit, and C	Calibrated
Update schedule	Estimated before training, then frozen	Calibrated
Evaluation and reporting		
AIME 25/26	16 samples, temperature 0.6	Shared
LiveCodeBench v5/v6	5 samples, temperature 0.6	Shared
IFEval / IFBench	1 sample, temperature 0.6	Shared
Checkpoint reported	Final; no evaluation-based selection	Shared
Training seeds	5	Shared

B.5 HYPERPARAMETERS

Table 5 summarizes the training, allocation, calibration, and evaluation settings. The shared training configuration follows the underlying MOPD implementation and is kept fixed across the compared distillation methods, without method-specific tuning. The allocation exponent is set to $\gamma = 7$ in the main experiments and varied over $\{3, 5, 7, 9\}$ in the sensitivity analysis reported in Appendix C.5.

Calibration settings apply to TrustMOPD and the proxy ablations that use calibrated scores. For each model setup and prompt set, we estimate μ_k using the response-length limit and candidate-set size of the corresponding training configuration. Changes to either setting require recalibration using Algorithm 1.

Algorithm 1 Offline calibration of teacher scales

Require: Initial student π_{θ_0}; teachers $\{\pi_k\}_{k=1}^K$; reference π_{ref}; unlabeled prompts $\mathcal{D}_{\mathrm{cal}}$; candidate-set size C; number of calibration batches $B_{\mathrm{cal}} = 8$
Ensure: Frozen calibration constants $\{\mu_k\}_{k=1}^K$

$S_k \leftarrow 0$ for all k; $N \leftarrow 0$
for $b = 1, \ldots, B_{\mathrm{cal}}$ **do**
  Sample a prompt batch $\mathcal{B}_b$ from $\mathcal{D}_{\mathrm{cal}}$
  for each prompt $x_i \in \mathcal{B}_b$ **do**
    Generate $y_i \sim \pi_{\theta_0}(\cdot \mid x_i)$
    for $t = 1, \ldots, |y_i|$ **do**
      $s_t \leftarrow (x_i, y_{i,<t})$; $\mathcal{A}_t \leftarrow \mathrm{TopC}(\pi_{\theta_0}(\cdot \mid s_t))$
      Compute teacher distributions $\{q_{k,t}\}_{k=1}^K$ and reference $q_{\mathrm{ref},t}$, normalized on $\mathcal{A}_t$
      $S_k \leftarrow S_k + \|\mathrm{center}(\log q_{k,t} - \log q_{\mathrm{ref},t})\|_2$ for all k
      $N \leftarrow N + 1$
    end for
  end for
end for
$\mu_k \leftarrow S_k / N$ for all k ▷ Mean over all response tokens
return $\{\mu_k\}_{k=1}^K$

Algorithm 2 TrustMOPD training

Require: Student π_θ; fixed teachers $\{\pi_k\}_{k=1}^K$; fixed reference π_{ref}; unlabeled prompts $\mathcal{D}$; frozen scales $\{\mu_k > 0\}_{k=1}^K$ from Algorithm 1; exponent $\gamma > 0$; candidate-set size C

for each training step **do**
  Sample a prompt batch $\mathcal{B}$ from $\mathcal{D}$
  for each prompt $x_i \in \mathcal{B}$ **do**
    Generate $y_i \sim \pi_\theta(\cdot \mid x_i)$
    for $t = 1, \ldots, |y_i|$ **do**
      $s_t \leftarrow (x_i, y_{i,<t})$; $\mathcal{A}_t \leftarrow \mathrm{TopC}(\pi_\theta(\cdot \mid s_t))$
      Compute student distribution $p_{\theta,t}$, normalized on $\mathcal{A}_t$
      Compute teacher distributions $\{q_{k,t}\}_{k=1}^K$ and reference $q_{\mathrm{ref},t}$, normalized on $\mathcal{A}_t$
      $\rho_{k,t} \leftarrow \|\mathrm{center}(\log q_{k,t} - \log q_{\mathrm{ref},t})\|_2$ for all k
      $w_{k,t} \leftarrow \mathrm{stopgrad}\left(\frac{(\rho_{k,t}/\mu_k)^\gamma}{\sum_{j=1}^K (\rho_{j,t}/\mu_j)^\gamma}\right)$ for all k
      $\widetilde{q}_{\mathrm{mix},t} \leftarrow \mathrm{softmax}_{\mathcal{A}_t}\left(\sum_{k=1}^K w_{k,t} \log q_{k,t}\right)$
      $\ell_{i,t}(\theta) \leftarrow D_{\mathrm{KL}}(p_{\theta,t} \| \widetilde{q}_{\mathrm{mix},t})$
    end for
    $\mathcal{L}_i(\theta) \leftarrow |y_i|^{-1} \sum_{t=1}^{|y_i|} \ell_{i,t}(\theta)$
  end for
  $\mathcal{L}(\theta) \leftarrow |\mathcal{B}|^{-1} \sum_{x_i \in \mathcal{B}} \mathcal{L}_i(\theta)$
  Update θ using $\nabla_\theta \mathcal{L}(\theta)$
end for

B.6 Algorithms

Algorithm 1 estimates frozen teacher scales by averaging displacement magnitudes over all response tokens generated by the initial student. Algorithm 2 trains the evolving student using these scales, averaging the distillation loss first within each response and then across the batch. Both procedures use unlabeled prompts. During differentiation, sampled trajectories and candidate sets are held fixed, and teacher scores and allocation weights are detached. Token loops describe computations at each state; model scoring can be batched across states.

C Additional Experimental Results

This appendix provides detailed numerical results supporting the analyses in §4.3 and §4.4. The first two subsections report calibrated-score diagnostics and calibration constants, both obtained without distillation training. §C.3–C.5 present detailed results for supervision allocation, ablations, sensitivity analyses, and transfer experiments, including per-domain scores and standard deviations.

Table 6: **Calibrated teacher scores before distillation.** Scores are computed from 256 initial-student rollouts per domain on held-out prompts. Entries report means over response states, with standard deviations across prompts. Rows identify teachers, and columns identify prompt domains. Within each column, the highest mean score is tinted.

	Prompt domain		
Teacher	Mathematics	Code	Instruction following
RL-math	1.120 ±0.210	0.987 ±0.076	0.541 ±0.223
RL-code	0.846 ±0.221	1.343 ±0.424	0.472 ±0.236
RL-IF	0.719 ±0.067	0.714 ±0.072	6.868 ±3.812

Table 7: **Calibration constants in the default experiments.** Values are the μ_k of equation 3 for the SmolLM3-3B teacher pool on the two training prompt sets. Both settings use the same initial student and a maximum response length of 8192 tokens.

Training prompt set	μ_{math}	μ_{code}	μ_{IF}
SINGLECAP	3.640	7.482	1.722
MULTICAP	3.809	8.223	1.269

C.1 THE CALIBRATED SCORE BEFORE ANY DISTILLATION

Table 6 evaluates the calibrated scores $r_{k,t}$ of equation 3 on held-out prompts using rollouts from the initial student. This complements **Obs. 3** in §4.3, which examines allocation weights from a trained TrustMOPD run.

Within each prompt-domain column, the matching specialist has the highest mean calibrated score. The separation is particularly pronounced for instruction-following prompts, where the instruction-following teacher scores 6.868, compared with 0.541 and 0.472 for the mathematics and code teachers. These results show that the allocation signal already exhibits domain-aligned structure before distillation, with scores measured relative to each teacher's calibration mean.

C.2 CALIBRATION CONSTANTS

Table 7 reports the calibration constants μ_k used in the default experiments. The code teacher's mean displacement is approximately 4.3 times that of the instruction-following teacher on SINGLECAP and 6.5 times on MULTICAP. Dividing by μ_k expresses each teacher's displacement relative to its own calibration baseline.

The constants also differ between the two training prompt sets under the same model setup. We therefore estimate them separately for each model setup and training prompt set, following §B.4, including in the backbone and data-source transfer experiments.

C.3 ALLOCATION ANALYSIS

Table 8 reports the rollout-level allocation statistics underlying Figure 3(b–d). On SINGLECAP, the domain-matched teacher receives the largest rollout-averaged weight on 98%, 91%, and 100% of mathematics, code, and instruction-following rollouts, respectively. The alignment therefore holds across most individual rollouts as well as in the group means. On MULTICAP, the two teachers corresponding to each capability composition receive a combined mean weight of 0.89, 0.75, and 0.73 for mathematics–instruction following, code–instruction following, and mathematics–code, respectively.

Figure 7 complements these aggregate statistics with token-level examples from all three compositions. The examples show allocation varying across positions within a response. The response-level ablation in Table 9 evaluates the contribution of this positional variation.

Table 8: **Rollout-level allocation statistics underlying Figure 3(b–d).** Teacher columns report group means of $\bar{w}_{i,k} = |y_i|^{-1} \sum_t w_{i,k,t}$; each row sums to one across teachers, up to rounding. Tinted entries identify domain-matched teachers for SINGLECAP and the two composition-matched teachers for MULTICAP; *Relevant mass* is their combined mean weight. For SINGLECAP, we also report the mean and positive fraction of the margin $m_i = \bar{w}_{i,d_i} - \max_{k \neq d_i} \bar{w}_{i,k}$, where d_i is the prompt domain, as in Figure 3(c). The shaded row gives uniform allocation; its relevant mass is shown for SINGLECAP / MULTICAP, and its margin statistics apply to SINGLECAP. Group labels are used only for this analysis, not for allocation.

Data	Prompt group	Mean weight $\bar{w}_{i,k}$			Relevant mass	Mean margin m_i	$\Pr[m_i > 0]$
		RL-math	RL-code	RL-IF			
	Uniform allocation $1/K$	0.33	0.33	0.33	0.33 / 0.67	0.00	0%
SINGLECAP	Mathematics	0.69	0.19	0.12	0.69	0.46	98%
SINGLECAP	Code	0.30	0.65	0.05	0.65	0.34	91%
SINGLECAP	Instruction following	0.00	0.00	0.99	0.99	0.99	100%
MULTICAP	Math + IF	0.51	0.11	0.38	0.89	—	—
MULTICAP	Code + IF	0.25	0.47	0.28	0.75	—	—
MULTICAP	Math + Code	0.27	0.46	0.27	0.73	—	—

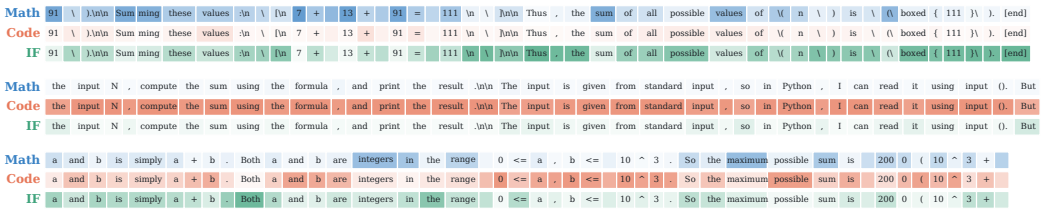

Figure 7: **Token-level allocation across all three MULTICAP compositions.** This figure extends the mathematics–instruction-following example in Figure 3(e) with examples covering all three compositions. Each block displays the same response tokens once per teacher; darker shading indicates a higher allocation weight at that position.

C.4 CORE COMPONENT ABLATIONS

C.4.1 DEFINITIONS OF THE ABLATED VARIANTS

This subsection defines the ablation variants in Figure 4. Throughout, $q_{k,t}$ and $p_{\theta,t}$ denote teacher and student distributions normalized on the student's top-C candidate set $\mathcal{A}_t$, following §B.4. Full TrustMOPD combines equation 2, equation 3, and equation 4:

$$\underbrace{\rho_{k,t}}_{\text{displacement}} \longrightarrow \underbrace{r_{k,t} = \rho_{k,t}/\mu_k}_{\text{calibration}} \longrightarrow \underbrace{w_{k,t}}_{\text{allocation}} . \tag{30}$$

Each variant changes the component specified below while retaining the shared training settings in §B.4.

Proxy substitutions. These variants replace the displacement score $\rho_{k,t}$ with one of three alternative signals:

$$\text{teacher likelihood:} \quad \rho^{\text{lik}}_{k,t} := q_{k,t}(y_t), \tag{31}$$

$$\text{negative entropy:} \quad \rho^{\text{negH}}_{k,t} := \log C - H(q_{k,t}), \qquad H(q) := -\sum_{v \in \mathcal{A}_t} q(v) \log q(v), \tag{32}$$

$$\text{teacher–student divergence:} \quad \rho^{\text{tsd}}_{k,t} := D_{\text{KL}}\big(q_{k,t} \,\|\, p_{\theta,t}\big). \tag{33}$$

Here, y_t is the token sampled from the student at position t. The negative-entropy score is shifted by $\log C$ to make it nonnegative. The divergence score uses the teacher-to-student KL direction, opposite to the reverse-KL distillation loss in equation 5.

Each alternative score is calibrated separately using Algorithm 1, with $\rho_{k,t}$ replaced by $\rho^{\text{alt}}_{k,t}$. The resulting scale μ^{alt}_k gives $r^{\text{alt}}_{k,t} = \rho^{\text{alt}}_{k,t}/\mu^{\text{alt}}_k$ an empirical mean of one over the calibration tokens. The calibrated scores are then converted into allocation weights using equation 4 with the same exponent γ.

Uncalibrated scores. "Without μ_k" uses raw displacement scores without teacher-specific calibration:

$$\mu_k := 1 \quad \text{for every } k, \qquad r_{k,t} = \rho_{k,t}, \qquad w_{k,t} = \frac{\rho^{\gamma}_{k,t}}{\sum_{j=1}^{K} \rho^{\gamma}_{j,t}}. \tag{34}$$

Response-level allocation. This variant computes the token-level weights of full TrustMOPD using equation 4, then replaces them with their mean within each sampled response:

$$\bar{w}_{i,k} := \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} w_{i,k,t}, \qquad w^{\text{resp}}_{i,k,t} := \bar{w}_{i,k} \quad \text{for all } t \le |y_i|. \tag{35}$$

The averaged weights remain nonnegative and sum to one across teachers. For every teacher k and rollout i, they also satisfy

$$\sum_{t=1}^{|y_i|} w^{\text{resp}}_{i,k,t} = \sum_{t=1}^{|y_i|} w_{i,k,t}. \tag{36}$$

Thus, on a given rollout, this transformation preserves each teacher's total allocation weight while removing its variation across token positions. The ablation tests the contribution of within-response allocation.

The control. UNIFORM, $w_{k,t} = 1/K$, is defined in §B.3 and included in Table 9 as the reference for assessing whether each alternative scoring rule improves performance over equal teacher weighting.

C.4.2 RESULTS

Table 9 reports the absolute scores corresponding to Figure 4, which shows changes relative to full TrustMOPD. All ablated variants have lower mean overall scores than full TrustMOPD on both training sets, although some individual domain scores improve.

All three alternative scoring rules underperform reference-relative displacement after separate calibration. Teacher–student divergence performs close to UNIFORM on SINGLECAP and below it on MULTICAP. These comparisons support the choice of displacement over the tested alternative signals.

Response-level allocation leaves the mean overall score nearly unchanged on SINGLECAP (0.05 points lower), but reduces it by 0.92 points on MULTICAP. This pattern suggests a larger benefit from within-response allocation when training prompts combine multiple capabilities.

Table 9: **Core component ablations: absolute scores.** Results correspond to Figure 4; variant definitions are given in §C.4.1. Scores are means ± standard deviations over five training seeds, evaluated at the final checkpoint. Within each training-set block, Δ_{full} is the difference in mean overall score from full TrustMOPD.

Data	Ablated component	Math	Code	IF	Overall ↑	Δ_{full}
SingleCap	Uniform $w_{k,t} = 1/K$ (control)	20.14 ±0.23	18.67 ±0.38	45.04 ±1.23	27.95 ±0.59	−2.09
	TrustMOPD (full)	21.91 ±0.61	20.67 ±0.75	47.56 ±0.48	**30.04** ±0.47	0.00
	proxy → teacher likelihood $q_{k,t}(y_t)$	20.02 ±0.92	18.90 ±0.67	46.29 ±0.57	28.41 ±0.42	−1.63
	proxy → negative entropy $-H(q_{k,t})$	19.96 ±0.72	18.79 ±0.69	46.47 ±0.72	28.41 ±0.25	−1.63
	proxy → teacher–student $D_{\text{KL}}(q_k \Vert p_\theta)$	19.66 ±0.84	17.82 ±0.37	46.40 ±0.32	27.96 ±0.20	−2.08
	without μ_k (set $\mu_k = 1$)	20.79 ±0.50	20.39 ±0.64	47.99 ±0.82	29.73 ±0.49	−0.31
	response-level $\bar{w}_{i,k} \to \widetilde{w}_{i,k}$	21.57 ±1.29	20.42 ±1.10	47.97 ±1.02	29.99 ±0.74	−0.05
MultiCap	Uniform $w_{k,t} = 1/K$ (control)	20.20 ±0.43	18.87 ±0.52	43.88 ±1.42	27.65 ±0.53	−2.76
	TrustMOPD (full)	22.92 ±1.40	20.69 ±0.50	47.62 ±1.41	**30.41** ±0.60	0.00
	proxy → teacher likelihood $q_{k,t}(y_t)$	20.44 ±0.81	19.10 ±0.83	45.46 ±2.06	28.33 ±0.88	−2.08
	proxy → negative entropy $-H(q_{k,t})$	20.54 ±1.01	19.47 ±0.65	46.78 ±0.77	28.93 ±0.38	−1.48
	proxy → teacher–student $D_{\text{KL}}(q_k \Vert p_\theta)$	19.04 ±0.72	17.11 ±0.64	45.91 ±2.21	27.35 ±0.85	−3.06
	without μ_k (set $\mu_k = 1$)	20.79 ±0.68	20.78 ±0.39	45.83 ±1.95	29.13 ±0.54	−1.28
	response-level $\bar{w}_{i,k} \to \widetilde{w}_{i,k}$	21.21 ±0.36	19.56 ±0.78	47.71 ±1.45	29.49 ±0.79	−0.92

Table 10: **Sensitivity to the allocation sharpness exponent γ.** Absolute scores and allocation concentration corresponding to Figure 5(a,b). Within each training-set block, Δ_{Uni} is the difference in mean overall score from Uniform. Concentration is measured by $\mathbb{E}_{s_t}[\max_k w_{k,t}]$, which equals $1/K$ for Uniform. Scores are means ± standard deviations over five training seeds, evaluated at the final checkpoint.

Data	Allocation rule	Math	Code	IF	Overall ↑	Δ_{Uni}	$\mathbb{E}[\max_k w_k]$
SingleCap	Uniform $w_{k,t} = 1/K$	20.14 ±0.23	18.67 ±0.38	45.04 ±1.23	27.95 ±0.59	—	0.33
	TrustMOPD, $\gamma = 3$	22.36 ±0.65	19.96 ±0.89	47.79 ±0.98	30.04 ±0.27	+2.09	0.66
	TrustMOPD, $\gamma = 5$	21.65 ±1.00	20.60 ±0.40	47.76 ±0.90	30.00 ±0.52	+2.05	0.76
	TrustMOPD, $\gamma = 7$ *(default)*	21.91 ±0.61	20.67 ±0.75	47.56 ±0.48	30.04 ±0.47	+2.09	0.83
	TrustMOPD, $\gamma = 9$	22.24 ±0.84	20.40 ±0.96	48.22 ±0.80	**30.29** ±0.40	+2.34	0.86
MultiCap	Uniform $w_{k,t} = 1/K$	20.20 ±0.43	18.87 ±0.52	43.88 ±1.42	27.65 ±0.53	—	0.33
	TrustMOPD, $\gamma = 3$	21.07 ±0.97	20.13 ±0.59	46.71 ±1.73	29.31 ±0.97	+1.66	0.56
	TrustMOPD, $\gamma = 5$	21.79 ±1.07	20.28 ±0.40	47.16 ±1.74	29.74 ±0.74	+2.09	0.67
	TrustMOPD, $\gamma = 7$ *(default)*	22.92 ±1.40	20.69 ±0.50	47.62 ±1.41	**30.41** ±0.60	+2.76	0.74
	TrustMOPD, $\gamma = 9$	22.24 ±1.14	20.59 ±0.55	46.44 ±1.01	29.76 ±0.43	+2.11	0.78

C.5 Allocation sharpness and transfer

Table 10 reports the scores and allocation concentration corresponding to Figure 5(a,b). TrustMOPD achieves higher mean overall scores than Uniform at every tested $\gamma \in \{3, 5, 7, 9\}$ on both training sets.

Increasing γ produces more concentrated allocation, but does not consistently improve performance. From $\gamma = 7$ to $\gamma = 9$, the mean overall score increases by 0.25 points on SingleCap and decreases by 0.65 on MultiCap. Among the tested values, the default $\gamma = 7$ achieves the highest mean score on MultiCap and remains close to the best result on SingleCap.

Table 11 reports absolute scores for the transfer experiments in Figure 5(c,d). The backbone-transfer setting uses DeepSeek-R1-Distill-Qwen-7B as the student initialization and shared reference, together with three public RL-finetuned teachers. The data-source transfer setting retains the default models and uses independently sourced training prompts.

Table 11: **Transfer across model backbones and prompt sources.** Absolute scores corresponding to Figure 5(c,d). Shaded rows denote UNIFORM controls. Within each model and training-data setting, Δ_{Uni} reports the mean overall-score gain over the matched UNIFORM control. Scores for trained methods are means ± standard deviations over five training seeds at the final checkpoint.

Data	Allocation rule	Math	Code	IF	Overall ↑	Δ_{Uni} ↑
Default setting: SmolLM3-3B models and the original training prompt sets (Table 1)						
INIT (student initialization)		17.02	15.97	41.66	24.89	—
SINGLECAP	UNIFORM	20.14 ±0.23	18.67 ±0.38	45.04 ±1.23	27.95 ±0.59	—
	TrustMOPD	21.91 ±0.61	20.67 ±0.75	47.56 ±0.48	30.04 ±0.47	+2.09 ±0.50
MULTICAP	UNIFORM	20.20 ±0.43	18.87 ±0.52	43.88 ±1.42	27.65 ±0.53	—
	TrustMOPD	22.92 ±1.40	20.69 ±0.50	47.62 ±1.41	30.41 ±0.60	+2.76 ±0.26
(c) Backbone → DeepSeek-R1-Distill-Qwen-7B: new student initialization, shared reference, and public RL teachers						
INIT (its own student initialization)		43.02	26.99	28.13	32.71	—
SINGLECAP	UNIFORM	49.67 ±3.11	30.00 ±1.24	32.83 ±1.70	37.50 ±1.33	—
	TrustMOPD	52.87 ±0.54	30.91 ±0.72	32.91 ±0.46	38.90 ±0.45	+1.40 ±0.97
MULTICAP	UNIFORM	50.55 ±2.13	30.37 ±1.03	31.68 ±1.31	37.53 ±1.36	—
	TrustMOPD	52.68 ±0.71	31.42 ±0.83	31.93 ±1.28	38.68 ±0.83	+1.15 ±0.66
(d) Prompt source → independent corpus: student, teachers and INIT *as in the reference block*						
SINGLECAP	UNIFORM	19.30 ±0.62	18.22 ±0.47	44.12 ±1.51	27.22 ±0.41	—
	TrustMOPD	21.36 ±1.44	19.69 ±0.42	47.27 ±1.04	29.44 ±0.77	+2.22 ±0.61
MULTICAP	UNIFORM	20.03 ±0.98	18.47 ±0.62	44.50 ±1.29	27.67 ±0.46	—
	TrustMOPD	22.33 ±2.52	19.74 ±0.48	46.78 ±1.15	29.62 ±1.08	+1.95 ±0.73

Table 12: **Computational cost of distillation and calibration.** Measurements use 8×H20 GPUs under the protocol in §B.4. *Scored models* counts distinct teacher and reference models evaluated per step, excluding the student. For label-routed MOPD, each example uses one teacher, while mixed-domain batches invoke all three teachers. *Total GPU-h* is elapsed time multiplied by eight, summed over five training seeds and including evaluation. Peak memory is reported per GPU. †Offline calibration is measured separately and performed once per prompt set; its percentage is relative to the reported TrustMOPD GPU-hour total.

Allocation rule	Scored models / step		Total GPU-h	GPU-h vs. UNIFORM	Peak mem (GB/GPU)
	Teachers	Reference			
Per-step training cost					
Single teacher $\mathbf{e}_{k_0}$	1	0	149	−16.8%	69.2
MOPD $\mathbf{e}_{d(x)}$	3	0	187	+4.5%	79.8
UNIFORM $\frac{1}{K}\mathbf{1}$	3	0	179	0.0%	87.5
RANDOM $\mathrm{Unif}(\Delta^{K-1})$	3	0	183	+2.2%	88.4
TrustMOPD $\propto \mathbf{r}_t^{\gamma}$	3	1	192	+7.3%	92.6
One-off cost, incurred once per prompt set before training					
Offline calibration of μ_k (8 batches)	3	1	1.3	+0.7%†	92.6

Relative to the matched UNIFORM controls, TrustMOPD improves mean overall scores by 1.40 and 1.15 points on the additional backbone, and by 2.22 and 1.95 points on the independent training corpus, for SINGLECAP and MULTICAP, respectively. These gains show that the benefit of adaptive allocation extends to the additional backbone and data-source settings tested here.

D COMPUTATIONAL OVERHEAD

Table 12 reports GPU-hours and peak GPU memory for the SmolLM3-3B setting on 8×H20 GPUs, following the training protocol in §B.4.

Case 1: local arithmetic supervision SmolLM3-3B, step 0

Problem. *56 lines are drawn on a plane such that no three are concurrent. If the lines intersect at exactly 594 points, what is the maximum number of them that could have the same slope?*

Student prefix (token 2526, inside the reasoning block):

```
...1558 - (Sum k_i²)/2 = 594
Therefore, (Sum k_i²)/2 = 1558 -594 =
```

Local criterion: $1558 - 594 = 964$. Among the 128 candidates, `"964"` and `"96"` match prefixes of this result; 39 other digit tokens have incompatible prefixes.

Teacher	Top-1 in $\mathcal{A}_t$		P(matching prefix) ↑	P(incompatible digits) ↓	$r_{k,t}$	$w_{k,t}$
math	`"964"`	✓	**0.538**	**0.235**	**3.02**	**0.999**
code	`"962"`	✗	0.234	0.666	0.63	0.000
IF	`"962"`	✗	0.124	0.716	1.19	0.001
the student itself	`"960"`	✗	0.125	0.693		

Distilled target $\tilde{q}_w$ at this position	Token-level	Response-level
Probability on the verified continuation `964` ↑	**0.537**	0.344

The mathematics teacher's top-1 token is `"964"`, whereas both other teachers prefer `"962"`. Token-level allocation assigns the mathematics teacher a weight of 0.999. The sampled student rollout emitted `"960"` at this position.

Figure 8: **Case 1: supervision at a local arithmetic position.** The displayed subtraction evaluates to 964. Matching-prefix mass sums the probabilities of `"964"` and `"96"`; incompatible-digit mass covers the 39 digit candidates that cannot begin this result. The mathematics teacher has the correct top-1 token and receives the largest allocation weight.

Training cost. Relative to UNIFORM and RANDOM, which already score all K teachers, TrustMOPD adds reference scoring and token-level allocation computations. Compared with UNIFORM, the measured training-and-evaluation GPU cost increases by 7.3%, and peak memory increases by 5.8%. Offline calibration costs an additional 1.3 GPU-hours per prompt set, with the resulting scales reused across training seeds.

Deployment. Deployment uses only the distilled student. Teachers, the reference, and the allocation procedure are not required, so TrustMOPD introduces no additional inference-time overhead relative to serving the same student architecture.

E CASE STUDIES OF TOKEN-LEVEL SUPERVISION

We examine two positions in MULTICAP rollouts where teacher preferences can be assessed using explicit local criteria: the result of a subtraction in Figure 8, and compliance with a no-comma constraint in Figure 9. At each position, we compare token-level allocation with the response-level control defined in equation 35.

Shared setup. Both cases use SmolLM3-3B rollouts from the initial student at step 0. Scores and weights are recomputed offline using the training implementation, with $\gamma = 7$, $C = 128$, and the frozen MULTICAP calibration constants. All reported probabilities and top-1 tokens are evaluated within the student's candidate set $\mathcal{A}_t$. The target $\tilde{q}_w$ is the normalized geometric mixture in equation 28; the response-level control uses the same rollout's mean weights $\bar{w}_k$.

These examples illustrate allocation favoring different specialists at different positions. In both cases, the token-level target improves the reported local probability metric relative to the response-

Case 2: local compliance with a no-comma constraint SmolLM3-3B, step 0

Problem. A quadratic-graph question with instruction constraints. The constraint examined here is: *"In your entire response refrain from the use of any commas."*

Student prefix (token 3608, in the evaluated final response; the scored region starts at token 3384):

```
...3.  **Determining the Vertex**:
   - The vertex of the quadratic function \( q(x) \) is at \( x = 7.5
\
```

Decidable by the constraint: Local criterion: a token containing a comma violates the constraint. Seven of the 128 candidates contain commas, including `","` and `"),"`.

Teacher	Top-1 in $\mathcal{A}_t$		P(forbidden) ↓	$r_{k,t}$	$w_{k,t}$
math	`"),"`	✗	0.469	1.04	0.392
code	`"),"`	✗	0.481	0.58	0.006
IF	`").\n"`	✓	**0.337**	**1.11**	**0.602**
the student itself	`","`	✗	0.493		

Distilled target $\tilde{q}_w$ at this position	**Token-level**	**Response-level**
Probability on forbidden tokens ↓	**0.390**	0.406

The instruction-following teacher is the only teacher whose top-1 token contains no comma. Its token-level weight is 0.602, compared with a response mean of $\bar{w}_{\mathrm{IF}} = 0.486$. The token-level target assigns 0.390 probability to comma-bearing tokens, compared with 0.406 under response-level allocation.

Figure 9: **Case 2: supervision under a no-comma constraint.** Forbidden-token mass sums probabilities over the seven comma-bearing candidates. Among the teachers, the instruction-following teacher assigns the least mass to these tokens and receives the largest allocation weight. Token-level allocation reduces the target's forbidden-token mass relative to the response-level control.

level control. They provide qualitative evidence of alignment with local supervision quality; the aggregate performance comparison is reported in Table 9.

F Limitations

Our findings are established on 3B and 7B backbones with mathematics, code, and instruction-following specialists; larger models and broader domains remain directions for extending this evaluation. MultiCap enables controlled comparisons of capability composition, while its programmatic construction captures only part of the variation in naturally blended user requests. Our experiments support displacement-based allocation within the evaluated specialist pools. The theoretical utility-contrast interpretation assumes globally optimal KL-regularized teachers, while practical RL uses finite-budget optimization. Finally, the current formulation assumes a shared pre-RL reference and incurs reference-scoring overhead during training (Appendix D), with extensions to teachers of heterogeneous provenance left to future work.