CONTEXT DISTILLATION DISTILLS MORE THAN SKILLS: NAVIGATING SKILL GAINS AND CROSS-DOMAIN RETENTION WITH STATE-SHARED DRIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Skill documents provide language model agents with reusable procedural knowledge, but retrieving and processing them at inference time incurs computational cost. On-policy context distillation can internalize such knowledge by training a document-free student to match a document-conditioned teacher. However, this process can improve the target skill while degrading instruction and rule following in unrelated domains. We identify *state-shared drift*: systematic token-score shifts that recur across prediction contexts with context-dependent magnitudes. Under heterogeneous distillation, we decompose this drift into document-induced and model-induced sources whose directions and scales differ substantially. Larger state-shared drift coincides with greater cross-domain degradation, while progressively attenuating it traces a trade-off between document-free skill gains and cross-domain retention. Based on this, we propose Drift-Corrected Skill Distillation (DCSD), which separately estimates the two source-specific drift components and uses them as correction directions. Control documents and selective masking help preserve skill guidance, while correction strength and correcting one or both sources adjust the balance between skill gains and cross-domain retention. Experiments with Qwen3 models on SRA-Bench and six cross-domain instruction- and rule-following benchmarks show that DCSD substantially improves cross-domain retention while preserving useful document-free skill gains.

1 INTRODUCTION

Large language model (LLM) agents often rely on procedural guidance to carry out specialized tasks, such as software development and document processing. Skill documents provide this guidance through step-by-step workflows, tool-use instructions, examples, and verification criteria, often accompanied by scripts and reference files (Anthropic, 2025; Zhang & Qi, 2026). These documents are modular and reusable, but applying them requires selecting relevant skills and supplying their contents during inference (Su et al., 2026). Because skill documents can span thousands of tokens and include additional supporting references, repeated use across queries and interaction steps can increase input-processing cost and inference latency (Gao et al., 2026; Yu et al., 2026; Zou et al., 2024). These costs motivate *skill internalization*: transferring the behavior taught by skill documents into model weights or adapters so that agents can apply learned procedures to new task instances without receiving the original skill text (Lu et al., 2026; Yu et al., 2026). Figure 1 contrasts document-free skill gains with the additional prompt cost of retrieval-based skill use.

Existing skill internalization methods learn procedural behavior through adapter training, reinforcement learning, or skill-conditioned distillation (Zhang & Qi, 2026; Yu et al., 2026; Lu et al., 2026; Wang et al., 2026a). However, improved target-skill performance does not imply that only the intended procedural behavior has been internalized. Distillation may also transfer output preferences that persist beyond the situations in which the skill is useful. We find that context-distillation-based skill internalization can substantially improve document-free skill use while degrading performance in domains not used during internalization, particularly on tasks requiring the model to follow new instructions or rules. This reveals a failure mode that target-skill accuracy alone does not capture.

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.

To understand this cross-domain degradation, we separate the supervision of on-policy context distillation (OPCD) (Ye et al., 2026) into two sources. The *document-induced shift* captures changes caused by conditioning the teacher on the skill document and therefore includes the procedural guidance to be internalized. The *model-induced shift* captures differences between the teacher and the initial student that remain even without the skill document. Because these two sources arise from different causes and exhibit different behavior, we analyze and estimate them separately.

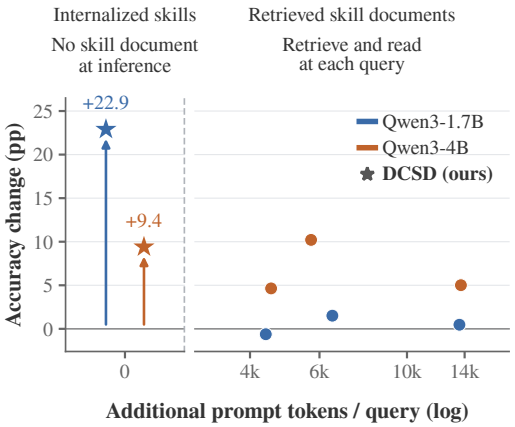


Figure 1: **Skill gains versus inference-time prompt cost.** SRA-Bench accuracy gain (pp) versus additional prompt tokens for skill use. Retrieval requires skill-document tokens at inference, whereas internalized skills require none. Circles denote retrieval and stars denote DCSD (ours). Details are in Appendix C.1.

Under heterogeneous distillation, both sources exhibit systematic shifts in token preferences that recur across generation steps within a response. We call this recurring behavior *state-shared drift*. Larger state-shared drift coincides with greater degradation in cross-domain instruction and rule following, while progressively attenuating the estimated drift improves cross-domain retention at the eventual cost of part of the acquired skill gain. State-shared drift therefore provides recurring structure that characterizes the trade-off between skill gains and cross-domain retention. We use its estimated directions to control how much recurring supervision is retained during distillation.

Building on this analysis, we propose *Drift-Corrected Skill Distillation* (DCSD), which separately estimates state-shared components in the document-induced and model-induced shifts and uses them as source-specific correction directions. Varying the correction strength and correcting one or both sources adjusts the balance between skill gains and cross-domain retention. Control documents help distinguish recurring effects associated with document conditioning from target-skill guidance, while selective correction masking limits correction where the shared component may be useful for the current prediction. The student is then trained on the resulting teacher distributions over its own responses.

Experiments with Qwen3 models (Yang et al., 2025) evaluate document-free skill use on a skill-stratified split of SRA-Bench (Su et al., 2026) and cross-domain instruction and rule following on six benchmarks, including IFEval (Zhou et al., 2023) and KOR-Bench (Ma et al., 2025). Standard OPCD substantially improves document-free skill performance but can also degrade cross-domain instruction and rule following, with substantially larger degradation under heterogeneous distillation. Across the evaluated settings, larger state-shared components coincide with larger cross-domain degradation. Varying correction strength and correcting one or both sources traces different balances between skill gains and cross-domain retention. DCSD uses this structure to preserve substantial document-free skill gains while improving cross-domain retention.

Our contributions are as follows:

- Cross-domain degradation in instruction and rule following is established as a failure mode of context-distillation-based skill internalization.
- We identify state-shared drift in document-induced and model-induced supervision and show that its attenuation traces the trade-off between document-free skill gains and cross-domain retention.
- Drift-Corrected Skill Distillation (DCSD) uses these source-specific components as correction directions, allowing correction strength and source-specific correction to adjust this trade-off.

2 Related Work

Agent Skills and Skill Internalization. Agent skills provide reusable procedural guidance (Su et al., 2026), while skill internalization transfers the behaviors they induce into model parameters.

One line of work preserves skills as modular adapters. Skill-to-LoRA learns a separate adapter for each skill (Zhang & Qi, 2026), while LatentSkill and Parametric Skills use hypernetworks to generate adapters from textual skill descriptions (Yu et al., 2026; Zhao et al., 2026). These approaches retain modularity but require the appropriate adapter to be selected or generated at inference. Another line of work incorporates skill-guided behavior into a shared policy. SKILL0 progressively removes skill context during reinforcement learning (Lu et al., 2026), while Skill-SD uses skill summaries derived from trajectories as training-only teacher context (Wang et al., 2026a). DCSD likewise internalizes skills into a shared policy, but uses recurring structure in the distillation signal to characterize and adjust the trade-off between target-skill gains and cross-domain instruction and rule following.

Context and On-Policy Distillation. Context distillation transfers behaviors elicited by instructions, examples, or intermediate reasoning into a model that no longer requires the original context at inference (Snell et al., 2022). On-policy distillation further reduces the mismatch between training prefixes and those encountered under the student policy. GKD obtains teacher supervision on student-generated sequences (Agarwal et al., 2024), while MiniLLM combines reverse-KL distillation with on-policy optimization (Gu et al., 2024). OPCD combines context and on-policy distillation by matching a context-conditioned teacher on student rollouts (Ye et al., 2026). DCSD builds directly on OPCD, retaining its rollout protocol and reverse-KL objective while modifying the teacher targets used for distillation.

Related methods also modify which parts of teacher supervision are transferred. OPD^2 constructs its distillation signal from the log-probability difference between a reasoning-tuned teacher and its pre-tuning base model (Heo et al., 2026), aiming to emphasize capabilities acquired during tuning rather than preferences already present in the base model. Flux-OPD instead weights contextual corrections according to conflicts among evolving contexts (Wang et al., 2026b). DCSD differs in the structure it targets: it identifies supervision that recurs across generation steps within a student response and estimates this structure separately for document-induced and model-induced shifts. Thus, OPD^2 distinguishes supervision across model checkpoints, Flux-OPD through interactions among contexts, and DCSD through recurring structure across generation steps. Appendix E discusses broader connections to agent memory, context compression, prediction correction, and interference during learning.

3 Problem Setup and Preliminaries

Skill internalization. We study the internalization of multiple skills into a single shared policy that can apply the acquired procedures without skill documents at inference. For each skill $k \in \{1, \dots, K\}$, let D_k denote its skill document and $\mathcal{X}_k$ the associated training queries. The internalization set is $\mathcal{D}_{\text{int}} = \{(x, D_k) \mid x \in \mathcal{X}_k,\ k = 1, \dots, K\}$. Let θ_0 and θ denote the initial and current student parameters, with corresponding policies $\pi_{S_0} = \pi_{\theta_0}$ and $\pi_S = \pi_\theta$. The student is trained using pairs (x, D_k), but at inference receives only x. The query may still contain task-specific instructions or rules, while the skill document used during internalization is omitted. We evaluate document-free skill gains on the internalization tasks and performance retention on cross-domain tasks, which are not used for checkpoint or hyperparameter selection.

On-policy context distillation. A straightforward approach to skill internalization is supervised imitation of fixed demonstrations generated by a document-conditioned teacher. Training on teacher-generated response prefixes, however, can mismatch the prefixes encountered under the student policy at inference. On-policy distillation reduces this mismatch by obtaining teacher supervision on student-generated sequences (Agarwal et al., 2024). For skill internalization, this provides document-conditioned guidance at the generation states visited by the document-free student. We adopt on-policy context distillation (OPCD) (Ye et al., 2026) as our starting point.

At each training step, the student samples $y = (y_1, \dots, y_T) \sim \pi_S(\cdot \mid x)$ without the document. At position t, the generation state is $s_t = (x, y_{<t})$. A frozen teacher evaluates the same state with access to D, yielding $\pi_T(\cdot \mid D, s_t)$.

Following OPCD, we compute distillation on the student's top-K next-token candidates at each state. The student and teacher probabilities are restricted to the same token set and renormalized, yielding p_t^S and p_t^T. For $(x, D) \sim \mathcal{D}_{\text{int}}$ and a student response y, OPCD minimizes the mean

token-wise reverse-KL loss

$$\mathcal{L}_{\mathrm{OPCD}}(\theta; x, D, y) = \frac{1}{T}\sum_{t=1}^{T} \mathrm{KL}\big(p_t^S \,\|\, p_t^T\big)\,. \tag{1}$$

Training averages this loss over sampled pairs and current-student rollouts. During each update, the sampled response and teacher targets are fixed, and gradients flow only through the student.

4 Analysis

We examine how OPCD changes task performance and what recurring structure appears in its supervision, using the evaluation settings in Section 6.

Skill gains and cross-domain degradation. Figure 2 shows that successful skill internalization can coexist with degraded performance in other domains. Both students improve on new queries that require skills encountered during internalization. Because all evaluation queries are excluded from training, these gains indicate that the students can apply the learned procedures to new inputs. Appendix C.2 describes the measurements.

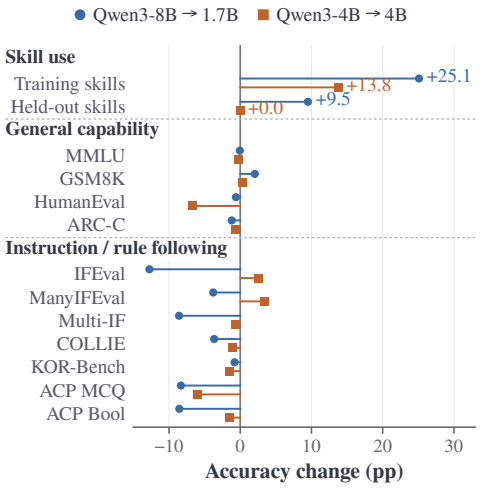


Figure 2: **Skill gains can coexist with cross-domain degradation.** Performance changes after OPCD relative to the initial student. Skill performance improves, while several instruction- and rule-following benchmarks decline.

These gains do not extend consistently to tasks that provide their own instructions and rules. Performance decreases on several such benchmarks even though they do not require the internalized skills. Evaluating skill acquisition alone therefore misses an important consequence of internalization. This observation motivates examining what OPCD transfers alongside useful skill guidance.

Recurring structure in teacher supervision. We first separate two sources of change in the teacher supervision. At each generation state s_t, let $\mathcal{V}_t$ denote the student's top-K next-token candidates used for distillation. Let $\ell_T(D, s_t)$ and $\ell_T(\emptyset, s_t)$ denote the teacher's full-vocabulary next-token log-probability vectors with and without the skill document, respectively, and let $\ell_{S_0}(s_t)$ denote the frozen initial student's full-vocabulary next-token log probabilities. Here, $\emptyset$ denotes the absence of a skill document, and $\cdot|_{\mathcal{V}_t}$ restricts a vector to the tokens in $\mathcal{V}_t$. The *document-induced shift* measures the change caused by providing the skill document, while the *model-induced shift* measures the remaining difference between the document-free teacher and the frozen initial student:

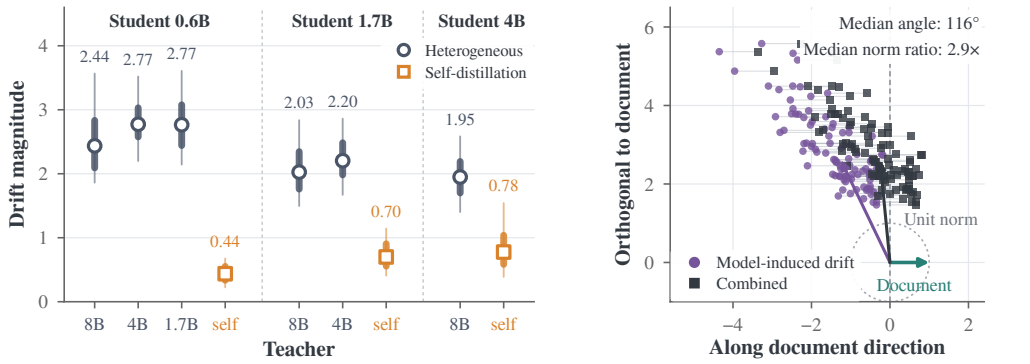
$$\Delta_t^{\mathrm{doc}} = \big[\ell_T(D, s_t) - \ell_T(\emptyset, s_t)\big]|_{\mathcal{V}_t}, \qquad \Delta_t^{\mathrm{model}} = \big[\ell_T(\emptyset, s_t) - \ell_{S_0}(s_t)\big]|_{\mathcal{V}_t}. \tag{2}$$

Their sum equals the difference between the document-conditioned teacher and the initial student on the same token set. The document-induced shift therefore captures changes introduced by skill conditioning, while the model-induced shift captures teacher–student differences that already exist without the skill document. The latter becomes zero when the teacher and initial student are identical.

We next ask whether parts of these shifts recur across generation states within the same response. For each source $a \in \{\mathrm{doc}, \mathrm{model}\}$, we represent the recurring component as

$$b_t^a = c_t^a\big(d_y^a|_{\mathcal{V}_t}\big), \qquad \Delta_t^a = b_t^a + r_t^a, \tag{3}$$

where d_y^a is a response-level shared direction, c_t^a controls its strength at state s_t, and r_t^a captures the remaining state-specific variation. We refer to b_t^a as *state-shared drift*. Empirically, shifts are more aligned across generation states within the same response than across different responses, supporting the presence of response-level recurring structure. The corresponding recurrence measurements are reported in Appendix C.4.



(a) State-shared drift across teacher–student pairs

(b) Drift from the two sources, 8B → 1.7B

Figure 3: **State-shared drift across distillation settings and source geometry.** (a) State-shared drift in the document-conditioned teacher–student shift is substantially larger under heterogeneous distillation than under self-distillation. (b) Relative geometry of the document- and model-induced response-level shifts for Qwen3-8B → Qwen3-1.7B. Each document-induced vector is aligned with the horizontal axis and normalized to unit length, with the same transformation applied to its model-induced counterpart. The two sources differ substantially in both direction and magnitude.

State-shared drift across distillation settings and source geometry. We first examine how state-shared drift changes across distillation settings. Panel (a) estimates the state-shared component of the document-conditioned teacher–student shift, aggregates it across generation states within each response, and reports the median across responses of its root-mean-square magnitude (Appendix C.3). Drift is small under self-distillation, where the teacher and initial student are identical, but becomes substantially larger under heterogeneous distillation. Thus, strong state-shared drift is associated with teacher–student mismatch in addition to document conditioning. The increase can reflect both the model-induced shift and changes in how different teacher models respond to the document.

We next examine how this larger drift is structured in the heterogeneous setting. Using Eq. 2, we separate the teaching signal into document-induced and model-induced shifts and aggregate each source across generation states within a response. Figure 3(b) shows their relative geometry for Qwen3-8B → Qwen3-1.7B. Each response is rotated and scaled so that its document-induced vector points horizontally with unit norm. The model-induced vector is shown under the same transformation, so its length and angle directly indicate its scale and direction relative to the document-induced vector.

The two sources differ substantially. The model-induced vector has a median norm 2.9 times that of the document-induced vector and a median relative angle of 116°. They therefore cannot be viewed as two magnitudes of a common recurring direction. Although their strengths may change together across model pairs, their relative directions and magnitudes within the heterogeneous setting remain distinct. This motivates estimating their state-shared components separately rather than estimating a single direction from the combined teaching signal. These observations identify state-shared drift as a candidate structure through which distillation may affect both acquired skill behavior and cross-domain performance. We next use source-specific correction to test how attenuating this structure changes the balance between the two.

5 Drift-Corrected Skill Distillation

DCSD estimates recurring drift separately for the document-induced and model-induced shifts, fits its contribution at each generation state, and corrects the teacher targets while limiting the removal of useful skill guidance. The student then learns from these corrected targets on its own responses. Algorithm 1 summarizes the procedure.

Estimating state-shared drift. Directly estimating recurring structure from the original document-induced shift may also capture preferences that are essential to the skill procedure itself. We therefore construct *control documents* by injecting Gaussian noise into the teacher's next-token logits when perturbing the original skill document. This makes procedure-specific token choices less con-

Algorithm 1 DCSD training step

Require: Current student π_S, frozen teacher π_T, frozen initial student π_{S_0}
Require: Query x, skill document D, control documents $\{C_m\}_{m=1}^{M}$, support size K, correction strength α

```
y ~ π_S(· | x)                                   ▷ Student rollout without the skill document
for each generation state s_t = (x, y_<t) do
    V_t ← TopK(π_S(· | s_t))
    Compute Δ_t^doc, Δ_t^model, and Δ_t^ctrl
end for
for each generation state s_t do
    Estimate d̂_t^doc and d̂_t^model by token-matched LOO      ▷ Estimate state-shared drift
    Compute c_t^a and b̂_t^a using Eq. 6
    Construct m_t                                ▷ Apply selective correction masking
    B̂_t ← m_t ⊙ b̂_t^doc + b̂_t^model
    ℓ̃_t ← ℓ_T(D, s_t)|_{V_t} − α B̂_t
    p̃_t^T ← softmax_{V_t}(ℓ̃_t)
    Renormalize π_S(· | s_t) on V_t to obtain p_t^S
end for
Update π_S using Eq. 8                           ▷ Optimize the student on corrected targets
```

sistently preserved across controls, so the recurring signal estimated from them is less dominated by guidance tied to the exact procedure and more reflective of preferences that persist under document perturbation. For each skill document D, we generate M controls $C_1, \ldots, C_M$ offline by perturbing the frozen teacher's next-token scores along the original document prefixes and replacing tokens with the resulting predictions. The controls are generated once and reused throughout training. Appendix A.2 gives the full construction procedure.

At each generation state s_t, we measure the teacher shift induced by each control relative to the document-free teacher and average across controls:

$$\Delta_t^{\mathrm{ctrl}} = \frac{1}{M} \sum_{m=1}^{M} \left[\ell_T(C_m, s_t) - \ell_T(\emptyset, s_t) \right] \big|_{\mathcal{V}_t}. \tag{4}$$

We then estimate the recurring pattern from other positions in the same response:

$$\hat{d}_t^{\mathrm{doc}} = \mathrm{LOO}_t \left(\{\Delta_u^{\mathrm{ctrl}}\}_{u \neq t} \right), \qquad \hat{d}_t^{\mathrm{model}} = \mathrm{LOO}_t \left(\{\Delta_u^{\mathrm{model}}\}_{u \neq t} \right). \tag{5}$$

Here, LOO_t denotes token-matched leave-one-out averaging over the other generation states. We exclude the current state so that its local shift cannot directly determine the recurring pattern used to correct it. Thus, $\hat{d}_t^a$ captures preferences supported by their recurrence at other positions rather than by the current prediction alone. Because the top-K support varies across states, each token is averaged only over positions where the same vocabulary token is present. Appendix B gives the exact matching rule.

The estimated pattern determines what recurs across states, while its contribution at the current state is fitted to the original source-specific shift:

$$c_t^a = \frac{\langle \Delta_t^a, \hat{d}_t^a \rangle}{\|\hat{d}_t^a\|_2^2 + \varepsilon}, \qquad \hat{b}_t^a = c_t^a \hat{d}_t^a, \qquad a \in \{\mathrm{doc}, \mathrm{model}\}. \tag{6}$$

In other words, $\hat{d}_t^a$ determines which token preferences recur across states, and c_t^a determines how strongly that pattern appears at the current state. This separates direction estimation from local strength: the recurring direction identifies what persists across states, while fitting to the original shift measures how strongly that pattern is present in the actual supervision at state s_t. For the document source, the recurring direction is estimated from the control-document shifts, while its strength is fitted to the original document-induced shift at the current state. For the model source, the recurring direction is estimated from model-induced shifts at other states, and its strength is fitted to the model-induced shift at the current state. Thus, control documents are used only for estimating

the document-side recurring direction. Unlike the document-induced shift, the model-induced shift is defined without conditioning on the skill document, so no analogous document perturbation is required for its recurring-direction estimate.

Selective correction masking. Recurring document-side preferences can still overlap with useful procedural guidance. To reduce the chance of removing such guidance at the current prediction, we mask the document-induced correction at two tokens: the sampled student token y_t, when it belongs to $\mathcal{V}_t$, and the teacher-preferred token within $\mathcal{V}_t$. These tokens correspond to the student's realized action and the teacher's strongest document-conditioned preference at the current state. Masking both preserves their score difference under the document-side correction. We apply this protection only to the document-side correction because this source directly contains the skill-conditioned guidance that the procedure is intended to internalize. Neither token is assumed to be correct, and the model-induced correction can still modify their relative scores.

Let m_t be zero at the masked tokens and one elsewhere. For correction strength $\alpha \geq 0$, the combined correction and corrected teacher scores are

$$\hat{B}_t = m_t \odot \hat{b}_t^{\text{doc}} + \hat{b}_t^{\text{model}}, \qquad \tilde{\ell}_t(\alpha) = \ell_T(D, s_t)|_{\mathcal{V}_t} - \alpha \hat{B}_t. \tag{7}$$

The source-specific coefficients determine how much document- and model-induced drift is fitted at each state, while α controls the overall correction strength. At $\alpha = 0$, the target reduces to the original OPCD target, and increasing α progressively attenuates the estimated recurring supervision. The source-specific decomposition also allows either the document- or model-induced term to be omitted when constructing $\hat{B}_t$. Thus, the correction strength and the corrected sources can be varied independently. Section 6 evaluates how these choices affect skill gains and cross-domain retention.

Training. We normalize the corrected scores as $\tilde{p}_t^T(\alpha) = \text{softmax}_{\mathcal{V}_t}(\tilde{\ell}_t(\alpha))$ and renormalize the current student's probabilities on the same support to obtain p_t^S. For a sampled training pair (x, D) and student response y, DCSD minimizes the mean token-wise reverse-KL loss

$$\mathcal{L}_{\text{DCSD}}(\theta; \alpha, x, D, y) = \frac{1}{T} \sum_{t=1}^{T} \text{KL}\big(p_t^S \,\|\, \tilde{p}_t^T(\alpha)\big). \tag{8}$$

Training averages this loss over sampled pairs and current-student rollouts. The teacher and initial-student reference models remain fixed, while their predictions, drift estimates, and corrected targets are recomputed for each new response. During each update, the sampled response and targets are fixed, and gradients flow only through the current student. Inference requires no skill documents.

Effect of correction on student updates. DCSD subtracts the estimated correction $\alpha\hat{B}_t$ from the teacher scores, which changes the relative preference assigned to tokens during distillation. Let z_t denote the student's logits on $\mathcal{V}_t$, $\mathcal{L}_t(\alpha) = \text{KL}(p_t^S \| \tilde{p}_t^T(\alpha))$, and $\mu_t = \mathbb{E}_{v \sim p_t^S}[\hat{B}_t(v)]$. Holding the current state, support, and estimated correction fixed, the gradient difference relative to uncorrected OPCD, $\alpha = 0$, is

$$\nabla_{z_t} \mathcal{L}_t(\alpha) - \nabla_{z_t} \mathcal{L}_t(0) = \alpha p_t^S \odot \left(\hat{B}_t - \mu_t \mathbf{1}\right). \tag{9}$$

The derivation is provided in Appendix B.2.

Thus, increasing the correction strength suppresses tokens whose estimated drift is larger than the student-weighted average, while relatively favoring tokens with smaller correction values. The effect is weighted by the student's current probability, so tokens that the student already considers likely receive larger updates. Because the gradient difference depends on $\hat{B}_t - \mu_t \mathbf{1}$, any token-independent offset in the correction cancels, and only relative differences in the estimated drift affect the update. Eq. 9 therefore connects state-shared drift to the student update: DCSD reduces the transfer of preferences receiving large drift corrections rather than uniformly shifting the teacher distribution.

6 Experiments

Benchmarks. We use four SRA-Bench (Su et al., 2026) domains for skill-internalization training and in-domain evaluation: MedCalc-Bench, TheoremQA, LogicBench, and CHAMP. In-domain performance in the main results is the mean accuracy across these domains on the full set of 1,546

Table 1: Skill internalization and cross-domain performance. SRA-Bench reports the mean accuracy across the four domains. Cross-domain Avg. is the mean over the six benchmarks after averaging multiple metrics within each benchmark. Evaluation details are in Appendix A. **Bold** denotes the best distillation method per column within each pairing.

Method	In-domain	Cross-domain										
	SRA-Bench	ACPBench		COLLIE	ManyIFEval	Multi-IF			KOR-Bench	IFEval		Avg.
		MCQ	Bool			t1	t2	t3		Prompt	Inst.	
Heterogeneous: Qwen3-8B → Qwen3-1.7B												
Initial student	30.6	47.0	59.2	26.4	28.0	68.8	48.5	31.1	33.5	68.4	76.0	43.8
Off-policy CD	**54.6**	35.7	47.8	**25.9**	18.7	49.4	31.8	16.9	19.4	49.7	60.3	32.2
OPD^2	39.5	31.9	51.1	18.2	18.7	50.7	34.2	18.3	26.0	51.2	62.7	32.6
OPCD	53.1	38.7	50.7	22.8	**24.2**	56.5	38.5	22.5	32.7	55.6	68.2	37.6
DCSD (ours)	53.5	**40.7**	**56.9**	22.5	23.3	**62.9**	**41.5**	**25.8**	**38.6**	**63.6**	**72.7**	**40.8**
Self-distillation: Qwen3-4B → Qwen3-4B												
Initial student	51.6	55.6	61.9	35.3	44.9	80.1	63.8	48.2	48.4	78.2	84.5	55.4
Off-policy CD	60.8	**56.2**	64.2	30.2	43.7	80.6	58.2	39.3	**49.5**	80.6	86.5	54.4
OPD^2	44.8	54.9	**74.9**	27.7	22.4	60.1	40.6	24.9	42.7	60.4	69.3	44.1
OPCD	**61.9**	49.6	60.4	**34.2**	**48.3**	80.3	**63.4**	**47.5**	46.9	80.8	86.5	55.3
DCSD (ours)	61.0	50.9	62.8	33.8	47.6	**82.2**	62.8	45.9	47.0	**81.3**	**86.9**	**55.5**

SRA-Bench test queries (Table 2), evaluated with no skill document in context for the initial student and all context-distilled students. Cross-domain evaluation measures instruction following with IFEval (Zhou et al., 2023), COLLIE (Yao et al., 2023), ManyIFEval (Harada et al., 2025), and Multi-IF (He et al., 2024), planning reasoning with ACPBench (Kokel et al., 2024), and rule following with KOR-Bench (Ma et al., 2025). These benchmarks are used only for evaluation and retain their native instructions and rules without any SRA skill document. For the analysis in Section 4, we additionally partition the SRA-Bench test queries according to whether their annotated skills appeared during internalization. Details are provided in Appendix C.2. Cross-domain Avg. is the unweighted mean of these six benchmark scores and is reported only when all six are available. General-capability benchmarks, data splits, and evaluation protocols are described in Appendix A.

Implementation details. We evaluate Qwen3 models (Yang et al., 2025) in heterogeneous distillation from Qwen3-8B to Qwen3-1.7B and self-distillation from Qwen3-4B to Qwen3-4B. For OPCD (Ye et al., 2026) and DCSD, we compute the reverse-KL loss on the student's top-K support with $K = 256$. DCSD uses $M = 3$ controls per skill document. Table 1 reports $\alpha = 1$ with both corrections, while Figure 4 evaluates how correction strength and single- versus two-source correction affect the trade-off between skill gains and cross-domain retention. Optimization, control generation, checkpoint selection, and evaluation details are provided in Appendix A.

Main results. Table 1 reports the initial student's performance before skill-internalization training and compares DCSD with three baselines. Off-policy context distillation (Off-policy CD) transfers document-conditioned teacher behavior without on-policy student rollouts (Snell et al., 2022). We adapt OPD^2 (Heo et al., 2026) using the teacher's log-probability difference with and without the skill document. It also changes the optimization rule, so this comparison does not isolate signal construction. OPCD (Ye et al., 2026) is the direct baseline for target correction because DCSD retains its student rollouts and reverse-KL objective. Baseline implementations are detailed in Appendix A.4.

Under heterogeneous distillation, OPCD raises SRA-Bench accuracy from 30.6 to 53.1 but lowers the cross-domain average from 43.8 to 37.6. DCSD preserves essentially the same skill gain at 53.5 while recovering the cross-domain average to 40.8, restoring 3.2 of the 6.2 points lost under OPCD. The recovery is partial because some individual benchmarks remain below the initial student. Off-policy CD further illustrates this trade-off, attaining the highest SRA-Bench score in this setting at 54.6 while reducing the cross-domain average to 32.2. Thus, skill acquisition and cross-domain retention must be evaluated jointly.

In contrast, when cross-domain interference is weak, correction changes the aggregate results only marginally. Under self-distillation, OPCD raises SRA-Bench accuracy from 51.6 to 61.9 while leaving the cross-domain average nearly unchanged from 55.4 to 55.3. DCSD yields a similar balance, reaching 61.0 on SRA-Bench and 55.5 on the cross-domain average. This contrast is consistent with

the much smaller state-shared drift under self-distillation in Figure 3(a), while both document- and model-induced components can vary across model pairings. DCSD's main benefit therefore appears in the setting where the cross-domain degradation it targets is substantial.

Effect of correction strength. The correction-strength sweep tests how attenuation of the estimated state-shared component changes the trade-off between skill gains and cross-domain retention. We focus on Qwen3-8B $\rightarrow$ Qwen3-1.7B, where Table 1 shows substantial cross-domain degradation and therefore makes this trade-off visible. At $\alpha = 0$, DCSD reduces to OPCD, while larger α attenuates the estimated state-shared component more strongly.

Within each variant, the sweep changes only α and uses the same SRA validation rule for checkpoint selection. Cross-domain scores are used neither for checkpoint selection nor for hyperparameter selection. Figure 4 shows a continuous trade-off as the estimated state-shared component is attenuated more strongly. Increasing α from zero to one improves the cross-domain average while retaining the OPCD-level skill gain. Increasing it further to $\alpha = 1.5$ produces additional cross-domain recovery but lowers in-domain accuracy below both $\alpha = 1$ and uncorrected OPCD, while still remaining above the initial student. Correction strength therefore determines the balance between preserving acquired skill gains and improving cross-domain retention.

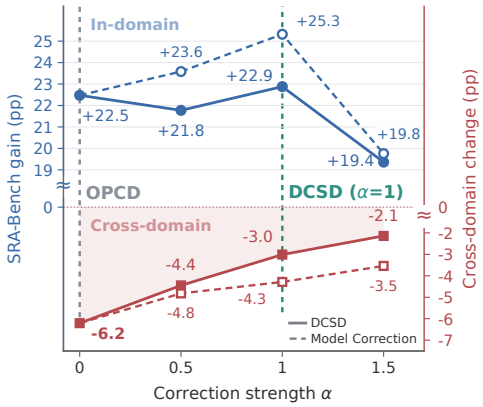


Figure 4: **Correction-strength and source-specific sweeps.** Changes from the initial student for Qwen3-8B $\rightarrow$ Qwen3-1.7B. In-domain and cross-domain metrics follow Table 1. Solid lines correct both sources and dashed lines only the model-induced source. $\alpha = 0$ corresponds to OPCD.

Source-specific correction. Correcting one or both sources provides a second degree of control because the document- and model-induced components contribute differently to the trade-off. Figure 4 therefore compares DCSD, which corrects both sources, with Model Correction, which corrects only the model-induced component at matched values of α.

Across nonzero strengths, Model Correction yields higher in-domain performance than DCSD but less cross-domain recovery. Leaving out document-side correction therefore preserves more signal useful for skill acquisition, while including it provides greater cross-domain recovery. This matched-strength comparison complements the source geometry in Figure 3 and shows that single- versus two-source correction changes the balance between skill gains and cross-domain retention.

Together, these sweeps show that correction strength and source-specific correction use the estimated state-shared directions to adjust the trade-off between skill gains and cross-domain retention. Unlike the fixed-state gradient comparison in Eq. 9, these sweeps measure the outcome of full training, including changes in the states visited by the student.

Additional results. Appendix D reports the full per-benchmark correction-strength sweep, source-specific and selective-masking ablations, broader general-capability retention, and training cost. It also applies the model-side correction to document-free heterogeneous on-policy distillation, showing that the same correction principle extends beyond OPCD-based skill internalization.

7 Conclusion

Skill gains from context distillation can coexist with degraded cross-domain performance, particularly under heterogeneous distillation. We identify state-shared drift in document- and model-induced supervision and show that its attenuation traces the trade-off between skill gains and cross-domain retention. Varying correction strength and correcting one or both sources changes the balance between document-free skill gains and cross-domain retention, with stronger correction eventually sacrificing part of the acquired skill gain for further cross-domain improvement. We further find that the same correction principle improves cross-domain retention in plain heterogeneous on-policy distillation, extending DCSD beyond OPCD-based skill internalization.

AI USE STATEMENT

Generative AI tools assisted manuscript editing, implementation and debugging of research code, and preparation of scientific figures. All AI-assisted outputs were reviewed and verified by the authors, who take responsibility for the final content of this work.

ETHICS STATEMENT

This study uses existing publicly available language models and benchmarks and involves neither human-subject experiments nor collection of new personal data. The reported reductions in cross-domain degradation concern the evaluated tasks and do not establish broader safety or fairness guarantees.

REPRODUCIBILITY STATEMENT

Sections 5 and 6 and Appendices A and B describe the method, data splits, and training and evaluation settings. We plan to release the experimental code.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Anthropic. Equipping agents for the real world with Agent Skills. Anthropic Engineering, 2025. URL https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Tianlei Chen, Jiao Ou, Ziyuan Liu, Ruiming Tang, Jian Liang, and Han Li. Counteraction-aware multi-teacher on-policy distillation for general capability recovery with domain preservation. *arXiv preprint arXiv:2605.27115*, 2026.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. TheoremQA: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7889–7901. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.489. URL https://aclanthology.org/2023.emnlp-main.489/.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting Language Models to Compress Contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://aclanthology.org/2023.emnlp-main.232/.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. DoLa: Decoding by Contrasting Layers Improves Factuality in Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.03883.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Gen Dong, Yanjie Gao, Liqun Li, Tianyin Xu, Yu Hua, and Fan Yang. Agent skills can be harmful: An empirical study of skill-induced failures in llm agents. *arXiv preprint arXiv:2608.11888*, 2026.

Shiyuan Feng, Huan-ang Gao, Haohan Chi, Hanlin Wu, Zhilong Zhang, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Weak-to-strong generalization via direct on-policy distillation. *arXiv preprint arXiv:2607.05394*, 2026.

Yudong Gao, Zongjie Li, Yuanyuan Yuan, Zimo Ji, Pingchuan Ma, and Shuai Wang. SkillReducer: Optimizing LLM agent skills for token efficiency. *arXiv preprint arXiv:2603.29919*, 2026. URL https://arxiv.org/abs/2603.29919.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context Autoencoder for Context Compression in a Large Language Model. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.06945.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8ac015d409635f196f9e3e9dcfb9a94e-Paper-Conference.pdf.

Keno Harada, Yudai Yamazaki, Masachika Taniguchi, Edison Marrese-Taylor, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. When instructions multiply: Measuring and estimating LLM capabilities of multiple instructions following. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 16506–16526. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.896. URL https://aclanthology.org/2025.findings-emnlp.896/.

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, et al. Multi-IF: Benchmarking LLMs on multi-turn and multilingual instructions following. *arXiv preprint arXiv:2410.15553*, 2024. URL https://arxiv.org/abs/2410.15553.

Roee Hendel, Mor Geva, and Amir Globerson. In-Context Learning Creates Task Vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023. URL https://arxiv.org/abs/2310.15916.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020. URL https://arxiv.org/abs/2009.03300.

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation. *arXiv preprint arXiv:2607.15161*, 2026.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface Form Competition: Why the Highest Probability Answer Isn't Always Right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://aclanthology.org/2021.emnlp-main.564/.

Nikhil Khandekar, Qiao Jin, Guangzhi Xiong, Soren Dunn, Serina S Applebaum, Zain Anwar, Maame Sarfo-Gyamfi, Conrad W Safranek, Abid A Anwar, Andrew Zhang, et al. MedCalc-Bench: Evaluating large language models for medical calculations. *arXiv preprint arXiv:2406.12036*, 2024. URL https://arxiv.org/abs/2406.12036.

Taeil Kim, Kangsan Kim, and Sung Ju Hwang. Agent Memory Distillation: Empowering Small LLM Agents with Hierarchical Teacher Memory. *arXiv preprint arXiv:2608.07169*, 2026. URL https://arxiv.org/abs/2608.07169.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017. URL https://doi.org/10.1073/pnas.1611835114.

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*, 2024. URL https://arxiv.org/abs/2402.03898.

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*, 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

Harsha Kokel, Michael Katz, Kavitha Srinivas, and Shirin Sohrabi. ACPBench: Reasoning about action, change, and planning. *arXiv preprint arXiv:2410.05669*, 2024. URL https://arxiv.org/abs/2410.05669.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2202.10054.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive Decoding: Open-ended Text Generation as Optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023. URL https://arxiv.org/abs/2210.15097.

Zhizhong Li and Derek Hoiem. Learning without Forgetting. In *European Conference on Computer Vision*, 2016. URL https://arxiv.org/abs/1606.09282.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive Knowledge Distillation through Imitation Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020. URL https://aclanthology.org/2020.emnlp-main.494/.

Hongxiang Lin, Zhirui Kuai, Erpeng Xue, and Lei Wang. SKILLC: Learning Autonomous Skill Internalization in LLM Agents via Contrastive Credit Assignment. *arXiv preprint arXiv:2605.27899*, 2026a. URL https://arxiv.org/abs/2605.27899.

Huizu Lin, Chengkai Huang, Tianqi Gao, Tao Huang, Daijiao Liu, Tongxin Li, Xiaoyan Sun, and Lina Yao. AUSO: Action-Level Unified Skill Optimization from Internalization to Utilization. *arXiv preprint arXiv:2608.21292*, 2026b. URL https://arxiv.org/abs/2608.21292.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. URL https://arxiv.org/abs/1711.05101.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. SKILL0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*, 2026. URL https://arxiv.org/abs/2604.02268.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023. URL https://arxiv.org/abs/2308.08747.

Kaijing Ma, Xinrun Du, Yunran Wang, Haoran Zhang, Zhoufutu Wen, Xingwei Qu, Jian Yang, Jiaheng Liu, Minghao Liu, Xiang Yue, Wenhao Huang, and Ge Zhang. KOR-Bench: Benchmarking language models on knowledge-orthogonal reasoning tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SVRRQ8goQo.

Yujun Mao, Yoon Kim, and Yilun Zhou. CHAMP: A competition-level dataset for fine-grained analyses of LLMs' mathematical reasoning capabilities. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13256–13274. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.785. URL https://aclanthology.org/2024.findings-acl.785/.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to Compress Prompts with Gist Tokens. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2304.08467.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13679–13707. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.739. URL https://aclanthology.org/2024.acl-long.739/.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! In *International Conference on Learning Representations*, 2024. URL https://openreview.net/pdf?id=hTEGyKf0dZ.

Tao Ren, Weiyao Luo, Hui Yang, Rongzhi Zhu, Xiang Huang, Yuchuan Wu, Bingxue Chou, Jieping Ye, Jiafeng Liang, Yongbin Li, and Yijie Peng. Scaling Self-Evolving Agents via Parametric Memory. *arXiv preprint arXiv:2606.04536*, 2026. URL https://arxiv.org/abs/2606.04536.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen-tau Yih. Trusting Your Evidence: Hallucinate Less with Context-aware Decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, 2024. URL https://aclanthology.org/2024.naacl-short.69/.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022. URL https://arxiv.org/abs/2209.15189.

Hongwen Song et al. More skills, worse agents? skill shadowing degrades performance when expanding skill libraries. *arXiv preprint arXiv:2605.24050*, 2026.

Weihang Su, Jianming Long, Qingyao Ai, Qiaozhi He, Yichen Tang, Changyue Wang, Yiteng Tu, Yingbo Wang, and Yiqun Liu. Skill retrieval augmentation for agentic AI. *arXiv preprint arXiv:2604.24594*, 2026.

Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15731–15740. IEEE, 2024.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function Vectors in Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.15213.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint arXiv:2305.16291*, 2023a. URL https://arxiv.org/abs/2305.16291.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-SD: Skill-conditioned self-distillation for multi-turn LLM agents. *arXiv preprint arXiv:2604.10674*, 2026a. URL https://arxiv.org/abs/2604.10674.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal Subspace Learning for Language Model Continual Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023b. URL https://arxiv.org/abs/2310.14152.

Yuran Wang, Zekun Wang, Bohan Zeng, Ruixu Zhang, Wenxuan Liu, Liu Yang, Yifan Dai, Yang Shi, Bozhou Li, Chengzhuo Tong, et al. Flux-opd: On-policy distillation with evolving contexts. *arXiv preprint arXiv:2607.28022*, 2026b.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent Workflow Memory. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/wang25bx.html.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Howard Chen, Austin W. Hanjie, Runzhe Yang, and Karthik Narasimhan. COLLIE: Systematic construction of constrained text generation tasks. *arXiv preprint arXiv:2307.08689*, 2023. URL https://arxiv.org/abs/2307.08689.

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

Aofan Yu, Chenyu Zhou, Tianyi Xu, Zihan Guo, Rong Shan, Zhihui Fu, Jun Wang, Weiwen Liu, Yong Yu, Weinan Zhang, et al. LatentSkill: From in-context textual skills to in-weight latent skills for LLM agents. *arXiv preprint arXiv:2606.06087*, 2026.

Tianyi Zhang and Zhonghao Qi. Skill-to-LoRA: From using skills to learning behaviors for token-efficient LLM agents. *arXiv preprint arXiv:2606.16769*, 2026. URL https://arxiv.org/abs/2606.16769.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM Agents Are Experiential Learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. URL https://arxiv.org/abs/2308.10144.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate Before Use: Improving Few-Shot Performance of Language Models. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. URL https://arxiv.org/abs/2102.09690.

Xuan Zhao, Haonan He, Qingyu Yang, Minglei Li, Jingqi Ye, Zelin Tan, Bo Wan, and Peng Ye. Parametric skills. *arXiv preprint arXiv:2606.30015*, 2026.

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 arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. Self-Discover: Large Language Models Self-Compose Reasoning Structures. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e41efb03e20ca3c231940a3c6917ef6f-Abstract-Conference.html.

Jiapeng Zhu, Jianxiang Yu, Yibo Zhao, Chengcheng Han, Qi Gu, Xunliang Cai, Xiang Li, and Weining Qian. Skill0.5: Joint Skill Internalization and Utilization for Out-of-Distribution Generalization in Agentic Reinforcement Learning. *arXiv preprint arXiv:2605.28424*, 2026. URL https://arxiv.org/abs/2605.28424.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. ToolQA: A dataset for LLM question answering with external tools. *arXiv preprint arXiv:2306.13304*, 2023. URL https://arxiv.org/abs/2306.13304.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. *arXiv preprint arXiv:2406.15877*, 2024. URL https://arxiv.org/abs/2406.15877.

Jiaru Zou, Mengyu Zhou, Tao Li, Shi Han, and Dongmei Zhang. PromptIntern: Saving inference costs by internalizing recurrent prompt during large language model fine-tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10288–10305. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.602. URL https://aclanthology.org/2024.findings-emnlp.602/.

APPENDIX CONTENTS

A Experimental setup **17**

A.1 Benchmarks 17

A.2 Models and distillation setup 19

A.3 Training and evaluation details 19

A.4 Baselines 19

B Method details and derivation **20**

B.1 DCSD estimation details 20

B.2 Effect of Correction on Distillation 21

C Measurement details for main figures **22**

C.1 Retrieval cost and skill gains 22

C.2 Skill-gain and cross-domain metrics 22

C.3 State-shared drift across teacher–student pairs 23

C.4 Document and model shift geometry 24

D Additional experiments **25**

D.1 General-capability evaluation 25

D.2 Correction strength sweep 25

D.3 Correction ablation 26

D.4 Training cost 27

D.5 Extending DCSD to On-Policy Distillation 27

E Additional related work **28**

Table 2: **Benchmark suite and evaluation roles.** SRA-Bench is split into training, validation, and in-domain test queries. All other benchmarks are used only for evaluation with their native task inputs and no SRA skill documents.

Benchmark	Evaluates	Queries	Metric
In-domain			
SRA-Bench (Su et al., 2026)	skill use without the document	1,156 train / 128 val / 1,546 test (4 domains)	mean accuracy (4 domains)
Cross-domain: contextual instruction following and related capabilities			
KOR-Bench (Ma et al., 2025)	novel in-context rules (5 categories)	1,250	accuracy
IFEval (Zhou et al., 2023)	verifiable instructions	541	prompt- and instruction-level strict acc.
ManyIFEval (Harada et al., 2025)	many instructions per prompt	2,160	prompt-level all-follow
Multi-IF (He et al., 2024)	multi-turn instructions	909 English (896 with a third turn)	turn-1/2/3 strict all-follow
COLLIE (Yao et al., 2023)	compositional constraints	2,080	constraint satisfaction
ACPBench (Kokel et al., 2024)	action and planning	900 MCQ / 900 Bool	accuracy
General capabilities			
MMLU (Hendrycks et al., 2020)	general knowledge and reasoning	14,042	accuracy
GSM8K (Cobbe et al., 2021)	mathematical reasoning	1,319	exact match (strict)
HumanEval (Chen et al., 2021)	code generation	164	pass@1
ARC-C (Clark et al., 2018)	scientific reasoning	1,172	accuracy

A Experimental setup

A.1 Benchmarks

We use SRA-Bench (Su et al., 2026) for skill-internalization training and in-domain evaluation, six benchmarks for cross-domain evaluation of contextual instruction following and related capabilities, and four benchmarks for broader general capabilities. Table 2 summarizes their roles. SRA-Bench is the only benchmark used for training or checkpoint selection. All other benchmarks are evaluation only. The main-text cross-domain average uses the six cross-domain benchmarks, while general-capability results are reported separately.

SRA-Bench (Su et al., 2026) evaluates the use of skill documents provided in context. The original benchmark contains six domains: MedCalc-Bench (Khandekar et al., 2024), TheoremQA (Chen et al., 2023), LogicBench (Parmar et al., 2024), CHAMP (Mao et al., 2024), BigCodeBench (Zhuo et al., 2024), and ToolQA (Zhuang et al., 2023). We use MedCalc-Bench, TheoremQA, LogicBench, and CHAMP. SRA-Bench contains both single- and multi-skill mappings, so we refer to the benchmark-provided gold skill annotation rather than assuming one gold skill per query.

Because the original SRA-Bench is evaluation only, we construct a query-level split stratified by these gold skill annotations. Within each annotation group, half of the associated queries are assigned to the training pool and the remainder to test. We use fixed seed 42 for the split, and annotation groups containing a single query are assigned to test. This produces a training pool of 1,284 queries and a test set of 1,546 queries with no query overlap. We then split the training pool by domain at a 9:1 ratio into 1,156 training and 128 validation queries. The main results report the mean accuracy across the four domains on all 1,546 test queries, without skill documents at infer-

ence. The test set contains both queries whose annotated skill information was represented during internalization and queries whose annotation was not. Section 4 separates these cases when analyzing skill gains. Because this is a custom split, its scores are not directly comparable to the official SRA-Bench leaderboard.

We exclude ToolQA because it requires tool execution, which is outside our document-only internalization setting. We exclude BigCodeBench because its long skill documents can produce teacher prompts beyond our 4,096-token training context limit, reaching up to 12,300 tokens in our preprocessing.

IFEval (Zhou et al., 2023) evaluates compliance with programmatically verifiable instructions over 541 prompts and 25 instruction types. We report prompt-level strict accuracy, which requires all instructions in a prompt to be satisfied, and instruction-level strict accuracy, which scores individual instructions. Their arithmetic mean defines the IFEval contribution to the cross-domain aggregate. We evaluate IFEval with lm-evaluation-harness using the model's chat template in non-thinking mode, greedy decoding, and at most 1,280 new tokens.

KOR-Bench (Knowledge-Orthogonal Reasoning) (Ma et al., 2025) evaluates whether a model can apply newly invented rules supplied in context rather than relying on pretrained knowledge alone. It contains 1,250 questions across five categories, with 25 rules and 10 questions per rule in each category. We evaluate all questions in the rule-provided setting and report accuracy.

ACPBench (Kokel et al., 2024) evaluates reasoning about actions, change, and planning from natural-language descriptions of planning domains and problem states. It contains multiple-choice (MCQ) and yes/no (Bool) questions across seven task types, with 900 questions per format. An MCQ response is correct when its final option letter matches the reference answer, while a Bool response is correct when its yes/no answer matches the reference.

COLLIE (Yao et al., 2023) evaluates constrained text generation. Each prompt requires text satisfying one or more constraints at the word, sentence, paragraph, or passage level. Constraint satisfaction is checked deterministically, and a response is correct only when all specified constraints are satisfied.

ManyIFEval (Harada et al., 2025) extends IFEval to prompts containing multiple simultaneous instructions. It constructs 10 variants for each of 216 task descriptions by varying the number of mutually compatible instructions from 1 to 10, yielding 2,160 examples. We evaluate all examples and count a response as correct only when every instruction in that example is satisfied.

Multi-IF (He et al., 2024) evaluates instruction following across three conversational turns, with new verifiable instructions introduced at each turn. We use all 909 English conversations, 896 of which contain a non-empty third-turn prompt in the released data. At each turn, the model receives the conversation history including its own earlier responses. The main tables report strict all-follow accuracy for turns 1, 2, and 3 on 909, 909, and 896 conversations, respectively. Their arithmetic mean defines the Multi-IF contribution to the cross-domain aggregate. This English strict all-follow setting differs from the benchmark's overall multilingual score.

General capabilities. We evaluate MMLU (Hendrycks et al., 2020), GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021), and ARC-Challenge (Clark et al., 2018) using the default lm-evaluation-harness configurations.

Cross-domain aggregate. We first average multiple reported metrics within each benchmark: MCQ and Bool accuracy for ACPBench, turn-1/2/3 accuracy for Multi-IF, and prompt- and instruction-level strict accuracy for IFEval. COLLIE, ManyIFEval, and KOR-Bench each contribute one score. Cross-domain Avg. is the unweighted arithmetic mean of these six benchmark scores, so each benchmark receives equal weight. It excludes SRA-Bench and the general-capability benchmarks.

For cross-domain evaluation, the model receives only each benchmark's native inputs and no SRA skill documents. These benchmarks do not affect checkpoint selection. Generative evaluations use greedy decoding.

A.2 Models and distillation setup

We use Qwen3 models (Yang et al., 2025) in heterogeneous distillation from Qwen3-8B to Qwen3-1.7B and self-distillation from Qwen3-4B to Qwen3-4B. The self-distillation teacher and student share the same architecture and initialization. The heterogeneous teacher is larger than the student. Teacher parameters remain fixed throughout training. We disable thinking mode for all models during training and evaluation to study document-free skill use without the additional generation cost of an explicit thinking phase.

Following OPCD (Ye et al., 2026), we evaluate the reverse-KL objective on the student's top-K support with $K = 256$, renormalizing the teacher and student distributions on this support. For DCSD, we generate $M = 3$ control documents per skill document using independent random seeds. Controls are generated offline once per skill document and reused throughout training. We tokenize the original document and run the frozen teacher on its unmodified prefixes. For each position after the first, we add independent Gaussian noise with $\sigma = 2$ to the corresponding next-token logits and take the argmax. The first original token is retained. Each prediction is conditioned on the original prefix, not on previously resampled tokens. We decode the resulting token sequence with special tokens removed and insert the text into the same skill-document prompt template. This preserves the number of token positions before decoding, but does not guarantee unchanged token length after retokenization, surface structure, or procedural content. Table 1 uses $\alpha = 1$ with both source-specific corrections. Figure 4 evaluates the trade-off across correction strengths and single- versus two-source correction. The sweep keeps the estimator, precomputed control documents, and training protocol fixed and selects each checkpoint using the same SRA-Bench validation rule.

A.3 Training and evaluation details

All training runs use a single H200 GPU, a batch size of 128, and AdamW (Loshchilov & Hutter, 2017) with a learning rate of 1×10^{-6}. All context-distilled models, including off-policy CD, follow the same checkpoint-selection rule. We evaluate every 100 steps using the mean accuracy across the four SRA-Bench validation domains. Training is capped at 2,000 steps with an early-stopping patience of 5, and we select the checkpoint with the highest validation accuracy.

For in-domain evaluation, the initial student and context-distilled models are evaluated on the SRA-Bench test queries (Table 2) without skill documents. Cross-domain evaluations preserve each benchmark's native instructions, rules, or constraints without providing any SRA-Bench skill document. We compare performance with the initial student to assess skill gains and cross-domain changes.

A.4 Baselines

On-policy context distillation (OPCD) (Ye et al., 2026) serves as the base training framework for our method. The student generates responses to queries without the skill document, while the teacher, conditioned on the document, evaluates the student-generated responses under teacher forcing. The student is then trained to minimize the token-wise reverse KL divergence between its own distribution and the teacher distribution. Our method follows the same training procedure as OPCD but modifies the distillation target.

Off-policy context distillation (Off-policy CD) (Snell et al., 2022) is a method in which the teacher, conditioned on a skill document, pre-generates responses to queries, and the student is trained to reproduce these fixed trajectories without the document while matching the teacher distribution. We directly use the baseline pipeline provided in the official OPCD repository (Ye et al., 2026).

OPD^2 (Heo et al., 2026) isolates the changes induced by reasoning fine-tuning through the log-probability difference between a tuned teacher and its pre-tuning base model. It centers this delta using the student's probability weights and suppresses sampled-token rewards whose signs conflict with the centered teacher–student signal. For skill internalization, we replace the tuning-induced delta with the same teacher's log-probability difference with and without the skill document, retaining the centering and sign-consistency rule. Our implementation approximates the student-weighted sums on a top-1,024 support that includes the sampled token, uses reward weight 0.1, and applies policy-gradient updates to sampled-token log probabilities. DCSD instead optimizes reverse KL

over the top-256 support. This comparison therefore evaluates the adapted methods as a whole, rather than isolating the choice of distillation signal.

B METHOD DETAILS AND DERIVATION

B.1 DCSD ESTIMATION DETAILS

This section provides additional details for the estimation procedure in Algorithm 1. For each sampled response, we first compute the teacher and initial-student predictions and the corresponding shifts at every generation state $s_t = (x, y_{<t})$ on the current student's top-K support $\mathcal{V}_t$. The recurring directions and corrected teacher targets are then computed using the full sampled response. The teacher and initial student remain frozen throughout training, while their predictions and all response-dependent quantities are recomputed for each new student rollout.

Direction inputs. The document-induced and model-induced shifts, Δ_t^{doc} and $\Delta_t^{\mathrm{model}}$, are defined in Eq. 2. The recurring direction for the model-induced shift is estimated directly from $\Delta_t^{\mathrm{model}}$. For the document-induced shift, we instead use the precomputed control documents. Let $D_1^{\mathrm{ctrl}}, \ldots, D_M^{\mathrm{ctrl}}$ denote the M controls generated from skill document D. Their average teacher shift relative to the document-free teacher is

$$\Delta_t^{\mathrm{ctrl}} = \frac{1}{M} \sum_{m=1}^{M} \left[\ell_T(D_m^{\mathrm{ctrl}}, s_t) - \ell_T(\emptyset, s_t) \right]\big|_{\mathcal{V}_t}. \tag{10}$$

As in Eq. 2, these shifts are computed from full-vocabulary log probabilities and then restricted to $\mathcal{V}_t$, without renormalization before taking the difference. Appendix A.2 describes how the control documents are constructed.

Token-matched LOO averaging. To keep the current state's shift from determining its own recurring direction, we estimate it using the other generation states in the same response. For a token $v \in \mathcal{V}_t$, let $U_t(v) = \{u \neq t : v \in \mathcal{V}_u\}$ denote the other positions whose top-K supports contain the same token. For any sequence of shift vectors $\{\Delta_u\}$, token-matched leave-one-out averaging is defined as

$$\mathrm{LOO}_t(\{\Delta_u\})(v) = \begin{cases} \dfrac{1}{|U_t(v)|} \displaystyle\sum_{u \in U_t(v)} \Delta_u(v), & |U_t(v)| > 0, \\ 0, & |U_t(v)| = 0. \end{cases} \tag{11}$$

The recurring directions are $\hat{d}_t^{\mathrm{doc}} = \mathrm{LOO}_t(\{\Delta_u^{\mathrm{ctrl}}\})$ and $\hat{d}_t^{\mathrm{model}} = \mathrm{LOO}_t(\{\Delta_u^{\mathrm{model}}\})$. Token matching uses vocabulary identity rather than top-K rank, and both earlier and later positions in the completed response can contribute. If a token appears only at the current position, its recurring-direction estimate is zero.

Coefficient fitting. The recurring direction determines which token preferences recur across states, while Eq. 6 fits how strongly that direction appears at the current state. We use $\varepsilon = 10^{-6}$. If $\hat{d}_t^a = 0$, then $c_t^a = \hat{b}_t^a = 0$. For the document-induced shift, the recurring direction is estimated from the control-document shifts, while c_t^{doc} is fitted to the original Δ_t^{doc}. For the model-induced shift, the recurring direction is estimated from model-induced shifts at the other states, while c_t^{model} is fitted to $\Delta_t^{\mathrm{model}}$ at the current state.

Selective correction masking. The document-side correction is masked at the sampled student token y_t, when it belongs to $\mathcal{V}_t$, and at the teacher-preferred token

$$v_t^T = \arg\max_{v \in \mathcal{V}_t} \ell_T(D, s_t)(v).$$

We define

$$m_t(v) = \begin{cases} 0, & v \in \{y_t, v_t^T\} \cap \mathcal{V}_t, \\ 1, & v \in \mathcal{V}_t \setminus \{y_t, v_t^T\}. \end{cases} \tag{12}$$

Because the document-side correction is zero at both tokens, selective correction masking preserves their score difference under document-side subtraction. The model-induced correction remains active at these tokens, and normalization can still change their final probabilities. Table 8 evaluates the effect of removing selective correction masking.

Target construction and optimization. The document- and model-induced corrections are combined as in Eq. 7 and subtracted from the document-conditioned teacher scores. The corrected scores are normalized on $\mathcal{V}_t$ to obtain $\tilde{p}_t^T$, and the current student's probabilities are renormalized on the same support to obtain p_t^S. The student is then updated using Eq. 8. During the update, the sampled response, top-K supports, masks, and corrected teacher targets are held fixed. Gradients flow only through the current student.

B.2 Effect of Correction on Distillation

This section derives Eq. 9, which characterizes how the correction changes the local student gradient. We compare different correction strengths at the same student state while holding the sampled response, top-K support, correction mask, and estimated correction fixed. The analysis therefore describes the conditional token-wise update and does not differentiate through the rollout or target construction.

Local gradient difference. For brevity, let $\ell_t^T = \ell_T(D, s_t)|_{\mathcal{V}_t}$ and $\hat{B}_t = m_t \odot \hat{b}_t^{\text{doc}} + \hat{b}_t^{\text{model}}$. Let z_t be the current student's logits on $\mathcal{V}_t$, with $p_t^S = \text{softmax}_{\mathcal{V}_t}(z_t)$. The corrected teacher target is

$$\tilde{p}_t^T(\alpha) = \text{softmax}_{\mathcal{V}_t}\left(\ell_t^T - \alpha \hat{B}_t\right).$$

Relative to uncorrected OPCD, $\alpha = 0$, the corrected scores satisfy

$$\tilde{\ell}_t(\alpha) = \tilde{\ell}_t(0) - \alpha \hat{B}_t. \tag{13}$$

Let

$$Z_t(\alpha) = \sum_{v \in \mathcal{V}_t} \exp\Big(\ell_t^T(v) - \alpha \hat{B}_t(v)\Big)$$

and $C_t(\alpha) = \log Z_t(\alpha) - \log Z_t(0)$. After normalization,

$$\log \tilde{p}_t^T(v; \alpha) - \log \tilde{p}_t^T(v; 0) = -\alpha \hat{B}_t(v) - C_t(\alpha). \tag{14}$$

For $\mathcal{L}_t(\alpha) = \text{KL}(p_t^S \| \tilde{p}_t^T(\alpha))$, the loss difference becomes

$$\mathcal{L}_t(\alpha) - \mathcal{L}_t(0) = \alpha \mu_t + C_t(\alpha), \qquad \mu_t = \sum_{v \in \mathcal{V}_t} p_t^S(v) \hat{B}_t(v). \tag{15}$$

Because the teacher target and correction are fixed during the student update, $C_t(\alpha)$ is constant with respect to z_t. Using the softmax Jacobian gives

$$\frac{\partial \mu_t}{\partial z_t(j)} = p_t^S(j)\left(\hat{B}_t(j) - \mu_t\right). \tag{16}$$

Therefore,

$$\nabla_{z_t} \mathcal{L}_t(\alpha) - \nabla_{z_t} \mathcal{L}_t(0) = \alpha p_t^S \odot \left(\hat{B}_t - \mu_t \mathbf{1}\right), \tag{17}$$

which recovers Eq. 9.

Equivalently, for a score-space gradient-descent step $\Delta z_t(\alpha) = -\eta \nabla_{z_t} \mathcal{L}_t(\alpha)$,

$$\Delta z_t(\alpha) - \Delta z_t(0) = -\eta \alpha p_t^S \odot \left(\hat{B}_t - \mu_t \mathbf{1}\right). \tag{18}$$

Relative to uncorrected OPCD, positive correction strength suppresses tokens with $\hat{B}_t(v) > \mu_t$ and relatively favors those with $\hat{B}_t(v) < \mu_t$. The effect is weighted by the student's probability, so corrections on low-probability tokens have little direct influence on the local update. Because $\hat{B}_t$ is centered by μ_t, any token-independent offset cancels. These statements concern relative local updates and do not imply an absolute change in a token's logit after a parameter update.

Table 3: **Retrieval versus internalization.** SRA-Bench gains are relative to the corresponding initial student on the 1,546 test queries. Additional tokens include all measured inference prompt processing, including the selection call for LLM Selection.

Setting	Gain (pp)	Additional tokens/query
Qwen3-1.7B student		
BM25 top-1	-0.62	4,385.5
LLM Selection	$+1.51$	6,469.1
BM25 top-3	$+0.47$	13,581.7
DCSD (ours)	$+22.9$	0
Qwen3-4B student		
BM25 top-1	$+4.64$	4,524.0
LLM Selection	$+10.21$	5,713.3
BM25 top-3	$+5.01$	13,701.1
DCSD (ours)	$+9.4$	0

C Measurement details for main figures

C.1 Retrieval cost and skill gains

Figure 1 compares skill gains and additional inference prompt tokens under retrieval and DCSD internalization. Table 3 records the plotted measurements.

Compared systems. Retrieval uses the initial Qwen3-1.7B or Qwen3-4B student. DCSD uses the corresponding student trained with a Qwen3-8B or frozen Qwen3-4B teacher, respectively, and answers without skill documents. The retrieval pool contains 26,065 documents, consisting of the 286 training skills, the 153 gold skills of the two unevaluated SRA-Bench domains (BigCodeBench and ToolQA), and 25,626 general skill documents from the SRA-Bench library (Su et al., 2026). Documents are ranked by BM25. BM25 top-1 and top-3 insert the one or three highest-ranked documents, and LLM Selection lets the student pick one document from the top 50 candidates before answering. All settings answer the same 1,546 test queries.

Plotted metrics. The vertical axis is the change in mean accuracy across the four SRA-Bench domains from the initial student (30.60% for 1.7B and 51.56% for 4B). DCSD gains use the rounded scores in Table 1. The horizontal axis is the increase in server-recorded prompt tokens per recorded query over document-free inference. LLM Selection includes both selection and answer-generation calls. This measures prompt processing, excluding retrieval computation, latency, and training cost. Positive costs use a logarithmic axis, with zero-cost DCSD points shown separately.

C.2 Skill-gain and cross-domain metrics

Figure 2 reports accuracy changes from each initial student after OPCD, separating skill acquisition from changes on other tasks.

Skill-query groups. A query belongs to the *training-skills* group if any annotated skill appeared during internalization, and to the *held-out-skills* group otherwise. The groups contain 1,367 and 179 queries, respectively, all excluded from training. Accuracy is pooled within each group rather than averaged equally across domains, because held-out-skill queries occur only in TheoremQA (165) and CHAMP (14).

Other-task metrics. The cross-domain entries use IFEval prompt-level strict accuracy, Multi-IF third-turn accuracy, and separate ACPBench MCQ and Bool scores. Table 1 gives the absolute scores. The general-capability group covers MMLU, GSM8K, HumanEval, and ARC-C, with protocols in Appendix A.1 and scores in Table 6. Figure 2 displays individual cross-domain metrics, whereas Figure 4 averages submetrics within each benchmark and then takes the mean over the six benchmarks, following Table 1.

C.3 STATE-SHARED DRIFT ACROSS TEACHER–STUDENT PAIRS

Figure 3(a) compares the magnitude of state-shared drift across teacher–student pairs. Here we describe the measurement used for this comparison.

Fixed responses. We evaluate all model pairs on the same 96 responses, comprising 39,321 generation states, also used for the source-geometry analysis in Appendix C.4. The responses are generated once by Qwen3-4B without a skill document for 96 SRA-Bench test queries (24 per domain; temperature 1.0; maximum 1,024 tokens) and then held fixed across all settings. This provides a controlled comparison on the same response states rather than confounding the measurement with different rollouts. For each setting, a frozen initial student S_0 (Qwen3-0.6B, 1.7B, or 4B) is paired with a document-conditioned teacher that is either a larger model of the same family (Qwen3-8B, 4B, or 1.7B; six heterogeneous pairs) or the student itself (three self-distillation pairs). At each state s_t, $\mathcal{V}_t$ is the top-K support of the corresponding student, with $K = 256$.

Measured shift. At each state, we measure the full document-conditioned teacher–student shift

$$\Delta_t^{\mathrm{sum}} = \left[\ell_T(D, s_t) - \ell_{S_0}(s_t)\right]\big|_{\mathcal{V}_t} = \Delta_t^{\mathrm{doc}} + \Delta_t^{\mathrm{model}}, \tag{19}$$

corresponding to the full teaching signal in Eq. 2. Under self-distillation, $T = S_0$, so $\Delta_t^{\mathrm{model}} = 0$ and the measured shift reduces to Δ_t^{doc}. As in Appendix C.4, each shift is centered on its support before comparison.

State-shared drift estimate. For this diagnostic, we treat Δ_t^{sum} as a single source and apply the same recurring-component estimator used in DCSD. The recurring direction is estimated by the token-matched leave-one-out average

$$\hat{d}_t = \mathrm{LOO}_t\left(\{\Delta_u^{\mathrm{sum}}\}_{u \neq t}\right), \tag{20}$$

and its state-specific coefficient is obtained by the least-squares projection in Eq. 6, yielding

$$\hat{b}_t = c_t \hat{d}_t. \tag{21}$$

Unlike DCSD training, this measurement does not use control documents, because its purpose is to quantify the recurring component of the full teaching signal rather than to separate the document- and model-induced sources.

Response-level magnitude. For each response y, we aggregate $\hat{b}_t$ across generation states using the same token-matched procedure as Eq. 23. For each retained token $v \in \mathcal{V}_2$, we average over the positions whose support contains v and then recenter over $\mathcal{V}_2$, producing a response-level drift vector

$$\bar{b}_y \in \mathbb{R}^{|\mathcal{V}_2|}.$$

We summarize its magnitude by the root-mean-square entry

$$m_y = \frac{\|\bar{b}_y\|_2}{\sqrt{|\mathcal{V}_2|}}. \tag{22}$$

Figure 3(a) reports the median m_y over the 96 responses, together with the interquartile and 5th–95th percentile ranges.

Results. Table 4 reports the median response-level drift magnitude for each teacher–student pair. Heterogeneous pairs consistently exhibit substantially larger state-shared drift than self-distillation. For example, the 1.7B student yields values of 2.03–2.20 with heterogeneous teachers, compared with 0.70 under self-distillation, while the 0.6B student yields 2.44–2.77 versus 0.44. The magnitude is not monotonic in teacher size, as the 1.7B and 4B teachers produce drift comparable to or larger than the 8B teacher for the 0.6B student.

Under self-distillation, the model-induced shift vanishes. The larger drift observed for heterogeneous pairs therefore reflects effects associated with the teacher–student model gap, either through the model-induced component itself or through changes in how a different teacher responds to the skill document. Appendix C.4 separates these two sources for Qwen3-8B $\rightarrow$ Qwen3-1.7B.

Table 4: **State-shared drift across teacher–student pairs.** Median response-level drift magnitude m_y over the 96 shared responses. Self denotes self-distillation, where the teacher and initial student are the same model.

	Teacher			
Student	Qwen3-8B	Qwen3-4B	Qwen3-1.7B	Self
Qwen3-0.6B	2.44	2.77	2.77	0.44
Qwen3-1.7B	2.03	2.20	–	0.70
Qwen3-4B	1.95	–	–	0.78

C.4 Document and model shift geometry

Figure 3(b) compares the two sources' mean directions within each response. We also test whether position-wise shifts align more closely within the same response than across responses.

Data and centering. We score 96 fixed responses (39,321 generation states) with a Qwen3-8B teacher and frozen Qwen3-1.7B reference student. At each generation position t, let S_t denote the top-K support of the frozen reference student, with $K = |S_t| = 256$. We center each shift on this support, $\Delta^a_{t,\mathrm{ctr}} = \Delta^a_t - \frac{1}{K}\sum_{v\in S_t}\Delta^a_t(v)\,\mathbf{1}$, since a constant offset cancels under the softmax but can distort cosine similarities. The subscript ctr denotes this centering, not a DCSD correction.

Response-level vectors. For a response y, let $n_y(v) = |\{t : v \in S_t\}|$ denote the number of positions whose support contains token v. We define the retained vocabulary as $\mathcal{V}_2 = \{v : n_y(v) \geq 2\}$, keeping only tokens that appear in at least two supports. For each source a and each retained token, we average the centered shift $\Delta^a_{t,\mathrm{ctr}}(v)$ over the positions containing it and then recenter over the retained vocabulary:

$$h^a(v) = \frac{1}{n_y(v)}\sum_{t:v\in S_t}\Delta^a_{t,\mathrm{ctr}}(v), \qquad v \in \mathcal{V}_2,$$
$$\bar{\Delta}^a = h^a - \frac{1}{|\mathcal{V}_2|}\sum_{v\in\mathcal{V}_2} h^a(v)\,\mathbf{1}. \tag{23}$$

Here $h^a(v)$ is the mean centered shift of token v, and $h^a \in \mathbb{R}^{|\mathcal{V}_2|}$ collects these means over the retained vocabulary. $\bar{\Delta}^a$ is then the centered response-level vector for source a. Since both sources use the same supports and $\mathcal{V}_2$, their combined shift satisfies $\bar{\Delta}^{\mathrm{sum}} = \bar{\Delta}^{\mathrm{doc}} + \bar{\Delta}^{\mathrm{model}}$.

Plot coordinates. For each response, let θ_y be the angle between $\bar{\Delta}^{\mathrm{doc}}$ and $\bar{\Delta}^{\mathrm{model}}$, and let $\rho_y = \|\bar{\Delta}^{\mathrm{model}}\|/\|\bar{\Delta}^{\mathrm{doc}}\|$. In Figure 3(b), each response's two vectors are rotated and scaled so that the document vector becomes $(1, 0)$ and the model vector becomes $(\rho_y\cos\theta_y, \rho_y\sin\theta_y)$, preserving their angle and norm ratio. The combined vector is their sum. The arrows use the median angle (116°) and median norm ratio (2.9) across responses and therefore do not depict a single measured response. These centered response-level vectors are used only to describe source geometry.

Within-response recurrence. We compare each centered shift with the three directions in Table 5, matching token identities. The same-response mean excludes the current position. Cosine similarities are averaged within each response, then summarized by their median across responses. The same- versus other-response mean comparison supports response-specific estimation. Averaging itself reduces position-specific variation, so comparison with random position pairs alone is insufficient.

Table 5: **Within-response recurrence of document- and model-induced shifts.** Values are medians of response-level mean cosine similarities.

Comparison	**Document-induced**	**Model-induced**
Mean of other positions in the same response	0.40	0.46
Randomly paired positions in the same response	0.18	0.17
Mean from another response	0.19	0.27

D ADDITIONAL EXPERIMENTS

D.1 GENERAL-CAPABILITY EVALUATION

Table 6 evaluates how well general knowledge, mathematical reasoning, code generation, and commonsense reasoning are preserved after skill internalization. DCSD has the highest four-benchmark average among the evaluated internalized methods in both model pairings, although the margins are small in the heterogeneous setting and individual benchmarks favor different methods. Its heterogeneous average remains close to the initial student. Under self-distillation, DCSD reduces the average less than the other internalization methods, but a gap from the initial student remains, most notably on HumanEval. These results complement the cross-domain instruction- and rule-following evaluation without implying that every general capability is preserved.

Table 6: **General-capability evaluation after skill internalization.** Metrics are accuracy for MMLU and ARC-C, strict exact match for GSM8K, and pass@1 for HumanEval. Avg. is their arithmetic mean. **Bold** denotes the best distillation method per column within each pairing.

Method	MMLU	GSM8K	HumanEval	ARC-C	**Avg.**
Heterogeneous: Qwen3-8B → Qwen3-1.7B					
Initial student	55.7	68.6	40.2	39.9	51.1
Off-policy CD	53.5	66.3	38.4	37.5	48.9
OPD[2]	**56.0**	64.4	39.0	**41.4**	50.2
OPCD	55.7	**70.7**	39.6	38.7	51.2
DCSD (ours)	55.6	68.2	**40.9**	40.3	**51.3**
Self-distillation: Qwen3-4B → Qwen3-4B					
Initial student	68.3	85.6	72.0	50.7	69.2
Off-policy CD	68.3	85.3	64.6	51.9	67.5
OPD[2]	68.0	80.7	62.2	**53.3**	66.1
OPCD	68.1	**85.9**	65.2	50.0	67.3
DCSD (ours)	**68.4**	85.4	**67.7**	50.9	**68.1**

D.2 CORRECTION STRENGTH SWEEP

Table 7 provides the per-benchmark results underlying Figure 4, showing how correction strength affects skill acquisition and cross-domain performance for both DCSD and Model Correction. For DCSD, all evaluated positive strengths improve the cross-domain average over uncorrected OPCD, although the improvement varies across benchmarks. Up to $\alpha = 1$, SRA-Bench performance remains close to or above OPCD. Increasing the strength to $\alpha = 1.5$ yields the highest reported cross-domain average but lowers SRA-Bench accuracy from 53.5 to 50.0 relative to $\alpha = 1$. Thus, stronger correction can further improve cross-domain performance at the cost of some in-domain skill gains.

Model Correction shows a consistent complementary pattern. At every matched positive strength, correcting only the model-induced source yields higher SRA-Bench accuracy than DCSD, but a lower cross-domain average. For example, at $\alpha = 1$, Model Correction reaches 55.9 on SRA-Bench compared with 53.5 for DCSD, while its cross-domain average is lower (39.5 vs. 40.8). The higher skill score is consistent with the document-side correction also attenuating signal useful for skill acquisition. The lower cross-domain recovery shows that model-only correction does not match the recovery obtained when both sources are considered. Together with the source geometry in Figure 3, this matched-strength comparison shows that single- versus two-source correction changes the trade-off between skill gains and cross-domain retention.

Table 7: **Correction-strength sweep under heterogeneous distillation** (Qwen3-8B → Qwen3-1.7B). DCSD corrects both sources, whereas Model Correction uses only the model-induced source. $\alpha = 0$ corresponds to OPCD. Metrics follow Table 1. **Bold** denotes the best correction setting per column.

	In-domain	**Cross-domain**										
α	SRA-Bench	ACPBench		COLLIE	ManyIFEval	Multi-IF			KOR-Bench	IFEval		**Avg.**
		MCQ	Bool			t1	t2	t3		Prompt	Inst.	
Initial student	30.6	47.0	59.2	26.4	28.0	68.8	48.5	31.1	33.5	68.4	76.0	43.8
0 (OPCD)	53.1	38.7	50.7	22.8	24.2	56.5	38.5	22.5	32.7	55.6	68.2	37.6
DCSD (both sources)												
0.5	52.4	39.9	54.4	23.2	23.8	59.6	39.4	23.0	35.8	60.4	70.5	39.3
1	53.5	40.7	56.9	22.5	23.3	62.9	41.5	25.8	**38.6**	63.6	72.7	40.8
1.5	50.0	37.9	**59.6**	**24.8**	**24.5**	**63.9**	**42.9**	**26.9**	38.5	**64.3**	**73.3**	**41.7**
Model Correction (model-induced source only)												
0.5	54.2	40.2	53.4	22.0	23.9	59.8	39.6	23.0	35.4	59.7	70.1	39.0
1	**55.9**	**43.8**	54.6	22.1	23.2	60.5	39.8	23.8	35.9	60.1	70.4	39.5
1.5	50.4	40.4	57.8	23.1	23.0	61.7	42.0	24.0	38.0	60.6	70.9	40.3

D.3 Correction ablation

Table 8 further isolates the effects of the two source-specific corrections and selective correction masking. Recall from Eq. 7 that DCSD combines the two correction terms as

$$\hat{B}_t = m_t \odot \hat{b}_t^{\text{doc}} + \hat{b}_t^{\text{model}}.$$

Document Correction retains only the document-induced term, $\hat{B}_t = m_t \odot \hat{b}_t^{\text{doc}}$, whereas Model Correction retains only the model-induced term, $\hat{B}_t = \hat{b}_t^{\text{model}}$. Without selective masking instead sets $m_t = \mathbf{1}$, giving $\hat{B}_t = \hat{b}_t^{\text{doc}} + \hat{b}_t^{\text{model}}$. In all cases, the corrected teacher target remains $\tilde{\ell}_t(\alpha) = \ell_T(D, s_t)|_{\mathcal{V}_t} - \alpha\hat{B}_t$. For a matched-strength comparison, all correction variants use $\alpha = 1$.

The two source-specific variants exhibit different trade-offs. Model Correction achieves higher SRA-Bench accuracy than DCSD (55.9 vs. 53.5), but a lower cross-domain average (39.5 vs. 40.8). The higher skill score is consistent with document-side correction attenuating some recurring signal that is useful for skill acquisition. Document Correction also remains below DCSD in cross-domain performance (39.4 vs. 40.8). Neither single-source variant therefore matches the cross-domain recovery obtained by correcting both sources in this setting. These results support accounting for recurring contributions from both sources.

Selective correction masking is also important for preserving useful guidance. Without selective masking reduces SRA-Bench accuracy from 53.5 to 48.9 and the cross-domain average from 40.8 to 38.6 relative to DCSD at the same correction strength. This degradation is consistent with the motivation for selective correction masking because recurring document-induced preferences can overlap with useful procedural guidance, and unrestricted correction can attenuate signal needed for skill acquisition. Together, these ablations support combining corrections from both sources with selective correction masking in DCSD.

Table 8: **DCSD ablations under heterogeneous distillation** (Qwen3-8B → Qwen3-1.7B). Document Correction and Model Correction retain only the document-induced and model-induced terms of the correction in Eq. 7, respectively, while Without selective masking sets $m_t = \mathbf{1}$. All correction variants use $\alpha = 1$. Metrics and cross-domain aggregation follow Table 1. Bold denotes the best internalized setting per column.

Method	In-domain	Cross-domain										
	SRA-Bench	ACPBench		COLLIE	ManyIFEval	Multi-IF			KOR-Bench	IFEval		Avg.
		MCQ	Bool			t1	t2	t3		Prompt	Inst.	
Heterogeneous: Qwen3-8B → Qwen3-1.7B												
OPCD (no correction)	53.1	38.7	50.7	22.8	**24.2**	56.5	38.5	22.5	32.7	55.6	68.2	37.6
Document Correction	51.7	38.9	**58.8**	**22.9**	23.4	60.5	41.1	25.6	33.4	60.8	70.4	39.4
Model Correction	**55.9**	**43.8**	54.6	22.1	23.2	60.5	39.8	23.8	35.9	60.1	70.4	39.5
Without selective masking	48.9	39.6	56.1	22.1	23.8	59.5	38.2	22.4	34.3	58.2	69.2	38.6
DCSD	53.5	40.7	56.9	22.5	23.3	**62.9**	**41.5**	**25.8**	**38.6**	**63.6**	**72.7**	**40.8**

D.4 Training cost

Table 9 reports the training cost of each distillation method. DCSD incurs a higher per-step cost than OPCD because estimating the correction requires additional teacher scoring at each generation state. Since its selected checkpoint arrives earlier (step 500 versus 600), the cost to the selected checkpoint is 15.0 GPU-hours, compared with 7.5 for OPCD, 19.2 for off-policy CD (including teacher response generation), and 32.7 for OPD[2]. DCSD thus roughly doubles the training cost of OPCD in this setting while remaining below the other distillation baselines.

Table 9: **Training cost under heterogeneous distillation** (Qwen3-8B → Qwen3-1.7B) on a single H200 GPU. We report median step time, teacher-scoring time within each step, and GPU-hours to the checkpoint selected by early stopping.

Method	s / step	teacher scoring (s)	GPU h to selected ckpt
Off-policy CD	24.1	12.8	19.2
OPD[2]	97.6	43.6	32.7
OPCD	44.3	14.1	7.5
DCSD (ours)	108.2	60.3	15.0

D.5 Extending DCSD to On-Policy Distillation

The benefit of correcting state-shared drift extends beyond document-conditioned skill distillation. We find that the model-side correction of DCSD also improves cross-domain retention in heterogeneous on-policy distillation even when no document is provided. To examine this setting, we consider plain on-policy distillation (*OPD*), where the student is trained on its own samples to match the document-free teacher distribution. Its target is

$$\ell_T(\emptyset) = \ell_{S_0} + \Delta^{\text{model}},$$

so the teaching signal contains the model-induced change but no document-induced component. Nevertheless, OPD still reduces cross-domain performance, showing that such degradation can also arise when optimizing the student toward a different teacher through the on-policy distillation objective, independently of document conditioning.

We therefore apply the model-side correction of DCSD directly to OPD. The corrected target is

$$\ell_T(\emptyset) - \alpha\, \hat{b}^{\text{model}},$$

where $\hat{b}^{\text{model}}$ is estimated using the same leave-one-out procedure as in Section 5. For this comparison, we use $\alpha = 1$, with the remaining training and evaluation settings following Table 1.

Table 10: **DCSD under document-free on-policy distillation.** Results for heterogeneous Qwen3-8B → Qwen3-1.7B distillation without document conditioning. DCSD reduces to model-side correction in this setting. Metrics and cross-domain aggregation follow Table 1. **Bold** denotes the best distilled model per column.

Method	In-domain	Cross-domain											Avg.
	SRA-Bench	ACPBench		COLLIE	ManyIFEval	Multi-IF			KOR-Bench	IFEval			
		MCQ	Bool			t1	t2	t3		Prompt	Inst.		
Heterogeneous: Qwen3-8B → Qwen3-1.7B													
Initial student	30.6	47.0	59.2	26.4	28.0	68.8	48.5	31.1	33.5	68.4	76.0		43.8
OPD	42.6	43.0	**61.9**	24.5	24.0	62.9	45.1	27.5	36.3	**62.7**	72.2		41.6
DCSD	**43.3**	**47.0**	61.0	**25.4**	**24.7**	**63.8**	**45.7**	**28.0**	**41.3**	62.5	**72.5**		**43.1**

Table 10 shows that OPD increases SRA-Bench accuracy from 30.6 to 42.6 while reducing the cross-domain average from 43.8 to 41.6. Since no document is used, this degradation occurs in a setting where the teacher target differs from the initial student only through the model-induced component. Applying model correction recovers 1.5 of the 2.2 lost cross-domain points, raising the average to 43.1, while preserving the in-domain improvement (43.3 versus 42.6).

These results extend DCSD from OPCD-based skill internalization to plain heterogeneous on-policy distillation. Both OPCD and plain OPD minimize the same on-policy reverse-KL objective over student-generated responses, differing in whether the teacher distribution is conditioned on a skill document. In OPCD, DCSD corrects state-shared drift arising from both document- and model-induced changes, whereas in document-free OPD it naturally reduces to correcting only the model-induced component. The improvement in cross-domain retention in both settings shows that the correction principle applies more broadly to heterogeneous on-policy distillation.

E Additional related work

Agent memory and skill internalization. External memory supports skill reuse without updating model parameters. Voyager stores executable skills, Reflexion retains verbal feedback, and ExpeL extracts insights from experience (Wang et al., 2023a; Shinn et al., 2023; Zhao et al., 2024). Agent Workflow Memory, Self-Discover, and Agent Memory Distillation preserve reusable workflows or textual experience (Wang et al., 2025; Zhou et al., 2024; Kim et al., 2026), while context-compression methods encode documents into compact activations or memory slots (Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2024). These approaches retain guidance for inference, whereas document-free skill internalization incorporates the behavior into model parameters.

Other methods move skill knowledge into parameters through different supervision. SkillC, Skill0.5, and AUSO use skill contrasts or outcome-based learning, while TMEM performs online LoRA updates within an episode (Lin et al., 2026a; Zhu et al., 2026; Lin et al., 2026b; Ren et al., 2026). Reports of skill shadowing and skill-induced failures study interference when skill libraries or documents remain available at inference (Song et al., 2026; Dong et al., 2026). DCSD studies a different locus of interference: supervision that is absorbed into a shared student and affects other tasks after the skill documents are removed.

Distillation and prediction correction. Distillation methods differ in which states and teacher signals drive learning. DAgger and ImitKD motivate learner-induced states, DistiLLM and DistiLLM-2 modify objectives and data sources, and Direct-OPD transfers an RL-induced policy change rather than the teacher's full final policy (Ross et al., 2011; Lin et al., 2020; Ko et al., 2024; 2025; Feng et al., 2026). DCSD keeps OPCD's rollout protocol and reverse-KL objective fixed, and instead modifies the target according to structure that recurs across generation states.

Prediction correction also appears at inference. Contextual calibration and domain-conditional PMI compensate for answer preferences (Zhao et al., 2021; Holtzman et al., 2021), while Contrastive Decoding, context-aware decoding, and DoLa contrast predictions across models, contexts, or layers (Li et al., 2023; Shi et al., 2024; Chuang et al., 2024). Logit standardization adjusts teacher and

student score scales before distillation (Sun et al., 2024). DCSD uses recurrence across generation states to identify shared training-time supervision, estimates document-induced and model-induced directions separately, and controls how strongly their fitted contributions enter the distillation target.

Interference after specialization. Continual-learning methods limit interference through parameter-importance constraints, old-model supervision, or orthogonal task subspaces, as in elastic weight consolidation, Learning without Forgetting, and O-LoRA (Kirkpatrick et al., 2017; Li & Hoiem, 2016; Wang et al., 2023b). Empirical studies likewise show that specialization can alter behavior outside the target task (Luo et al., 2023; Kumar et al., 2022; Qi et al., 2024), motivating preservation of non-target capabilities. CaMOPD addresses this goal through multiple teachers, alternating updates, and gap-based sample selection (Chen et al., 2026). DCSD instead intervenes directly in the supervision used for internalization and does not train on the held-out cross-domain benchmarks.

Shared structure can also carry useful task information. Function vectors and task vectors encode reusable task information in internal activations (Todd et al., 2024; Hendel et al., 2023). DCSD's shared directions instead describe token-aligned log-probability shifts that recur within a response. This distinction matters because the recurring supervision also carries useful skill information. In DCSD, correction strength and single- versus two-source correction determine how much of this supervision is retained, tracing the observed trade-off between skill gains and cross-domain retention.