PAY FOR ROLLOUTS ONLY WHEN NEEDED: COMPUTE-AWARE ROUTING FOR LANGUAGE MODEL DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation addresses exposure bias by training a student language model on its own generated responses, with supervision from a teacher. However, generating and scoring these responses, specially with frontier models, is more expensive compared to off-policy distillation using cached teacher responses. We investigate whether selectively allocating on-policy supervision can reduce distillation cost while maintaining student quality. In samples where a student favors one teacher-supported choice while ignoring others (i.e., anchored collapse), off-policy training with cached teacher probabilities can help recover those alternatives. In contrast, when a student assigns substantial probability to choices the teacher considers unlikely (i.e., support escape), the student's trajectory may not match the cached prefixes in off-policy distillation, making fresh rollouts via on-policy distillation helpful. We formalize this in our proposed compute-aware routing (CAR) approach for distillation, choosing among three actions: skipping an update, cached off-policy and on-policy distillation. CAR compares student and teacher probabilities on cached prefixes to predict action utilities that account for compute costs. In particular, we compute directional support statistics from cached prefixes to distinguish anchored collapse, for which off-policy may suffice, from support escape, for which fresh on-policy supervision may provide additional value. We also provide a routing-regret bound conditional on utility-estimation accuracy and an idealized support-corridor bound linking escape probability to loss-surrogate discrepancy under a common trajectory distribution. In a one-seed GSM8K study, CAR achieved 62.40% accuracy versus 62.24% for vanilla on-policy distillation while routing 71.02% of samples to fresh on-policy supervision, indicative of cost savings.

1 INTRODUCTION

Large language models (LLMs) increasingly power online assistants in diverse domains such as shopping, recruiting, finance, healthcare, education and legal services. However, serving frontier LLMs at scale is costly. Knowledge distillation offers a way to reduce inference costs by transferring a capable teacher model's behavior to a smaller student tailored to a specific application. In autoregressive language modeling, standard sequence-level knowledge distillation (SeqKD) trains the student on responses generated by the teacher Kim & Rush (2016). Because these responses and their token-level teacher distributions can be cached before training, SeqKD supports efficient parallel teacher-forced updates. Its central weakness is distribution shift: at inference time, the student conditions on its own previous outputs rather than the teacher's prefixes.

On-policy distillation (OPD) addresses this mismatch by generating trajectories from the current student and asking the teacher to supervise the states the student actually visits Lin et al. (2020); Agarwal et al. (2024); Gu et al. (2024). This can improve long-horizon generation, but it replaces a relatively efficient supervised training loop with a costly online step containing student autoregressive rollout, teacher scoring, and student optimization. Recent methods improve the objectives used on these trajectories Jia et al. (2026); Jin et al. (2026), tolerate stale trajectories Rang et al. (2026); Kang et al. (2026), or intervene when a student trajectory strays Wang et al. (2026). They largely retain the assumption that student rollouts should be collected broadly or according to a fixed systems schedule.

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

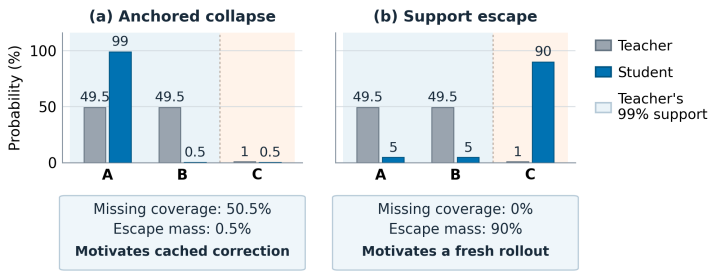


Figure 1: Anchored collapse (a) concentrates student probability on one teacher-supported outcome while neglecting another; support escape (b) places substantial probability outside teacher support. Cached corrections can mitigate anchored collapse while fresh roll-outs can mitigate support escape.

We ask a different question: *"For which prompts is a fresh student rollout worth its incremental compute over cached teacher-forced distillation?"*. A naive answer is to go on-policy whenever teacher and student token probabilities differ. This confounds two distinct cases. First, the student may have collapsed onto one valid teacher-supported pathway while underweighting other teacher modes. At a cached prefix exhibiting this pattern, forward-KL supervision provides a signal to increase underweighted teacher-supported alternatives without first collecting a fresh student rollout. Second, the student may allocate substantial probability to tokens outside the teacher's plausible support. Such choices can lead to downstream prefixes absent from the cache, making a fresh rollout valuable. We refer to these cases as *anchored collapse* and *support escape*, respectively. Figure 1 shows an illustrative toy example for anchored collapse and support escape. Consider learning a teacher distribution (0.495, 0.495, 0.01) over outcomes A, B, and C (Figure 1). Its 99% effective support is $\{A, B\}$. A student with probabilities (0.99, 0.005, 0.005) exhibits anchored collapse: it concentrates on A while neglecting the teacher-supported alternative B. Cached forward-KL supervision can restore coverage at the observed prefix without a fresh rollout. In contrast, a student with probabilities (0.05, 0.05, 0.90) exhibits support escape: 90% of its mass lies outside the teacher's effective support. In autoregressive generation, such choices can lead to prefixes absent from the cache, motivating fresh on-policy supervision. This suggests that on-policy distillation may not be needed for all samples, and it may be possible to reduce the additional cost versus cached forward-KL supervision.

We formalize this distinction through a weighted bidirectional trajectory divergence and cost-penalized action selection. The forward direction measures omitted teacher mass and can be estimated from cached teacher trajectories. The reverse direction measures student behavior unsupported by the teacher and intrinsically depends on student trajectories. This yields a compute-aware oracle that chooses among skip, cached distillation, and on-policy distillation according to expected policy-gap reduction minus a compute-cost penalty.

Our main contributions can be summarized as follows.
• We formulate language-model distillation as cost-aware reduction of a weighted bidirectional trajectory-policy gap.
• We derive the one-step oracle routing rule and define the signed learning advantage of on-policy updates and the corresponding routing regret.
• We propose cheap directional support features, computed using only a teacher-forced student pass on cached teacher trajectories, that distinguish anchored mode collapse from support escape.
• We prove a generic utility-estimation regret bound and an idealized support-corridor bound relating loss-surrogate discrepancy under a common trajectory distribution to cumulative escape probability.
• We report positive results from a one-seed GSM8K study where CAR achieves accuracy close to fully on-policy distillation while routing only 71.02% samples to on-policy distillation.

2 Related Work

Offline sequence-level distillation. SeqKD trains a student on teacher-generated sequences using maximum likelihood Kim & Rush (2016). More general sequence-level formulations connect

distillation to forward, reverse, and symmetric f-divergences Wen et al. (2023). Offline teacher trajectories are inexpensive to reuse but expose the student only to teacher prefixes.

Imitation learning and on-policy distillation. The distribution-shift problem is closely related to behavioral cloning and interactive imitation learning Ross et al. (2011). ImitKD mixes teacher and student trajectories Lin et al. (2020). GKD trains on student-generated sequences and permits different local divergences between teacher and student distributions Agarwal et al. (2024). MiniLLM directly optimizes a reverse-KL sequence objective with an on-policy estimator Gu et al. (2024).

Adaptive on-policy objectives. AOPD routes token-level supervision between an advantage-weighted policy-gradient update and localized forward-KL correction, while remaining on-policy throughout Jia et al. (2026). Entropy-aware OPD adds forward KL when teacher uncertainty is high to preserve diversity Jin et al. (2026). Adaptive KL objectives also combine forward and reverse directions according to distributional statistics Wu et al. (2024). Our method is orthogonal: the primary routed object is the *trajectory source and associated compute*, not only the divergence used after a trajectory has been collected.

Efficient and stale on-policy training. Near-Policy Distillation decouples generation and training and filters highly off-distribution samples Rang et al. (2026). AsyncOPD studies stale rollout caches and shows that KL direction affects robustness to staleness Kang et al. (2026). MOTAB monitors an ongoing student trajectory and backtracks when it crosses a safety boundary Wang et al. (2026). Our goal is to predict whether a fresh rollout is needed *before* paying for it, using statistics available from cached teacher prefixes.

3 Problem Formulation

Let $x \sim \mathcal{D}$ be a prompt, π_T a fixed teacher policy, and π_θ a student policy on a shared finite vocabulary. Both policies generate until an end-of-sequence token is emitted or a common finite maximum generation length is reached. A trajectory $y = (y_1, \ldots, y_{|y|})$ includes the end-of-sequence token when emitted and induces states $s_t = (x, y_{<t})$ and trajectory distributions

$$P_T^x(y) = \prod_{t=1}^{|y|} \pi_T(y_t \mid s_t), \qquad P_\theta^x(y) = \prod_{t=1}^{|y|} \pi_\theta(y_t \mid s_t).$$

All trajectory sums run over the generated positions. We assume that the displayed KL divergences and their expectations over prompts are finite.

3.1 A bidirectional trajectory-policy gap

We define

$$\mathcal{G}_\alpha(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\alpha F_x(\theta) + (1-\alpha) R_x(\theta)\right], \tag{1}$$

where $\alpha \in [0, 1]$ and

$$F_x(\theta) = D_{\mathrm{KL}}(P_T^x \| P_\theta^x), \tag{2}$$

$$R_x(\theta) = D_{\mathrm{KL}}(P_\theta^x \| P_T^x). \tag{3}$$

The forward component F_x penalizes teacher-supported trajectories omitted by the student. The reverse component R_x penalizes trajectories generated by the student that receive little teacher support. Neither direction alone expresses both desired behaviors: reverse KL can accept collapse onto one teacher mode, while forward KL evaluated only on teacher trajectories does not directly observe states reached after student deviations.

Proposition 1 (Autoregressive decomposition)**.** *For any prompt x for which the divergences are finite,*

$$F_x(\theta) = \mathbb{E}_{y\sim P_T^x} \sum_t D_{\mathrm{KL}}\left(\pi_T(\cdot \mid s_t) \| \pi_\theta(\cdot \mid s_t)\right), \tag{4}$$

$$R_x(\theta) = \mathbb{E}_{y\sim P_\theta^x} \sum_t D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid s_t) \| \pi_T(\cdot \mid s_t)\right). \tag{5}$$

We have provided the proof in Appendix A. Equation (4) can be estimated using cached teacher trajectories and logits. Equation (5) instead averages over student-visited prefixes and can be estimated directly using trajectories sampled from the current student. This asymmetry motivates our routing question: when is the additional learning value of supervising student-visited prefixes worth the cost of collecting and scoring a fresh rollout?

3.2 ACTIONS AND COMPUTE COSTS

At each training decision, the learner chooses an action $a \in \mathcal{A} = \{\text{SKIP}, \text{CACHE}, \text{ON}\}$. SKIP leaves the parameters unchanged. CACHE performs a teacher-forced update on a cached teacher trajectory, using sequence cross-entropy or cached top-K forward KL. ON generates a fresh student trajectory, evaluates it with the teacher, and applies an on-policy objective such as GKD, OPD, or AOPD.

Let c_a denote the expected compute cost of executing action a, excluding the cost of selecting the action. We consider the regime $c_{\text{SKIP}} = 0 < c_{\text{CACHE}} < c_{\text{ON}}$. Costs are scalar quantities expressed in a common unit, such as GPU-seconds or FLOPs, or explicitly specified compute proxies. Let $\widehat{g}_a(x;\theta)$ be the stochastic update direction produced by action a and

$$\theta_a^+ = \theta - \eta \widehat{g}_a(x;\theta).$$

Here $\eta > 0$ is the step size. Define its expected policy-gap reduction

$$\Delta_a(x,\theta) = \mathbb{E}\left[\mathcal{G}_\alpha(\theta) - \mathcal{G}_\alpha(\theta_a^+)\right]. \tag{6}$$

The expectation is over the randomness in the update, conditional on the current prompt x and parameters θ. For SKIP, we set $\widehat{g}_{\text{SKIP}} = 0$, giving $\Delta_{\text{SKIP}} = 0$. A more expensive action is worthwhile only if its additional learning benefit justifies its cost. For a compute price $\lambda \geq 0$, define utility

$$U_a(x,\theta) = \Delta_a(x,\theta) - \lambda c_a. \tag{7}$$

The price λ specifies how much policy-gap reduction compensates for one unit of compute. At $\lambda = 0$, actions are compared solely by their expected learning benefit; increasing λ penalizes expensive actions more strongly. The price controls the benefit–cost trade-off but does not enforce a hard compute budget by itself. Selecting an action can also require computation, such as evaluating the signals used by the router. We treat this selection cost as common to the three choices while deciding. Subtracting the same cost penalty from every utility does not change their ranking, so we omit it from Equation (7). It must be included in total training compute, even when the selected action is SKIP.

Proposition 2 (Myopic compute-aware oracle)**.** *For a fixed compute price $\lambda \geq 0$, an optimal one-step action satisfies*

$$a^*(x,\theta) \in \arg\max_{a \in \mathcal{A}} U_a(x,\theta).$$

Ties are resolved by a fixed rule. A fresh rollout is strictly preferred to cached distillation if and only if

$$\Delta_{\text{ON}}(x,\theta) - \Delta_{\text{CACHE}}(x,\theta) > \lambda(c_{\text{ON}} - c_{\text{CACHE}}). \tag{8}$$

For ON *to be the unique optimal action, it must also beat skipping, requiring $U_{\text{ON}}(x,\theta) > 0$.*

The left-hand side of Equation (8) is the *signed learning advantage* of a fresh on-policy update over a cached update. A positive value indicates greater expected policy-gap reduction from the on-policy update; a negative value favors the cached update before accounting for compute costs. Our practical goal is to predict this quantity without first paying for the rollout.

Smoothness-based interpretation. If $\mathcal{G}_\alpha$ is L-smooth along update segments and $\mathbb{E}\|\widehat{g}_a\|^2 < \infty$,

$$\Delta_a \geq \eta \left\langle \nabla \mathcal{G}_\alpha, \mathbb{E}[\widehat{g}_a] \right\rangle - \frac{L\eta^2}{2} \mathbb{E}\|\widehat{g}_a\|^2. \tag{9}$$

The gradient is evaluated at θ, and expectations are conditional on x and θ. This lower bound highlights two factors: alignment of the expected update with the policy-gap gradient, and the update's second moment, which includes both magnitude and variability. Neither trajectory source is guaranteed to yield the larger policy-gap reduction, motivating action-specific utility prediction.

Missing coverage	Escape	Suggested action
Low	Low	Skip or sparse replay
High	Low	Cached forward-KL KD
Low	High	Fresh on-policy refresh
High	High	Fresh on-policy refresh

Table 1: Conceptual routing regimes.

4 SUPPORT-AWARE ROUTING

The exact utilities in Equation (7) are unavailable before taking actions. We construct a cheap feature representation from one student teacher-forced pass over cached teacher trajectories and logits.

4.1 DIRECTIONAL SUPPORT MISMATCH

For a distribution p and mass level $\rho \in (0, 1)$, let $\mathcal{N}_\rho(p)$ denote a smallest nucleus set whose cumulative probability is at least ρ, i.e., the fewest highest-probability tokens needed to reach mass ρ, with ties broken by a fixed token order. At a cached teacher state $s_t^T = (x, y_{<t}^T)$, define

$$A_T^t = \mathcal{N}_\rho(\pi_T(\cdot \mid s_t^T)), \tag{10}$$

$$A_S^t = \mathcal{N}_\rho(\pi_\theta(\cdot \mid s_t^T)). \tag{11}$$

We define two directional scores:

$$m_t = 1 - \sum_{v \in A_S^t} \pi_T(v \mid s_t^T), \tag{12}$$

$$e_t = 1 - \sum_{v \in A_T^t} \pi_\theta(v \mid s_t^T). \tag{13}$$

Throughout, m_t measures teacher probability mass not covered by the student's effective support, while e_t measures student mass outside the teacher's effective support. Here "support" means the nucleus-defined effective support, not the set of tokens with nonzero probability. Thus, escape does not necessarily indicate an invalid token. Even when teacher and student distributions coincide, $m_t = e_t$ equals the probability mass outside their common nucleus, which lies between 0 and $1 - \rho$. The product

$$a_t = m_t(1 - e_t) \tag{14}$$

is an *anchored-collapse score*: it is large when the student omits teacher alternatives but remains mostly inside teacher support. In contrast, high e_t indicates possible support escape and motivates observing a student rollout. We aggregate token-level scores using a robust upper-tail statistic rather than a raw maximum. Let $\mathrm{TailMean}_q$ average the largest $\lceil qn \rceil$ scores for a sequence of $n \geq 1$ completion positions, where $q \in (0, 1]$. We define

$$C_x = \mathrm{TailMean}_q(\{m_t\}_t), \tag{15}$$

$$E_x = \mathrm{TailMean}_q(\{e_t\}_t). \tag{16}$$

Thus, C_x summarizes missing teacher coverage, and E_x summarizes escape. We use these features jointly instead of aggregating the anchored-collapse score a_t. Other candidate features include cached forward KL, teacher and student entropy, top-token agreement, student loss on the cached response, sequence length, time since last on-policy refresh, and changes in these quantities across checkpoints.

4.2 INTERPRETING THE ROUTING REGIMES

Table 1 summarizes the intended behavior. These regimes motivate the routing features, rather than determine the optimal action: the final choice depends on predicted learning benefit and compute cost. Consider again the toy example from Section 1. For $p_T = (0.495, 0.495, 0.01)$, a student $p_S = (0.99, 0.005, 0.005)$ is collapsed but anchored. A cached forward-KL update directly increases the missing second teacher mode. A student $p_S = (0.05, 0.05, 0.90)$ instead places most mass outside teacher support. Teacher forcing can correct its probabilities at the cached state, but does not observe

Algorithm 1 Compute-Aware Routed Distillation

Require: Teacher π_T, student π_θ, prompt set $\mathcal{D}$, cached teacher traces/logits $\mathcal{C}$, compute price λ

for training step $k = 1, \ldots, K$ **do**
  Sample prompt minibatch B
  Run one teacher-forced student pass on cached traces in B
  Compute cheap features $\phi(x, \theta)$, including C_x and E_x
  Route each prompt to SKIP, CACHE, or ON using $\widehat{U}_a$
  Apply cached forward-KL/SeqKD updates to B_{CACHE}
  Generate fresh student trajectories for B_{ON}
  Score fresh trajectories with the teacher and apply OPD/AOPD/GKD
  if calibration is enabled and step k is a probe step **then**
    Estimate counterfactual action utilities on a small probe subset
    Update/calibrate utility predictors $f_{a,\psi}$
  end if
end for

the downstream states reached after selecting the third token; this is where on-policy collection is expected to add value. For $\rho = 0.99$, the teacher nucleus is $\{1, 2\}$. The anchored student's nucleus is $\{1\}$, giving $(m_t, e_t) = (0.505, 0.005)$. The escaping student's nucleus contains all three tokens, giving $(m_t, e_t) = (0, 0.9)$. Thus, the first student has substantial missing coverage but little escape, whereas the second places most of its probability outside the teacher nucleus. All probabilities are positive, so both local KL directions are finite.

4.3 UTILITY PREDICTION AND ROUTING

We use the cheap features $\phi(x, \theta)$ to predict action utilities:

$$\widehat{U}_a(x, \theta) = f_{a,\psi}(\phi(x, \theta)), \tag{17}$$

and select

$$\widehat{a}(x, \theta) \in \arg\max_{a \in \mathcal{A}} \widehat{U}_a(x, \theta). \tag{18}$$

We again use a fixed rule to break ties. The utility predictors can use fixed coefficients or be calibrated from data. Our "Prior-Guided" CAR experiments use fixed coefficients (details in Section 6, with features recomputed under the current student throughout training. An extension is to calibrate the predictors using periodic counterfactual probes: starting from the same model and optimizer state, apply alternative actions to separate copies and estimate their policy-gap reductions on a held-out probe set. Between probes, routing uses predicted utilities without executing the alternative actions. Compute comparisons must include the cost of these probes when calibration is enabled.

Let $\widehat{F}_x$ denote the cached per-token forward-KL feature used by the router. Its length normalization and cache approximation distinguish it from the trajectory divergence F_x; the implementation is specified in the experimental details. A transparent parameterization is

$$\widehat{U}_{\text{CACHE}} = \beta_0 + \beta_F \widehat{F}_x + \beta_C C_x - \beta_E E_x - \lambda c_{\text{CACHE}}, \tag{19}$$

$$\widehat{U}_{\text{ON}} = \gamma_0 + \gamma_E E_x + \gamma_{FE} \widehat{F}_x E_x - \lambda c_{\text{ON}}, \tag{20}$$

$$\widehat{U}_{\text{SKIP}} = 0, \tag{21}$$

with coefficients fixed in the evaluated Prior-Guided CAR variant. The form encodes the prior that cached correction is useful for missing coverage with low escape, whereas fresh supervision is useful for high escape. These scores are heuristic utility estimates; optional probe-based calibration could fit their coefficients to measured action values.

5 THEORETICAL ANALYSIS

Routing regret from utility-estimation error Let a_k^* be the oracle action and $\widehat{a}_k$ the predicted action at decision k. We define the cumulative routing regret as:

$$\mathrm{Regret}(K) = \sum_{k=1}^{K} \left[U_{a_k^*}(x_k, \theta_k) - U_{\widehat{a}_k}(x_k, \theta_k) \right]. \tag{22}$$

Theorem 1 (Plug-in routing regret)**.** *Suppose for decision k the utility estimates satisfy $\max_{a \in \mathcal{A}} |\widehat{U}_a - U_a| \le \epsilon_k$, then*

$$U_{a_k^*} - U_{\widehat{a}_k} \le 2\epsilon_k, \qquad \mathrm{Regret}(K) \le 2 \sum_{k=1}^{K} \epsilon_k.$$

The proof is in Appendix B. The bound compares the predicted action with the best action at the same current parameters θ_k. It does not compare the final models from two different training trajectories. It shows that accurate utility estimates imply small local decision regret, but does not establish that our support-based predictors satisfy the assumed error bound. In particular, the fixed coefficients used by Prior-Guided CAR are not accompanied by a guarantee on ϵ_k. The theorem applies to the pointwise utility-maximizing router. Budget-Constrained CAR imposes batch-level action quotas and would require a separate comparison against an oracle subject to those same quotas (details in Section 6).

An idealized support-corridor bound We fix a prompt x and student parameters θ, and let T denote the finite maximum generation length. We bound the probability of leaving a teacher-supported corridor and the expected discrepancy between two loss functions that agree inside it. For each covered state s_t, let $A_T(s_t)$ be a teacher-supported action set. Let $\mathcal{C}_x$ be the set of trajectories that remain within $A_T(s_t)$ at every generated position. This corridor includes all such trajectories, not merely those present in the cache. Assume that for every covered prefix at depth t,

$$\Pr_{v \sim \pi_\theta(\cdot|s_t)}[v \notin A_T(s_t)] \le \bar{e}_t. \tag{23}$$

Here $\bar{e}_t \in [0, 1]$ must bound escape at every prefix reachable without previously leaving the corridor.

Proposition 3 (Escape-probability bound)**.** *The probability that a student trajectory exits the teacher-supported corridor by horizon T satisfies*

$$\Pr_{y \sim P_\theta^x}[y \notin \mathcal{C}_x] \le 1 - \prod_{t=1}^{T}(1 - \bar{e}_t) \le \sum_{t=1}^{T} \bar{e}_t. \tag{24}$$

Let ℓ and $\widetilde{\ell}$ be integrable loss functions on the same trajectory space. Suppose they agree inside $\mathcal{C}_x$ and satisfy $|\ell(y) - \widetilde{\ell}(y)| \le L_{\max}$ outside it, for a finite $L_{\max} \ge 0$. Then

$$\left| \mathbb{E}_{Y \sim P_\theta^x}[\ell(Y)] - \mathbb{E}_{Y \sim P_\theta^x}[\widetilde{\ell}(Y)] \right| \le L_{\max} \Pr_{Y \sim P_\theta^x}[Y \notin \mathcal{C}_x] \le L_{\max} \sum_{t=1}^{T} \bar{e}_t.$$

The proof is in Appendix C. Our assumptions are deliberately strong. In practice, we observe e_t only on cached teacher prefixes, not every state in the corridor, so these measurements do not establish the required uniform escape bounds. Moreover, the loss bound compares two functions under the same student trajectory distribution; it does not directly bound the learning advantage $\Delta_{\text{ON}} - \Delta_{\text{CACHE}}$. That connection would require additional assumptions linking the losses to the resulting parameter updates. Proposition 3 should therefore be interpreted as motivation for the estimator and as a target for empirical validation, not as a distribution-free guarantee for arbitrary language-model trajectories. Token-level support is a tractable surrogate for latent trajectory modes; Appendix D.1 describes a mode-level extension that quotients out harmless lexical and reasoning-path variation.

6 EXPERIMENTS

We evaluate whether selective on-policy supervision retains the accuracy of fully on-policy distillation, how fixed-prior and calibrated routers allocate learning, and whether reduced rollout use translates to lower observed training time.

Method	Accuracy (%)	Parse rate (%)	Limit hits
Qwen3.5 9B teacher	89.61	99.32	9
Qwen3.5 0.8B base	38.29	84.84	29
SeqKD/SFT	53.45	98.94	14
All-cache forward-KL	58.68	98.56	19
Fresh OPD	62.24	93.18	89
Budget-Constrained CAR	60.35	94.92	63
Prior-Guided CAR	62.40	94.01	76
Calibrated CAR, smaller probe, $\lambda = 0.1$	51.02	93.10	90
Calibrated CAR, larger probe, $\lambda = 0.01$	62.55	95.30	60
Calibrated CAR, larger probe, $\lambda = 0.1$	54.66	98.48	20

Table 2: GSM8K test results for validation-selected checkpoints. Parse rate measures structured-answer extraction; limit hits count responses reaching 512 tokens. Results use one training seed and do not establish significance for small differences. Smaller-probe checkpoint 20 precedes updates governed by the fitted router.

Models and data: we use a frozen Qwen3.5 9B teacher and a capacity-constrained Qwen3.5 0.8B student, which share a tokenizer, chat-template family, and base architecture, to avoids introducing vocabulary or model-family mismatch into the distillation comparison. We use OpenMathInstruct-1, derived from GSM8K and MATH training data. After deduplication and removal of overlaps with the official GSM8K test set, we obtain 11,553 training and 1,444 validation prompts. A frozen Qwen3.5 9B teacher generates trajectories; the 8,959 training trajectories with correct answers form the SeqKD and CAR caches. Trained models are chosen based on the best exact-answer accuracy on the validation set. Appendix E.4 details the checkpoint selection and evaluation protocol.

Baselines and routed variants: For our experiments, we evaluate the Qwen3.5 9B teacher and base Qwen3.5 0.8B under the same test protocol. The baselines and routing variants are listed below.

- **SeqKD/SFT:** trained on cached teacher responses with completion-only maximum likelihood.
- **All-cache forward-KL:** this off-policy baseline trains on cached teacher trajectories by matching the teacher's stored top-K token probabilities and residual tail mass with forward KL.
- **Fresh OPD:** every example receives a fresh student rollout, live teacher scoring, and equal-weighted forward- and reverse-KL supervision.
- **Budget-Constrained CAR:** support features rank examples within fixed budgets of up to 25% ON and 50% CACHE; the remainder is SKIP. Budgets constrain frequencies, not which examples receive each action.
- **Prior-Guided CAR:** the cost-penalized rule in Algorithm 1 uses fixed coefficients that favor CACHE for anchored coverage error and ON for support escape. Recomputed student features yield an adaptive action mix without learnt coefficients or action quotas.
- **Probe-Calibrated CAR:** every 20 accumulation windows, including all-SKIP windows, cloned student/optimizer updates estimate action benefits on held-out prompts and refit the router. We reserve 128 training prompts for scoring and compare probes with one trial example per rank and 16 scoring prompts ($\lambda = 0.1$) against four per rank and 32 prompts ($\lambda \in \{0.01, 0.1\}$), using eight ranks. The initialized prior governs routing until the first probe.

Results: Table 2 reports official GSM8K test accuracy. All distilled 0.8B models improve over the 38.29% initialization. SeqKD reaches 53.45%, while all-cache forward-KL reaches 58.68%, narrowing the gap to fresh OPD from 8.79 to 3.56 percentage points. Cached distribution matching is therefore a stronger offline comparator in this study.

Prior-Guided CAR reaches 62.40% and larger-probe calibrated CAR at $\lambda = 0.01$ reaches 62.55%, versus 62.24% for fresh OPD. The calibrated result differs from Prior-Guided CAR by two answers and from OPD by four; these are similar observed scores, not established improvements. Budget-Constrained CAR reaches 60.35% with a substantially smaller on-policy allocation. Calibration does not consistently improve quality: the smaller-probe run selects checkpoint 20 at 51.02%, whereas larger-probe $\lambda = 0.1$ selects its final model at 54.66%. These outcomes show sensitivity to calibration settings and training trajectory, not a uniform benefit from fitted routing.

Variant	Checkpoint	ON (%)	CACHE (%)	SKIP (%)	Training H100 hours	Probe H100 hours
All-cache forward-KL	140	0.00	100.00	0.00	3.68	–
Fresh OPD	Final	100.00	0.00	0.00	27.90	–
Budget-Constrained CAR	100	24.94	50.00	25.06	8.97	–
Prior-Guided CAR	120	69.35	0.98	29.66	12.01	–
Calibrated CAR, smaller probe, $\lambda = 0.1$	20	13.08	0.07	86.85	4.32	3.98
Calibrated CAR, larger probe, $\lambda = 0.01$	120	68.97	31.03	0.00	11.19	22.82
Calibrated CAR, larger probe, $\lambda = 0.1$	Final	13.90	52.89	33.21	6.82	23.03

Table 3: Routing counts through each selected checkpoint. ON uses fresh student trajectories; CACHE uses cached off-policy forward KL. Counts reflect processed samples, not optimizer steps or matched training volumes. H100-hours measure training run, including continuations after checkpoint selection.

Compute cost: Table 3 reports H100-hours computed from wall-clock phase durations recorded in the training pod logs. For calibrated CAR, we partition the observed runtime into training and counterfactual probing. H100-hours equal each phase duration multiplied by the eight allocated H100 GPUs; they measure elapsed allocated compute rather than utilization-integrated GPU activity. Larger-probe CAR at $\lambda = 0.01$ routes 68.97% of examples to ON and 31.03% to CACHE, reducing fresh student rollouts and live teacher calls by 31.03% relative to OPD. Excluding calibration, its observed training cost is 11.19 H100-hours versus 27.90 for OPD, a 59.9% reduction. However, counterfactual probing adds 22.82 H100-hours, bringing the combined cost to 34.01 H100-hours. Thus, routing substantially reduces the core on-policy training cost, but the current probing procedure more than offsets these savings.

7 DISCUSSION

We introduced compute-aware routing (CAR) to decide when fresh student rollouts justify their additional cost over cached off-policy distillation. Our formulation uses a weighted bidirectional trajectory-policy gap to capture both omitted teacher modes and student behavior with little teacher support. It yields an oracle rule based on cost-penalized marginal gap reduction and motivates directional support features that distinguish anchored collapse from support escape using a student forward pass on cached teacher prefixes. In our experiments, Prior-Guided CAR and larger-probe $\lambda = 0.01$ attain accuracy close to OPD, while all-cache forward-KL provides a strong off-policy baseline. The results also reveal a distinction between answer correctness and output regularity: SeqKD and all-cache forward-KL have parse rates above 98% with only 14 and 19 token-limit hits, yet lower accuracy than OPD. Larger-probe $\lambda = 0.1$ exhibits a similar pattern, producing reliable output structure without matching OPD's accuracy (insights in Appendix D.2).These observations suggest complementary roles for cached and fresh supervision, and motivate the following directions for future work.

Calibration reliability and compute-price robustness. The smaller-probe run becomes SKIP-dominated only after its first calibration event. Noisy or miscalibrated policy-gap estimates are a plausible explanation for suppressing useful updates, not an experimentally isolated cause. Its selected checkpoint 20 precedes SKIP-domination, and the poor quality is attributable to incomplete training. Larger probes sustain learning but need some tuning to improve quality. Future work can focus on making routing less brittle to λ and calibration noise through uncertainty-aware action selection, more stable calibration, and safeguards against prolonged skipping. More efficient probing will also help bring down the probe time which is very expensive in the current version. Adaptive probe cadence and cost models grounded in measured GPU time could improve the calibration-compute tradeoff.

Beyond one-step look-ahead. The present oracle values immediate policy-gap reduction, which need not predict long-horizon answer accuracy. Multi-step counterfactual updates or a finite-horizon decision rule could capture the exploratory value of fresh trajectories and the retention benefits of cached replay. Outcome- or semantic-mode signals could further recognize distinct valid solution paths rather than requiring token-level agreement.

AI USE STATEMENT

Generative AI tools were used to adapt the manuscript to the ICLR 2027 template and to edit the abstract for clarity and consistency with the reported results. They were not used to generate experimental data or reported numerical results as part of this formatting and editing pass. The authors take responsibility for the final content of the submission, including all AI-assisted text and formatting changes.

REFERENCES

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

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

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level. *arXiv preprint arXiv:2605.06387*, 2026.

Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjun Kang, Sanghyun Park, Donghoon Kim, Minjae Lee, Minseo Kim, Rishabh Tiwari, Yuchen Zeng, Hyung Il Koo, and Kangwook Lee. AsyncOPD: How stale can on-policy distillation be? *arXiv preprint arXiv:2606.24143*, 2026.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016.

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.

Miao Rang, Zhenni Bi, Hang Zhou, Kai Han, Xuechun Wang, An Xiao, Xinghao Chen, Yunhe Wang, and Hanting Chen. Near-policy: Accelerating on-policy distillation via asynchronous generation and selective packing. *arXiv preprint arXiv:2605.05940*, 2026.

Stéphane 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*, pp. 627–635, 2011.

Bing Wang, Shaotian Yan, Chen Shen, Kaiyuan Liu, Sinan Fan, Ximing Li, Rui Miao, Xiaosong Yuan, Zhanming Shen, and Jieping Ye. Backtracking when it strays: Mitigating dual exposure biases in LLM reasoning distillation. *arXiv preprint arXiv:2605.19433*, 2026.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation. *arXiv preprint arXiv:2307.15190*, 2023.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking kullback–leibler divergence in knowledge distillation for large language models. *arXiv preprint arXiv:2404.02657*, 2024.

A AUTOREGRESSIVE DECOMPOSITION

Proposition 1 states that, for any prompt x for which the divergences are finite,

$$F_x(\theta) = \mathbb{E}_{y\sim P_T^x} \sum_t D_{\mathrm{KL}}\left(\pi_T(\cdot \mid s_t) \| \pi_\theta(\cdot \mid s_t)\right), \tag{25}$$

$$R_x(\theta) = \mathbb{E}_{y\sim P_\theta^x} \sum_t D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid s_t) \| \pi_T(\cdot \mid s_t)\right). \tag{26}$$

Proof. By the definition of KL divergence and autoregressive factorization,

$$F_x(\theta) = \mathbb{E}_{Y\sim P_T^x} \sum_{t=1}^{|Y|} \log \frac{\pi_T(Y_t \mid s_t)}{\pi_\theta(Y_t \mid s_t)}.$$

At each active prefix s_t, the next token follows $\pi_T(\cdot \mid s_t)$, so

$$\begin{aligned} D_{\mathrm{KL}}(\pi_T(\cdot \mid s_t)\|\pi_\theta(\cdot \mid s_t)) &= \sum_v \pi_T(v \mid s_t) \log \frac{\pi_T(v \mid s_t)}{\pi_\theta(v \mid s_t)} \\ &= \mathbb{E}_{V\sim\pi_T(\cdot|s_t)} \left[\log \frac{\pi_T(V \mid s_t)}{\pi_\theta(V \mid s_t)}\right]. \end{aligned}$$

Applying the tower property at each generated position and summing over t gives Equation (25). The bounded trajectory length justifies exchanging expectation and summation. Reversing the log-ratio and taking expectations under P_θ^x gives Equation (26). □

B PLUG-IN ROUTING REGRET

Theorem 1 (plug-in routing regret) states that: suppose for decision k the utility estimates satisfy

$$\max_{a\in\mathcal{A}} |\widehat{U}_a - U_a| \le \epsilon_k.$$

Then

$$U_{a_k^*} - U_{\widehat{a}_k} \le 2\epsilon_k, \qquad \mathrm{Regret}(K) \le 2\sum_{k=1}^{K} \epsilon_k.$$

Proof. By optimality under the predicted utilities, $\widehat{U}_{\widehat{a}_k} \ge \widehat{U}_{a_k^*}$. Therefore

$$U_{a_k^*} - U_{\widehat{a}_k} \le (U_{a_k^*} - \widehat{U}_{a_k^*}) + (\widehat{U}_{\widehat{a}_k} - U_{\widehat{a}_k}) \le 2\epsilon_k.$$

Summing proves the result. □

C ESCAPE PROBABILITY BOUND

The escape-probability bound (proposition 3) states that the probability that a student trajectory exits the teacher-supported corridor by horizon T satisfies

$$\Pr_{y\sim P_\theta^x}[y \notin \mathcal{C}_x] \le 1 - \prod_{t=1}^{T}(1-\bar{e}_t) \le \sum_{t=1}^{T} \bar{e}_t. \tag{27}$$

Let ℓ and $\widetilde{\ell}$ be integrable loss functions on the same trajectory space. Suppose they agree inside $\mathcal{C}_x$ and satisfy $|\ell(y) - \widetilde{\ell}(y)| \le L_{\max}$ outside it, for a finite $L_{\max} \ge 0$. Then

$$\left|\mathbb{E}_{Y\sim P_\theta^x}[\ell(Y)] - \mathbb{E}_{Y\sim P_\theta^x}[\widetilde{\ell}(Y)]\right| \le L_{\max} \Pr_{Y\sim P_\theta^x}[Y \notin \mathcal{C}_x] \le L_{\max} \sum_{t=1}^{T} \bar{e}_t.$$

Proof. Let S_t be the event that no escape occurs through position $\min\{t, |Y|\}$. Then $\Pr(S_0) = 1$ and

$$\Pr(S_t) \ge (1-\bar{e}_t)\Pr(S_{t-1}),$$

since active trajectories have conditional escape probability at most $\bar{e}_t$, and trajectories already terminated inside the corridor cannot subsequently escape. Iterating gives $\Pr(S_T) \ge \prod_{t=1}^{T}(1-\bar{e}_t)$. Taking complements and using $1 - \prod_t(1-\bar{e}_t) \le \sum_t \bar{e}_t$ proves the escape bound.

For the loss bound, with expectations under P_θ^x,

$$\begin{aligned} \left|\mathbb{E}[\ell(Y)] - \mathbb{E}[\widetilde{\ell}(Y)]\right| &\le \mathbb{E}[|\ell(Y) - \widetilde{\ell}(Y)|] \\ &\le L_{\max} \Pr[Y \notin \mathcal{C}_x]. \end{aligned}$$

The last inequality follows because the losses agree inside the corridor and differ by at most $L_{\max}$ outside it. □

D Additional Analysis

D.1 From token support to semantic modes

Token-level support is a tractable surrogate for latent trajectory modes. Let $z = g(x, y)$ map a trajectory to an outcome or semantic mode, inducing distributions $Q_T(z \mid x)$ and $Q_\theta(z \mid x)$. The ideal objective would replace Equation (1) with

$$\mathcal{G}_\alpha^{\text{mode}} = \mathbb{E}_x \left[\alpha D_{\text{KL}}(Q_T \| Q_\theta) + (1 - \alpha) D_{\text{KL}}(Q_\theta \| Q_T) \right]. \tag{28}$$

This quotients out harmless lexical and reasoning-path variation. Because g is generally unknown, our primary method operates on token distributions, while experiments estimate mode-level gaps using answer normalization, verifier labels, and clustering of multiple sampled rationales.

D.2 Output diagnostics

Table 2 separates answer quality from formatting. SeqKD has the highest parse rate and fewest token-limit hits among distilled models, but trails all-cache forward-KL and fresh OPD in accuracy. OPD improves accuracy by 8.79 percentage points over SeqKD despite reducing parse rate by 5.76 points and increasing token-limit hits from 14 to 89; its advantage is not explained by better parseability.

Larger-probe $\lambda = 0.1$ reaches a 98.48% parse rate with 20 token-limit hits, but only 54.66% accuracy. Larger-probe $\lambda = 0.01$ instead reaches 62.55% accuracy with a 95.30% parse rate and 60 hits. Reliable structure and termination therefore do not imply better mathematical reasoning.

E Experimental Details

E.1 Training configuration

All methods use the same frozen Qwen3.5 9B teacher, Qwen3.5 0.8B initialization, math prompt template, and seed. SeqKD trains for one epoch with completion-only negative log likelihood, a learning rate of 2×10^{-5}, cosine decay, an effective batch size of 16, and a maximum sequence length of 2,048. It runs on one H100.

Fresh OPD trains for one epoch with a 50/50 mixture of forward and reverse KL, top-$K = 40$ distribution matching, a learning rate of 7×10^{-6}, and cosine decay after 10 warmup steps. Each update samples a fresh student trajectory with temperature 0.6, top-p 0.9, and at most 256 generated tokens. OPD runs on eight H100s.

Budget-Constrained CAR and Prior-Guided CAR train for 140 optimizer steps on eight H100s with a learning rate of 7×10^{-6}, cosine decay with a 3% warmup ratio, an effective batch size of 64, and the OPD rollout distribution and 256-token limit. Fresh OPD instead uses 1,443 optimizer steps, an effective batch size of eight, and 10 fixed warmup steps. Calibrated CAR retains the eight-GPU, batch-64 setup but reserves training prompts for probe scoring; all-SKIP windows do not advance its optimizer or scheduler.

E.2 Teacher caches and routing

SeqKD, all-cache forward-KL, and CAR use the correctness-filtered cache of 8,959 teacher trajectories; calibrated CAR additionally reserves 128 training prompts for probe scoring. Fresh OPD schedules updates over all 11,553 training prompts. CAR stores the teacher's top-128 token probabilities and residual tail mass at each completion position. The ON objective uses top-40 teacher and student logits to limit the cost of live distribution matching.

Support sets use nucleus mass $\rho = 0.99$, and sequence-level support features average the largest 10% of token-level values. The cached gap $\widehat{F}_x$ is the mean, over completion positions, of the per-token forward KL in nats between the cached teacher distribution and the current student. The teacher's stored top-128 probabilities are evaluated exactly, while the remaining probability mass is represented as one residual-tail event.

Budget-Constrained CAR ranks examples using cached forward KL, anchored collapse, and support escape, then fills the configured action allocation. Prior-Guided CAR instead uses

$$\widehat{U}_{\text{CACHE}} = \widehat{F}_x + C_x - E_x - 0.1, \tag{29}$$

$$\widehat{U}_{\text{ON}} = E_x + \widehat{F}_x E_x - 0.3, \tag{30}$$

$$\widehat{U}_{\text{SKIP}} = 0. \tag{31}$$

These expressions correspond to a compute price of 0.1, normalized CACHE cost 1, normalized ON cost 3, and unit feature coefficients. The costs are operation-count proxies: CACHE requires a teacher-forced student update, whereas ON additionally requires autoregressive student rollout and live teacher scoring. The coefficients are fixed throughout training.

Table 3 reports full-run action percentages over real processed examples, excluding distributed padding. Calibrated routing percentages exclude held-out scoring and cloned trial updates; the table reports their probe H100-hours separately.

E.3 ROUTING DURING TRAINING

Table 3 counts live per-example decisions throughout training, excluding counterfactual trials. A CAR accumulation window contains up to 64 examples across eight ranks and can mix all three actions; an all-SKIP window advances the data stream but not the optimizer.

Budget-Constrained CAR uses all three actions at every optimizer step, with nearly constant allocation. Prior-Guided CAR uses ON at every step, but SKIP rises from 8.44% of examples in steps 1–20 to 40.66% in steps 101–120; CACHE remains sparse. Through checkpoint 120, 71.02% of its decisions are ON. As training progresses, the router assigns lower predicted marginal value to additional on-policy updates, increasing the SKIP share from 8.44% to 40.66%.

Smaller-probe $\lambda = 0.1$ starts off with ON-heavy, but after the first probe, switches entirely to skip. This is likely due to noise in the smaller probe resulting in the perception of no value from further training. Larger-probe $\lambda = 0.01$ starts overwhelmingly ON-heavy and develops a more balanced ON/CACHE mixture without SKIP; the transition fluctuates rather than proceeding monotonically. Larger-probe $\lambda = 0.1$ instead moves from an ON-heavy start through a SKIP-heavy phase toward predominantly cached updates. The shift from ON to CACHE with a higher λ is due to the increasing cost of ON.

E.4 CHECKPOINT SELECTION AND EVALUATION

We select checkpoints using only the 1,444-example OpenMathInstruct validation split. SeqKD checkpoint 550 is selected by teacher-forced validation loss; for cached forward-KL and CAR, cached-loss screening identifies candidates that are then compared by generated exact-answer accuracy. OPD endpoints and checkpoints are compared directly by generated accuracy. The selected models are evaluated on all 1,319 examples of the reserved official GSM8K test split using greedy decoding (temperature zero, top-p one), at most 512 generated tokens, seed 42, and thinking disabled. We report normalized exact-answer accuracy, count parsing failures as incorrect, and persist predictions and normalized answers before aggregation.