# SURGE: SPARSE UPDATE ROUTING VIA GATED ENTROPY FOR EFFICIENT ACTIVE DISTILLATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

On-policy distillation (OPD) is widely used to transfer capabilities from large teacher language models to smaller student models. However, supervising every token incurs substantial computational cost without distinguishing where guidance is truly needed and can induce unnecessary updates that disrupt existing student capabilities. To address these limitations, we introduce **SURGE**, an active distillation framework that provides selective control over student adaptation and knowledge transfer by using two separate gating mechanisms for updates. A token gate applies teacher supervision only at high-entropy positions, while a step gate forces updates to remain inside low-scoring generation steps. Additionally, a decoupled KL objective is used to apply updates selectively on such active tokens while retaining regularization toward a frozen reference student over the entire sequence. Across the evaluated reasoning, scientific knowledge, and hate speech tasks, SURGE achieves the highest reported accuracy among the compared baselines. When distilling Gemma4-31B into Gemma4-E4B, SURGE improves over the base student by a relative **8.61%** on average and achieves a mean recovery of **40.93%** of the initial teacher–student accuracy gap. Relative to OPD, estimated distillation FLOPs fall by **49.89%** on average, and the share of that gap SURGE closes is **66.60%** larger. Average response length falls by **45.21%**, while the mean format-failure rate falls by **24.78%** relative to OPD. Finally, we assess the generalization of our method by distilling Qwen3.5-122B-A10B into Qwen3.5-2B on these benchmarks, also noting improvements across accuracy and training FLOPs.

## 1 INTRODUCTION

Distillation has become a practical route for training smaller language models with the capabilities of larger teachers (Hinton et al., 2015). When adapted to specific tasks, smaller models can support practical agent deployments (Bhathal and Gupta, 2025; Saad-Falcon et al., 2026) at a cheaper cost. On-policy distillation (OPD) has been notably used in Qwen3's training regime, as well as in Thinking Machines Lab's work on reasoning and instruction following (Yang et al., 2025; Lu and Thinking Machines Lab, 2025). It trains on student-generated responses, allowing the teacher to guide learning on trajectories the student actually produces (Agarwal et al., 2024). Still, supervising every token incurs computation where additional guidance may have little value. In this work, we examine how to allocate teacher guidance while controlling changes to the student's existing behavior.

Selective knowledge transfer requires both useful feedback and controlled adaptation. Process supervision can be used to assess entire reasoning steps (Lightman et al., 2023). For example, Weaver (Saad-Falcon et al., 2025) combines weak verifiers for selecting candidate responses. Still, a poorly scored step may contain many predictable tokens. On the other hand, uncertainty over the next token can occur anywhere in an acceptable step, making both signals useless on their own. Combining step quality with local uncertainty provides a strong basis for selective teacher updates. Selection alone, however, does not isolate the effects of learning: updates to shared parameters can also alter predictions at unselected positions through back-propagation. Sparse supervision therefore motivates an explicit mechanism for regularizing updates throughout the entire response.

We introduce **SURGE** (**S**parse **U**pdate **R**outing via **G**ated **E**ntropy), a compute-efficient active distillation framework that performs selective updates against teacher logits while still regularizing



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.

the student model across the full response. The framework sparsely learns from student-generated rollouts scored by its teacher counterpart (Figure 1). Its core components are:

1. **Doubly-gated updates:** SURGE selects token positions based on both a macro-level and a micro-level gate. The first uses a frozen reference policy as a process reward model (PRM) for scoring coarser reasoning steps, while the latter gates updates strictly to high-entropy tokens at a fine-grained lens. Their intersection routes supervision exclusively to uncertain tokens within low-scoring generation steps. We note such gates leave us with largely distinct tokens to update on, and their combination outperforms either gate alone in our ablations.
2. **Decoupled KL objective.** Though our method restricts teacher-derived rewards to selected positions, we retain the reference-policy KL regularization objective across all generated tokens. This regularizes the student at unselected positions without using teacher log-probabilities for the sake of efficiency and stability. In our ablations, this objective achieves higher accuracy than variants that mask the full loss or omit reference regularization.

In our experiments, we evaluate SURGE by distilling Gemma4-31B into Gemma4-E4B across five benchmarks. Our method outperforms the evaluated Supervised Fine-tuning (SFT), Group-Relative Policy Optimization (GRPO), PRM, and OPD baselines in average task accuracy. Relative to OPD, SURGE recovers the teacher-student gap **66.6%** more on average, reporting higher accuracy and lower training FLOPs than OPD on all evaluated benchmarks. We detail our methods in Appendix B.

## 2 Background and Problem Formulation

**On-policy teacher supervision.** Let $x$ be a prompt and $y = (y_1, \ldots, y_T)$ a response sampled from a policy $\pi_{\theta_{\text{old}}}$. Write $h_t = (x, y_{<t})$ for its prefix. A frozen teacher computes log-probabilities conditioned on this same prefix. Comparing teacher and student log-probabilities supplies the distillation signal. The sequence-level reverse KL motivates this supervision (Gu et al., 2024):

$$D_{\text{KL}}(\pi_\theta(\cdot \mid x) \| \pi_T(\cdot \mid x)) = \mathbb{E}_{y \sim \pi_\theta}\left[\sum_t \log \frac{\pi_\theta(y_t \mid h_t)}{\pi_T(y_t \mid h_t)}\right]. \tag{1}$$

For fixed sampled prefixes, the immediate teacher signal is

$$r_t = \log \pi_T(y_t \mid h_t) - \log \pi_{\theta_{\text{old}}}(y_t \mid h_t). \tag{2}$$

A clipped policy-gradient update uses the probability ratio

$$\rho_t(\theta) = \frac{\pi_\theta(y_t \mid h_t)}{\pi_{\theta_{\text{old}}}(y_t \mid h_t)} \tag{3}$$

and the surrogate

$$\psi(\rho, a) = \min\left(\rho a,\ \text{clip}(\rho, 1 - \epsilon_c, 1 + \epsilon_c) a\right). \tag{4}$$

The advantage estimate $a$ is held fixed during differentiation (Schulman et al., 2017). We set the reward discount factor $\gamma$ to zero, so each token's update uses its immediate reward without accumulating rewards from later tokens. Sampled prefixes are treated as fixed training contexts. The resulting update is a local surrogate and does not recover the exact gradient of the sequence-level reverse KL in Equation 1. Gating and reward normalization further modify this surrogate, while clipping can affect the update when probability ratios leave the clipping interval. Incorporating rewards from later tokens would also require temporal credit assignment (Saligram et al., 2025).

**Step-level rewards and token entropy.** Process supervision supplies intermediate assessments of a response (Lightman et al., 2023). It partitions the response into steps $s_1, \ldots, s_K$. Let $k(t)$ be the step containing token $t$. The scorer returns $q_k \in [0, 1]$ for step $k$ given the prompt and its available step context. We use $q_k$ as a score, without assuming it is a calibrated probability of correctness.

On the other hand, the predictive entropy of a token is defined as:

$$H_t = -\sum_{v \in \mathcal{V}} \pi_{\theta_{\text{old}}}(v \mid h_t) \log \pi_{\theta_{\text{old}}}(v \mid h_t). \tag{5}$$

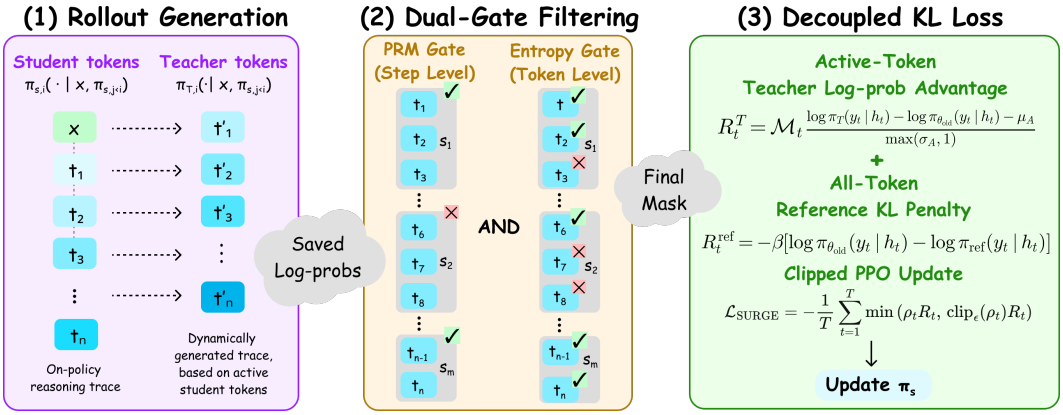


Figure 1: **The SURGE framework.** In (1), the student's tokens are sampled from the smaller policy and scored against the teacher policy. (2) A frozen student reference policy acting as a PRM grades sampled sequences at the step level, while an entropy gate determines the token-level update routing for uncertain tokens. (3) At the selected positions, the teacher and student's log-probabilities compute a reverse-KL feedback, with rewards only applying to the surviving tokens, while reference KL regularization is still applied on the full response. We further expand on the method on Section 3.1.

With natural logarithms, $H_t$ is measured in nats. It describes the uncertainty over possible continuations, including multiple valid alternatives being considered. As an analogy, JSON formatting boilerplate token positions contain extremely low entropy, but are not a measure of correctness.

**Related works.** Prior works in selective distillation use student entropy (SE-KD; Tavor et al., 2026), token importance and disagreement (TIP; Xu et al., 2026), teacher-verified token acceptance (SelecTKD; Huang et al., 2025), support compatibility (TA-OPD; Wang et al., 2026), and prompt-level teacher reliability (TGOPD; Zhang et al., 2026) to minimize updates. SURGE adds a step-quality context to token selection while retaining reference feedback at excluded positions. Here, we use dense OPD as the full-supervision counterpart to gauge the accuracy–compute trade-off of our work.

## 3 The SURGE Framework

SURGE uses three models: the trainable student $\pi_\theta$, a frozen reference student $\pi_{\text{ref}}$, and a frozen teacher $\pi_T$. The reference is a copy of the initial student and serves two roles: it supplies model-based process scores for the macro-level gate and anchors the KL regularization. We use PRM to denote this process-scoring role. Reusing the student for feedback is inspired by on-policy self-distillation (Zhao et al., 2026); in SURGE, the frozen reference supplies step-quality scores while the larger teacher supplies next-token log-probabilities for reverse-KL feedback. Figure 1 summarizes the framework.

SURGE combines step quality and token uncertainty to route teacher updates within each response. The step gate supplies context beyond token uncertainty, while response-wide reference regularization anchors the excluded positions.

### 3.1 Dual-Gated Distillation

SURGE constructs a binary routing mask $\mathcal{M}_{\text{SURGE}} \in \{0,1\}^T$ from two complementary filters:

$$\mathcal{M}_{\text{SURGE}}(t) = b_t \, M_{\text{PRM}}(t) \, M_{\text{Entropy}}(t), \tag{6}$$

where $b_t$ identifies valid response positions, excluding prompt and padding tokens.

The macro-level gate broadcasts the score of each reasoning step to its constituent tokens:

$$M_{\text{PRM}}(t) = \mathbb{I}\big(q_{k(t)} < \tau_{\text{prm}}\big), \tag{7}$$

where $q_{k(t)}$ is the process score for the step containing token $t$. For model-scored steps, the frozen reference supplies a two-class score from its next-token logits for + and −. The scoring prompt

includes the task input, preceding student steps, and the candidate step; no reference solution or gold answer is supplied. Task-specific checks can assign a score before model evaluation; Appendix E.1 specifies both paths. The micro-level gate selects positions with high predictive entropy:

$$M_{\text{Entropy}}(t) = \mathbb{I}(H_t > \tau_{\text{entropy}})\,. \tag{8}$$

Teacher supervision is applied at positions satisfying both criteria: their containing step receives a score below the PRM threshold, and the entropy of their next-token distribution exceeds the entropy threshold $\tau_{\text{entropy}}$. Training settings and gate thresholds are given in Appendix B.

### 3.2 Decoupled KL Objective

Let $\mathcal{S}_{\text{active}} = \{t : \mathcal{M}_{\text{SURGE}}(t) = 1\}$. At each selected position, the teacher computes log-probabilities over the next-token vocabulary, conditioned on the student-generated prefix. Comparing teacher and student log-probabilities under this same prefix provides our reverse-KL feedback. The macro-level gate separately uses a step-quality score from the PRM scorer or task-specific checks. For the sampled token $y_t$, the raw teacher reward is:

$$r_{\text{raw}}(t) = \log \pi_T(y_t \mid h_t) - \log \pi_{\theta_{\text{old}}}(y_t \mid h_t), \tag{9}$$

where $h_t = (x, y_{<t})$ and $\pi_{\theta_{\text{old}}}$ is the student policy that generated the token sequence. The log-probability difference in Equation 9 is the negative sampled reverse-KL contribution. Taking the norm of this value supplies the teacher component of the update:

$$\bar{r}_{\text{norm}}(t) = \frac{r_{\text{raw}}(t) - \mu_{\text{active}}}{s_{\text{active}}}, \qquad t \in \mathcal{S}_{\text{active}}. \tag{10}$$

Here $\mu_{\text{active}}$ is the mean of the active token rewards and $s_{\text{active}}$ is the normalization scale. These help pool active positions across the training batch. We define $\bar{r}_{\text{norm}}(t) = 0$ outside the active set.

Finally, the per-token reward combines the selected token updates with the $\pi_{\text{ref}}$ regularization penalty:

$$R_{\text{SURGE}}(t) = \mathcal{M}_{\text{SURGE}}(t)\,\bar{r}_{\text{norm}}(t) - \beta d_{\text{ref}}(t), \tag{11}$$

where:

$$d_{\text{ref}}(t) = \log \pi_{\theta_{\text{old}}}(y_t \mid h_t) - \log \pi_{\text{ref}}(y_t \mid h_t) \tag{12}$$

is a sampled reference log-ratio. We use $\beta = 0.05$.

Student parameters are optimized using the clipped PPO surrogate from Equation 4:

$$\mathcal{L}_{\text{SURGE}}(\theta) = -\frac{1}{Z}\sum_{i,t} b_{it}\,\psi(\rho_{it}(\theta), \text{sg}[R_{\text{SURGE},it}])\,, \qquad Z = \sum_{i,t} b_{it}, \tag{13}$$

where $i$ indexes sequences in the mini-batch.

We detail our optimization hyper-parameters in Appendix C, and further expand on teacher conditioning, reward normalization, and implementation methods for this loss on Appendix E.

## 4 Experiments

We perform ablations and benchmark SURGE's performance against other known methods, such as SFT, GRPO, our PRM baseline, and regular dense OPD, always grounding our results in baseline student and teacher model performance. First, we evaluate the accuracy and estimated compute against these methods, then examine whether the gates make complementary decisions and how adaptation changes the student's representations and copying behavior. Finally, two ablations remove reference supervision from excluded positions or from the entire response to test its role in learning with sparse teacher guidance. Within each benchmark, the compared runs use the same evaluation prompts, answer-extraction procedure, and generation limits. Baseline definitions and experimental details appear in Appendices B and C.

Table 1: **Average performance on benchmarks.** Arithmetic means on accuracy for distillation runs from Gemma4-31B Dense into Gemma4-E4B. Benchmarks evaluated were: GPQA Diamond, Competition Math, HateXplain, HateCheck and DynaHate. Training FLOPs saving averages treat OPD as the relative baseline, with negative values indicating additional compute. Appendix D defines our FLOPs estimation method, while Table 5 gives the full benchmark breakdown.

| Method / Model | Mean accuracy (%) | Mean distillation FLOPs | Mean FLOP saving (%) | Mean output length | Mean format failure (%) |
|---|---|---|---|---|---|
| Base Gemma4-E4B | 59.98 | – | – | 500.8 | 11.43 |
| Teacher Gemma4-31B | 73.56 | – | – | 285.3 | 7.70 |
| Standard SFT | 61.51 | $1.50 \times 10^{17}$ | 60.7 | 410.0 | 11.20 |
| GRPO ($G = 4$) | 62.38 | $6.65 \times 10^{17}$ | -74.6 | 385.0 | 10.73 |
| PRM Baseline | 62.15 | $\mathbf{8.74 \times 10^{16}}$ | **72.8** | 378.6 | 10.87 |
| OPD | 63.04 | $3.63 \times 10^{17}$ | 0.0 | 515.8 | 14.04 |
| **SURGE (Ours)** | **64.96** | $1.89 \times 10^{17}$ | 49.9 | **282.6** | **10.56** |

### 4.1 Accuracy and Compute-Efficiency Results

We compare accuracy and estimated FLOPs under dense OPD and SURGE. Response length and format failure assess output compactness and compliance with the requested format.

Across all five benchmarks with Gemma4, SURGE achieves the highest final model accuracy and has lower estimated FLOPs than OPD (Tables 1 and 5). The largest accuracy gains over OPD occur on Competition Math and HateXplain. These gains span reasoning, scientific knowledge, and hate speech detection capabilities.

FLOP savings relative to OPD range from **32.8%** to **74.9%** across tasks. The FLOP values reported are computed by tracking the computations during the corresponding distillation runs, including student generation and updates, reference evaluation, process scoring, and teacher computation. Appendix D defines these workloads and the distinction between token selection and computation. FLOP reductions do not directly measure wall-clock speedups.

Figure 2 shows the accuracy–compute trade-offs for selected Gemma and Qwen benchmarks. Within each displayed benchmark, SURGE lies on the observed Pareto frontier and achieves higher accuracy at lower estimated FLOPs than OPD.

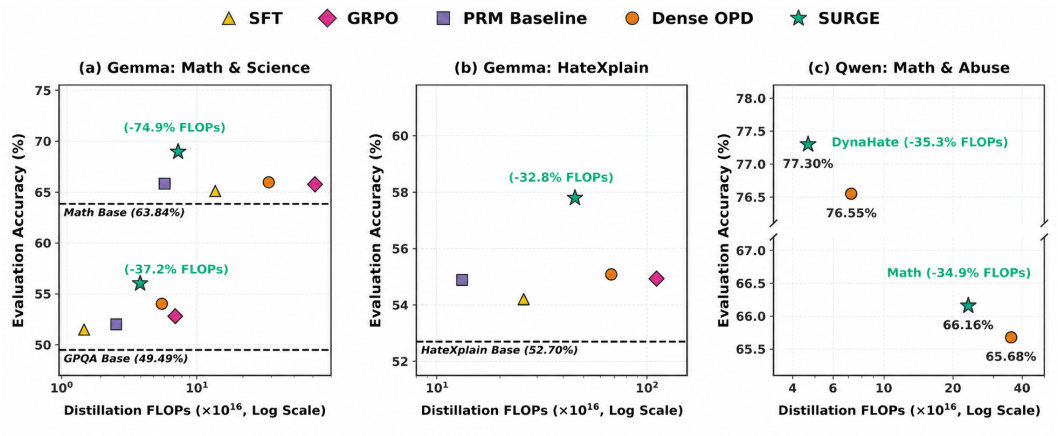


Figure 2: **Accuracy–compute trade-offs.** Evaluation accuracy versus distillation FLOPs for (a) Gemma4-E4B on Competition Math (upper cluster) and GPQA Diamond (lower cluster), (b) Gemma4-E4B on HateXplain, and (c) Qwen3.5-2B on DynaHate (upper axis) and Competition Math (lower axis). FLOPs are shown on the logarithmic scale. Dashed horizontal lines in (a,b) mark base-student accuracy. Green annotations report SURGE's FLOP reduction relative to dense OPD.

SURGE also produces shorter responses than OPD on every task. On GPQA Diamond, response length falls by 55.8% relative to OPD while accuracy improves and format failures decrease. These results show that more compact outputs can accompany better task performance under these settings.

### 4.2 Gate Selection and Complementarity

Step quality and token uncertainty provide different evidence about where teacher guidance may be useful: a low-scoring step can contain predictable tokens, while uncertainty can occur within an acceptable step. This ablation tests whether SURGE benefits from requiring both signals. Macro-only masking omits the entropy gate, while micro-only masking omits the step-score gate. We compare the overlap between their exclusions and evaluate the single-gate variants against full SURGE. The total blocked rate counts positions rejected by either gate, with shared exclusions counted once. Otherwise, all remaining positions remain active.

Table 2: **Gate filtering rates over response tokens.** Macro-level and micro-level gating blocked token rates. Active positions pass both gates. Tokens blocked by both gates are also recorded.

| Dataset | Active (%) | Total blocked (%) | Macro blocked (%) | Micro blocked (%) | Both blocked (%) |
|---|---|---|---|---|---|
| Competition Math | 63.16 | 36.84 | **23.36** | 14.19 | 0.71 |
| GPQA Diamond | 58.89 | 41.11 | **30.59** | 12.20 | 1.68 |
| MMLU Pro | 55.56 | 44.44 | 19.06 | **28.04** | 2.66 |
| HateXplain | 49.77 | 50.23 | 25.19 | **27.94** | 2.90 |
| DynaHate | 28.51 | 71.49 | 13.83 | **66.72** | 9.06 |

We note the macro-level gate blocks more tokens on Math and GPQA Diamond, while the micro gate blocks more on MMLU Pro, HateXplain, and DynaHate. This is potentially reinforced by mathematical reasoning enabling more process-level verification than the other tasks evaluated. Notably, only 0.71%–9.06% of response tokens are blocked by both, which reinforces their joint importance in optimizing feedback allocation from the teacher model.

Moreover, full SURGE gating outperforms both single-gate variants on all three reported benchmarks (Figure 3). Relative to the stronger single-gate result, accuracy improves by 1.52 percentage points on Math (68.92% versus 67.40%), 0.58 on HateXplain (57.80% versus 57.22%), and 5.55 on GPQA Diamond (56.06% versus 50.51%). Together with the exclusion analysis in Table 2, these gains support the combination of the gates for routing updates to the student model.

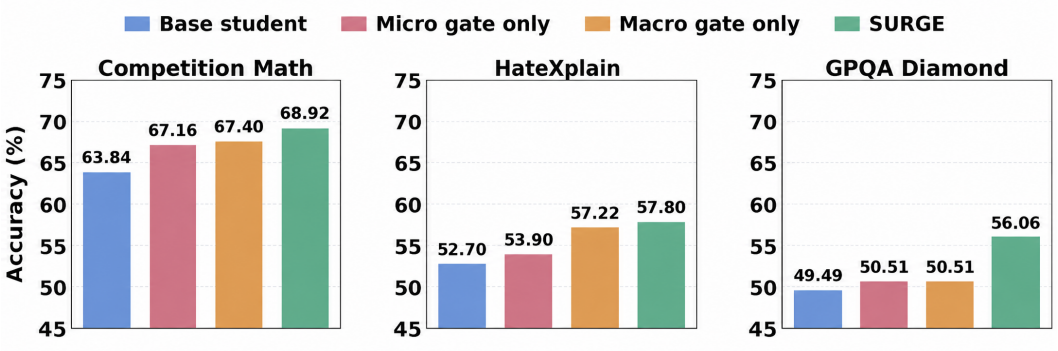


Figure 3: **Gate ablation across benchmarks.** Task accuracy comparison for the base student model, individual gates, and double-gating (SURGE). On the left, Competition Math; on the middle, HateXplain; and on the right, GPQA Diamond. The single-gate results are listed in detail on Table 7 in Appendix A.1, while base-student and SURGE results are reported on Table 5.

### 4.3 Layerwise Representation Similarity

To characterize adaptation, we use linear centered kernel alignment (CKA) (Kornblith et al., 2019) to compare corresponding layers of the base and adapted models. Values near one indicate similar rela-

tionships among token representations; lower values indicate greater change. All model checkpoints receive identical prompts. Figure 4 spans the full model depth (42 transformer layers), testing where representations begin to change. We describe the CKA protocol used in Appendix F.

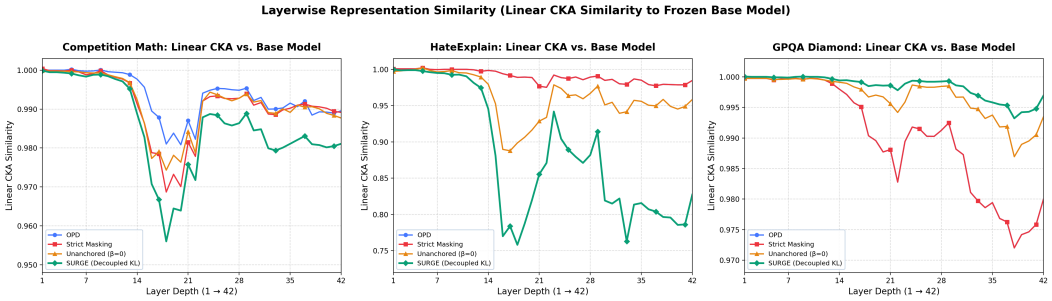


Figure 4: **Representation similarity over layers.** Linear CKA between adapted models and the frozen base model at corresponding layers. SURGE exhibits larger middle-to-late-layer changes on Competition Math and HateXplain, while remaining closer to the base model on GPQA Diamond.

Early-layer CKA remains near one on all three tasks. In later layers, SURGE departs further from the base model than OPD on Competition Math and HateXplain, but remains closer on GPQA Diamond. Accuracy gains thus coexist with different degrees of representation change across tasks. Behavioral evaluation is needed to determine how these changes affect capabilities beyond the measured tasks.

### 4.4 Activation Geometry and In-Context Copying

To test whether distillation degrades the internal representations or core in-context behaviors, we track the stable rank of final-layer activations and evaluate copying accuracy on repeated token patterns. Figure 5 reports both diagnostics, complementing the CKA analysis in Figure 4.

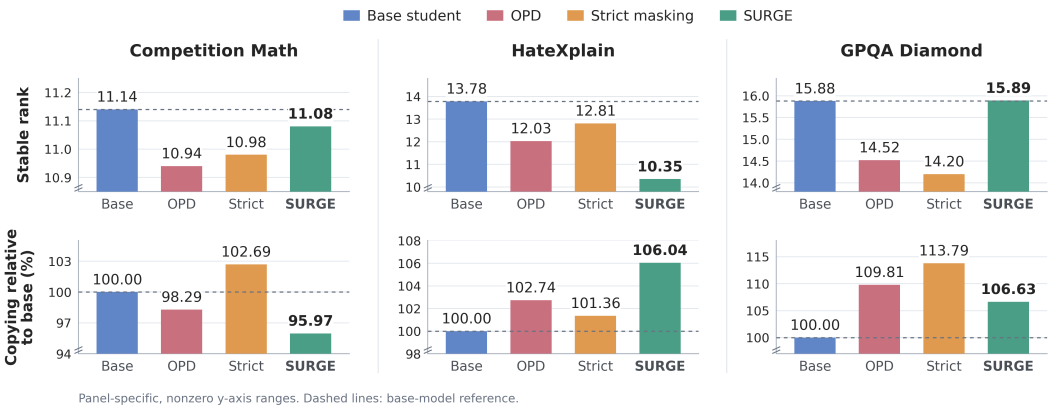


Figure 5: **Stable rank and repeated-pattern copying.** Left to right: Competition Math, HateXplain, GPQA Diamond. Activations from a 16-prompt probe are mean-centered to compute final-layer stable rank. For copying, we report the mean target-token probability on ten synthetic repeated sequences, normalized to the base model (dashed lines). Exact values are listed in Table 8 in Appendix A.2.

**Stable rank.** Let $H$ contain the final-layer token representations pooled across the 16 prompts. Subtracting each feature's mean across token rows gives $H_c$. We summarize its variation using the stable rank (Sanyal et al., 2020):

$$\mathrm{srank}(H_c) = \frac{\|H_c\|_F^2}{\|H_c\|_2^2}.$$

A lower ratio means that the strongest direction accounts for a larger share of variation in $H_c$. SURGE retains near-base stable rank on Competition Math and GPQA Diamond. On HateXplain, the lower

stable rank and larger CKA change indicate more concentrated, altered representations, without establishing better representations or lower computational cost.

**Repeated-pattern copying.** To test in-context retrieval, we insert a token pair $(A, B)$ early in a synthetic sequence, repeat $A$ downstream, and evaluate the probability of predicting $B$ next (scaled so the base model equals 100). SURGE achieves 95.97%, 106.04%, and 106.63% copying relative to the base model on Math, HateXplain, and GPQA Diamond, respectively. Copying improves over the base model on HateXplain and GPQA Diamond, but decreases slightly on Math, even though downstream accuracy increases across all these benchmarks. Although this probe suggests basic in-context copying is retained, broader evaluation across longer contexts remains necessary.

### 4.5 Decoupled vs. Strict Loss Masking

This experiment tests whether sparse teacher supervision should also imply sparse reference regularization. Excluding a token from the loss does not hold its prediction fixed, since updates at selected positions also change shared parameters. Strict masking removes the entire per-position objective at inactive tokens, including reference feedback, and averages over active tokens. SURGE, on the other hand, masks only the teacher term, retaining reference feedback throughout the response, and averages over all valid response tokens.

Table 3: **Ablation on loss decoupling.** Accuracy with the full loss masked versus teacher-only masking with response-wide reference feedback. SURGE's gain is reported in percentage points.

| Dataset | Base (%) | Strict Masking (%) | SURGE (%) | Gain over strict (pp) |
|---|---|---|---|---|
| GPQA Diamond | 49.49 | 31.82 | **56.06** | +24.24 |
| DynaHate | 76.40 | 73.30 | **81.25** | +7.95 |
| MMLU Pro | 57.46 | 54.80 | **60.78** | +5.98 |
| HateXplain | 52.70 | 55.30 | **57.80** | +2.50 |
| Competition Math | 63.84 | 66.60 | **68.92** | +2.32 |

SURGE exceeds strict masking by 2.32–24.24 percentage points across all five tasks (Table 3). Strict masking falls below the base on GPQA Diamond, DynaHate, and MMLU Pro; SURGE improves over the base on every task. The largest gap occurs on GPQA Diamond: 31.82% versus 56.06%. Loss normalization also differs between configurations (Appendix E), so the gains cannot be attributed solely to inactive-position regularization. The results support SURGE's complete objective, which joins selectively routing updates and supervising with the reference model across the response.

### 4.6 Ablating KL Reference Regularization

We next test whether gated teacher feedback is sufficient for reliable adaptation without the reference anchor. Setting $\beta = 0$ in Equation 11 removes reference supervision everywhere while retaining the gated teacher term. We track accuracy, format failures, and active-token counts to distinguish successful learning from gate inactivity accompanied by degraded outputs.

Table 4: **Outcomes without reference regularization.** Full SURGE ($\beta = 0.05$) compared with the variant without reference regularization ($\beta = 0$). Diagnostics characterize changes in task behavior.

| Dataset | SURGE | $\beta = 0$ | Diagnostics |
|---|---|---|---|
| Competition Math | 68.92% | 0.00% | Active tokens drop to 0; 9.0% format failure; corrupted derivations. |
| MMLU Pro | 60.78% | 0.00% | Active tokens drop to 0; 100% parse failure; HTML repetition loops. |
| HateXplain | 57.80% | 23.02% | Active tokens oscillate (0–117); 54% format failure; below uniform three-class guessing (33.3%). |
| GPQA Diamond | 56.06% | 53.03% | Smaller accuracy decline; training-budget effects are not isolated. |

SURGE achieves higher accuracy in all four comparisons (Table 4). Without reference regularization, accuracy falls to 0.00% on Math and MMLU Pro and to 23.02% on HateXplain. MMLU Pro has 100% parse failure, whereas Math has only 9.0% format failure, indicating degradation beyond formatting. GPQA shows a smaller decline, from 56.06% to 53.03%.

On Math and MMLU Pro, zero active-token counts coexist with zero accuracy. Gate inactivity therefore does not always indicate successful learning: low-entropy repetition can remain incorrect while receiving no teacher feedback. Regularized SURGE also exhibits comparatively low, stable measured reference divergence over the plotted training window (Appendix H, Figure 6). These comparisons do not isolate training-budget effects or the cause of each failure, nor establish longer-run stability. Within the evaluated runs, the consistent accuracy advantage and divergence diagnostics support reference regularization as a practical safeguard for sparse distillation.

## 5 Conclusion and Limitations

Our results overall support using SURGE for compute-efficient and more stable on-policy distillation regimes. We prove that, across the Gemma4 model family, SURGE improves both accuracy and compute costs relative to regular OPD. Ablations performed to isolate the gate and loss design confirm the strength of the design. Together, these findings suggest that model knowledge transfer in distillation benefits from extremely precise control of the specific knowledge being used in updates.

Still, the gates are imperfect proxies for supervision needs. Entropy can be high for valid alternatives and low for confident errors, while process scores depend on the frozen student's judgments, step segmentation, prompts, and task-specific checks. The scorer receives no reference solutions or gold answers. Ground-truth classification labels are used only for evaluation, so it is impossible for the PRM scorer's accuracy to be fully trusted. We further acknowledge that the presence of reference regularization in the decoupled KL objective may also constrain beneficial departures from the initial student.

Empirically, we cover two separate teacher–student pairings across two different model families and five benchmarks. The small representation and copying probes characterize adaptation but do not establish broad capability preservation. Likewise, format compliance measures only a narrow aspect of instruction following. Generalization to other domains and larger model scales, as well as longer-run stability, requires further evaluation.

The ablations also leave attribution limits. Strict masking changes loss normalization as well as regularization coverage, and the reference-removal comparisons do not isolate training-budget effects. Comparisons with random routing and other selective distillation methods at matched supervision and training budgets would better distinguish selection quality from supervision quantity.

## AI USE STATEMENT

Generative AI tools assisted with the revisions of the paper, code assistance, visualization development, and mathematical derivations. We remain responsible for checking the mathematical statements, implementation descriptions, and experimental claims.

## REPRODUCIBILITY STATEMENT

The appendices document process-scoring rules, loss normalization, baseline definitions, evaluation partitions, diagnostic procedures, and metric aggregation. Training settings are marked approximate where final run configurations were not retained. Evaluation prompts, answer extraction, and generation limits are shared across compared runs within each benchmark. The FLOP values reported are computed by tracking the computations during the corresponding distillation runs. Appendix D defines the workloads and reporting conventions. The approximate configuration ranges do not provide exact per-run settings, and component-level profiles and wall-clock timings are not reported. Accuracies are point estimates; variability across training seeds is not quantified.

## 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. URL https://openreview.net/forum?id=3zKtaqxLhW.

Tanvir Bhathal and Asanshay Gupta. WebSight: A Vision-First Architecture for Robust Web Agents. *arXiv preprint arXiv:2508.16987*, 2025. URL https://arxiv.org/abs/2508.16987.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge Distillation of Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Haiduo Huang, Jiangcheng Song, Yadong Zhang, and Pengju Ren. SelecTKD: Selective Token-Weighted Knowledge Distillation for LLMs. *arXiv preprint arXiv:2510.24021*, 2025. URL https://arxiv.org/abs/2510.24021.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050*, 2023. URL https://arxiv.org/abs/2305.20050.

Kevin Lu and Thinking Machines Lab. On-Policy Distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. URL https://thinkingmachines.ai/blog/on-policy-distillation/.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14867–14875, 2021. doi: 10.1609/aaai.v35i17.17745. URL https://ojs.aaai.org/index.php/AAAI/article/view/17745.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

Amartya Sanyal, Philip H. S. Torr, and Puneet K. Dokania. Stable Rank Normalization for Improved Generalization in Neural Networks and GANs. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1enKkrFDB.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. HateCheck: Functional Tests for Hate Speech Detection Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.4. URL https://aclanthology.org/2021.acl-long.4/.

Jon Saad-Falcon, E. Kelly Buchanan, Mayee F. Chen, Tzu-Heng Huang, Brendan McLaughlin, Tanvir Bhathal, Shang Zhu, Ben Athiwaratkun, Frederic Sala, Scott Linderman, Azalia Mirhoseini, and Christopher Ré. Shrinking the Generation-Verification Gap with Weak Verifiers. *arXiv preprint arXiv:2506.18203*, 2025. URL https://arxiv.org/abs/2506.18203.

Jon Saad-Falcon, Avanika Narayan, Robby Manihani, Tanvir Bhathal, Herumb Shandilya, Hakki Orhun Akengin, Gabriel Bo, Andrew Park, Matthew Hart, Caia Costello, Chuan Li, Christopher Ré, and Azalia Mirhoseini. OpenJarvis: Personal AI, On Personal Devices. *arXiv preprint arXiv:2605.17172*, 2026. URL https://arxiv.org/abs/2605.17172.

Prady Saligram, Tanvir Bhathal, and Robby Manihani. 2048: Reinforcement Learning in a Delayed Reward Environment. *arXiv preprint arXiv:2507.05465v1*, 2025. URL https://arxiv.org/abs/2507.05465v1. Version 1; the paper was subsequently withdrawn in version 2 on July 24, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Almog Tavor, Itay Ebenspanger, Neil Cnaan, and Mor Geva. Rethinking Selective Knowledge Distillation. *arXiv preprint arXiv:2602.01395*, 2026. URL https://arxiv.org/abs/2602.01395.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. Learning from the Worst: Dynamically Generated Datasets to Improve Online Hate Detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.132. URL https://aclanthology.org/2021.acl-long.132/.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not All Disagreement Is Learnable: Token Teachability in On-Policy Distillation. *arXiv preprint arXiv:2605.26844*, 2026. URL https://arxiv.org/abs/2605.26844.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.01574. Datasets and Benchmarks Track.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token Importance in On-Policy Distillation. *arXiv preprint arXiv:2604.14084*, 2026. URL https://arxiv.org/abs/2604.14084.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Zhiwei Zhang, Zechen Sun, Fei Zhao, Kang Peng, Bin Liang, Huayu Deng, Yao Hu, Kam-Fai Wong, and Mu Chuan. Verify Before You Distill: Prompt-Level Teacher Gating for On-Policy Distillation. *arXiv preprint arXiv:2609.02998*, 2026. URL https://arxiv.org/abs/2609.02998.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. *arXiv preprint arXiv:2601.18734*, 2026. URL https://arxiv.org/abs/2601.18734.

# A Full Benchmark Results

Table 5 provides the per-benchmark results summarized in Table 1. The Competition Math, GPQA Diamond, and HateXplain accuracy–compute comparisons are visualized in Figure 2(a,b).

Table 5: **Full benchmark comparison.** Results across five benchmarks from distilling Gemma4-31B Dense into Gemma4-E4B. FLOPs savings are relative to OPD. †Teacher model accuracy.

| Dataset | Method / Model | Accuracy (%) | Distillation FLOPs | Avg Output Len | Format Failure (%) |
|---|---|---|---|---|---|
| **Competition Math** | Base Gemma4-E4B | 63.84 → 78.52† | – | 489 | 23.40 |
| | Standard SFT | 65.12 | $1.28 \times 10^{17}$ | 350 | 22.80 |
| | GRPO ($G = 4$) | 65.76 | $5.85 \times 10^{17}$ | 338 | 22.40 |
| | PRM Baseline | 65.80 | $\mathbf{6.32 \times 10^{16}}$ | **330** | 22.20 |
| | OPD | 65.96 | $2.96 \times 10^{17}$ | 342 | 22.20 |
| | **SURGE (Ours)** | **68.92** | $7.43 \times 10^{16}$ (-74.9%) | 334 | **21.20** |
| **GPQA Diamond** | Base Gemma4-E4B | 49.49 → 70.20† | – | 1075 | 2.02 |
| | Standard SFT | 51.52 | $\mathbf{1.43 \times 10^{16}}$ | 850 | 1.52 |
| | GRPO ($G = 4$) | 53.03 | $6.21 \times 10^{16}$ | 780 | **1.01** |
| | PRM Baseline | 52.02 | $2.30 \times 10^{16}$ | 740 | **1.01** |
| | OPD | 54.04 | $5.45 \times 10^{16}$ | 1370 | 17.68 |
| | **SURGE (Ours)** | **56.06** | $3.42 \times 10^{16}$ (-37.2%) | **606** | 2.02 |
| **MMLU Pro** | Base Gemma4-E4B | 57.46 → 72.95† | – | 589 | 31.57 |
| | Standard SFT | 58.62 | $1.52 \times 10^{17}$ | 580 | 31.49 |
| | GRPO ($G = 4$) | 59.33 | $6.97 \times 10^{17}$ | 560 | 30.16 |
| | PRM Baseline | 58.91 | $\mathbf{2.19 \times 10^{16}}$ | 575 | 31.08 |
| | OPD | 60.41 | $3.69 \times 10^{17}$ | 593 | 29.58 |
| | **SURGE (Ours)** | **60.78** | $1.15 \times 10^{17}$ (-68.8%) | **299** | **29.21** |
| **HateXplain** | Base Gemma4-E4B | 52.70 → 61.59† | – | 169 | **0.00** |
| | Standard SFT | 54.21 | $2.59 \times 10^{17}$ | 160 | 0.05 |
| | GRPO ($G = 4$) | 54.94 | $1.11 \times 10^{18}$ | 145 | 0.05 |
| | PRM Baseline | 54.89 | $\mathbf{1.32 \times 10^{17}}$ | 154 | 0.05 |
| | OPD | 55.09 | $6.76 \times 10^{17}$ | 176 | 0.52 |
| | **SURGE (Ours)** | **57.80** | $4.54 \times 10^{17}$ (-32.8%) | **77** | 0.26 |
| **DynaHate** | Base Gemma4-E4B | 76.40 → 84.55† | – | 182 | 0.15 |
| | Standard SFT | 78.10 | $1.98 \times 10^{17}$ | 110 | 0.15 |
| | GRPO ($G = 4$) | 78.85 | $8.70 \times 10^{17}$ | 102 | 0.05 |
| | PRM Baseline | 79.15 | $\mathbf{1.97 \times 10^{17}}$ | **94** | **0.00** |
| | OPD | 79.70 | $4.18 \times 10^{17}$ | 98 | 0.20 |
| | **SURGE (Ours)** | **81.25** | $2.69 \times 10^{17}$ (-35.6%) | 97 | 0.10 |

We also perform an additional experiment across the Qwen 3.5 model family on the same benchmarks. Here, we distill from a Qwen3.5-122B-A10B teacher into a Qwen3.5-2B student model to showcase generalization across larger model gaps. Table 6 reports these results. Remarkably, we see a similar pattern arise, in which SURGE outperforms OPD across all tasks but GPQA Diamond (tie) with nearly 70% of the original compute on average.

Table 6: **Benchmarking another model family.** Distillation results from Qwen3.5-122B-A10B into Qwen3.5-2B across five benchmarks. All OPD and SURGE runs use two training epochs. FLOPs reductions are calculated relative to total OPD FLOPs.

| Benchmark | Method / Model | Accuracy (%) | Distillation FLOPs |
|---|---|---|---|
| **Competition Math** | Base Qwen3.5-2B (Student) | 61.40 | – |
| | Qwen3.5-122B-A10B (Teacher) | 88.28 | – |
| | OPD | 65.68 | $3.58 \times 10^{17}$ |
| | **SURGE (Ours)** | **66.16** | $\mathbf{2.33 \times 10^{17}}$ (-34.9%) |
| **DynaHate** | Base Qwen3.5-2B (Student) | 63.35 | – |
| | Qwen3.5-122B-A10B (Teacher) | 82.50 | – |
| | OPD | 76.55 | $7.20 \times 10^{16}$ |
| | **SURGE (Ours)** | **77.30** | $\mathbf{4.66 \times 10^{16}}$ (-35.3%) |
| **GPQA Diamond** | Base Qwen3.5-2B (Student) | 42.50 | – |
| | Qwen3.5-122B-A10B (Teacher) | 77.50 | – |
| | OPD | **52.50** | $4.54 \times 10^{15}$ |
| | **SURGE (Ours)** | **52.50** | $\mathbf{3.69 \times 10^{15}}$ (-18.7%) |
| **HateCheck** | Base Qwen3.5-2B (Student) | 77.08 | – |
| | Qwen3.5-122B-A10B (Teacher) | 96.65 | – |
| | OPD | 93.70 | $2.04 \times 10^{16}$ |
| | **SURGE (Ours)** | **94.50** | $\mathbf{1.24 \times 10^{16}}$ (-39.2%) |
| **HateExplain** | Base Qwen3.5-2B (Student) | 47.66 | – |
| | Qwen3.5-122B-A10B (Teacher) | 52.86 | – |
| | OPD | 51.72 | $1.57 \times 10^{17}$ |
| | **SURGE (Ours)** | **52.08** | $\mathbf{1.02 \times 10^{17}}$ (-35.0%) |

### A.1 Single-Gate Results

Table 7: **Single-gate accuracy.** Accuracy results run using individual gates of token-entropy and step-PRM only.

| Dataset | Gating architecture | Accuracy (%) |
|---|---|---|
| GPQA Diamond | Micro-Gate Only | 50.51 |
| | Macro-Gate Only | 50.51 |
| Competition Math | Micro-Gate Only | 67.16 |
| | Macro-Gate Only | 67.40 |
| HateXplain | Micro-Gate Only | 53.90 |
| | Macro-Gate Only | 57.22 |

### A.2 Stable-Rank and Copying Results

Table 8: **Stable rank and repeated-pattern copying.** Stable rank uses final-layer activations from the 16-prompt probe, with each feature's mean subtracted across pooled token positions. Copying is the mean target-token probability on ten fixed synthetic sequences, normalized so that the base model scores 100.

| Benchmark | Model | Stable rank | Copying relative to base (%) |
|---|---|---|---|
| Math | Base | 11.14 | 100.00 |
| | OPD | 10.94 | 98.29 |
| | Strict Masking | 10.98 | 102.69 |
| | SURGE | 11.08 | 95.97 |
| HateXplain | Base | 13.78 | 100.00 |
| | OPD | 12.03 | 102.74 |
| | Strict Masking | 12.81 | 101.36 |
| | SURGE | 10.35 | 106.04 |
| GPQA Diamond | Base | 15.88 | 100.00 |
| | OPD | 14.52 | 109.81 |
| | Strict Masking | 14.20 | 113.79 |
| | SURGE | 15.89 | 106.63 |

# B Training Configurations

For the Gemma experiments, we initialize the student from Gemma4-E4B and use Gemma4-31B as the frozen teacher. Reference regularization uses a frozen copy of the initial student. We fine-tune all student parameters in JAX/Flax using bfloat16 and GSPMD across 16 TPU v5e chips. Training uses AdamW with weight decay of approximately $10^{-4}$–$10^{-2}$, constant or cosine learning-rate schedules, and global gradient-norm clipping at 1.0. The AdamW moment coefficients are $(0.9, 0.999)$ with $\varepsilon = 10^{-8}$. On-policy training uses one update per rollout batch without gradient accumulation.

Table 9 summarizes approximate training settings from the available records. The listed generation caps and optional length-penalty settings describe general ranges or nominal limits, rather than exact final values for individual runs. These entries should not be interpreted as a single configuration used jointly in every experiment.

Table 9: **Approximate Gemma training settings.** Batch size counts sequences per optimizer update. Response caps give approximate limits on tokens generated during training, rather than exact final per-run settings; paired values (seen split by a "/") denote reasoning and classification tasks, respectively.

| Method | Learning rate | Batch | Epochs | Response cap | $\beta$ | $\epsilon_c$ |
|---|---|---|---|---|---|---|
| SFT | $2\times10^{-5}$ | 4–16 | 1–3 | 512–1024 | — | — |
| GRPO | $2\times10^{-6}$–$10^{-5}$ | 4–16 | 1 | 512 / 128 | 0.05 | 0.20 |
| PRM | $5\times10^{-7}$–$2\times10^{-6}$ | 16 | 1 | 512 / 128 | 0.01–0.20 | — |
| OPD | $2\times10^{-6}$–$10^{-5}$ | 16 | 1–2 | 256–512 / 128 | 0.05–0.10 | 0.20 |
| SURGE | $2\times10^{-6}$ | 16 | 1–2 | 256–512 / 128 | 0.05 | 0.20 |

GRPO uses 1–4 prompts with $G = 4$ responses per prompt. SFT sequences are padded to approximately 1,024–2,048 tokens. Typical settings are $\beta = 0.10$ for PRM and learning rate $2 \times 10^{-6}$ with $\beta = 0.05$ for OPD.

**SFT.** The teacher generates one complete response per prompt, including reasoning and a final answer. We fine-tune the student on these teacher responses using cross-entropy over all response tokens, excluding prompt and padding positions.

**GRPO.** Our GRPO baseline (Shao et al., 2024) generates $G = 4$ responses per prompt, and the teacher acts as a reward model, assigning a scalar score to each response. Rewards are normalized within each group using the population standard deviation and a numerical safeguard of $10^{-8}$, then used in a clipped policy update with reference regularization.

**PRM baseline.** The process scorer uses the frozen reference student and task-specific checks described in Appendix E.1. The same frozen model also supplies reference log-probabilities. Each

step score is assigned to its constituent tokens and supplies an immediate training reward ($\gamma = 0$), with reference regularization and an unclipped policy-gradient update.

**OPD.** Following Lu and Thinking Machines Lab (2025), the student generates on-policy responses and the teacher computes next-token log-probabilities conditioned on the student-generated prefixes. Comparing teacher and student log-probabilities for the sampled response tokens provides immediate token-level reverse-KL feedback with $\gamma = 0$. Teacher rewards are normalized across valid response tokens in the batch with a standard-deviation floor of 1; the update includes a frozen-reference term and PPO clipping.

**SURGE.** Our method uses the same frozen reference student for step-level scoring and reference regularization. Teacher feedback applies exclusively to tokens passing both the step-score and entropy gates, with a PRM threshold of 0.5 and approximate entropy thresholds of 1.2–1.5 nats. The nominal configuration uses 1.5 nats. Teacher-reward statistics pool active positions across the batch, with a standard-deviation floor of 1. Reference regularization remains active across all valid response positions, and the clipped policy objective averages over valid response tokens, as specified in Equations 11–13.

**Sampling and length control.** Student rollout temperatures are approximately 0.7–1.0 during training, with an estimated top-$p$ of 0.95 for GRPO. Optional length control for GRPO, OPD, and SURGE uses a very small auxiliary penalty, with a nominal weight around 0.01, to discourage collapse into repetitive or nonsensical continuations that consume the generation budget. Approximate activation thresholds are around 400 tokens for reasoning tasks and 128 tokens for classification tasks. These describe general settings rather than final values for individual runs.

# C Data and Evaluation

**Benchmark partitions.** Table 10 lists the experimental partitions used in this study. These counts describe the study's training and evaluation partitions, rather than the default partitions distributed by each dataset source.

Table 10: **Benchmark partitions used in the experiments.**

| Benchmark | Training examples | Evaluation examples |
|---|---|---|
| Competition Math (Hendrycks et al., 2021) | 10,000 | 2,500 |
| GPQA Diamond (Rein et al., 2023) | 178 | 198 |
| MMLU Pro (Wang et al., 2024) | 9,625 | 2,407 |
| HateXplain (Mathew et al., 2021) | 15,383 | 1,924 |
| DynaHate (Vidgen et al., 2021) | 8,000 | 2,000 |

**Evaluation.** All evaluations use greedy decoding with temperature 0. Within each benchmark, all compared runs use the same evaluation prompts, answer-extraction procedure, and maximum generation length. These settings are shared within a benchmark; different benchmarks can require different output formats and generation limits. The approximate training-cap ranges in Table 9 describe training settings and do not specify the exact benchmark-specific evaluation limits. Ground-truth classification labels are used only to evaluate predictions. Accuracy, mean response length in tokens, and format-failure rate are reported separately. Accuracy assesses the task answer, while format failure concerns compliance with the required output format; correctness and format validity need not coincide. Cross-benchmark aggregation and rounding conventions are defined in Appendix G.

## C.1 Prompts

All benchmark prompts followed a common structure: the task input, a task-specific instruction, and an explicit response-format specification within a user turn, followed by the model turn. The task wording, permitted answers, and requested response fields were adapted to each benchmark's requirements, so the prompt text was not identical across benchmarks. Within each benchmark, the same prompt template and output requirements were used across the compared methods.

The following example is a three-class classification prompt. The input post replaces `{post_text}`. The requested JSON response contains concise reasoning and a final class label.

```
<|turn>user
Post: {post_text}

Classify the post above into one of the following categories:
hatespeech, normal, or offensive.
Respond ONLY in the following JSON format:
{
  "thinking_steps": "your step-by-step reasoning (keep it concise)",
  "label": "the category name (hatespeech, normal, or offensive)"
}<turn|>
<|turn>model
```

## D Compute Definitions

**Reported FLOPs.** The FLOP values reported are computed by tracking the computations during the corresponding distillation runs. FLOPs measure floating-point operation counts; they are distinct from elapsed time, memory traffic, and communication overhead. This appendix defines the workloads included in the comparison and their interpretation. It does not reconstruct the reported totals from approximate hyperparameter ranges or evaluation response lengths.

**Distillation workloads.** For method $m$, we write its total distillation workload as

$$F_m = F_{\text{rollout},m} + F_{\text{update},m} + F_{\text{ref},m} + F_{\text{process},m} + F_{\text{teacher},m}. \tag{14}$$

Here $F_{\text{rollout},m}$ counts student prompt processing and auto-regressive response generation; $F_{\text{update},m}$ counts student training forward and backward work; $F_{\text{ref},m}$ counts frozen-reference forward evaluation for response-wide reference log-probabilities; $F_{\text{process},m}$ counts the separate process-scoring forward passes; and $F_{\text{teacher},m}$ denotes teacher response generation (SFT), sequence reward evaluation (GRPO), or token log-probability computation (OPD and SURGE), as required by the method. A component is zero when the method does not use that workload. Each component sums the corresponding work across the training examples, generated responses, and updates processed by that run. Evaluation after training is separate from distillation.

Table 11: **Workloads included in distillation FLOPs.** The components define the accounting boundary for each method.

| Method | Distillation workloads |
|---|---|
| SFT | Offline teacher-completion generation and student forward/backward training on the resulting responses. |
| GRPO | Student generation, teacher sequence reward evaluation, reference evaluation, and student updates for the sampled response groups. |
| PRM | Student generation and updates, reference evaluation, and process scoring using the frozen reference student. |
| OPD | Student generation and updates, reference evaluation, and teacher log-probability computation at valid response positions. |
| SURGE | Student generation and updates, response-wide reference evaluation, process scoring, and teacher computation required to obtain log-probabilities at selected response positions. |

For SFT, teacher completions are generated once and reused across student training epochs. For GRPO, $G = 4$ specifies responses per prompt; the corresponding workloads count the responses actually processed. Process scoring and reference regularization share model weights, but evaluate different inputs and perform distinct jobs. Process-scoring work includes the task input, preceding student steps, and candidate step in each model-evaluated scoring prompt. Deterministic checks that return a score without invoking the model do not incur a model-scoring forward pass.

**Teacher conditioning and computation.** For each selected position $t$, the teacher computes next-token log-probabilities conditioned on $(x, y_{<t})$. The teacher and student log-probabilities for $y_t$

are then compared to form the raw reverse-KL feedback in Section 3.2. The conditioning prefix contains all preceding student tokens, including tokens excluded from teacher supervision. The teacher computes its own hidden states using its frozen weights. OPD and SURGE train different student policies and can generate different token sequences and lengths, even when their prompts and generation limits are shared.

Teacher log-probability computation includes both context processing and output computation:

$$F_{\text{teacher},m} = F_{\text{teacher body},m} + F_{\text{teacher output},m}. \tag{15}$$

The body term counts the teacher's forward computation through the prompt and student-token contexts required for the queried positions, including preceding tokens excluded by the supervision mask. The output term counts vocabulary projections and probability normalization at every position where those operations are evaluated. Masking a loss does not subtract an already evaluated projection. Repeated context evaluation contributes each time it is performed; reusing a computed context does not add another forward pass. Both terms are added to student rollout and update work, reference evaluation, and process scoring in Equation 14.

The supervision mask specifies where teacher feedback enters the loss; it is not a multiplier on the teacher-body cost or the full distillation total. Selecting fewer output positions retains the context required to compute their log-probabilities. Differences in processed training lengths, numbers of updates, and evaluated output positions affect FLOPs; different probability values alone do not change operation counts.

**Gate statistics and training workloads.** For the population of responses used in the gate analysis, let $A_i$ be the number of active positions and $T_i$ the valid response length. The token-weighted active fraction is

$$\alpha_{\text{gate}} = \frac{\sum_i A_i}{\sum_i T_i}. \tag{16}$$

It describes the fraction of response tokens selected for teacher supervision. For DynaHate, Table 2 reports 28.51% active tokens. This routing statistic is distinct from the fraction of total distillation FLOPs saved: student updates, reference evaluation, process scoring, and required teacher context processing also contribute to the total. Training FLOPs depend on the training workloads processed by each method. The mean output lengths in the results tables are evaluation statistics and do not substitute for training token counts.

**Savings relative to OPD.** For benchmark $d$, the reported saving for method $m$ is

$$s_{md}(\%) = 100\left(1 - \frac{F_{md}}{F_{Od}}\right), \tag{17}$$

where $F_{Od}$ is the OPD total for that benchmark. For example, the displayed DynaHate entries give $100(1 - 2.69/4.18) \approx 35.6\%$. The mean of the five Gemma benchmark savings is 49.9%, as defined in Appendix G. This average assigns equal weight to benchmarks and differs from the reduction in summed FLOPs. The operation-count comparison does not directly establish a wall-clock speedup, which also depends on hardware utilization, memory movement, and communication.

# E SURGE Implementation Details

Section 3 specifies the core teacher-alignment and reference-regularization objective. This appendix records process scoring, teacher log-probability computation, normalization, reduction conventions, and the small optional length penalty used as an auxiliary training safeguard.

## E.1 Process Scoring

**Model-based score.** For a step that reaches model evaluation, the frozen reference student runs one forward pass on a scoring prompt $c_k$. Let $\ell_{+,k}$ and $\ell_{-,k}$ be its next-token logits at the final prompt position for the designated + and − token IDs. The score is

$$q_k^{\text{model}} = \sigma(\ell_{+,k} - \ell_{-,k}) = \frac{\exp(\ell_{+,k})}{\exp(\ell_{+,k}) + \exp(\ell_{-,k})}. \tag{18}$$

This is the softmax restricted to the two designated tokens. It is used as a routing score, without assuming calibration as step-correctness probability. The scorer reads the logits directly; it does not generate a numerical rating.

**Scoring context.** Each newline-delimited candidate step is evaluated in a prompt containing the original input, all preceding student steps, and the current step. The role instruction is an expert math tutor for Math, GPQA Diamond, and MMLU Pro, or a content moderator for HateXplain and DynaHate. Across all reported experiments, the scorer receives the task question or input and student-generated steps, with no reference solution or gold answer. The prompt instructs the model to respond with + for a correct step or − for an incorrect step. Previous steps are numbered, with an empty history indicated for the first candidate.

**Deterministic checks.** Before invoking the model, the scorer applies task-specific checks that can return a fixed score and skip the scoring forward pass:

| Condition | Assigned $q_k$ |
|---|---|
| Malformed JSON | 0.05 |
| Syntax-only line | 1.0 |
| Repetitive step with more than 15 words and fewer than 35% unique words | 0.02 |

If no check returns a fixed score, $q_k = q_k^{\text{model}}$. The resulting $q_k$ is broadcast to the step's tokens before applying the entropy gate. With $\tau_{\text{prm}} = 0.5$, low scores pass the macro gate and remain eligible for teacher supervision; the micro gate must also pass. Ground-truth classification labels are used only for evaluation; they do not determine process scores, training rewards, or routing masks. No reference solutions are supplied to the model-based scorer.

### E.2 Teacher Feedback and Loss Conventions

**Teacher log-probabilities and conditioning.** At each active position, the large teacher computes next-token log-probabilities given $h_t = (x, y_{<t})$. This prefix contains all preceding student tokens, including those excluded from teacher supervision. The update subtracts the recorded student log-probability of $y_t$ from the teacher log-probability of that same token to obtain $r_{\text{raw}}(t)$. This provides sampled reverse-KL feedback for the teacher-derived reward. The macro gate's step-quality score is computed separately by the process scorer or task-specific checks. Both teacher feedback and process scoring use the task input and student-generated context without supplied reference solutions. Prompt and padding positions are excluded from the response objective. Teacher log-probability computation leaves the sampled response unchanged.

**Reward normalization.** Let $\mathcal{U}$ contain the active positions in the normalization batch, $A = |\mathcal{U}|$, and $r_{it} = r_{\text{raw},i}(t)$. The statistics in Equation 10 are

$$\mu_{\text{active}} = \frac{\sum_{(i,t)\in\mathcal{U}} r_{it}}{A + \varepsilon}, \tag{19}$$

$$v_{\text{active}} = \frac{\sum_{(i,t)\in\mathcal{U}} (r_{it} - \mu_{\text{active}})^2}{A + \varepsilon}, \tag{20}$$

$$s_{\text{active}} = \max(\sqrt{v_{\text{active}}}, 1)\,, \tag{21}$$

with $\varepsilon = 10^{-8}$. The denominator floor prevents amplification of small reward differences, so the normalized rewards need not have unit variance. An empty active set has zero teacher contributions and a single active reward has zero centered signal (apart from numerical safeguards).

**Reference regularization and optimization.** The reference term uses the sampled log-ratio defined in Section 3.2. An individual value can be negative; at a fixed prefix, its expectation under the behavior policy is the corresponding categorical KL to the reference. Rewards are held fixed during differentiation. Equation 13 averages the loss over valid response tokens, with $Z = \sum_{i,t} b_{it}$. When a fresh rollout batch receives one update starting at $\theta = \theta_{\text{old}}$, the probability ratios are one and PPO clipping does not alter that update's gradient.

At inactive positions, the teacher reward is zero while reference regularization and any enabled length penalty remain. All student parameters are shared across positions, so updates from active

positions can also change predictions at inactive positions. Strict masking instead multiplies the entire per-position objective by the routing mask and divides the sum by the active-token count, $A = \sum_{i,t} \mathcal{M}_{\text{SURGE},it}$. For $A > 0$, its reduction uses $A$ in place of SURGE's $Z = \sum_{i,t} b_{it}$. This changes the scale of the retained loss terms in addition to removing reference feedback at inactive positions. A comparison that isolates reference coverage must hold this reduction fixed.

**Small auxiliary length penalty.** Where enabled, a very small length penalty serves as a safeguard against output collapse into repetitive or nonsensical continuations that persist to the generation cap. For a response of length $T_i$, the penalty at each valid response position is

$$p_i = w_{\text{len}} \, \mathbb{I}[T_i > L_{\text{len}}] \, \frac{T_i - L_{\text{len}}}{\max(T_i, 1)}. \tag{22}$$

The penalty is zero up to the threshold and satisfies $0 \le p_i \le w_{\text{len}}$. With a nominal weight around $0.01$, the per-token reward subtraction is therefore bounded by approximately $0.01$, even for very long continuations. For SURGE, this auxiliary term enters Equation 13 through the training reward $R_{\text{SURGE},it} - p_i$. Setting $w_{\text{len}} = 0$ disables it. The weight and thresholds around $400$ tokens for reasoning tasks and 128 for classification summarize approximate general settings, rather than exact final per-run values. They do not specify a fixed pairing with the approximate generation caps in Table 9. It is also important to note that the soft penalty and hard cap are separate controls.

# F Diagnostic Protocols

**CKA.** The representation probe uses 16 fixed, first-in-order prompts per task, with at most 256 input tokens per prompt. Intermediate entries provide residual states between blocks, after the attention and MLP residual additions and before the next block's input normalization. The final probe uses the model's returned final hidden state. Padding positions are removed using the attention mask. For each layer, corresponding non-padding token rows are pooled into $X \in \mathbb{R}^{N \times d_X}$ and $Y \in \mathbb{R}^{N \times d_Y}$. Center each feature across these rows:

$$X_c = X - \mathbf{1}\bar{X}, \qquad Y_c = Y - \mathbf{1}\bar{Y}. \tag{23}$$

For nonzero denominators,

$$\text{CKA}(X, Y) = \frac{\|Y_c^\top X_c\|_F^2}{\|X_c^\top X_c\|_F \, \|Y_c^\top Y_c\|_F}. \tag{24}$$

The index $N$ counts token rows; it is not the number of independent prompts. The implementation returns 1.0 when the denominator is below $10^{-12}$; this numerical fallback must be distinguished from a nondegenerate similarity estimate.

Figure 4 reports the layerwise results across all 42 Transformer blocks, with model comparisons made at the corresponding layer depths.

**Stable rank.** For the centered final-layer matrix $H_c$, the implementation forms

$$C = H_c^\top H_c / N \tag{25}$$

and computes its eigenvalues $\lambda_j$. The reported ratio is

$$\frac{\sum_j \lambda_j}{\max_j \lambda_j}, \tag{26}$$

with numerical negative eigenvalues clamped to zero. It equals $\|H_c\|_F^2 / \|H_c\|_2^2$ when the denominator is positive. The implementation also returns 1.0 for a near-zero maximum eigenvalue; this convention does not establish nonzero effective dimensionality.

**Copying.** The probe uses NumPy's `RandomState(42)` to sample ten sequences of length 64, with integer token IDs in $[500, 50000)$. Using zero-based positions, the reported construction places $A$ at an early index $p_1$ in the range 5–32 and $B$ at $p_1 + 1$, then repeats $A$ at a later index $p_2$ in the range 37–62. The input prefix contains positions 0 through $p_2$, inclusive, and the next-token probability of

$B$ is evaluated at its final position. The raw and relative scores are

$$C(\pi) = \frac{1}{10}\sum_{j=1}^{10} \pi(B_j \mid \text{prefix}_j, A_j), \tag{27}$$

$$C_{\text{relative}}(\pi) = 100\,\frac{C(\pi)}{C(\pi_{\text{base}})}. \tag{28}$$

This probe measures next-token probability on repeated patterns. The relative score does not reveal the absolute copying probability.

## G Metric Definitions

Table 1 reports unweighted arithmetic means across the five Gemma benchmarks in Table 5. Each benchmark contributes equally to mean accuracy, FLOPs, output length, and format-failure rate. The Qwen results are separate.

**Accuracy improvement and gap recovery.** For benchmark $d$, let $a_{0d}$, $a_{Td}$, $a_{Sd}$, and $a_{Od}$ be base, teacher, SURGE, and OPD accuracies in percentage points. The mean absolute gain and mean relative gain over the base are

$$\Delta^{\text{pp}}_{\text{accuracy}} = \frac{1}{5}\sum_d (a_{Sd} - a_{0d}), \tag{29}$$

$$\Delta^{\text{rel}}_{\text{accuracy}}(\%) = \frac{100}{5}\sum_d \frac{a_{Sd} - a_{0d}}{a_{0d}}. \tag{30}$$

The reported 8.6% is the mean of the five relative gains; it is not the relative change between mean accuracies.

Define per-benchmark recovery of the initial teacher–student gap as

$$r_{Sd} = \frac{a_{Sd} - a_{0d}}{a_{Td} - a_{0d}}, \qquad r_{Od} = \frac{a_{Od} - a_{0d}}{a_{Td} - a_{0d}}. \tag{31}$$

With $\bar{r}_S = \frac{1}{5}\sum_d r_{Sd}$ and $\bar{r}_O = \frac{1}{5}\sum_d r_{Od}$, SURGE's mean gap recovery is $100\bar{r}_S = 40.93\%$. Its improvement in mean gap recovery relative to OPD is

$$100\left(\frac{\bar{r}_S}{\bar{r}_O} - 1\right) = 66.6\%.$$

These are descriptive benchmark averages under the evaluation protocol in Appendix C.

**FLOP savings.** For a method $m$ with reported FLOPs $F_{md}$ and OPD FLOPs $F_{Od}$, the mean saving is

$$\Delta_{\text{FLOPs},m}(\%) = \frac{100}{5}\sum_d \left(1 - \frac{F_{md}}{F_{Od}}\right). \tag{32}$$

Thus, the 49.9% SURGE value averages benchmark-specific percentage savings relative to OPD. It differs from $100(1 - \sum_d F_{Sd}/\sum_d F_{Od})$. The computation definitions and accounting boundary are specified in Appendix D.

**Response length and format failure.** Let $\ell_{md}$ and $f_{md}$ denote mean response length and format-failure rate for method $m$ on benchmark $d$, and define

$$\bar{\ell}_m = \frac{1}{5}\sum_d \ell_{md}, \qquad \bar{f}_m = \frac{1}{5}\sum_d f_{md}. \tag{33}$$

Reductions relative to OPD use ratios of benchmark means:

$$\Delta_{\text{length}}(\%) = 100\left(1 - \frac{\bar{\ell}_S}{\bar{\ell}_O}\right) = 45.2\%, \tag{34}$$

$$\Delta_{\text{format}}(\%) = 100\left(1 - \frac{\bar{f}_S}{\bar{f}_O}\right) = 24.8\%. \tag{35}$$

Accuracy differences are expressed in percentage points; relative changes are percentages.

**Rounding and numerical precision.** For the Gemma results, accuracy differences and teacher–student gap recovery are calculated from the benchmark accuracies displayed in the results tables, with rounding applied to the final derived value. Accuracies and percentage-point gains are displayed to two decimal places. Relative changes use the precision shown. Total FLOPs are displayed to three significant figures. The aggregation rules above specify whether a summary uses a mean of benchmark-specific changes or a change between benchmark means.

## H Reference-Divergence Dynamics

Figure 6 reports measured KL divergence from the frozen reference during training with sparse teacher feedback. Without reference regularization ($\beta = 0$), divergence at active positions rises sharply and exhibits large fluctuations during the highlighted interval, while inactive-position divergence remains comparatively low. SURGE with reference regularization ($\beta = 0.05$) exhibits low, stable divergence across the plotted training window. These training dynamics complement the ablation outcomes reported in Section 4.6 and Table 4.

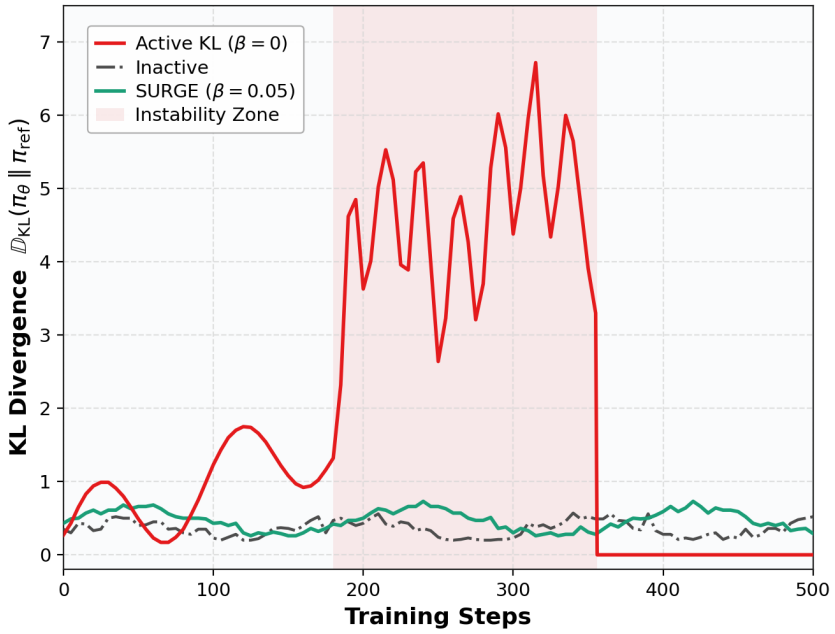


Figure 6: **Reference-divergence dynamics during training.** Measured KL divergence from the frozen reference. The red curve shows active-position divergence without reference regularization ($\beta = 0$), the gray curve shows inactive-position divergence, and the green curve shows SURGE with $\beta = 0.05$. The shaded interval highlights the period of elevated and fluctuating divergence at unregularized active positions. SURGE maintains comparatively low and stable reference divergence.