SEAM: SHARED-EVENT ALIGNMENT FOR CROSS-TOKENIZER ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation across tokenizers requires predictions in different vocabularies to express a common observation. Shared token identities provide incomplete and asymmetric correspondence, while broader alignment interfaces must still determine how predictions are conditioned at token boundaries. We introduce Shared-Event Alignment via Marginalization (SEAM), which matches local next-byte and control-event distributions through the models' native outputs. At each student boundary, compatible tokens either extend beyond the query or end there and contribute a native continuation. Marginalizing these routes on both sides preserves earlier native histories and requires no shared token inventory or auxiliary prediction head. We derive the probability decomposition, native-logit gradients, and a bound for thresholded teacher lookahead. Across three teacher–student pairs and five mathematics/programming benchmarks, SEAM achieves the highest average score among the evaluated methods, with gains of 0.47–1.09 percentage points over the strongest baselines. Distribution diagnostics reveal changes in supervision even at common boundaries. A complementary phi-4 (14B) comparison reports a higher mean for the full construction than for teacher-only covering. The results support shared events with explicit local conditioning as a useful interface for cross-tokenizer supervision.

1 INTRODUCTION

On-policy distillation (OPD) trains a student with teacher feedback on its own generated prefixes (Agarwal et al., 2024; Gu et al., 2024). Beyond teacher-written responses (Kim & Rush, 2016), distribution matching transfers preferences among alternative continuations. Across tokenizers, these alternatives use different prediction units: a word in one vocabulary may span several tokens in another. Supervision therefore requires a common interpretation.

Shared token identities offer one direct correspondence, but their coverage is incomplete and asymmetric. Figure 1 shows that an intersection can contain most teacher token types while covering much less of the student's inventory. Qwen3-32B $\rightarrow$ Gemma-2-2B-it has the lowest coverage on both sides among our three pairs and smaller gains under existing methods. This observation motivates examining supervision beyond exact token matches.

Dual-Space Knowledge Distillation (DSKD) uses learned projections, Approximate Likelihood Matching (ALM) matches aligned likelihoods, Simple Cross-Tokenizer OPD (SimCT) scores continuations, and Byte-Prefix Marginalization (BPM) maps byte prefixes to extend correspondence beyond shared tokens (Zhang et al., 2024; Minixhofer et al., 2025; Sun et al., 2026; Wang et al., 2026). These constructions raise two questions: *what future event should the models predict, and how should each model condition that prediction on the observed bytes?* A common output representation alone leaves the second question open.

Consider a local prefix `a`. It can be the complete token `a`, or the beginning of a longer token such as `ab`. The native next-token row after `a` conditions on the token having ended. A prediction conditioned on a covering token beginning with those bytes also includes the longer alternative. This distinction exists even at a boundary shared by the two observed tokenizations: agreement on the realized segmentation does not eliminate other compatible native tokens.

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.

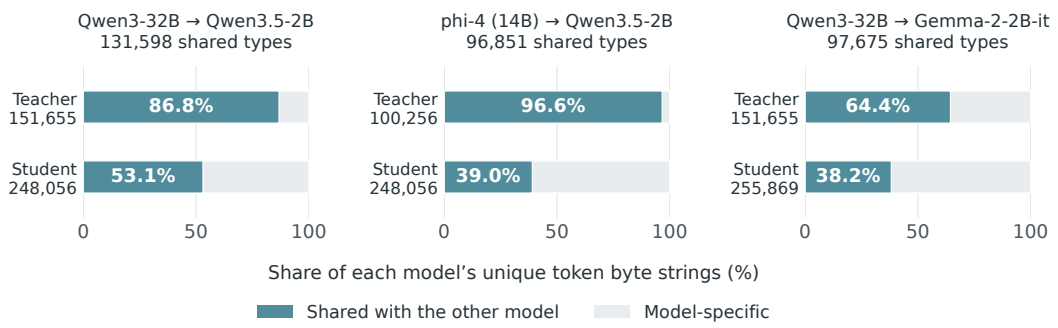

Figure 1: **Shared-token coverage is incomplete and asymmetric.** Exact overlap of unique token byte strings for our three teacher–student pairs, after excluding tokenizer-declared special tokens and collapsing byte aliases. Each bar uses its own model's vocabulary as the denominator; labels at left give that vocabulary size. Shared types cover 64.4–96.6% of teacher vocabularies but only 38.2–53.1% of student vocabularies. These are unweighted type counts, not probability mass or a measured loss of supervision.

We introduce Shared-Event Alignment via Marginalization (SEAM) to connect a common observation with an explicit local conditional. Both models predict the next byte or a control event through their own native vocabularies. Compatible tokens that extend across the query contribute the next byte directly; those ending there contribute their continuation distribution. SEAM marginalizes this covering decision on both sides, keeping earlier native histories fixed, and matches the resulting event distributions at student boundaries using Jensen–Shannon divergence (JSD). The construction uses existing language-model heads, reuses adjacent student rows, and limits teacher lookahead through a probability-mass threshold.

We evaluate transfer across three model pairs, with SEAM achieving the highest five-task mean in each. Distribution diagnostics measure the effect of covering on supervision at common boundaries. Gemma-2-2B-it ablations compare conditional distribution construction, supervised boundary coverage, and loss weighting within one event alphabet; a phi-4 (14B) comparison evaluates the full construction against teacher-only covering. These analyses examine both downstream performance and the design choices underlying the shared-event interface.

Our contributions are: (i) a shared-event supervision interface that makes local boundary conditioning explicit on both teacher and student; (ii) its probability identities, native-logit gradients, and teacher-lookahead error bound; and (iii) an empirical study combining three-pair benchmark results, distribution diagnostics, Gemma-2-2B-it objective comparisons, and a complementary phi-4 (14B) comparison.

2 Background and Motivation

Limited and asymmetric vocabulary overlap. Figure 1 measures exact overlap between unique non-special token byte strings. For inventories B_T and B_S, teacher and student coverage are $|B_T \cap B_S|/|B_T|$ and $|B_T \cap B_S|/|B_S|$. Coverage is 86.8%/53.1% for Qwen3-32B → Qwen3.5-2B, 96.6%/39.0% for phi-4 (14B) → Qwen3.5-2B, and 64.4%/38.2% for Qwen3-32B → Gemma-2-2B-it. Gemma-2-2B-it thus has the lowest coverage on both sides, while phi-4 (14B) shows why low student coverage alone is insufficient to characterize transfer. These unweighted type counts describe the available exact correspondences. Appendix D.1 gives the counting protocol.

Extending supervision beyond exact matches. Classical distillation compares probabilities over common labels (Hinton et al., 2015). Shared-token OPD applies this idea to the vocabulary intersection. Existing cross-tokenizer methods also use information outside that intersection: Universal Logit Distillation (ULD) matches probability ranks, DSKD learns cross-space projections, and ALM matches aligned likelihoods (Boizard et al., 2025; Zhang et al., 2024; Minixhofer et al., 2025). General On-Policy Logit Distillation (GOLD) combines alignment and distribution matching, while

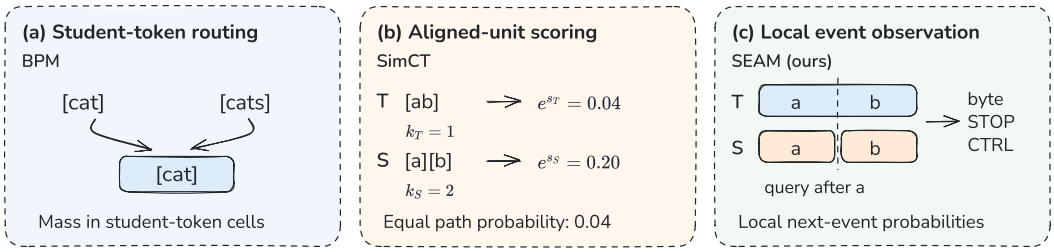

Figure 2: **Different interfaces compare different prediction objects.** (a) BPM routes teacher mass to student-token cells; the toy student vocabulary contains `cat` but not `cats` (Wang et al., 2026). (b) SimCT scores shared tokens and aligned units using $s_M(u) = k_M^{-1} \sum_j \log p_M(v_j \mid \cdot)$, then normalizes over candidates (Sun et al., 2026). Equal prescribed-path probabilities can yield different scores when token counts differ. (c) SEAM compares local next-byte/control observations at student queries. Paths and probabilities are illustrative; method-specific handling is omitted.

SimCT augments shared tokens with short continuations (Patiño et al., 2025; Sun et al., 2026). Figure 2 compares these interfaces through their prediction objects.

Routing and conditioning. BPM routes teacher mass to the longest matching student vocabulary prefix, with residual handling (Wang et al., 2026). This extends correspondence into student-token coordinates. Longest-prefix routing can differ from ranked byte-pair encoding (BPE), which applies merges in priority order (Sennrich et al., 2016); Appendix D gives an example. A separate choice remains even with exact first-token routing: a mapped next-token distribution retains the assumption that the preceding native token ended. Accounting for compatible tokens that cross the query requires marginalizing the covering decision.

Scoring and event probabilities. Continuation scoring defines coordinates through its candidate set and scoring rule. Prefix events for candidates `a` and `ab`, for example, overlap; length normalization further transforms path likelihoods. These scores can be useful learning targets without representing disjoint native observations. SEAM instead specifies an observation with mutually exclusive outcomes—the next byte or a control event—and computes its local conditional probability. This separates the shared output representation from the boundary assumption used to evaluate it.

Relation to byte interfaces and marginalization. Bytes provide a vocabulary-independent alphabet (Xue et al., 2022). Exact byte-level probabilities and token-to-character language models establish marginalization as a way to remove segmentation dependence (Phan et al., 2025; Vieira et al., 2025); tokenization bias also arises under text-prefix conditioning (Phan et al., 2024). Byte-level distillation heads and cross-tokenizer likelihood scoring are closely related alternatives (Singh et al., 2026; Phan et al., 2026). Building on this work, SEAM studies a local covering conditional at student OPD queries, its symmetric realization through existing native logits, and its empirical behavior as supervision.

3 Shared-Event Alignment via Marginalization

SEAM defines a shared next-byte/control observation, computes its local covering conditional through each native vocabulary, and matches the event distributions on student-generated responses. Appendix B provides formal statements and derivations.

3.1 Native histories and a shared observation

Let T denote a frozen teacher and S_θ the student. A student rollout supplies response bytes y together with the sampled native token sequence. The teacher encodes the same response using its own tokenizer and task template. These two native histories define queries at response start and subsequent student content-token boundaries. Length-truncated responses are discarded before

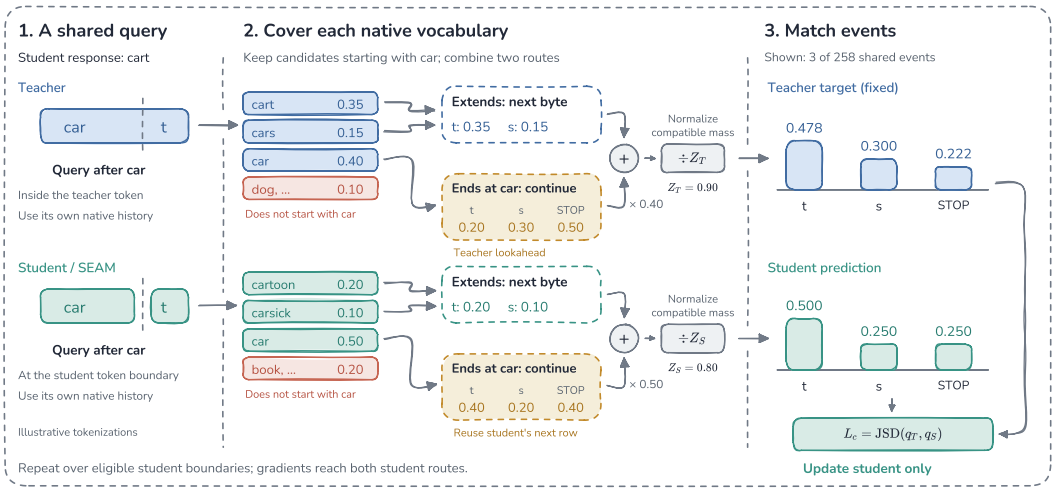

Figure 3: **SEAM illustrated on "cart".** After `car`, the query lies inside the teacher token and at a student boundary. Each side conditions its native candidate mass on this prefix. Extending tokens contribute their next byte; an exact `car` token contributes its weighted continuation. Summing and renormalizing produces distributions on the same events despite different vocabularies. The teacher uses lookahead; the student reuses its next row. JSD updates both student routes with the teacher fixed. Tokenizations and probabilities are illustrative; only three of 258 events have mass here. Algorithms 1–2 specify thresholding, aliases, and response-token normalization.

optimization. For retained responses, a terminal query is included when generation ends with a stop event.

For model $M \in \{T, S\}$, let $b_M(v)$ be the lossless bytes of content token v. Non-content rows have declared roles STOP or CTRL. The observation alphabet is

$$\mathcal{E} = \{0, \ldots, 255\} \cup \{\mathrm{STOP}, \mathrm{CTRL}\}. \tag{1}$$

Each content token is nonempty. Its first byte determines a root event, whereas a stop or control row determines its corresponding atom. The root distribution at native history h is

$$R_M(e \mid h) = \sum_{v:\, \phi_M(v)=e} p_M(v \mid h), \tag{2}$$

where ϕ_M implements these role and first-byte assignments. It is a full probability distribution over $\mathcal{E}$. A control event ends this observation; it need not end a serving conversation. Lossless byte decoding is essential: singleton Unicode decoding can replace incomplete UTF-8 sequences and change the event labels.

3.2 The covering-token conditional

At byte offset $c > 0$, choose the native token containing byte $c-1$. Let $a_M < c$ be its start, h_M the native history before that token, and $P_M = y[a_M : c]$ the observed local prefix of length ℓ_M. For the student this is its token ending at c; for the teacher the selected token may end at c or continue beyond it. The two anchors can differ.

Define the compatible, extending, and exact-token sets

$$\begin{aligned} \mathcal{C}_M(P) &= \{v : b_M(v) \text{ starts with } P\}, \\ \mathcal{X}_M(P, e) &= \{v \in \mathcal{C}_M(P) : |b_M(v)| > |P|,\ b_M(v)[|P|] = e\}, \\ \mathcal{A}_M(P) &= \{v : b_M(v) = P\}, \qquad Z_M(P) = \sum_{v \in \mathcal{C}_M(P)} p_M(v \mid h_M). \end{aligned} \tag{3}$$

The extending set is empty for control outcomes. With $w_M(v) = p_M(v \mid h_M)/Z_M(P_M)$, the local event marginal is

$$q_M(e; c) = \sum_{v \in \mathcal{X}_M(P_M, e)} w_M(v) + \sum_{v \in \mathcal{A}_M(P_M)} w_M(v) R_M(e \mid h_M v). \tag{4}$$

At $c = 0$ we use Equation 2. The two terms in Figure 3 correspond to the two ways a compatible token reaches the query: it either supplies the next byte directly or ends and invokes a further native model decision.

The two routes produce a normalized distribution over shared events using each model's own compatible tokens. Their coordinates refer to the same observations even without shared teacher–student token identities. The marginal reconsiders one covering decision with earlier native history fixed, which makes the computation local rather than a sum over all segmentations of the response prefix. Appendix B states the assumptions and proves the decomposition.

Relation to ordinary boundary prediction. If $\alpha_M = \sum_{v \in \mathcal{A}_M} w_M(v)$ and there is one exact-token continuation Q_M, then

$$q_M = (1 - \alpha_M) U_M + \alpha_M Q_M, \tag{5}$$

where U_M is the normalized next-byte distribution of the extending tokens. Ordinary next-row prediction uses Q_M alone, whereas the mixture also retains the probability of longer alternatives. Those alternatives can exist at a common boundary. The same construction therefore applies to the student, whose observed token ends at every noninitial query but need not exhaust the compatible covering tokens.

3.3 Symmetric supervision through native outputs

Let D_{KL} denote Kullback–Leibler (KL) divergence. We match teacher events $t = q_T$ and student events $s = q_{S_\theta}$ using JSD:

$$L_c = \mathrm{JSD}(t, s) = \tfrac{1}{2} D_{\mathrm{KL}}(t \| m) + \tfrac{1}{2} D_{\mathrm{KL}}(s \| m), \qquad m = (t + s)/2. \tag{6}$$

The teacher target is fixed during optimization. On the student side, both routes in Equation 4 contribute gradients: compatible-token weights determine how probability is divided between extending and ending routes, while the next row determines continuation after an ending token. This trains the student under the same local event construction used by the teacher, through its existing language-model head. Sampled token IDs and alignment choices remain fixed during the update. Appendix B.5 derives the native-logit gradients.

Teacher computation depends on the query position. At a common boundary, the continuation is available in the native forward pass. At an interior boundary, an exact token spelling P_T requires a separate lookahead after that token. We evaluate it only when its conditional mass α_T exceeds $\tau = 10^{-3}$; otherwise, we use the extending-token distribution U_T. This retains normalized targets and limits extra teacher calls. The approximation bound and proof are in Appendix B.3.

When several native tokens have identical bytes, the implementation aggregates their mass and shares one continuation: the observed token's row on the main path or the highest-logit alias for teacher lookahead. Appendix B.4 specifies this representative policy and its approximation error.

3.4 Implementation and objective normalization

The local decomposition also organizes the computation. Lexicographically sorted token bytes place compatible tokens in a contiguous interval. A prefix lookup identifies $\mathcal{C}_M$, its exact-token subset, and the next-byte groups without constructing a cross-vocabulary matrix; stable softmax within the interval gives w_M. Teacher rows are scored in chunks, and lookahead branches attend to their own native prefixes. The student reuses adjacent causal rows from its full forward pass.

For a batch of responses with eligible boundary sets $\mathcal{B}_i$, the objective is

$$\mathcal{L}(\theta) = \frac{\sum_i \sum_{c \in \mathcal{B}_i} L_c(\theta)}{N_{\mathrm{response}}}. \tag{7}$$

Here $N_{\text{response}} = \sum_i T_i$, where T_i is the number of native student tokens in retained response i of the optimizer batch. It includes an emitted stop token and excludes prompt and padding tokens. Discarded length-truncated responses contribute neither to the loss numerator nor to this token count. This count is taken before selecting supervised boundaries and does not use teacher-token lengths. All-boundary and common-boundary variants use the same definition, so expanding the query set changes the total supervision weight for a fixed response. The equal-weight control scales each response's all-boundary sum by its common-to-all boundary-count ratio. Appendix C details the distributed reduction.

We evaluate event losses in chunks and accumulate their logit gradients before the ordinary student backward pass, reducing temporary autograd storage. Appendix B.6 gives the surrogate construction and its gradient-equivalence argument. Probability aggregation uses double precision and model execution uses bfloat16. Appendix C gives the complete update in Algorithm 1 and the event-marginal routine in Algorithm 2.

4 Experiments

We evaluate the interface at three levels: downstream transfer across model pairs, changes in the supervised event distributions, and alternative event objectives. Gemma-2-2B-it provides comparisons of conditioning, boundary coverage, and loss weighting, and phi-4 (14B) provides an additional comparison of the full and teacher-only constructions.

4.1 Experimental setup

We study Qwen3-32B $\rightarrow$ Qwen3.5-2B, phi-4 (14B) $\rightarrow$ Qwen3.5-2B, and Qwen3-32B $\rightarrow$ Gemma-2-2B-it (Yang et al., 2025; Abdin et al., 2024; Qwen Team, 2026; Gemma Team, 2024). Within each pair, methods share a teacher-specific supervised fine-tuning (SFT) initialization; all Gemma-2-2B-it methods use the same lossless byte codec. Training uses 10,000 mixed mathematics/programming prompts, two prompt passes, effective batch 256, AdamW at 10^{-6}, cosine decay, and 5% warmup. Student rollouts use temperature 0.6, top-$p = 0.95$, a 4,096-token response limit, and nonthinking mode.

Each method starts from the same SFT checkpoint within a pair and receives frozen-teacher supervision on responses generated by its current student. The prompt population and rollout settings are shared. Responses truncated at the maximum response length are discarded, and an update uses 256 retained responses. Appendix E specifies the training configurations and response-filtering rule.

We compare BPM, DSKD, GOLD, and SimCT under native on-policy transport, retaining their method-specific objectives and normalizers. Appendix E describes the baseline formulations and training settings; Appendix F.4 discusses transfer patterns for Gemma-2-2B-it.

Evaluation covers Grade School Math (GSM8K; 1,319 examples), MATH-500 (500), Mostly Basic Programming Problems (MBPP; 257), LiveCodeBench-v6 (LCB-v6; 1,055), and Topics in Algorithmic Code Generation (TACO; Easy100+Medium100, 200) (Cobbe et al., 2021; Hendrycks et al., 2021; Austin et al., 2021; Jain et al., 2024; Li et al., 2023). The primary aggregate is the equally weighted five-task average (Avg5). Table 1 gives the main comparison, and Table 2 reports the component controls. A separate 800-example, four-stratum validation panel tracks training trajectories. Appendix F provides supplementary transfer analysis and validation trajectories.

4.2 Transfer across vocabulary-overlap settings

Table 1 reports Avg5 scores of 47.24%, 45.76%, and 31.70% for SEAM. These exceed the strongest baselines by 0.47, 0.58, and 1.09 percentage points (pp), respectively: BPM is the strongest baseline for Qwen3-32B $\rightarrow$ Qwen3.5-2B and Qwen3-32B $\rightarrow$ Gemma-2-2B-it, and SimCT for phi-4 (14B) $\rightarrow$ Qwen3.5-2B. SEAM improves over the corresponding SFT starts by 2.85, 3.24, and 2.03 pp, and over Base by 4.19, 2.71, and 2.53 pp.

Transfer under lower observed vocabulary coverage. Qwen3-32B $\rightarrow$ Gemma-2-2B-it has the lowest teacher- and student-side type coverage in Figure 1. GOLD, DSKD, SimCT, and BPM change Avg5 over SFT by +0.53, -0.73, +0.64, and +0.94 pp, less than their gains in either Qwen3.5-2B

Table 1: **Cross-tokenizer distillation performance.** Task pass@1 (%), mean ± sample standard deviation (SD) for decoding seeds 42–46 (Appendix E.6). Average is the five-task mean. Best and second-best means within each group are **bold** and underlined, including ties. SFT rows are gray; SEAM rows are blue.

Model	Method	Math		Code			Average
		GSM8K	MATH-500	MBPP	LCB-v6	TACO	
Qwen3-32B	Teacher	95.15 ± 0.22	84.76 ± 0.90	83.19 ± 0.84	43.03 ± 0.41	20.90 ± 1.52	65.41
phi-4 (14B)	Teacher	92.80 ± 0.16	81.28 ± 1.00	78.83 ± 1.15	31.17 ± 0.77	26.00 ± 1.46	62.02
Qwen3.5-2B	Base	74.68 ± 1.05	69.52 ± 1.71	52.22 ± 1.39	14.01 ± 0.74	4.80 ± 0.67	43.05
Gemma-2-2B-*it*	Base	61.76 ± 0.63	25.88 ± 1.57	47.08 ± 1.01	6.65 ± 0.62	4.50 ± 0.57	29.17
Qwen3-32B teacher ↓ Qwen3.5-2B student	SFT	77.57 ± 0.43	69.92 ± 1.60	53.31 ± 0.83	15.53 ± 0.50	5.60 ± 1.98	44.39
	GOLD	83.00 ± 0.55	70.92 ± 1.29	53.93 ± 1.34	16.89 ± 0.85	**8.20** ± 1.25	46.59
	DSKD	81.26 ± 0.46	68.60 ± 1.13	55.10 ± 0.85	15.68 ± 0.36	5.90 ± 0.65	45.31
	SimCT	82.61 ± 0.57	69.48 ± 0.89	55.56 ± 1.33	16.82 ± 0.44	6.00 ± 1.27	46.09
	BPM	83.06 ± 0.96	71.92 ± 1.20	55.10 ± 2.31	**17.06** ± 0.86	6.70 ± 0.45	46.77
	SEAM	**83.15** ± 0.34	**72.60** ± 0.96	**56.42** ± 0.73	16.72 ± 0.83	7.30 ± 1.40	**47.24**
phi-4 (14B) teacher ↓ Qwen3.5-2B student	SFT	76.82 ± 0.92	64.76 ± 1.28	51.21 ± 0.52	13.52 ± 0.95	6.30 ± 1.15	42.52
	GOLD	80.53 ± 0.67	67.56 ± 0.67	53.85 ± 1.66	15.87 ± 1.08	6.80 ± 1.48	44.92
	DSKD	78.39 ± 0.40	67.04 ± 0.95	51.21 ± 2.00	15.49 ± 0.72	5.70 ± 0.57	43.57
	SimCT	80.83 ± 0.53	66.24 ± 1.06	**55.95** ± 1.39	**16.27** ± 0.78	6.60 ± 0.82	45.18
	BPM	**81.35** ± 0.54	**68.28** ± 1.03	53.62 ± 2.44	15.39 ± 0.42	6.00 ± 1.54	44.93
	SEAM	81.17 ± 0.52	67.96 ± 0.82	55.72 ± 0.84	15.83 ± 0.50	**8.10** ± 0.82	**45.76**
Qwen3-32B teacher ↓ Gemma-2-2B-*it* student	SFT	59.70 ± 0.48	27.96 ± 0.88	49.57 ± 1.69	6.43 ± 0.23	4.70 ± 0.57	29.67
	GOLD	61.47 ± 0.96	28.80 ± 2.13	49.81 ± 1.74	6.54 ± 0.52	4.40 ± 0.42	30.20
	DSKD	56.86 ± 1.45	28.32 ± 1.55	48.17 ± 1.01	7.45 ± 0.37	3.90 ± 0.65	28.94
	SimCT	60.36 ± 0.94	28.76 ± 1.53	50.74 ± 0.44	6.90 ± 0.39	4.80 ± 0.76	30.31
	BPM	62.55 ± 1.99	30.12 ± 0.23	49.81 ± 2.22	6.67 ± 0.66	3.90 ± 0.65	30.61
	SEAM	**63.82** ± 1.01	**30.64** ± 1.47	**50.97** ± 0.95	**8.19** ± 0.42	**4.90** ± 0.96	**31.70**

setting. SEAM reaches +2.03 pp and has its largest margin over the strongest baseline here. Its own SFT-relative gain is still smaller than in the other two pairs. Gemma-2-2B-it thus shows the largest SEAM margin over the strongest baseline, despite smaller absolute transfer gains across methods.

Where the improvement occurs. SEAM has the highest task mean in nine of the fifteen pair–benchmark combinations. Against the baselines in Table 1, it leads all five Gemma-2-2B-it tasks and improves over Base by +2.06 pp on GSM8K, +4.76 on MATH-500, +3.89 on MBPP, +1.54 on LCB-v6, and +0.40 on TACO. Under Qwen3-32B → Qwen3.5-2B, it leads GSM8K, MATH-500, and MBPP, while GOLD and BPM lead TACO and LCB-v6. Under phi-4 (14B), its largest contribution over competing methods is on TACO. The average advantage therefore combines different task profiles across model pairs. Appendix I illustrates reasoning and code-generation behavior in one example per pair.

Adaptation trajectories. Appendix F.1 shows validation trajectories across the three model pairs. Under the shared response-filtering rule and two-pass prompt regime, Gemma-2-2B-it runs complete 78 updates for SEAM and DSKD, 66 for BPM, 71 for SimCT, and 77 for GOLD. All five methods complete 78 updates in each of the two Qwen3.5-2B settings. Appendix E specifies the filtering rule and batch construction.

Qualitative behavior. In the GSM8K case, SEAM includes all donations and answers the requested contribution; baselines omit an amount or return another quantity. The LiveCodeBench case distinguishes correct adjacent comparisons and circular indexing from condition and indexing errors. In the MBPP case, SEAM combines descending order with string conversion; the baselines make ordering, type, or variable-reference errors. Appendix I presents all problems and model responses.

4.3 DOES COVERING CHANGE THE SUPERVISION?

The performance comparison establishes a downstream pattern; Figure 4 examines whether the proposed conditional produces a different supervision signal. It compares Equation 5 with ordinary root-event distributions. Mean teacher total variation (TV) distances at common boundaries are 0.00943, 0.00717, and 0.01319 across the three pairs. Student shifts are also nonzero. Covering therefore changes event probabilities even where the realized token boundaries agree.

Student queries fall inside teacher tokens at rates of 5.84%, 9.10%, and 10.29%. Gemma-2-2B-it has the largest interior fraction, teacher shift at common boundaries, and student shift (0.00904). These measurements locate changes in supervision rather than quantify target quality. The following ablations evaluate task outcomes under the corresponding conditioning choices. Appendix G defines the logged statistics.

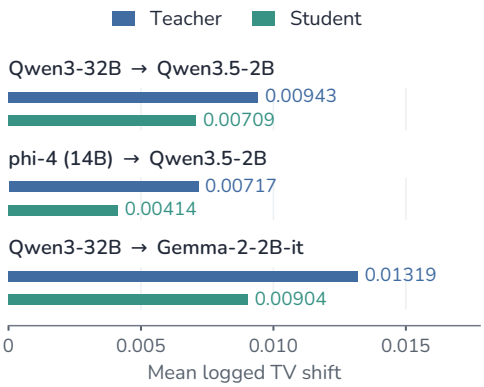

Figure 4: **Covering changes supervision.** Mean per-update TV shifts over 78 SEAM updates. Teacher: common boundaries; student: all supervised boundaries. The two series use different boundary populations.

Lookahead in practice. Mean logged dropped teacher branch mass is 3.03×10^{-6}, 2.13×10^{-6}, and 4.38×10^{-6}. These measurements characterize the threshold approximation analyzed in Appendix B.3; alias and history-conditioning differences remain separate. Appendix H reports numerical checks of the local probabilities and gradients.

4.4 EVENT-DISTRIBUTION AND WEIGHTING CONTROLS

Experimental design. We compare event-distribution and weighting choices in the Qwen3-32B $\rightarrow$ Gemma-2-2B-it setting, using the same SFT initialization, lossless codec, training seed, and prompt regime. The *root* control projects each model's native next-token distribution to shared byte/control events at common boundaries. *Teacher covering* replaces the teacher target with Equation 4; the student retains its root-event distribution. The *symmetric/common* control applies covering to both models at common boundaries. SEAM applies the covering construction to both models at all eligible student boundaries, using the response-token normalizer in Equation 7.

The *equal-weight* control retains SEAM's all-student query rule but scales each response's boundary coefficients by its common-to-all boundary-count ratio. It compares loss weighting under the same query rule. All ablation results are reported over five decoding seeds (42–46) as mean $\pm$ sample standard deviation. Each configuration is trained separately and evaluated at its final checkpoint. Root matching ends at 71 updates and the covering settings at 78. Appendix G, Table 10, reports the GPU-hours for the five configurations.

At common boundaries, replacing the student's root-event distribution with its covering conditional increases Avg5 from 30.58% to 31.04%, providing evidence for student-side covering under the same boundary rule (Table 2). GSM8K, MBPP, LCB-v6, and TACO improve by 1.12, 0.78, 1.15, and 0.40 pp, respectively, while MATH-500 decreases by 1.16 pp.

Full SEAM reaches 31.70% Avg5, compared with 31.04% for symmetric/common and 30.69% for the equal-weight all-boundary control. The latter's lower mean relative to symmetric/common indicates that extending boundary coverage alone does not ensure improvement under the tested weighting rule. Appendix F.2 compares these configurations and their computational costs.

Weighting under a fixed query rule. SEAM and the equal-weight control use the same all-student query rule. SEAM raises Avg5 from 30.69% to 31.70% (+1.01 pp), supporting the response-token-normalized objective without the per-response common-to-all rescaling.

Table 2: **Event-distribution and weighting controls on Gemma-2-2B-it.** Qwen3-32B → Gemma-2-2B-it, training seed 101. Task pass@1 (%), mean ± sample SD over five decoding seeds (42–46); Average is the five-task mean. Best and second-best means are **bold** and underlined, including ties. Root control is gray; SEAM is blue. Root uses 71 updates, others 78. †: one unresolved code execution counts as incorrect.

Method	Boundaries	Math		Code			Average
		GSM8K	MATH-500	MBPP	LCB-v6	TACO	
Root next-row	Common	52.80 ± 1.03	26.68 ± 1.49	47.78 ± 2.03	5.69 ± 0.33	3.70 ± 0.57	27.33
Teacher covering	Common	63.61 ± 1.15	29.96 ± 0.73	48.33 ± 1.24	6.60 ± 0.30	4.40 ± 0.42	30.58
Symmetric covering	Common	**64.73** ± 1.23	28.80 ± 1.03	49.11 ± 0.75	7.75 ± 0.39	4.80 ± 1.30	31.04
Symmetric covering (equal weight)	All student	63.76 ± 0.96	30.04 ± 0.80	48.02 ± 2.01	7.13 ± 0.34	4.50 ± 0.61	30.69
SEAM†	All student	63.82 ± 1.01	**30.64** ± 1.47	**50.97** ± 0.95	**8.19** ± 0.42	**4.90** ± 0.96	**31.70**

A complementary comparison on phi-4 (14B). Appendix F.3 reports 44.80% Avg5 for teacher-only covering and 45.76% for full SEAM, using five decoding seeds (42–46). Both runs use the same SFT initialization and complete 78 updates; teacher covering uses common boundaries and the student's root-event distribution. Full SEAM improves GSM8K, MATH-500, MBPP, LCB-v6, and TACO by 0.31, 0.56, 3.19, 0.11, and 0.60 pp, respectively, yielding a 0.96 pp Avg5 gain.

4.5 Computational cost

The shared event alphabet avoids a cross-vocabulary matching matrix, but covering still requires native-model computation. Recorded training-core costs are 11.99, 7.04, and 19.33 GPU-hours for SEAM across the three pairs, versus 3.65, 3.38, and 10.74 for BPM. They include generation, teacher scoring, student updates, and weight synchronization; checkpointing and evaluation are excluded. Appendix G reports all methods and controls. These trajectory-dependent totals characterize the observed accuracy–compute tradeoff.

For the Gemma-2-2B-it ablations, symmetric/common, equal-weight/all, and full SEAM cost 12.50, 14.14, and 19.33 GPU-hours, respectively. Full SEAM gains 0.66 pp Avg5 over symmetric/common: MATH-500, MBPP, LCB-v6, and TACO increase by 1.84, 1.86, 0.44, and 0.10 pp, while GSM8K decreases by 0.91 pp. The full configuration therefore attains the highest benchmark mean at additional compute cost; common-boundary covering offers a lower-cost alternative. All three configurations complete 78 updates on two H200 GPUs. Full SEAM also has the highest validation endpoint (39.00%), compared with 38.50% and 38.38% for symmetric/common and equal-weight/all, respectively (Appendix F.2).

5 Limitations and Conclusion

SEAM formulates cross-tokenizer supervision through a shared next-byte/control observation and an explicit local conditional on both teacher and student. Across three model pairs, it achieves the highest reported five-task mean among the evaluated methods. Diagnostics show that covering changes the supervised distributions, Gemma-2-2B-it ablations compare conditioning, boundary coverage, and loss weighting, and the phi-4 (14B) comparison adds evidence about the full construction relative to teacher-only covering. These results support treating event correspondence and local conditioning jointly when designing a distillation interface.

The study covers three model pairs and one training seed for the five-task benchmarks; the five decoding repeats measure generation variability. Vocabulary-overlap observations do not isolate its effect from differences between model pairs. Independent training repetitions and broader model comparisons would further characterize the robustness of the observed gains. The probability construction marginalizes one local covering decision, with the alias and lookahead approximations specified in the method.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript writing and polishing, including refining wording, improving clarity, and organizing the exposition. They were also used for literature retrieval and discovery, such as identifying potentially relevant related work. The authors take full responsibility for the accuracy of the manuscript, the appropriateness of its citations, and all scientific claims and conclusions.

REPRODUCIBILITY STATEMENT

Appendix A specifies the event and conditioning assumptions; Appendix B provides proofs and Appendix C gives pseudocode and implementation details. Model initialization, training configurations, evaluation populations, and checkpoint rules are given in Appendix E. Appendix F discusses transfer behavior and validation trajectories, and Appendix G defines the logged diagnostics and timing measurements.

ETHICS STATEMENT

The study concerns distillation of existing language models on mathematics and programming tasks. It does not evaluate the transfer of safety behavior, demographic fairness, or robustness to malicious instructions. A student that improves on these task metrics can still inherit undesirable teacher behavior; the reported results should not be interpreted as evidence of deployment safety.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024. URL https://arxiv.org/abs/2412.08905.

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.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

Nicolas Boizard, Kevin El Haddad, Céline Hudelot, and Pierre Colombo. Towards cross-tokenizer distillation: The universal logit distillation loss for LLMs. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=bwRxXiGO9A.

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

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

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://openreview.net/forum?id=5h0qf7IBZZ.

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. *arXiv preprint arXiv:2103.03874*, 2021. URL https://arxiv.org/abs/2103.03874.

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.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024. URL https://arxiv.org/abs/2403.07974.

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. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. TACO: Topics in algorithmic COde generation dataset. *arXiv preprint arXiv:2312.14852*, 2023. URL https://arxiv.org/abs/2312.14852.

Benjamin Minixhofer, Ivan Vulić, and Edoardo Maria Ponti. Universal cross-tokenizer distillation via approximate likelihood matching. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/720f9f5dc751eb56952ae4fee2398f73-Abstract-Conference.html.

Carlos Miguel Patiño, Kashif Rasul, Quentin Gallouédec, Ben Burtenshaw, Sergio Paniego, Vaibhav Srivastav, Thibaud Frere, Ed Beeching, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Unlocking on-policy distillation for any model family. Hugging Face Research Article, 2025. URL https://huggingfaceh4-on-policy-distillation.hf.space/. Includes the June 2026 byte-offset alignment update.

Buu Phan, Marton Havasi, Matthew Muckley, and Karen Ullrich. Understanding and mitigating tokenization bias in language models. *arXiv preprint arXiv:2406.16829*, 2024. URL https://arxiv.org/abs/2406.16829.

Buu Phan, Brandon Amos, Itai Gat, Marton Havasi, Matthew J. Muckley, and Karen Ullrich. Exact byte-level probabilities from tokenized language models for FIM-tasks and model ensembles. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zGej22CBnS.

Buu Phan, Ashish Khisti, and Karen Ullrich. Cross-tokenizer likelihood scoring algorithms for language model distillation. *arXiv preprint arXiv:2512.14954*, 2026. URL https://arxiv.org/abs/2512.14954v2. Version 2; first posted December 2025.

Qwen Team. Qwen3.5-2B model card. Hugging Face model repository, 2026. URL https://huggingface.co/Qwen/Qwen3.5-2B.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1715–1725. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162/.

Avyav Kumar Singh, Yen-Chen Wu, Alexandru Cioba, Alberto Bernacchia, and Davide Buffelli. Cross-tokenizer LLM distillation through a byte-level interface. *arXiv preprint arXiv:2604.07466*, 2026. URL https://arxiv.org/abs/2604.07466.

Jie Sun, Mao Zheng, Mingyang Song, Qiyong Zhong, Yilin Cheng, Bichuan Feng, Pengfei Liu, Junfeng Fang, and Xiang Wang. SimCT: Recovering lost supervision for cross-tokenizer on-policy distillation. *arXiv preprint arXiv:2605.07711*, 2026. URL https://arxiv.org/abs/2605.07711.

Tim Vieira, Ben LeBrun, Mario Giulianelli, Juan Luis Gastaldi, Brian DuSell, John Terilla, Timothy J. O'Donnell, and Ryan Cotterell. From language models over tokens to language models over characters. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2412.03719.

Hao Wang, Kun Yuan, Wenlin Zhong, Minglei Zhang, Han Xiao, Ming Sun, and Honggang Qi. Cross-tokenizer on-policy distillation via byte-prefix marginalization. *arXiv preprint arXiv:2607.22334*, 2026. URL https://arxiv.org/abs/2607.22334.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306, 2022. doi: 10.1162/tacl_a_00461. URL https://aclanthology.org/2022.tacl-1.17/.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. Dual-space knowledge distillation for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18164–18181. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.1010. URL https://aclanthology.org/2024.emnlp-main.1010/.

A Full Setup and Conditioning Assumptions

The supplementary material gives the shared-event assumptions, local probability and gradient proofs, and implementation and evaluation details. It combines Gemma-2-2B-it component evidence with the phi-4 (14B) comparison and recorded distribution diagnostics and costs. The theory concerns one covering-token decision with earlier native history fixed.

Content and controls. A model has a finite semantic output vocabulary V_M. Each legal row has exactly one role: a nonempty byte string, STOP, or CTRL. Roles are determined before probability aggregation; printable spellings of special tokens do not by themselves determine their semantics. Byte-bearing reasoning delimiters can remain content. A root observation maps every row to precisely one of 258 events. When a physical language-model head contains padding rows outside the semantic vocabulary, the event projection normalizes over the semantic vocabulary. This restriction is part of the declared model interface.

Codec contract. For eligible content sequences, decoding to bytes equals concatenating the per-token bytes. Byte-fallback tokens must retain their raw bytes, including bytes that cannot be decoded as standalone UTF-8 characters. Position alignment is performed on this byte stream. Histories containing literal control spellings that violate the transport contract are excluded by the native transport; no such skipped literal samples are recorded in the three principal SEAM seed-101 runs. Gemma-2-2B-it experiments in the main comparison use the same lossless byte codec for the proposed method and its controls.

History contract. The student response retains its generated token IDs. The teacher representation uses its native tokenization of the same response content. The prompt is rendered with the relevant native task template. A selected boundary fixes an earlier native history h_M, local prefix P_M, and compatible-token set $\mathcal{C}_M(P_M)$. These objects are held fixed during differentiation. They need not be identical between teacher and student. Their common element is the future event alphabet and the observed response content.

Local conditioning contract. The random experiment draws the next native token at h_M, conditional on its bytes starting with P_M. Thus it conditions on the next token being at least $|P_M|$ bytes long and on its first $|P_M|$ bytes. This is a local one-cover experiment. A full prefix likelihood would also sum native sequences whose first token ends before $|P_M|$, along with all compatible earlier histories. The distinction prevents the local identity from being misread as an unrestricted byte-prefix posterior.

Continuation and support contract. A native exact-token marginal uses the appropriate future state for each alias in $\mathcal{A}_M(P_M)$. Histories that emit the same bytes can still have different hidden states. The representative policy used in training is defined in Appendix B.4. Compatible sets have positive mass because the observed covering token is a member and its finite-logit probability is positive. Standard zero-mass conventions apply to JSD: $0 \log 0 = 0$. The synthetic derivatives use strictly positive probabilities; finite softmax logits provide positive mass for supported events.

B Probability Identities and Native Gradients

B.1 Local event probabilities

Proposition B.1 (Local event probabilities)**.** *Assume a fixed native history, a lossless nonempty content codec, $Z_M(P_M) > 0$, and the specified native continuation distribution for every exact-token branch. Equation 4 is the distribution of the next-byte/control observation conditional on the next native token beginning with P_M. Its coordinates are mutually exclusive, exhaustive, and sum to one.*

Proof. Fix M, h, P and condition on $C = \{v : b_M(v) \text{ starts with } P\}$. The native token distribution becomes $w_v = p_M(v \mid h)/Z$, $v \in C$, with $Z = \sum_{v \in C} p_M(v \mid h)$. The compatible set is the disjoint union of the exact set $A = \{v : b_M(v) = P\}$ and extending sets X_e for byte events e.

If $v \in X_e$, the next byte after the observed prefix is already determined: its event kernel is $K_v(e') = \mathbf{1}\{e' = e\}$. If $v \in A$, the token ends at the observation point. The subsequent event kernel is $K_v(e') = R_M(e' \mid hv)$. Both kinds of kernel are probability distributions. The event distribution is therefore

$$q(e) = \sum_{v \in C} w_v K_v(e) = \sum_{v \in X_e} w_v + \sum_{v \in A} w_v R_M(e \mid hv). \tag{8}$$

Every realization determines exactly one event: one next byte or one control atom. Distinct event labels cannot occur in the same realization. Summing gives

$$\sum_{e \in \mathcal{E}} q(e) = \sum_{v \in C} w_v \sum_{e} K_v(e) = \sum_{v \in C} w_v = 1. \tag{9}$$

Token aliases remain distinct v values in this identity. No longest-prefix routing rule or shared tokenizer is required. □

B.2 Worked covering-token example

Figure 3 uses the response `cart` with illustrative teacher tokenization [`cart`] and student tokenization [`car`][`t`]. The query after `car` asks for the next byte or control event. Thus teacher token `cart` and student token `cartoon` both contribute to event `t`, while `cars` and `carsick` contribute to event `s`. An exact `car` token contributes through its subsequent native event distribution.

At the teacher anchor, let $p_T(\texttt{car}, \texttt{cart}, \texttt{cars}) = (0.40, 0.35, 0.15)$; the remaining 0.10 is on tokens not starting with `car`. The compatible mass is $Z_T = 0.90$, and the conditional ending mass is $\alpha_T = 0.40/0.90$. Suppose the continuation after exact token `car` assigns $(0.20, 0.30, 0.50)$ to $(\texttt{t}, \texttt{s}, \text{STOP})$. Equation 8 gives

$$\begin{aligned} q_T(\texttt{t}) &= (0.35 + 0.40 \cdot 0.20)/0.90 = 43/90, \\ q_T(\texttt{s}) &= (0.15 + 0.40 \cdot 0.30)/0.90 = 0.30, \\ q_T(\text{STOP}) &= (0.40 \cdot 0.50)/0.90 = 2/9. \end{aligned} \tag{10}$$

The ordinary next row $(0.20, 0.30, 0.50)$ is a different conditional distribution: it fixes the preceding token to `car`, whereas covering also includes compatible longer tokens.

For the student, let $p_S(\texttt{car}, \texttt{cartoon}, \texttt{carsick}) = (0.50, 0.20, 0.10)$, with remaining incompatible mass 0.20, and let its continuation after `car` be $(0.40, 0.20, 0.40)$. Its compatible mass is $Z_S = 0.80$, giving

$$q_S(\texttt{t}, \texttt{s}, \text{STOP}) = \frac{(0.20, 0.10, 0) + 0.50(0.40, 0.20, 0.40)}{0.80} = (0.50, 0.25, 0.25). \tag{11}$$

Both distributions sum to one. All other byte events and CTRL have zero mass in this illustrative example; the actual event space remains 258-dimensional. The teacher interior continuation uses lookahead, while the student reuses its main next row. Figure 3 rounds the teacher probabilities to three decimals. The tokenizations and probabilities explain the construction and are not measurements of an experimental model.

B.3 Thresholded lookahead and its error bound

Proposition B.2 (Thresholded lookahead)**.** *For the mixture in Equation 5, replacing q_M by U_M incurs $\mathrm{TV}(q_M, U_M) = \alpha_M \,\mathrm{TV}(Q_M, U_M) \le \alpha_M$. Thus a skipped branch with $\alpha_M \le \tau$ changes the target by at most τ in total variation.*

Proof. Write $q = (1 - \alpha)U + \alpha Q$, where Q is the exact-token branch mixture. Replacing the exact branch by renormalized extending mass yields $\widetilde{q} = U$. Hence

$$\mathrm{TV}(q, \widetilde{q}) = \tfrac{1}{2} \sum_{e} |\alpha(Q(e) - U(e))| = \alpha \,\mathrm{TV}(Q, U) \le \alpha. \tag{12}$$

□

A skip decision is used only when $\alpha \leq \tau$. If $\alpha = 1$, a branch is necessary; renormalizing an empty extending set would be invalid. At a common boundary the main teacher row supplies the representative exact-token continuation without an extra branch. The threshold applies to interior lookahead.

The bound is a probability error bound. It is not a bound on the final task-score change, the parameter-update norm, or the difference between local and full-history conditionals. In particular, small target total variation does not provide a uniform bound on log-probability derivatives near zero student mass.

B.4 Alias continuations and the implementation policy

Let A contain several native tokens with identical bytes. Define $\alpha = \sum_{v \in A} w_v$ and $\pi_v = w_v/\alpha$. The exact continuation mixture is $Q = \sum_{v \in A} \pi_v R_M(\cdot \mid hv)$. Equal emitted bytes do not entail equal continuation distributions: the transformer state depends on native token IDs.

The implementation aggregates exact-token mass and uses one representative continuation $\widehat{Q}$. At a main-path boundary this is the next row following the observed token; at an interior teacher boundary it is the lookahead after the highest-logit exact-byte alias. All candidate weights, including those of aliases, remain in α. The resulting $\widehat{q} = (1-\alpha)U + \alpha\widehat{Q}$ is a normalized event law for that representative-kernel experiment. Relative to the exact native alias mixture,

$$\mathrm{TV}(q, \widehat{q}) = \alpha\, \mathrm{TV}(Q, \widehat{Q}) \leq \sum_{v \in A} w_v\, \mathrm{TV}\big(R_M(\cdot \mid hv), \widehat{Q}\big). \tag{13}$$

The right side vanishes for a singleton alias class or identical alias continuations. This alias term is distinct from threshold truncation. The logged dropped-mass diagnostic measures the latter only. Equation 13 makes the assumptions behind exact native probabilities explicit while describing the actual representative computation used for training.

B.5 Derivatives of JSD and conditional softmax

Teacher targets and sampled histories are fixed during differentiation. Student probabilities remain differentiable through the logits before the covering token and those after it. In the representative form, write $s = \sum_v w_v K_v$, with K_v a one-hot event for an extending token or the next-row distribution q for an exact token. For $g_e = \partial L_c/\partial s_e = \frac{1}{2}\log(s_e/m_e)$,

$$\frac{\partial L_c}{\partial z_v} = w_v\big(\langle g, K_v\rangle - \langle g, s\rangle\big), \qquad \frac{\partial L_c}{\partial u_e} = \alpha q_e\big(g_e - \langle g, q\rangle\big), \tag{14}$$

where z are compatible-token logits and u are event logits for the next-row distribution. Actual next-token logits receive the latter derivative through the native softmax and event aggregation. No gradient through discrete tokenization or rollout sampling is required. The following derivation also gives the native-vocabulary form.

With fixed teacher t and $m = (t+s)/2$, expand

$$L = \frac{1}{2}\sum_e t_e \log\frac{t_e}{m_e} + \frac{1}{2}\sum_e s_e \log\frac{s_e}{m_e}. \tag{15}$$

Differentiation of both terms, including m's dependence on s, gives $g_e = \partial L/\partial s_e = \frac{1}{2}\log(s_e/m_e)$. For fixed continuation kernels K_v, $s = \sum_v w_v K_v$ and $w = \mathrm{softmax}(z|_C)$. Applying the softmax Jacobian gives

$$\frac{\partial L}{\partial z_v} = w_v\big(\langle g, K_v\rangle - \langle g, s\rangle\big), \quad v \in C, \tag{16}$$

and zero for incompatible covering-row logits. This conditional softmax is algebraically identical to full-softmax probabilities divided by their compatible mass Z; the full-vocabulary normalizer cancels.

For the representative exact branch, let p^+ denote next-row native probabilities and $q_e = \sum_{j:\phi(j)=e} p^+_j$. With $s = n + \alpha q$ and extending mass n, the native next-row derivative is

$$\frac{\partial L}{\partial z^+_j} = \alpha p^+_j \left(g_{\phi(j)} - \sum_e p^+_{\phi^{-1}(e)} g_e \right) = \alpha p^+_j \big(g_{\phi(j)} - \langle g, q \rangle \big), \tag{17}$$

where $p^+_{\phi^{-1}(e)} = \sum_{k:\phi(k)=e} p^+_k$. This is the native-vocabulary version of Equation 14. The parameter gradient adds the Jacobian-transposed contributions of both rows. A row shared by multiple queries receives the sum of their contributions. Differentiation through earlier transformer computations remains part of ordinary backpropagation; it is not removed by the local probability interface.

B.6 Chunked surrogate correctness

Let $z(\theta)$ collect the native logits used in an event chunk and let $g^\star = \nabla_z L(z)|_{z=z(\theta_0)}$. Define

$$\widetilde{L}(\theta) = \langle z(\theta), \mathrm{stopgrad}(g^\star) \rangle - \mathrm{stopgrad}\langle z(\theta), \mathrm{stopgrad}(g^\star) \rangle + \mathrm{stopgrad}\, L(z(\theta_0)). \tag{18}$$

At the evaluation point, this has value $L(z(\theta_0))$ and gradient $J_z(\theta_0)^\mathsf{T} g^\star$. Summing chunks preserves the full logit derivative and avoids retaining their intermediate event graphs simultaneously. This is first-order gradient equivalence at the current parameters, not an assertion of identical higher derivatives through detached gradients.

C Algorithm and Implementation Details

C.1 Training update

Algorithm 1 summarizes one update on student-generated responses. RolloutAndFilter discards length-truncated responses and collects 256 retained responses. The native histories and forward rows are stored in $\mathcal{H}_M$; the teacher remains fixed while both student rows contribute gradients. The loss and response-token normalization are defined in Equations 6 and 7.

Algorithm 1 One SEAM update on student-generated responses

Input: Frozen teacher T, student S_θ, prompt batch $\mathcal{X}$, optimizer state ω, lossless byte codecs, and teacher lookahead threshold $\tau \in [0, 1)$
Output: Updated student parameters θ and optimizer state ω

$\mathcal{Y} \leftarrow$ RolloutAndFilter$(S_\theta, \mathcal{X})$ ▷ Retain sampled native IDs
Align teacher and student response IDs through the lossless response bytes
$\mathcal{H}_T, \mathcal{H}_S \leftarrow$ NativeForward$(T, S_\theta, \mathcal{X}, \mathcal{Y})$
$\mathcal{B}_i \leftarrow$ EligibleStudentBoundaries(y_i) for each response $y_i \in \mathcal{Y}$
$\mathcal{L} \leftarrow 0$
for all responses i and query offsets $c \in \mathcal{B}_i$ **do**
  $t \leftarrow \mathrm{stopgrad}\big($EventMarginal$(T, \mathcal{H}^{(i)}_T, c, \tau)\big)$
  $s \leftarrow$ EventMarginal$(S_\theta, \mathcal{H}^{(i)}_S, c, 0)$ ▷ Algorithm 2
  $\mathcal{L} \leftarrow \mathcal{L} + \mathrm{JSD}(t, s)$
end for
$\mathcal{L} \leftarrow \mathcal{L} / N_{\text{response}}$ ▷ Equation 7
$g_\theta \leftarrow$ ChunkedBackward$(\mathcal{L}, \mathcal{H}_S)$ ▷ Both native student rows
$(\theta, \omega) \leftarrow$ AdamWStep$(\theta, \omega, g_\theta)$
return (θ, ω)

C.2 Local event computation

Algorithm 2 supplies the EventMarginal call in Algorithm 1. Here n_M is unnormalized extending-event mass, so $\sum_e n_M(e) = 1 - \alpha_M$; its control entries are zero. The returned $\widehat{q}_M$ uses the representative-alias policy in Appendix B.4. The threshold only skips an interior teacher branch; $\tau < 1$ ensures that the skipped-branch denominator is positive. $\mathcal{H}$ supplies the native prompt history h^{start}_M and the main-path rows. ChunkedBackward in Algorithm 1 applies the surrogate

above and propagates through the student model. Training additionally records boundary coverage, conditioning shifts, dropped mass, branch frequency, loss, and execution time.

Algorithm 2 Local event marginal with representative continuations

Input: Model M, native histories and forward rows $\mathcal{H}$, query offset c, lossless byte map b_M, event map ϕ_M, and $\tau \in [0, 1)$
Output: Event distribution $\widehat{q}_M$ over $\mathcal{E}$; student computations retain gradients

function EVENTMARGINAL($M, \mathcal{H}, c, \tau$)
  if $c = 0$ **then**
    return $R_M(\cdot \mid h_M^{\text{start}})$ ▷ Response-start row; Equation 2
  end if
  $(h_M, P_M) \leftarrow$ COVERAT($\mathcal{H}, c$) ▷ Token containing byte $c - 1$
  $\mathcal{C}_M, \mathcal{A}_M, \{\mathcal{X}_M(e)\}_e \leftarrow$ PREFIXLOOKUP(P_M) ▷ Equation 3
  $w_M(v) \leftarrow p_M(v \mid h_M) / \sum_{u \in \mathcal{C}_M} p_M(u \mid h_M)$ for $v \in \mathcal{C}_M$
  $n_M(e) \leftarrow \sum_{v \in \mathcal{X}_M(e)} w_M(v)$ for $e \in \mathcal{E}$ ▷ Extending mass
  $\alpha_M \leftarrow \sum_{v \in \mathcal{A}_M} w_M(v)$ ▷ Exact-token mass
  if $\alpha_M = 0$ **then**
    return n_M ▷ No exact-token branch
  end if
  if M is the student **or** c is a native teacher boundary **then**
    $v_{\text{obs}} \leftarrow$ native token ending at c in $\mathcal{H}$
    $\widehat{Q}_M \leftarrow R_M(\cdot \mid h_M v_{\text{obs}})$ ▷ Reuse the main next row
  else if $\alpha_M \leq \tau$ **then**
    return $n_M / (1 - \alpha_M)$ ▷ Skip teacher lookahead; TV error $\leq \tau$
  else
    $v^\star \leftarrow \arg\max_{v \in \mathcal{A}_M} p_M(v \mid h_M)$ ▷ Exact-byte representative
    $\widehat{Q}_M \leftarrow$ ROOTAFTERBRANCH($M, h_M, v^\star$) ▷ Native causal lookahead
  end if
  return $n_M + \alpha_M \widehat{Q}_M$ ▷ Covering and continuation contributions
end function

Lookup and numerical precision. The prefix index sorts content-token byte strings, retaining duplicate byte strings as separate vocabulary IDs. Binary search locates the interval beginning with a prefix and the subinterval exactly equal to it. An indexed byte table assigns extending tokens to the event at depth $|P_M|$. The implementation subtracts each interval's maximum logit before exponentiation and accumulates event masses in float64. It processes teacher head projections in chunks of 256 rows and student event losses in chunks of 192 rows. These are computational subdivisions of the same objective.

Causal lookahead. Teacher main-path hidden states score the distribution before each native response token. An interior exact-token branch is attached before the observed covering token and substitutes an exact token spelling the local prefix. Its subsequent hidden state must attend to that branch's native past, not to the observed covering token or its suffix. The representative next-root event vector is returned with the query byte offset and threshold so that stale branch results are rejected.

Boundary sets. The all-boundary variant includes response start and each student content-token end before the response end. The endpoint is added when generation terminates with a stop event. The common-boundary variant uses byte offsets that are boundaries under both tokenizations, with the same finish rule. A content token that spans multiple bytes contributes only its token boundary as a query; SEAM does not train at every byte position in the response.

Response-token normalization. For response i, let P_i and T_i denote its native student prompt and emitted response lengths. The causal prediction mask selects rows $P_i - 1, \ldots, P_i + T_i - 2$, one for each generated response token. Prompt-token targets and padded rows have zero mask. An actually emitted terminal stop token contributes one to T_i. Only retained responses enter the optimizer batch; discarded length-truncated responses contribute neither event-loss terms nor normalization tokens. Thus $N_{\text{response}} = \sum_i T_i$ counts generated student tokens in retained responses before boundary selection, independently of the teacher's tokenization and the number of event-loss terms. With

D data-parallel ranks and A accumulation steps per rank, each microbatch numerator is divided by $N_{\text{response}}/(AD)$. Accumulation scaling by $1/A$ and gradient averaging over D ranks recover Equation 7. All boundary variants retain this token count; the equal-weight control changes only the per-response numerator coefficients.

Ablation implementations. The root control uses first-byte/control projections of native next rows at common boundaries. The teacher-only control replaces the teacher's root distribution by its covering conditional and keeps the student root distribution. The symmetric/common control uses covering on both sides at common boundaries. SEAM applies the covering construction on both sides at all eligible student boundaries. These settings use the response-token normalizer and JSD with equal mixture weights. The equal-weight control retains SEAM's query set but multiplies each response's sum by $|\mathcal{B}_i^{\text{common}}|/|\mathcal{B}_i^{\text{all}}|$. This changes boundary coefficients within each response while preserving the query locations.

What is held fixed. Teacher targets, token IDs, prefix intervals, token roles, branch choices, and validity masks are fixed during a gradient evaluation. The student conditional weights and next-row distribution are not fixed. No REINFORCE term through rollout generation is included. This matches the standard conditional-supervision use of on-policy trajectories rather than differentiating the probability of choosing the trajectory itself.

D Vocabulary Overlap and Illustrative Examples

D.1 Vocabulary-overlap statistics

The exact model releases are `Qwen/Qwen3-32B` (32B class; 32.8B parameters), `Qwen/Qwen3.5-2B` (2B), `microsoft/phi-4` (14B), and `google/gemma-2-2b-it` (2B). The Qwen3.5-2B checkpoint is the post-trained release; "Base" in Table 1 denotes the released student before our SFT stage.

We read the local tokenizer files for Qwen3-32B, Qwen3.5-2B, phi-4 (14B), and Gemma-2-2B-it used by the experiment setup. The inventory combines the base vocabulary and added-token records in `tokenizer.json` and `tokenizer_config.json`, excludes entries declared special, and retains ordinary added tokens. This inventory excludes physical language-model-head padding rows; it is an inventory statistic, not the method's control-role assignment. In particular, it does not count a shared stop/control event as an ordinary shared token.

We compare lossless token bytes, rather than token IDs or tokenizer-internal spellings. The Qwen3-32B, Qwen3.5-2B, and phi-4 (14B) ByteLevel alphabets are inverted to raw bytes; Gemma-2-2B-it's metaspace replacement and byte-fallback rules are applied, including replacement in ordinary added tokens. Nonempty byte strings are deduplicated before set intersection, so byte aliases do not inflate the overlap. Every valid-UTF-8 non-special entry is checked against its native singleton decoder; incomplete UTF-8 fragments remain raw bytes and are not replaced with a Unicode replacement character. Table 3 reports the effect of deduplication.

Table 3: Tokenizer inventory used for Figure 1. Alias excess is the number of non-special IDs minus the number of unique byte strings.

Model	Non-special IDs	Unique byte strings	Alias excess
Qwen3-32B	151,655	151,655	0
Qwen3.5-2B	248,056	248,056	0
phi-4 (14B)	100,256	100,256	0
Gemma-2-2B-it	255,994	255,869	125

The shared-type counts are 131,598 for Qwen3-32B $\rightarrow$ Qwen3.5-2B, 96,851 for phi-4 (14B) $\rightarrow$ Qwen3.5-2B, and 97,675 for Qwen3-32B $\rightarrow$ Gemma-2-2B-it. The count assigns equal weight to each unique token byte string and uses no text corpus or model probabilities. It measures type overlap, rather than the probability mass accessible to a distillation objective.

D.2 A LONGEST VOCABULARY PREFIX NEED NOT BE THE FIRST BPE TOKEN

Consider base symbols `a`, `b`, `c`, with merge `b+c→bc` ranked before `a+b→ab`. Both `ab` and `bc` are vocabulary entries. Tokenizing `abc` first forms `bc`, leaving `a|bc`. The longest vocabulary prefix of `abc` is nevertheless `ab`. Thus longest-prefix routing and ranked-merge tokenization are different operations. This counterexample identifies an assumption; it is not an estimate of the frequency or performance impact of that assumption for a particular model pair.

D.3 LOSSLESS BYTES AND SINGLETON UNICODE DECODING

Suppose two content tokens emit byte sequences `C3` and `A9`. Concatenation is the UTF-8 encoding of `é`, but neither byte sequence is a valid UTF-8 character on its own. A decode–encode interface that substitutes the replacement character for each invalid singleton changes both the bytes and their lengths. Native token-to-byte decoding is necessary before cross-tokenizer boundary alignment. Lossless byte decoding and event-probability alignment address distinct parts of the supervision interface.

D.4 OVERLAPPING CONTINUATION EVENTS

Let E_a be the event that future emitted bytes begin with `a` and let E_{ab} be the event that they begin with `ab`. Then $E_{ab} \subseteq E_a$. If their probabilities are 0.6 and 0.3, normalizing the vector gives $(2/3, 1/3)$, which cannot be the original probabilities of these events. A partition could instead contain E_{ab}, $E_a \setminus E_{ab}$, and the complement of E_a, but computing those probabilities requires specifying the events and the native paths they include. The next-byte/control partition supplies a fixed finite observation at every query.

D.5 SAME NEXT ROW, DIFFERENT EARLIER CONDITIONAL

A next-token row after an exact token fixes that token's identity. The covering marginal additionally uses the preceding row's mass on longer tokens sharing its bytes. Two native histories can have the same next row yet different longer-token mass at the anchor, producing different covering marginals. Conversely, equal anchor distributions do not determine the exact-token continuation after an interior branch. This explains why both rows are used for the student and why the teacher needs lookahead at some interior boundaries.

E TRAINING AND EVALUATION PROTOCOL

E.1 SFT DATA CONSTRUCTION AND CHECKPOINT SELECTION

The SFT corpus consists of teacher-generated responses to a mixture of mathematical reasoning and code-generation problems. Mathematics prompts are drawn from GSM8K, MATH with MATH-500 excluded, Orca-Math, and OpenMathInstruct; code prompts are drawn from CodeContests, TACO, KodCode, and OpenCodeInstruct. The prompt pool is filtered for valid problem statements and reference answers or executable tests, with duplicate and near-duplicate filtering against the evaluation collections.

For each prompt, we sample candidate responses from the corresponding teacher and apply response-quality and task-correctness checks. Mathematical answers are checked against the reference answer, and generated programs are checked using the associated tests. We retain the shortest verified response in teacher-token length as the target for each prompt and omit prompts without a valid candidate. For phi-4 (14B), the verified corpus is stratified by source to obtain an approximately 65% mathematics / 35% code mixture. Each retained example is formatted as a user prompt and an assistant response using the student's chat template. SFT minimizes cross-entropy on assistant-response tokens, with prompt tokens masked from the loss and a full-sequence length cap of 4,096 tokens.

The held-out validation panel contains 800 problems in four equally sized strata: 200 GSM8K problems; 200 MATH problems, with 40 from each of difficulty levels 1–5; 200 function-code problems (161 from MBPP and 39 from OpenCodeInstruct); and 200 contest-code problems (100

from CodeContests and 100 from TACO). The panel is constructed separately from the SFT/OPD prompt pool and screened for overlap with the GSM8K, MATH-500, MBPP, and LiveCodeBench test collections. Its validation score is the unweighted mean of the four stratum accuracies. We evaluate checkpoints saved during SFT and select the checkpoint with the highest mean validation score. All OPD methods within a teacher–student pair are initialized from this same checkpoint.

E.2 SFT AND OPD CONFIGURATIONS

Table 4: Unified SFT and OPD configuration. Slash-separated values follow Qwen3-32B → Qwen3.5-2B, phi-4 (14B) → Qwen3.5-2B, and Qwen3-32B → Gemma-2-2B-it. SFT checkpoint steps denote the selected initialization, not the full fit length. Objective-specific settings are specified below.

Setting	**SFT**	**OPD**
Training data	Verified teacher-response pairs	10,000 verifiable math/code prompts
Supervision	Response-token cross-entropy; prompt loss masked	Frozen-teacher supervision on student rollouts; method-specific objective
Trainable parameters	Full language model	Full student language model
Epochs / prompt passes	2 epochs	2 prompt passes; nominal 78 updates
Initialization step	Selected checkpoints: 20 / 80 / 180	The corresponding SFT checkpoint
Optimizer	Fused AdamW; $\beta_1 = 0.9$, $\beta_2 = 0.999$	AdamW; $\beta_1 = 0.9$, $\beta_2 = 0.98$
Learning rate	2×10^{-6}	10^{-6}
Scheduler / warmup	Cosine / 5%	Cosine to zero / 5%
Weight decay / clip norm	0 / 1.0	0 / 1.0
Effective batch	64 / 128 / 64	256
GPUs per run	4 NVIDIA H200 (all three pairs)	2 NVIDIA H200
Microbatch per GPU	2 / 4 / 4	2
Gradient accumulation	8 / 8 / 4	64
Precision / packing	Pure bfloat16 / no packing	bfloat16 / no packing
Gradient checkpointing	On / on / off	Method-specific; see text below
Training seed	42	101
Student length cap	4,096 tokens	Context 8,192; prompt 2,048; response 4,096
Teacher scoring context	Offline targets	32,768 / 16,384 / 32,768 tokens
Rollout sampling	Fixed teacher-response corpus	Temperature 0.6; top-$p = 0.95$; one accepted response/prompt
SEAM objective	—	JSD, $\beta = 0.5$; $\tau = 10^{-3}$; response-token normalization

Table 4 combines the SFT and OPD settings. All SFT runs use four NVIDIA H200 GPUs. The per-GPU microbatches and accumulation factors give effective batches of 64, 128, and 64 in the listed pair order. All three main SEAM configurations use symmetric covering at all student boundaries. Within each model pair, methods start from the same selected SFT checkpoint. The released Gemma-2-2B-it student is already instruction-tuned. Student generation uses nonthinking mode. The full SFT fits are configured for two epochs; their selected initialization checkpoints are earlier than the final fit checkpoints.

SFT uses gradient checkpointing for the two Qwen3.5-2B students and disables it for Gemma-2-2B-it. In OPD, SEAM, BPM, and DSKD enable gradient checkpointing, whereas SimCT disables it. GOLD disables it in the two Qwen3.5-2B settings and enables it for Gemma-2-2B-it. These computational settings are reflected in the reported GPU-hours. Gemma-2-2B-it training seed 202 is available only for the common/all symmetric validation controls; benchmark comparisons use seed 101.

E.3 On-policy training protocol

During OPD, responses truncated at the maximum response length (4,096 tokens) are discarded. An optimizer update is performed once a batch of 256 retained responses has been collected.

E.4 Baseline objectives and training settings

All five methods use native on-policy transport and a Hugging Face native teacher forward backend. SimCT and GOLD follow their published formulations (Sun et al., 2026; Patiño et al., 2025); their scoring, alignment, and normalization choices are treated as method definitions. We retain method-specific objectives within the shared training setup.

BPM. The baseline uses the official BPM release at commit `55093cff`, with lossless byte mappings for its vocabularies. It retains byte-prefix routing, the residual category, JSD, and response-token normalization. Appendix D gives a counterexample distinguishing longest-prefix routing from ranked BPE.

SimCT. SimCT augments shared-token correspondence with aligned continuation units and normalizes their scores to construct a cross-tokenizer distillation objective (Sun et al., 2026). Figure 2 illustrates the resulting prediction objects.

DSKD. DSKD uses alignment and cross-space projection to compare teacher and student predictions, together with reverse-KL loss terms (Zhang et al., 2024). It operates within the native on-policy training setup described above.

GOLD. GOLD combines alignment with distribution matching over matched and unmatched token groups (Patiño et al., 2025). We use the Hugging Face TRL implementation at commit `4e9ab9fa`, with matched generalized JSD ($\beta = 0.5$), unmatched sorted-probability terms, and response-level normalization. The on-policy sampling parameter is $\lambda = 1.0$, distinct from the JSD mixture coefficient.

The principal comparison comprises these four baselines and SEAM.

E.5 Evaluation populations and generation

The evaluation panel contains 3,331 problems per decoding seed. Five decoding seeds produce five pass@1 estimates for each evaluated checkpoint. Their arithmetic mean is the task score. Avg5 gives equal weight to GSM8K, MATH-500, MBPP, LiveCodeBench-v6, and TACO. The separate validation panel uses the four 200-problem strata described in Appendix E.1 and reports their mean accuracy.

Table 5: Evaluation populations and response caps in native generated tokens. A common token cap does not imply equal byte lengths across student families. Special-judge (SPJ) problems are excluded from TACO.

Benchmark	Examples	Response cap	Scope
GSM8K	1,319	4,096	Test split
MATH-500	500	4,096	500-problem panel
MBPP	257	2,048	Sanitized test
LiveCodeBench-v6	1,055	4,096	Fixed v6 panel
TACO	200	4,096	Easy100 + Medium100, non-SPJ

Math tasks use answer extraction and equivalence verification; code tasks use execution-based tests. The TACO subset excludes SPJ problems. Evaluation uses temperature 0.6, top-$p = 0.95$, and one response per example per decoding seed on the populations in Table 5.

E.6 Main-table decoding-seed variability

Table 1 reports task pass@1 (%) as the mean and sample standard deviation for decoding seeds 42–46. The sample standard deviation uses denominator $5 - 1$ and describes decoding-seed variability, rather than variation across independent training runs. The Average column is the equally weighted mean over GSM8K, MATH-500, MBPP, LiveCodeBench-v6, and TACO; it has no separate error term. Task scores and averages are displayed to two decimal places. Table 2 uses the same task-level reporting convention.

E.7 Ablation evaluation protocol

The validation accuracies for the five rows of Table 2, in table order, are 34.38%, 39.00%, 38.50%, 38.38%, and 39.00%. The equal-weight control scales each response by its common-to-all boundary-count ratio.

For Table 2, we fix training seed 101 and use each method's actual final checkpoint. Evaluation uses five decoding seeds (42–46). All Gemma-2-2B-it configurations use the same lossless byte codec.

The ablation evidence retains one unresolved code-execution result as incorrect, as indicated in Table 2. Treating that result as correct could raise the affected Avg5 by at most 0.0038 pp. This bound quantifies sensitivity to that execution outcome.

F Supplementary Transfer Analysis

Tables 1 and 2 report the main benchmark comparison and Gemma-2-2B-it ablations. This appendix presents validation trajectories, the boundary-coverage, loss-weighting, and compute comparison, the phi-4 (14B) control, and the relation between vocabulary overlap and observed transfer.

F.1 Validation trajectories

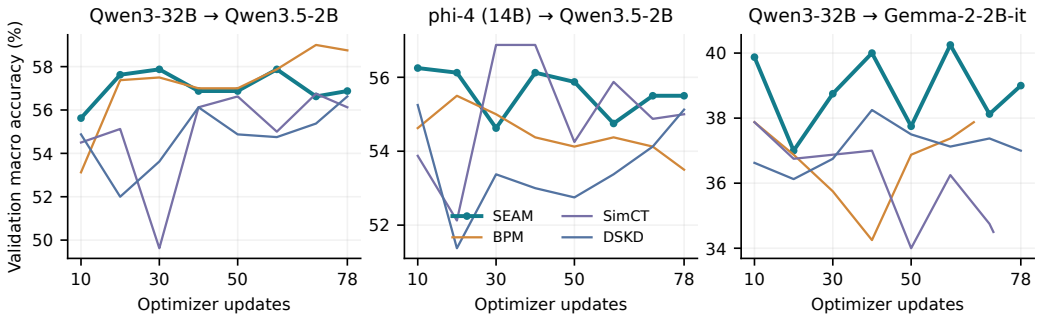

Figure 5: Validation accuracy versus optimizer updates for four methods in each model pair. Gemma-2-2B-it endpoints differ because of accepted-sample counts. The horizontal axis denotes update count, not computational cost.

Figure 5 evaluates checkpoints on the separate 800-example validation panel. This panel uses a different population and aggregate from the five-task benchmark, so their rankings can differ.

F.2 Boundary coverage, loss weighting, and compute

The symmetric/common control uses covering on both models at common boundaries. SEAM extends the query set to all eligible student boundaries. The equal-weight control uses the same all-student query rule as SEAM but scales each response's loss sum by its common-to-all boundary-count ratio, whereas SEAM uses Equation 7 directly. Table 6 compares these three configurations' benchmark means, validation endpoints, and recorded costs for training seed 101. All three runs complete 78 updates; benchmark results use five decoding seeds (42–46).

Table 6: **Boundary coverage, loss weighting, and compute.** Qwen3-32B → Gemma-2-2B-it. Validation is accuracy on the separate 800-example panel. Task means and SDs are in Table 2; GPU-hour accounting follows Table 10.

Objective	Avg5	Validation	GPU-hours
Symmetric covering (common)	31.04	38.50	12.50
SEAM (equal weight)	30.69	38.38	14.14
SEAM	**31.70**	**39.00**	19.33

The task-level and computational tradeoffs are discussed in Sections 4.4 and 4.5. On the separate validation panel, symmetric/common, equal-weight/all, and SEAM reach 38.50%, 38.38%, and 39.00%, respectively. The benchmark and validation means both favor full SEAM among these three configurations.

F.3 Teacher covering on phi-4 (14B)

The phi-4 (14B) → Qwen3.5-2B comparison examines the full-versus-teacher-only ordering in another model pair. Teacher covering starts from SFT checkpoint 80 with training seed 101 and completes 78 updates. It uses common boundaries and the student's root-event distribution. Full SEAM covers both sides at all eligible student boundaries.

Table 7: **Teacher covering and full SEAM on phi-4 (14B) → Qwen3.5-2B.** Task pass@1 (%), mean ± sample SD over five decoding seeds (42–46). Average is the five-task mean. The SEAM row reproduces Table 1; the higher mean in each column is **bold**.

Method	Boundaries	Math		Code			Average
		GSM8K	MATH-500	MBPP	LCB-v6	TACO	
Teacher covering	Common	80.86 ± 0.53	67.40 ± 1.69	52.53 ± 1.76	15.72 ± 0.25	7.50 ± 0.71	44.80
SEAM	All student	**81.17** ± 0.52	**67.96** ± 0.82	**55.72** ± 0.84	**15.83** ± 0.50	**8.10** ± 0.82	**45.76**

Table 7 reports 44.80% Avg5 for teacher covering and 45.76% for full SEAM (+0.96 pp). Task differences favor SEAM: +0.31 on GSM8K, +0.56 on MATH-500, +3.19 on MBPP, +0.11 on LiveCodeBench-v6, and +0.60 on TACO.

F.4 Vocabulary overlap, transfer, and conditioning on Gemma-2-2B-it

Gemma-2-2B-it combines the lowest observed vocabulary coverage with SEAM's largest baseline margin. Type counts motivate the comparison; shared-event diagnostics characterize the conditional distributions used for supervision.

Coverage and transfer across pairs. Teacher/student type coverage is 86.8%/53.1% for Qwen3-32B → Qwen3.5-2B, 96.6%/39.0% for phi-4 (14B) → Qwen3.5-2B, and 64.4%/38.2% for Qwen3-32B → Gemma-2-2B-it. Gemma-2-2B-it has the lowest coverage on both sides and smaller gains over SFT for every evaluated baseline. For SimCT, these gains are 1.70, 2.66, and 0.64 pp in the same pair order; for BPM, they are 2.38, 2.41, and 0.94 pp. Gemma-2-2B-it GOLD gains 0.53 pp and DSKD loses 0.73 pp. SEAM gains 2.85, 3.24, and 2.03 pp: it improves the Gemma-2-2B-it outcome but also transfers less there than in the Qwen3.5-2B settings.

What the overlap statistic measures. The statistic counts distinct token byte strings, rather than their probability on training prefixes. A small intersection can carry substantial model mass, and a large intersection does not establish compatible boundary conditioning. The phi-4 (14B) pair further shows that low student-side coverage can coexist with larger gains when teacher-side coverage is high. Methods that extend beyond exact token matches can also use information outside this intersection.

Net transfer and task tradeoffs. In Table 1, Gemma-2-2B-it Base and SFT score 29.17% and 29.67%. GOLD, SimCT, and BPM reach 30.20%, 30.31%, and 30.61%, while DSKD reaches 28.94%. Thus limited gains should not be read as general failure to transfer. SEAM reaches 31.70% and improves all five tasks over Base, with a 4.76 pp gain on MATH-500 and a 1.54 p-p gain on LCB-v6. SFT itself improves MATH-500, MBPP, and TACO while lowering GSM8K and LiveCodeBench. Aggregate and task-specific comparisons therefore capture different aspects of adaptation.

Event objectives and weighting. Table 2 uses the same shared byte/control alphabet across root matching, teacher covering, symmetric/common covering, and the two all-student objectives. The reported Avg5 scores are 27.33%, 30.58%, 31.04%, 30.69%, and 31.70%, respectively. Root matching completes 71 updates and the other configurations complete 78. The equal-weight control evaluates a different weighting rule under the same all-student query rule.

Implications for supervision design. Equation 5 connects the design to the evidence: covering combines token-ending continuation Q_M with longer-token alternatives through $(1-\alpha_M)U_M$. The diagnostics establish that this combination changes supervision, and the Gemma-2-2B-it comparisons report downstream outcomes for alternative event objectives. The phi-4 (14B) results add a second comparison of the complete constructions. BPM routing and SimCT scoring define different prediction objects; the shared-event ablations examine conditioning while retaining one event alphabet.

G Boundary Diagnostics and Computational Resources

Let $\mathcal{B}$ be the set of supervised student boundaries, and let $\mathcal{I} \subseteq \mathcal{B}$ contain those strictly inside a teacher content token. The per-example interior fraction is $|\mathcal{I}|/|\mathcal{B}|$. Training instrumentation aggregates diagnostic quantities over its local batches and distributed execution. Table 8 reports the arithmetic average of the recorded per-update metrics over the 78 updates of each principal seed-101 run. It does not reconstruct a single global boundary-weighted statistic.

Table 8: Logged diagnostics for the principal SEAM runs. TV shifts compare covering events with ordinary next-row root events. Teacher shifts use common boundaries; student shifts use the supervised student boundaries. Dropped mass measures thresholded teacher lookahead only.

Metric	Qwen3-32B → Qwen3.5-2B	phi-4 (14B) → Qwen3.5-2B	Qwen3-32B → Gemma-2-2B-it
Interior-boundary fraction (%)	5.838	9.100	10.292
Teacher conditioning shift (TV)	0.00943	0.00717	0.01319
Student conditioning shift (TV)	0.00709	0.00414	0.00904
Dropped teacher mass	3.03×10^{-6}	2.13×10^{-6}	4.38×10^{-6}
Lookahead branches/boundary (%)	1.545	0.996	2.192
Skipped literal samples	0	0	0

Boundary disagreement depends on generated text and the counting rule. These on-policy averages therefore characterize the observed training trajectories, whereas a fixed external corpus would measure disagreement under a different text distribution. Between-pair associations with validation performance remain descriptive because the models, initializations, and trajectories also differ.

For each OPD run, we compute

$$H_{\text{core}} = \frac{1}{3600} \sum_{u=1}^{U} G_u \, t_u^{\text{core}}. \tag{19}$$

Here $G_u = 2$ and t_u^{core} measures rollout generation, teacher scoring, student optimization, and rollout-weight synchronization for completed update u. The totals exclude checkpoint export, validation, offline benchmark evaluation, startup, idle time, and failed jobs. All completed updates contributing to the final checkpoint are included. The 15 OPD configurations contain 1,150 completed updates.

Table 9: GPU-hours for the recorded training computations. OPD rows sum core timings over all completed updates on two H200 GPUs. †: shared SFT preparation uses 0.40 GPU-hours for Qwen3-32B → Qwen3.5-2B and complete-fit Trainer runtimes with four GPUs for the other two pairs. SFT accounting excludes teacher-data generation and checkpoint-selection evaluation.

Method	Qwen3-32B → Qwen3.5-2B	phi-4 (14B) → Qwen3.5-2B	Qwen3-32B → Gemma-2-2B-it
GOLD	7.36	5.30	9.95
DSKD	4.00	2.54	7.45
SimCT	8.97	5.92	10.34
BPM	3.65	3.38	10.74
SEAM	11.99	7.04	19.33
Shared SFT fit†	0.40	0.78	0.30

SFT is counted once per model pair. The full-fit runtimes are 698.13 seconds on four GPUs for phi-4 (14B) → Qwen3.5-2B and 273.92 seconds on four GPUs for Qwen3-32B → Gemma-2-2B-it, corresponding to 0.78 and 0.30 GPU-hours. Qwen3-32B → Qwen3.5-2B uses 0.40 GPU-hours. These values cover the SFT fit, whereas the initialization checkpoints in Table 4 may precede its end. The OPD totals include the effects of generated response lengths and the computational settings in Appendix E.

G.1 Compute for the ablation configurations

Table 10: Training-core GPU-hours for the Gemma-2-2B-it ablation configurations in Table 2, using Equation 19. Each configuration uses two H200 GPUs. Costs sum the completed updates contributing to its final checkpoint. The SEAM entry repeats the same run as Table 9; it is not an additional training run.

Event construction	Boundaries	Updates	GPU-hours
Root next-row	Common	71	11.50
Teacher covering	Common	78	7.82
Symmetric covering	Common	78	12.50
Symmetric covering (equal weight)	All student	78	14.14
SEAM	All student	78	19.33

The five reported configurations contain 383 completed updates: 71 for root matching and 78 for each covering configuration. Table 10 reports their training-core costs under the same accounting rule as Table 9.

H SYNTHETIC VERIFICATION OF THE LOCAL FORMULAS

We evaluate 100 seeded synthetic cases of the local mixture and its JSD gradients. Each case samples finite covering logits, next-row logits, and a fixed teacher event distribution. Compatible candidates include extending tokens and exact tokens sharing a representative next-row distribution. We compare the analytical covering-row and next-row gradients with centered finite differences using step 10^{-5}. A separate random mixture in each case checks the thresholded total-variation inequality at $\tau = 10^{-3}$.

Table 11: Numerical checks of the declared local equations, float64 NumPy arithmetic, random seed 202709. These are synthetic mathematical checks, not full-model or GPU parity tests.

Quantity	Observed maximum
Absolute normalization error	2.22×10^{-16}
Absolute finite-difference logit-gradient error	1.03×10^{-11}
Truncation TV across sampled mixtures	6.47×10^{-4}
Truncation threshold	1.00×10^{-3}

These checks support the probability and gradient calculations under fixed histories, compatible sets, and continuation kernels. Byte-codec correctness, causal branch isolation in the full model, and training dynamics require distinct validation beyond this synthetic calculation.

I Qualitative Reasoning and Code-Generation Cases

The following GSM8K, LiveCodeBench-v6, and MBPP cases illustrate responses where SEAM succeeds and all four baselines fail, one per teacher–student pair. Within each case, methods receive the same problem at decoding seed 42. Math reasoning is condensed; code excerpts preserve the displayed logic while omitting surrounding prose and setup where indicated. Outcomes follow the archived evaluation. These cases illustrate output behavior rather than establish the causal effect of a loss component.

Case Study 1: Math Reasoning — **GSM8K**
Qwen3-32B → Qwen3.5-2B

Including every contribution and answering the requested quantity

Problem (ID 997): Walter is collecting money for charity. First he collects $500 from his neighbors. Then he collects $1500 from a fund he set up online. His lawyer offers to donate three times as much as everyone else donates. Walter is going to tell him about his neighbors and the online fund until his friend gives him $200 as well. How much is Walter's lawyer going to contribute?

Reference answer: $6,600.

Method	Generated reasoning (condensed)	Result
SEAM (ours)	$500 + 1500 + 200 = 2200$. Lawyer: $3 \times 2200 = 6600$. **Final answer: $6,600.** Includes the friend and returns the lawyer's contribution.	✓ Pass
BPM	$500 + 1500 = 2000$. Lawyer: $3 \times 2000 = 6000$. **Final answer: $6,000.** Omits the friend's $200 contribution.	✗ Fail
SimCT	$500 + 1500 = 2000$; $3 \times 2000 = 6000$. Then totals $2000 + 6000 = 8000$. **Final answer: $8,000.** Omits $200 and reports a combined total.	✗ Fail
DSKD	$500 + 1500 = 2000$. Lawyer: $3 \times 2000 = 6000$. **Final answer: $6,000.** Omits the friend's $200 contribution.	✗ Fail
GOLD	Identifies "everyone else" with $200. Lawyer: $3 \times 200 = 600$. **Final answer: $600.** Uses only the friend's contribution as the base.	✗ Fail

Analysis

The lawyer's contribution depends on all three other donations, including the friend's later $200 contribution. SEAM totals $2,200 and returns three times that amount. BPM and DSKD omit the friend's contribution and answer $6,000. SimCT also omits it, then answers the combined collection rather than the lawyer's contribution. GOLD computes three times only the friend's $200. Thus the four failures concern which amounts enter the calculation or which quantity is returned, rather than multiplication itself.

Case Study 2: Code Generation **LCB-v6**
phi-4 (14B) → Qwen3.5-2B

Checking adjacent colors and respecting circular indexing

Problem summary (ID 3463): Given a circular binary array `colors`, count all groups of three consecutive tiles whose middle color differs from both neighbors. The first and last tiles are adjacent ($3 \leq n \leq 100$).

Public examples:

```
assert Solution().numberOfAlternatingGroups([1,1,1]) == 0
assert Solution().numberOfAlternatingGroups([0,1,0,0,1]) == 3
```

Loop excerpts; n = len(colors), count = 0. Common setup, the method signature, comments, and final return are omitted.

Method	Generated code (key logic)	Result

SEAM (ours)	`for i in range(n):`
	`  if colors[i] != colors[(i + 1) % n] and colors[(i + 1) % n ] != colors[(i + 2) % n]:`
	`  count += 1`
	Correct adjacent comparisons with modulo indexing; 37/37 archived tests.
		✓ Pass

BPM	`for i in range(n):`
	`  if colors[i] != colors[(i + 1) % n] and colors[i] != colors[(i + 2) % n]:`
	`  count += 1`
	Compares the first tile with both others; second example returns 1.
		✗ Fail

SimCT	`for i in range(n):`
	`  if colors[i] != colors[(i + 1) % n] and colors[i] != colors[(i + 2) % n]:`
	`  count += 1`
	Repeats the same condition error; second example returns 1.
		✗ Fail

DSKD	`for i in range(n):`
	`  if colors[i] != colors[(i + 1) % n] and colors[i] != colors[(i + 2) % n]:`
	`  count += 1`
	Repeats the same condition error; second example returns 1.
		✗ Fail

GOLD	`for i in range(n):`
	`  is_alternating = True`
	`  for j in range(2):`
	`  if colors[i] == colors[i + j + 1] and colors[i + j + 1] == colors[i + j + 2]:`
	`  is_alternating = False`
	`  break`
	`  if is_alternating:`
	`  count += 1`
	Unwrapped i+j+1 / i+j+2 accesses raise IndexError.
		✗ Fail

Analysis

An alternating triple must satisfy two adjacent comparisons: the first color differs from the middle, and the middle differs from the last. SEAM implements both with modulo indexing and passes all 37 archived tests. BPM, SimCT, and DSKD instead compare the first tile against both remaining tiles; each returns 1 on the second public example, whose answer is 3. GOLD indexes beyond the array without wrapping and raises `IndexError`. Only SEAM passes the full archived evaluation for this example.

Case Study 3: Code Generation **MBPP**
Qwen3-32B → Gemma-2-2B-it

Preserving order and types in digit concatenation

Problem (ID 57): Write a python function to find the largest number that can be formed with the given list of digits.

Supplied tests:

```
assert find_Max_Num([1,2,3]) == 321
assert find_Max_Num([4,5,6,1]) == 6541
assert find_Max_Num([1,2,3,9]) == 9321
```

Method	Generated code (key logic)	Result

SEAM (ours)	`def find_Max_Num(digits):`
	`  digits.sort(reverse=True)`
	`  return int("".join(str(digit) for digit in digits))`
	Descending order and explicit string conversion.
		✓ Pass

BPM	`def find_Max_Num(digits):`
	`  digits.sort(reverse=True)`
	`  return int("".join(digits))`
	Joining integers raises TypeError.
		✗ Fail

SimCT	`def find_Max_Num(digits):`
	`  digits.sort()`
	`  return int("".join(str(d) for d in digits))`
	Ascending order produces 123 instead of 321.
		✗ Fail

DSKD	`def find_Max_Num(digits):`
	`  digits.sort()`
	`  return int("".join(str(d) for d in digits))`
	Also sorts ascending and produces 123.
		✗ Fail

GOLD	`from itertools import permutations`
	
	`def find_Max_Num(digits):`
	`  return max(int("".join(sorted(p)) ), key=lambda x: int(x))`
	References undefined p and raises NameError.
		✗ Fail

Analysis

The displayed SEAM solution sorts the digits in descending order and converts them to strings before concatenation. BPM has the required order but passes integers to `join`, causing a `TypeError`. SimCT and DSKD convert to strings but sort ascending, yielding 123 on the first test. GOLD references an undefined variable `p`, causing a `NameError`. This example exposes complementary ordering and type-handling errors in the generated programs.