BEYOND TOKEN ALIGNMENT: EVENT COMPLETION FOR CROSS-TOKENIZER ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) transfers knowledge between language models through teacher supervision on student-generated trajectories. With different tokenizers, a single teacher token may require multiple student tokens to generate, creating intermediate states where the event is entered but not yet completed. Existing cross-tokenizer methods align tokens or text spans to construct comparable prediction targets. We study a complementary problem after partial generation: once the student produces a prefix of a teacher token, multiple next tokens may complete the same remaining bytes, but the teacher only specifies the required completion rather than how probability should be divided among these valid continuations. We introduce **Event-Set Completion Distillation (ESCD)**, which complements cross-tokenizer probability alignment with completion-set supervision. ESCD aggregates prefix-related teacher events and supervises the **total probability of byte-compatible one-step student completions**, avoiding tokenizer-dependent probability splits among individual tokens. The method reuses student trajectories and predictions, requiring neither additional rollouts nor changes to the student vocabulary. Experiments demonstrate consistent gains in mathematics, code, and scientific reasoning across model families and tokenizers, extending to large-scale MoE distillation from a **1T teacher to a 35B student**. Local analyses show that retaining completion sets better matches the reference supervision, while one-step completion covers over 99% of observed compatible teacher mass after partial event entry in the studied tokenizer pairs. These findings support event entry and event completion as complementary supervision targets for cross-tokenizer knowledge transfer. Code will be released on GitHub.

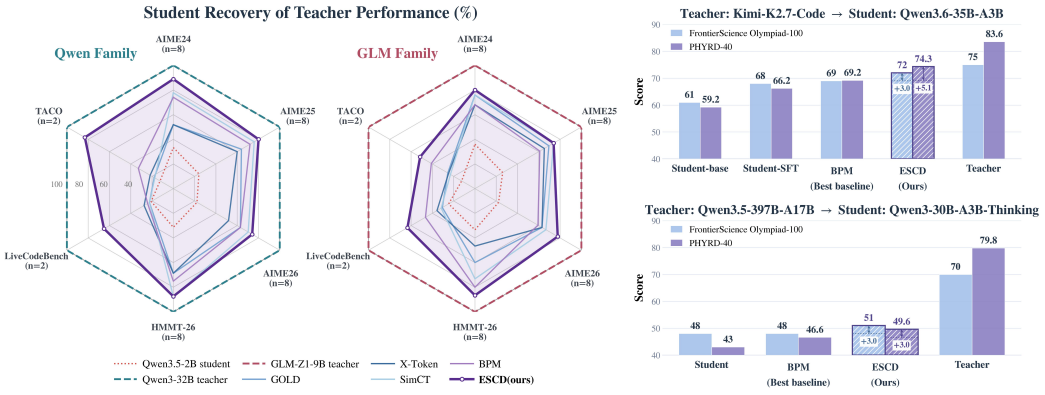


Figure 1: **Cross-tokenizer distillation results across model scales. Left:** Teacher-normalized pass@n scores on six mathematics and code benchmarks, with Qwen3.5-2B distilled from Qwen3-32B or GLM-Z1-9B. Recovery is the Student score divided by the corresponding teacher score, expressed as a percentage; 100% indicates matching teacher performance. We use $n = 8$ for mathematics and $n = 2$ for code. **Right:** Absolute scores on FrontierScience Olympiad and PHYRD-40 for two large-scale heterogeneous MoE pairs. Together, these results show consistent improvements across task domains, model families, and scales.

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.

1 INTRODUCTION

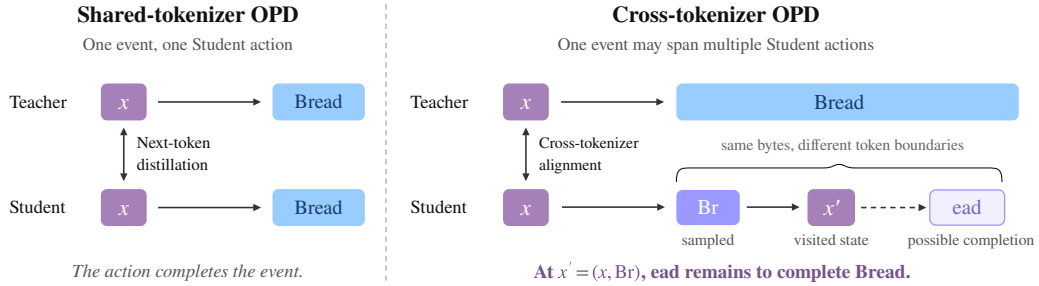


Figure 2: **On-policy distillation with shared and different tokenizers. Left:** a shared tokenizer allows direct next-token supervision. **Right:** a teacher token such as `Bread` may span multiple student actions. After sampling `Br`, the student reaches $x' = (x, \texttt{Br})$ with residual bytes `ead` still required to complete the event. The dashed arrow indicates a possible continuation, highlighting the distinction between entering an event and completing it.

The diversity of large language models creates opportunities for knowledge transfer across model families, architectures, and scales. A capable teacher and a smaller, deployment-efficient student may differ in model design, training data, and specialization. Knowledge distillation provides a framework for transferring these capabilities through teacher supervision (Hinton et al., 2015). On-policy distillation (OPD) is well suited to autoregressive generation because it obtains teacher feedback on student-generated trajectories (Lin et al., 2020; Gu et al., 2026; Agarwal et al., 2024). Training under the student-induced state distribution reduces the mismatch between training contexts and generation states, including prefixes the teacher would not typically generate independently.

Many OPD formulations assume a shared tokenizer and therefore a common next-token prediction space. This assumption becomes restrictive in cross-family distillation, where both vocabularies and token boundaries can differ substantially. The mismatch concerns not only which token represents a piece of text, but also how many generation steps are needed to produce it. Existing cross-tokenizer methods establish comparable supervision through token mappings, distribution matching, and text- or byte-level alignment (Boizard et al., 2025; Sun et al., 2026; Wang et al., 2026a). SimCT compares aligned continuation scores, while BPM also constructs conditional token targets inside a teacher token. Both therefore incorporate predictions beyond the first student action. We focus on a complementary question: *after a student action partially realizes a teacher event, how should supervision constrain the total probability of valid completions at the resulting state?*

To make this question concrete, consider the teacher token `Bread`, which the student may realize as `['Br', 'ead']` (Figure 2). After sampling `Br`, the rollout moves from x to $x' = (x, \texttt{Br})$, leaving the residual bytes `ead`. At x', the residual specifies which continuations complete the event, but this constraint alone does not determine their relative probabilities. Importantly, the teacher event defines a byte-level completion constraint rather than a tokenizer-dependent decomposition into individual student tokens. Therefore, assigning probabilities among multiple valid completions introduces an additional allocation choice that is determined by the student tokenizer rather than by the teacher supervision itself. Existing token-level objectives can supervise this state while prescribing such distributions over individual completions. We investigate whether supervising the total probability of valid completions provides an effective alternative for cross-tokenizer transfer. This frames the **event-completion gap** as a target-design problem: how to supervise the remaining byte constraint while leaving the allocation within its completion set unspecified.

More generally, let b_T and b_S denote token byte realizations, and consider a teacher content token v without a single-token student counterpart. If the student samples a content token a whose bytes are a strict prefix of $b_T(v)$, the bytes decompose into the generated prefix and residual bytes r:

$$b_T(v) = b_S(a) \,\|\, r, \qquad x' = (x, a), \tag{1}$$

where $\|$ denotes byte-string concatenation. Once the prefix is generated, the remaining requirement concerns the continuation from x'. Its valid one-step completions form a set of native student tokens:

$$\mathcal{C}(r) = \left\{ u \in V_S^{\text{cont}} : r \preceq b_S(u) \right\}, \tag{2}$$

where V_S^{cont} is the student content-token vocabulary, excluding special tokens, and $\preceq$ denotes byte-prefix inclusion. A valid token may complete the residual exactly or extend beyond its boundary: for residual `ead`, both `ead` and `eads` qualify if present in the vocabulary. These tokens satisfy the same residual byte constraint without necessarily representing equivalent full continuations. Matching separate token targets can additionally constrain their relative probabilities. This motivates a completion-set objective that supervises their aggregate probability while leaving within-set allocation to the remaining training objective.

We introduce **Event-Set Completion Distillation (ESCD)**, which combines byte-aligned root projection with explicit residual-event completion supervision. ESCD first aggregates prefix-related teacher events into groups, using the shortest member as a shared prefix constraint and summing their teacher probability mass. This aggregation preserves the included mass while coarsening longer members' constraints to the representative prefix. When a student action partially realizes this constraint, ESCD identifies the remaining bytes and constructs the valid one-step completion set at the naturally visited child state. For nonempty sets, ESCD applies a teacher-mass-weighted negative log-likelihood to their aggregate student probability, using probabilities from the native full-vocabulary distribution. The child objective supervises this aggregate probability without prescribing target probabilities for individual valid completions. The on-policy trajectory provides both the conditioning state and the student predictions needed by this objective. Completion candidates are evaluated through their probabilities at the visited child; they do not require separate sampled continuations. The auxiliary loss can therefore provide a completion signal even when the student's subsequently sampled token falls outside the valid set. ESCD requires neither counterfactual rollouts nor modifications to the student vocabulary. Its current formulation is local: residual events without a valid one-step completion contribute no auxiliary loss, while root-level supervision is retained.

Empirically, ESCD improves performance across model families, task domains, and scales. In controlled experiments with Qwen3.5-2B (Qwen Team, 2026a) as the student and Qwen3-32B (Qwen Team, 2025) or GLM-Z1-9B (GLM Team, 2024; Z.ai, 2025) as the teacher, ESCD matches or exceeds the strongest compared baseline on every reported metric. Gains reach 10.0 percentage points in mathematical reasoning and are particularly pronounced in code generation: under the Qwen teacher, pass@2 increases from 18.1% to 42.9% on LiveCodeBench and from 18.7% to 47.3% on TACO. Relative to the best baseline on each benchmark, **teacher performance recovery** rises from 62.0% to 79.8% for Qwen and from 64.0% to 72.2% for GLM. We further evaluate ESCD in large-scale MoE distillation with 397B Qwen and 1T Kimi teachers, transferring their capabilities to 30B and 35B students, respectively (Moonshot AI, 2026; Qwen Team, 2026b). ESCD improves scientific reasoning performance in both settings, with gains of 3.0–5.1 points over the strongest compared baseline. These results support cross-tokenizer OPD as a practical approach to transferring capabilities from large MoE teachers to smaller students for downstream deployment.

Our main contributions are summarized as follows:

- **From event entry to event completion.** We identify the **event-completion gap** in cross-tokenizer OPD: a student action can enter a teacher byte event without completing it, leaving a residual byte constraint at the visited student state. We show that completion should be separated from tokenizer-dependent probability allocation among individual student tokens, motivating completion-set supervision.
- **Event-Set Completion Distillation.** We propose ESCD, which aggregates prefix-related teacher events and uses their probability mass to supervise the total probability of byte-compatible student completions. ESCD complements root-level alignment with visited-child completion supervision, without prescribing individual completion probabilities, requiring counterfactual rollouts, or modifying the student vocabulary.
- **Cross-family transfer at scale.** ESCD achieves consistent improvements over compared baselines across mathematics, code, and scientific reasoning. The gains extend to heterogeneous MoE distillation with teachers up to 1T parameters, improving scientific reasoning benchmarks by 3.0–5.1 points. These results show the applicability of cross-tokenizer OPD across model families, architectures, and scales.

2 Related Work

2.1 Distillation for Language Models

Classical knowledge distillation transfers softened Teacher predictions to a smaller Student (Hinton et al., 2015), while SeqKD extends this approach to autoregressive generation through Teacher-decoded sequences (Kim & Rush, 2016). To reduce the mismatch between training and Student-induced states, ImitKD queries the Teacher on Student-generated prefixes (Lin et al., 2020), MiniLLM optimizes reverse KL on Student samples (Gu et al., 2026), and GKD supports mixtures of Student- and data-generated sequences with alternative divergences (Agarwal et al., 2024). Subsequent work explores generalized and skew divergences for sequence-level distillation (Wen et al., 2023; Ko et al., 2024). DistiLLM-2 further couples the objective with the source of training data through a contrastive formulation for Teacher- and Student-generated responses (Ko et al., 2025).

Recent studies investigate the reliability and efficiency of on-policy supervision. Entropy-Aware OPD augments reverse KL with forward KL at high-entropy Teacher predictions to preserve diversity (Jin et al., 2026). Other work identifies compatibility between Teacher and Student thinking patterns as a factor in OPD success and proposes off-policy cold starts and Teacher-aligned prompt selection (Li et al., 2026). Further analysis distinguishes rapid coverage of Student-visited states from the slower absorption of Teacher supervision (Fu et al., 2026). These findings motivate examining which states provide useful feedback and how feedback is translated into a training objective.

2.2 Cross-Tokenizer Distillation

Cross-tokenizer distillation must reconcile both vocabulary and sequence-boundary mismatches. FuseLLM approximately aligns heterogeneous token sequences using minimum edit distance (Wan et al., 2024), while DSKD uses learned projections to compare predictions in compatible output spaces (Zhang et al., 2024). ULD matches sorted probability profiles without retaining token identity (Boizard et al., 2025), and MultiLevelOT aligns logit distributions through token- and sequence-level optimal transport (Cui et al., 2025). GOLD extends cross-tokenizer supervision to on-policy training by merging text-aligned spans and combining direct matching on shared tokens with rank-based matching on unmatched tokens (Patiño et al., 2025).

Other approaches construct supervision through text or byte representations. ALM compares likelihoods of text-equivalent chunks (Minixhofer et al., 2025), CTLS develops cross-tokenization likelihood scoring algorithms (Phan et al., 2026), and BLD introduces an auxiliary byte-level prediction interface (Singh et al., 2026). SimCT builds minimal aligned multi-token units and compares their continuation scores (Sun et al., 2026), while X-Token constructs sparse vocabulary projections from token correspondences and multi-token decompositions (Sreenivas et al., 2026). BPM constructs Student-token targets through byte-prefix marginalization, including conditional targets at positions inside a teacher token (Wang et al., 2026a). More recently, ACTD addresses mapping noise through anchor supervision and regularization of the remaining Student vocabulary (Zhang et al., 2026).

Our contribution. Existing cross-tokenizer methods transfer supervision through aligned tokens, spans, or conditional distributions, but primarily construct comparable prediction targets rather than explicitly modeling completion after partial event entry. ESCD introduces **completion-set supervision** at naturally visited student child states, directly targeting the total probability of valid one-step completions while avoiding tokenizer-dependent probability allocation among individual completions. It aggregates prefix-related teacher events into representative prefix constraints and uses their combined probability mass to weight the completion loss. Combined with byte-aligned root projection, this objective couples **event entry and residual completion** without counterfactual rollouts.

3 Method

We introduce **E**vent-**S**et **C**ompletion **D**istillation (**ESCD**). When a sampled student token begins but does not complete a teacher byte event, ESCD supervises the total probability of next tokens completing the remaining bytes. It combines this completion target with probability alignment at the preceding state. We define the sampled action's residual (Sec. 3.1), then construct completion sets by aggregating prefix-related teacher events (Sec. 3.2), and define the training objective (Sec. 3.3).

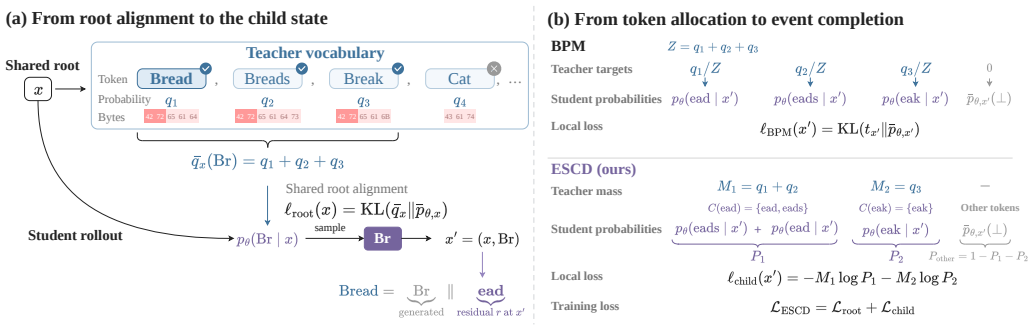


Figure 3: **ESCD: from event entry to event completion. (a) From event entry to residual completion.** At the shared root x, compatible teacher tokens contribute probability mass to the student action `Br`. Sampling `Br` reaches the visited state $x' = (x, \texttt{Br})$, where residual bytes such as `ead` remain to complete the teacher byte event `Bread`. **(b) From token allocation to event completion.** BPM constructs conditional targets for individual student tokens at the residual state, thereby specifying a probability allocation among possible completions. ESCD instead aggregates prefix-related teacher events and uses their combined mass to supervise the total probability of the byte-compatible one-step completion set. For example, both `ead` and `eads` satisfy the residual byte constraint `ead` without requiring a prescribed probability split. The final objective combines root alignment and completion supervision using student predictions at visited states, without additional rollouts.

3.1 From Event Entry to Event Completion

Let T be a frozen teacher and S_θ a trainable student with vocabularies V_T and V_S. At a student-visited root context x, the aligned predictions correspond to the same response-byte prefix, with next-token distributions $q_T(\cdot \mid x)$ and $p_\theta(\cdot \mid x)$. Here, "root" denotes the starting state of a local event, not the beginning of the response. Each model represents this shared byte prefix using its own tokenizer. Let b_T and b_S map tokens to their exact byte strings. A teacher event specifies the bytes of a teacher token, while a student action is a sampled student token. Let V_T^{cont} and V_S^{cont} denote the corresponding content-token vocabularies.

A teacher token may span multiple student actions. Figure 3(a) illustrates how sampling `Br` at the root x reaches the child state $x' = (x, \texttt{Br})$, leaving residual bytes `ead` for the teacher event `Bread`. Let $y = b_T(v)$ for a teacher content token $v \in V_T^{\text{cont}}$. If the student samples a content token a whose bytes form a nonempty strict prefix of y, the rollout reaches

$$x' = (x, a), \qquad r = y[|b_S(a)| :], \tag{3}$$

where $b_S(a) \prec y$ denotes nonempty strict byte-prefix inclusion: a produces part of y but leaves the event incomplete. The residual r is the suffix after removing $|b_S(a)|$ leading bytes from y.

Token-level alignment can supervise both event entry and intermediate student positions. In Figure 3(b), BPM assigns conditional target probabilities to individual student tokens, whereas ESCD supervises their total probability within each valid completion set. Thus, ESCD explicitly targets residual completion without requiring a particular probability split among valid next tokens. Appendix B.3 compares the supervision targets, and Appendix B.5 gives a numerical example.

3.2 Constructing Completion Sets

The single-event example extends to multiple teacher candidates, whose prefix-related byte strings can impose overlapping completion requirements. ESCD groups these candidates under a shared representative prefix and uses their combined teacher mass to weight its completion target. We use the full teacher vocabulary as the candidate set, $K_T(x) = V_T$, and retain content tokens whose complete byte strings have no exact single-token counterpart in the student content vocabulary:

$$\mathcal{V}(x) = \left\{ v \in K_T(x) \cap V_T^{\text{cont}} : \nexists u \in V_S^{\text{cont}},\ b_T(v) = b_S(u) \right\}. \tag{4}$$

We connect two candidates in $\mathcal{V}(x)$ whenever either byte string is a prefix of the other and use the resulting connected components as aggregation groups. For component g with members V_g, define

$$M_g = \sum_{v \in V_g} q_T(v \mid x), \qquad y_g = b_T(v_g^\star), \qquad v_g^\star \in \arg\min_{v \in V_g} |b_T(v)|, \tag{5}$$

with deterministic tie breaking. The shortest member defines a prefix shared by all group members. Aggregation preserves their total teacher mass while coarsening their byte constraints to this representative prefix. For example, grouping `Bread` and `Breads` retains their shared `Bread` requirement but does not require the additional `s`. Thus, M_g weights completion of the shared prefix, rather than exact reconstruction of every original teacher event.

For each group satisfying $b_S(a) \prec y_g$, the sampled action a partially realizes the representative event, leaving residual bytes $r_g = y_g[|b_S(a)| :]$. At the visited child state $x' = (x, a)$, we define the valid one-step completion set and its aggregate probability:

$$\mathcal{C}(r_g) = \{u \in V_S^{\text{cont}} : r_g \preceq b_S(u)\}, \qquad P_\theta(\mathcal{C}(r_g) \mid x') = \sum_{u \in \mathcal{C}(r_g)} p_\theta(u \mid x'), \tag{6}$$

where $\preceq$ denotes byte-prefix inclusion, allowing equality. Probabilities come from the student's native full-vocabulary distribution, without renormalizing over content tokens or the completion set. The sums play different roles: teacher aggregation determines the representative constraint's weight M_g, while student marginalization measures the probability of satisfying it in one token.

A valid completion must begin with the entire residual but may extend beyond it. In Figure 3(b), the `Bread`/`Breads` group has weight $M_1 = q_1 + q_2$. After sampling `Br`, both `ead` and `eads` complete the representative residual `ead`, giving $P_1 = p_\theta(\texttt{ead} \mid x') + p_\theta(\texttt{eads} \mid x')$ in the illustrative vocabulary. Satisfying this shared constraint does not imply equivalent full continuations. For a fixed action, distinct eligible groups have disjoint completion sets because neither representative prefix is a prefix of the other. Empty completion sets contribute no child loss.

3.3 Training Objective

We reuse BPM's byte alignment and root projection, without retaining its full interior- and spanning-position supervision. At an aligned root x, teacher mass is mapped to selected student tokens: Figure 3(a), for example, maps `Bread`, `Breads`, and `Break` to `Br`. The projected distributions $\bar{q}_x$ and $\bar{p}_{\theta,x}$ share these token entries and a complement $\perp$ holding their respective remaining mass (Appendix B.4). Without renormalizing over explicit entries, we apply forward KL:

$$\ell_{\text{root}}(x) = \text{KL}(\bar{q}_x \,\|\, \bar{p}_{\theta,x})\,. \tag{7}$$

For the sampled action a, let $\mathcal{G}(x, a)$ contain groups with $b_S(a) \prec y_g$ and nonempty completion sets $\mathcal{C}(r_g)$, where $r_g = y_g[|b_S(a)| :]$. At the visited child state $x' = (x, a)$, the completion loss is

$$\ell_{\text{child}}(x, a) = -\sum_{g \in \mathcal{G}(x,a)} M_g \log \left[\sum_{u \in \mathcal{C}(r_g)} p_\theta(u \mid x, a) \right]. \tag{8}$$

The weights M_g retain their root teacher mass without renormalization over eligible groups; student probabilities use the native full-vocabulary distribution at x'. This loss supervises completion-set probability without individual token targets, regardless of whether the next sampled token belongs to the set. It is zero when $\mathcal{G}(x, a)$ is empty. For $g \in \mathcal{G}(x, a)$, write $\mathcal{C}_g = \mathcal{C}(r_g)$ and $P_g = P_\theta(\mathcal{C}_g \mid x')$. Selecting a single valid completion $u^\star \in \mathcal{C}_g$ yields

$$-M_g \log p_\theta(u^\star \mid x') = -M_g \log P_g - M_g \log \frac{p_\theta(u^\star \mid x')}{P_g}. \tag{9}$$

The first term encourages completion-set probability; the second favors the selected token within the set. ESCD's child loss retains only the first term per group, supervising valid completions through their aggregate probability without prescribing individual token probabilities. The local gradient analysis examines how this target choice affects agreement with a byte-event reference gradient.

Table 1: Main results of cross-tokenizer on-policy distillation. The student is Qwen3.5-2B, with Qwen3-32B and GLM-Z1-9B as teachers. We report avg@k and pass@k (%) for each benchmark.

Method	Mathematics ($k = 8$)								Code ($k = 2$)			
	AIME24		AIME25		AIME26		HMMT-26		LiveCodeBench		TACO	
	avg@8	pass@8	avg@8	pass@8	avg@8	pass@8	avg@8	pass@8	avg@2	pass@2	avg@2	pass@2
Qwen3.5-2B (base)	10.4	30.0	10.0	20.0	5.0	20.0	4.2	15.2	11.5	13.7	5.3	7.1
Qwen3-32B†	79.6	90.0	69.2	83.3	72.1	90.0	27.7	48.5	59.1	65.9	53.2	56.9
GOLD	23.3	46.7	24.2	53.3	22.9	56.7	18.9	33.3	14.0	14.8	6.9	10.2
X-Token	28.8	46.7	26.7	50.0	24.2	46.7	14.0	33.3	12.6	18.1	8.0	12.4
SimCT	35.0	70.0	34.2	63.3	34.2	63.3	20.8	**42.4**	14.8	14.8	5.5	9.9
BPM	38.8	66.7	32.1	60.0	35.0	56.7	21.2	36.4	15.4	15.9	11.5	18.7
Ours	**46.7**	**80.0**	**37.9**	**66.7**	**43.8**	**66.7**	**27.7**	**42.4**	**31.0**	**42.9**	**30.0**	**47.3**
Δ vs. best baseline	+7.9	+10.0	+3.7	+3.4	+8.8	+3.4	+6.5	0.0	+15.6	+24.8	+18.5	+28.6
GLM-Z1-9B†	65.0	83.3	54.6	76.7	65.4	90.0	22.7	45.5	46.4	55.5	52.1	56.9
GOLD	30.8	63.3	26.3	53.3	27.5	56.7	12.5	27.3	13.2	17.0	9.9	14.5
X-Token	31.3	56.7	27.5	50.0	30.4	56.7	12.5	21.2	14.6	19.8	8.8	14.1
SimCT	31.3	63.3	27.5	**56.7**	29.6	60.0	19.7	33.3	14.8	17.6	9.0	13.1
BPM	30.8	56.7	25.4	46.7	32.1	53.3	20.8	36.4	22.0	25.8	16.4	23.3
Ours	**35.8**	**66.7**	**29.6**	**56.7**	**35.0**	**70.0**	**25.4**	**39.4**	**28.3**	**35.2**	**22.6**	**29.3**
Δ vs. best baseline	+4.5	+3.4	+2.1	0.0	+2.9	+10.0	+4.6	+3.0	+6.3	+9.4	+6.2	+6.0

Root and child losses are assigned to their respective prediction positions, with zero contributions where inapplicable, and summed before applying the training mask and reduction. Writing their contributions under this common reduction as $\mathcal{L}_{\text{root}}$ and $\mathcal{L}_{\text{child}}$, we obtain

$$\mathcal{L}_{\text{ESCD}} = \mathcal{L}_{\text{root}} + \mathcal{L}_{\text{child}}. \tag{10}$$

The terms are not separately averaged over eligible positions. Child supervision reuses student logits at visited states without additional rollouts. Algorithm 1 summarizes training.

4 Experiments

4.1 Experiment Settings

We use **Qwen3.5-2B** as the student with two teachers: **Qwen3-32B**, which differs in tokenizer within the Qwen family, and **GLM-Z1-9B**, which differs in both model family and tokenizer. Unless otherwise stated, our controlled algorithmic and cross-tokenizer analyses use these pairs. We evaluate mathematical reasoning on AIME 2024–2026 and HMMT 2026 (Balunović et al., 2026), and code generation on LiveCodeBench (Jain et al., 2024) and TACO (Li et al., 2023). For large-scale validation, we evaluate **Qwen3.5-397B-A17B** → **Qwen3-30B-A3B-Thinking** and **Kimi-K2.7-Code** → **Qwen3.6-35B-A3B** on FrontierScience Olympiad (Wang et al., 2026b) and our physics benchmark PHYRD-40. In result tables, † marks teacher scores as references for student performance. Benchmark details and evaluation protocols are in Appendix A.5.

4.2 Distillation Performance

Table 1 presents the main cross-tokenizer distillation results with Qwen3.5-2B as the student under two different teachers. ESCD consistently outperforms the representative baselines across both settings. With Qwen3-32B as the teacher, ESCD delivers clear gains on all mathematics benchmarks, with improvements of up to $+8.8$ and $+10.0$ points, while the advantage becomes substantially larger on code generation, reaching $+15.6/+24.8$ points on LiveCodeBench and $+18.5/+28.6$ points on TACO. The same trend remains with GLM-Z1-9B: ESCD improves over the strongest baselines across nearly all benchmarks, including gains of $+2.9/+10.0$ points on AIME26, $+6.3/+9.4$ points on LiveCodeBench, and $+6.2/+6.0$ points on TACO. These consistent improvements across two teachers of different model families and capacities demonstrate the robustness of ESCD, with **particularly strong benefits on code generation**.

Table 2 extends our dense-model evaluation to heterogeneous MoE distillation at Teacher–Student total-parameter ratios of $13.2\times$ (397B Qwen) and $28.6\times$ (1T Kimi). **ESCD improves over BPM in both settings**, using direct OPD for Qwen and SFT-initialized OPD for Kimi. In the Kimi setting,

Table 2: Cross-tokenizer OPD on heterogeneous MoE pairs with 397B and 1T teachers. In the Kimi setting, BPM and ESCD use the same SFT-initialized checkpoint due to direct OPD instability.

Method	FrontierScience Olympiad-100	PHYRD-40
Qwen3.5-397B-A17B†	70.0	79.8
Qwen3-30B-A3B-Thinking (base)	48.0	43.0
BPM	48.0	46.6
Ours	**51.0**	**49.6**
Δ vs. best baseline	+3.0	+3.0
Kimi-K2.7-Code†	75.0	83.6
Qwen3.6-35B-A3B (base)	61.0	59.2
Qwen3.6-35B-A3B (SFT)	68.0	66.2
BPM	69.0	69.2
Ours	**72.0**	**74.3**
Δ vs. best baseline	+3.0	+5.1

direct BPM-based OPD exhibits repetitive continuations and premature termination (Appendix C.6), highlighting a stability challenge in this configuration. Starting from the SFT checkpoint, ESCD further improves FrontierScience Olympiad accuracy from 68.0% to 72.0% and the PHYRD-40 mean score from 66.2 to 74.3. These results support effective cross-tokenizer transfer under both initialization regimes and **the complementary roles of SFT and OPD**: SFT provides the initialization, while subsequent OPD with ESCD yields additional performance gains.

Scope and training stability. ESCD addresses supervision across tokenizer boundaries, while OPD performance and stability also depend on student capacity, initialization, training data, and rollout quality. Event-completion supervision therefore complements initialization and optimization strategies rather than providing a general remedy for OPD collapse.

5 Ablation and Mechanism Analysis

We examine two ESCD questions: *how preserving the **completion set** affects agreement with a byte-event reference gradient*, and *how frequently **one-step completion** is available along on-policy student trajectories*. The first compares local objectives on fixed student predictions to isolate their immediate gradient effects; the second measures child-supervision availability.

On matched student contexts and teacher events, we compare three local objectives while holding student predictions, teacher masses, and BPM root supervision fixed. **Root** uses only root supervision, with no additional loss at the visited child. **Single** adds a child loss for one deterministically selected valid completion: its negative log-probability is weighted by the teacher event mass. **Event** uses the same mass weight and visited child, but applies the negative log to the summed probability of all valid one-step completions, without prescribing individual token probabilities. Thus, Root versus Event tests the effect of adding completion-set supervision, while Single versus Event tests the effect of preserving the set rather than selecting one member. Figure 6 illustrates these differences.

We evaluate supervision objectives on previously generated student trajectories without additional model training, using **Cross-Only Update Fidelity (COUF)** to measure agreement with a specified byte-event reference gradient. Holding student predictions, teacher events and their masses, conditioning contexts, and root supervision fixed enables a controlled comparison at a common model state. The reference differentiates an objective based on event probabilities summed over compatible student paths, including alternative first-token branches and multi-token realizations, whereas Event supervises one-step completion at the visited child. For each context x, candidate and reference gradients with respect to student logits at the root and the same depth-one states are concatenated in a common coordinate order into z_x and $z_x^\star$. COUF captures differences in both direction and magnitude by aggregating squared gradient errors across contexts and normalizing by total reference-gradient energy; Appendix C.1 further examines child-state selection:

$$\mathrm{COUF} = 1 - \frac{\sum_x \|z_x - z_x^\star\|_2^2}{\sum_x \|z_x^\star\|_2^2}. \tag{11}$$

Table 3: Local target comparison on contexts satisfying the COUF diagnostic criteria. We report reference-energy-weighted state-space COUF; Δ denotes Event minus Single.

Teacher → Student	Root	Single	Event	Δ
Qwen3 → Qwen3.5	0.7383	0.4323	**0.8495**	+0.4172
GLM-Z1 → Qwen3.5	0.8848	0.7493	**0.9341**	+0.1848

Table 4: Full-vocabulary child-supervision availability. Observed is relative to branchable teacher mass; 1-step and Deeper characterize representative residuals within visited mass.

Teacher → Student	Observed	1-step	Deeper
Qwen3 → Qwen3.5	81.83%	99.01%	0.99%
GLM-Z1 → Qwen3.5	83.29%	99.43%	0.57%

A value of 1 indicates exact reference agreement, 0 matches the zero-gradient baseline, and negative values indicate greater error than that baseline. This diagnostic evaluates local logit gradients, not downstream performance after independent training.

Table 3 shows higher reference agreement for **Event** than **Root**, increasing COUF from 0.7383 to 0.8495 for Qwen and from 0.8848 to 0.9341 for GLM. **Single** falls below Root in both settings. These results support preserving the completion set over selecting one valid token in the examined contexts. Event need not attain a score of 1: its root gradient, restriction to the visited child, and local completion objective differ from the reference.

The completion set also admits tokens extending beyond the residual boundary, such as `eads` alongside `ead` in Figure 3(b). In a separate diagnostic on naturally visited strict-prefix events with available child logits, boundary-crossing tokens account for 2.10% and 0.63% of completion probability, respectively. Removing them yields gradients with cosine similarities of 0.9832 and 0.9887 to the full-set gradients. These results indicate that boundary-crossing tokens have limited impact on the local gradient direction in the analyzed subsets.

5.1 On-Policy Availability of Child Supervision

We examine child-supervision availability using the full teacher vocabulary at aligned, non-whitespace positions in 16 frozen student trajectories per pair. After exact-shared and invalid-token filtering, we aggregate candidates into representative events. Among events admitting a strict-prefix student action, **Observed** measures the teacher mass fraction whose compatible child is actually visited. Conditional on this visited mass, **1-step** measures the fraction admitting completion by one additional student token; the remainder is reported as **Deeper**.

Table 4 shows that compatible children are visited for 81.83% and 83.29% of branchable teacher mass for Qwen and GLM, respectively. Conditional on these visited child states, representative residual constraints covering 99.01% and 99.43% of the visited teacher mass admit at least one student token that completes the residual bytes in one additional step. These results show that after partial event entry, the vast majority of observed residual constraints are covered by ESCD's one-step child supervision, supporting the practical scope of the proposed objective.

6 Conclusion

We identify the **event-completion gap** in cross-tokenizer on-policy distillation: a student action can enter a teacher byte event while leaving a residual byte constraint whose completion is not uniquely represented by the student tokenizer. We introduce **Event-Set Completion Distillation (ESCD)**, which complements root-level alignment with completion-set supervision at naturally visited student child states. ESCD aggregates prefix-related teacher events and supervises the total probability of byte-compatible student completions, without prescribing individual completion probabilities or requiring counterfactual rollouts. Experiments show consistent gains across mathematics, code, and scientific reasoning, including heterogeneous MoE distillation from a 1T teacher to a 35B student. Local analyses support preserving the completion set, while one-step-completable residual constraints cover over 99% of observed compatible teacher mass. These findings support event entry and event completion as complementary supervision targets for cross-tokenizer knowledge transfer.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing, translation assistance for author-written drafts, and limited code assistance (e.g., debugging and refactoring of training and evaluation scripts). We did not use generative AI tools to generate experimental results, fabricate data, or make final scientific claims. Separately from writing assistance, large language models appear in this work as research objects: all teacher and student models are publicly available checkpoints studied in our experiments. In addition, following common practice for open-ended scientific answers, we use DeepSeek-V4 Pro Preview as an automatic grader for FrontierScience Olympiad and PHYRD-40, conditioned on reference answers, reference solutions, and problem-specific rubrics; the grading protocol is described in Appendix A.5, and the same grader and prompts are applied to all compared models. All AI-assisted text, code, and suggestions were manually reviewed, revised, and verified by the authors. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies cross-tokenizer on-policy distillation, a technique for transferring capabilities from a teacher language model to a student with a different tokenizer. Such techniques can reduce the cost of deploying capable reasoning models, but distillation may also transfer undesirable behaviors of the teacher, including factual errors, social biases, and unsafe outputs; our evaluation focuses on reasoning accuracy and does not assess safety or bias, so distilled models should undergo separate safety evaluation before deployment. Cross-tokenizer distillation may further be used to replicate capabilities of models whose terms of use restrict distillation; all teacher and student models in this work are publicly released checkpoints used within their license terms, and we encourage practitioners to respect the licenses of the models they distill. Our training and evaluation data consist of publicly available mathematics, code, and science benchmarks together with SciDeriv, which is derived from mathematical and scientific documents and contains prompts only. These data do not involve personally identifiable information, and our work does not involve human-subject experiments. The PHYRD-40 problems were written by invited domain scientists, and only problem statements, reference solutions, and rubrics are used. Model-generated code is executed only in isolated sandboxed environments. Finally, our experiments require substantial GPU resources (Table 6); ESCD itself reuses on-policy trajectories and requires no additional rollouts.

REPRODUCIBILITY STATEMENT

We support reproducibility through detailed descriptions of our method, training and evaluation pipeline, and data in the main paper and appendix. Section 3 specifies the residual-event formulation, completion-set construction, and training objective, while Appendix B.4 provides further details, Appendix B.5 gives a worked example, and Algorithm 1 summarizes the training procedure. Appendix B.3 presents the compared baselines under a unified view, and Appendix A.1 describes their shared implementation framework. Chat-template rendering, retokenization, and special-token handling are detailed in Appendix A.2. Key OPD hyperparameters, rollout settings, and hardware configurations are summarized in Table 6; training data composition and benchmark filtering are described in Appendices A.4 and A.5; and decoding settings are given in Table 9. All teacher and student models are initialized from publicly available checkpoints. We will release our code, including the ESCD and baseline plugins, the SciDeriv prompts, the PHYRD-40 problem set with reference solutions and scoring rubrics, and the evaluation scripts and grading prompts to support reproduction of the reported results.

REFERENCES

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

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions, 2026. URL https://arxiv.org/abs/2505.23281.

Nicolas Boizard, Kevin El Haddad, Céline Hudelot, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for llms, 2025. URL https://arxiv.org/abs/2402.12030.

Xiao Cui, Mo Zhu, Yulei Qin, Liang Xie, Wengang Zhou, and Houqiang Li. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models, 2025. URL https://arxiv.org/abs/2412.14528.

Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan ang Gao, Yudong Wang, Zhiyuan Liu, Ning Ding, and Chaojun Xiao. Rethinking on-policy distillation of large language models ii: One training example, 2026. URL https://arxiv.org/abs/2609.04172.

GLM Team. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: On-policy distillation of large language models, 2026. URL https://arxiv.org/abs/2306.08543.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 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, 2024. URL https://arxiv.org/abs/2403.07974.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models, 2026. URL https://arxiv.org/abs/2603.07079.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation, 2016. URL https://arxiv.org/abs/1606.07947.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models, 2024. URL https://arxiv.org/abs/2402.03898.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. Distillm-2: A contrastive approach boosts the distillation of llms, 2025. URL https://arxiv.org/abs/2503.07067.

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, 2023. URL https://arxiv.org/abs/2312.14852.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe, 2026. URL https://arxiv.org/abs/2604.13016.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning, 2020. URL https://arxiv.org/abs/2009.07253.

Benjamin Minixhofer, Ivan Vulić, and Edoardo Maria Ponti. Universal cross-tokenizer distillation via approximate likelihood matching, 2025. URL https://arxiv.org/abs/2503.20083.

Moonshot AI. Kimi k2.7 code: An open-source, coding-focused agentic model built for long-horizon software engineering., July 2026. URL https://www.kimi.com/resources/kimi-k2-7-code.

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. https://huggingface.co/spaces/HuggingFaceH4/on-policy-distillation, 2025.

Buu Phan, Ashish Khisti, and Karen Ullrich. Cross-tokenizer likelihood scoring algorithms for language model distillation, 2026. URL https://arxiv.org/abs/2512.14954.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

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

Sharath Turuvekere Sreenivas, Adithyakrishna Venkatesh Hanasoge, Mingyu Yang, Ali Taghibakhshi, Saurav Muralidharan, Ashwath Aithal, and Pavlo Molchanov. X-token: Projection-guided cross-tokenizer knowledge distillation, 2026. URL https://arxiv.org/abs/2605.21699.

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, 2026. URL https://arxiv.org/abs/2605.07711.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. Knowledge fusion of large language models, 2024. URL https://arxiv.org/abs/2401.10491.

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

Miles Wang, Robi Lin, Kat Hu, Joy Jiao, Neil Chowdhury, Ethan Chang, and Tejal Patwardhan. Frontierscience: Evaluating ai's ability to perform expert-level scientific tasks, 2026b. URL https://arxiv.org/abs/2601.21165.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation, 2023. URL https://arxiv.org/abs/2307.15190.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Z.ai. GLM-Z1-9B-0414: Model card. Hugging Face model card, https://huggingface.co/zai-org/GLM-Z1-9B-0414, 2025.

Huiyi Zhang, Zijian Li, Xiaocheng Feng, Weitao Ma, Xiaoliang Yang, Yichong Huang, and Bing Qin. Actd: Anchor-based cross-tokenizer distillation with residual regularization, 2026. URL https://arxiv.org/abs/2608.29662.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. Dual-space knowledge distillation for large language models, 2024. URL https://arxiv.org/abs/2406.17328.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An llm post-training framework for rl scaling. https://github.com/THUDM/slime, 2025. GitHub repository. Corresponding author: Xin Lv.

A Training Framework and Settings

This appendix documents the training and evaluation implementation used throughout our experiments. We organize the details as follows:

1. **Training framework** (A.1): on-policy rollout, Teacher serving, Student optimization, and the unified implementation of cross-tokenizer baselines.
2. **Cross-model chat templates and special-token handling** (A.2): model-native prompt rendering, response retokenization, EOS bridging, truncation, and thinking markers.
3. **Training settings** (A.3): optimization hyperparameters, rollout budgets, and distributed-training configuration.
4. **Training data** (A.4): the mathematics and code prompts used for distillation.
5. **Benchmarks and evaluation** (A.5): benchmark construction and task-specific grading protocols.
6. **Common generation protocol** (A.6): decoding settings, sampling budgets, response-length limits, and truncation rules shared across compared models.

A.1 Training Framework

We conduct all on-policy distillation experiments with Slime (Zhu et al., 2025), which integrates Student rollout generation, Teacher scoring, and distributed Student optimization. At each step, the current Student generates responses with SGLang; these responses are retokenized and scored by a frozen Teacher, and only the Student is updated using the resulting cross-tokenizer supervision.

BPM, SimCT, GOLD, X-Token, and ESCD are implemented through a unified plugin interface. Within each Teacher–Student pair, all methods share the training data, rollout settings, and optimization budget, while each generates trajectories with its own Student and obtains Teacher predictions on those trajectories. The plugins implement method-specific alignment and supervision objectives. The Teacher is served with SGLang, while the Student is optimized with Megatron-LM (Shoeybi et al., 2019) through Slime's distributed actor backend. This setup provides a common infrastructure for rollout, scheduling, optimization, checkpointing, and evaluation while preserving on-policy training for each method.

A.2 Cross-Model Chat Templates and Special-Token Handling

Cross-family distillation introduces implementation details absent when Teacher and Student share a tokenizer and chat template. Directly copying token IDs or serialized prompts across models can misalign role boundaries, duplicate termination symbols, or misrepresent truncated responses as completed. We therefore separate two operations: (1) each model renders the structured conversation with its native chat template, and (2) model-specific special tokens are handled through explicit semantic rules rather than token-ID equality.

For the three model families used in our experiments, the native chat templates differ substantially in their role and control tokens. Figure 4 shows their template structures. In all cases, we retain the original structured conversation and let each model render it using its own native chat template with thinking enabled. We never translate model-specific template tokens between tokenizers.

A.2.1 Special-Token Handling

Different model families use different tokens to mark the end of a response. We follow each model's native generation semantics rather than assuming EOS token IDs or surface forms are shared across tokenizers. In particular, we recognize tokenizer-declared EOS tokens together with stop tokens declared by the model's generation configuration, while other special tokens are not treated as terminal by default. Table 5 summarizes the model-specific termination rules used in our experiments.

Termination and truncation. When a Student rollout ends with one of its recognized termination tokens, we remove that final token before decoding and retokenizing the response content for the Teacher. After retokenization, a Teacher-native bridge stop is appended to preserve the fact that the



*The Qwen3-32B template omits this opening <think> in thinking mode.

Figure 4: **Native chat-template prefixes across model families.** We illustrate a text-only conversation without tool calls, using each checkpoint's native generation prefix in its thinking configuration. Colored text denotes template-supplied prefixes; *generated text* begins after the displayed prefix. GLM-Z1-9B and Qwen3-32B do not prefill an opening <think>, whereas the other illustrated thinking configurations do. Model-specific role and control tokens remain local to their respective tokenizers; Student-generated response text is retokenized for Teacher scoring. Layout line breaks are schematic; exact whitespace follows each checkpoint's template.

Table 5: Model-specific termination handling. The bridge stop is the token used to represent a normally completed response after cross-tokenizer retokenization.

Model	Recognized termination	Preferred bridge stop
Qwen3 / Qwen3.5	model-declared EOS / stop tokens	<|im_end|>
GLM-Z1	<|endoftext|> and configured stops	<|endoftext|>
Kimi-K2.7-Code	[EOS], <|im_end|>	<|im_end|>

original response terminated normally. The bridge token is model specific: for example, GLM-Z1 uses <|endoftext|>, whereas Kimi-K2.7-Code uses <|im_end|> even though its tokenizer also defines [EOS]. In contrast, if generation reaches the maximum response length without producing a termination token, the response is treated as truncated and no EOS or bridge stop is added.

During cross-tokenizer supervision, these model-specific stop tokens are treated as the same semantic terminal event rather than as ordinary lexical content. Thus, termination can be aligned across models without requiring their EOS token IDs or textual forms to match. GLM-Z1 additionally declares role or interaction boundaries such as <|user|> and <|observation|> as generation stops; we respect these declarations when determining whether a rollout has terminated.

Thinking markers. We follow each checkpoint's native thinking-mode generation prefix (Figure 4). Template prefilling of <think> is checkpoint-specific: Qwen3-32B and GLM-Z1-9B do not prefill it, whereas the other illustrated configurations do. We do not impose a common opening marker during cross-model conversion. Any generated <think> or </think> is retained and retokenized with the surrounding text for Teacher scoring. Neither marker is treated as a termination event unless explicitly declared by the model configuration. If generation is truncated before </think>, we do not insert a closing marker. Prompt-side role and control markers remain model-specific, while generated reasoning markers are transferred as response text rather than copied as tokenizer-local IDs.

A.3 Training Settings

Table 6 summarizes the configurations for small-scale and large-scale OPD experiments. The small-scale experiments use Qwen3.5-2B as the Student with Qwen3-32B or GLM-Z1-9B as the Teacher. The large-scale experiments cover Qwen3.5-397B-A17B to Qwen3-30B-A3B-Thinking and Kimi-K2.7-Code to Qwen3.6-35B-A3B distillation. For the Kimi-K2.7-Code to Qwen3.6-35B-A3B setting, both BPM and ESCD are applied on the same SFT-initialized Student checkpoint to ensure a fair comparison. Within each Teacher–Student pair, compared methods share the training data,

Table 6: Training configurations for OPD experiments. GPU totals include Teacher serving, Student optimization, and Student rollout, excluding separately scheduled evaluation jobs.

Setting	**Small-scale**	**Large-scale**	
Teacher	Qwen3-32B / GLM-Z1-9B	Qwen3.5-397B-A17B	Kimi-K2.7-Code
Student	Qwen3.5-2B	Qwen3-30B-A3B-Thinking	Qwen3.6-35B-A3B
Training framework	Slime	Slime	Slime
Student optimization	Megatron-LM	Megatron-LM	Megatron-LM
Rollout and Teacher serving	SGLang	SGLang	SGLang
Training data	DAPO-Math + TACO	SciDeriv	SciDeriv
Number of training prompts	20,000	17,577	17,577
Global rollout batch size	16	16	16
Global training batch size	16	16	16
Responses per prompt	1	1	1
Micro-batch size	1	1	1
Maximum prompt length	5,120 tokens	4,096 tokens	4,096 tokens
Maximum response length	27,648 tokens	27,648 tokens	80,000 tokens
Maximum sequence length	33,280 tokens	33,280 tokens	84,112 tokens
Rollout temperature	1.0	1.0	1.0
Rollout top-p	0.95	0.95	0.95
Rollout top-k	20	20	20
Learning rate	5×10^{-7}	1×10^{-6}	1×10^{-6}
Weight decay	0.1	0.1	0.1
Tensor parallel size	2	2	2
Pipeline parallel size	2	4	2
Context parallel size	8	8	8
Checkpoint interval	50 steps	10 steps	10 steps
Evaluation interval	50 steps	10 steps	10 steps
Teacher GPUs	16	32	32
Student training GPUs	32	64	64
Student rollout GPUs	16	32	32
Total GPUs	64	128	128

rollout budget, optimizer settings, and distributed training configuration, while differing in their cross-tokenizer objectives. Training datasets are described in Section A.4.

A.4 Training Data

We use two training corpora depending on model scale. For small-scale distillation, we use a balanced mathematics–code mixture of 20,000 prompts: 10,000 DAPO mathematics problems with exact integer answers (Yu et al., 2025) and 10,000 TACO programming problems with recovered executable test suites (Li et al., 2023). Table 7 summarizes the composition.

For large-scale distillation, we construct **SciDeriv**, a scientific derivation reasoning corpus derived from mathematical and scientific documents. A document-to-prompt pipeline extracts formulas and intermediate derivation relations from source PDFs and converts them into structured reasoning problems. After filtering, SciDeriv contains 17577 prompts covering two complementary tasks: conclusion assessment and conclusion reconstruction (Table 8).

In the 10000 conclusion-assessment prompts, a candidate conclusion is provided, and the model must determine whether it is entailed by the supplied relations, merely compatible with them, or inconsistent with them. In the 7577 conclusion-reconstruction prompts, the endpoint is withheld, and the model must reconstruct the strongest conclusion justified by the available relations.

Both tasks require reasoning over intermediate transformations, including identifying relevant assumptions, ambiguities, normalization choices, and convention-dependent steps. They also re-

Table 7: Training data for small-scale distillation.

Task	Mathematics	Code
Source	DAPO-Math	TACO
Prompts	10,000	10,000

Table 8: Composition of SciDeriv for large-scale distillation.

Task type	Prompts
Conclusion assessment	10,000
Conclusion reconstruction	7,577
Total	17,577

quire recognizing when the supplied information is insufficient to determine a unique conclusion. SciDeriv contains prompts only, without stored reference completions or scalar reward labels.

Case 1: Endpoint exposed and verifiable

<|im_start|>system
You are a rigorous mathematical reasoning assistant. Analyze the supplied derivation, determine whether the proposed endpoint follows from the given relations, and explicitly identify any required assumptions or conventions.
<|im_end|>

<|im_start|>user
Consider the phase convention

$$S(\theta) = \epsilon e^{i\delta(\theta)}, \qquad \epsilon \in \{+1, -1\},$$

together with

$$\rho(i\pi - \theta) = \rho(i\pi + \theta),$$
$$\rho(\theta) - \rho(-\theta) = i\delta(\theta) + \frac{i\pi(1-\epsilon)}{2}\,\mathrm{sign}(\theta).$$

Define

$$\log F_{\min}(\theta) = \rho(\theta).$$

Determine whether the following endpoint relations are implied by the displayed identities:

$$F_{\min}(\theta) = F_{\min}(2\pi i - \theta), \qquad F_{\min}(\theta) = S(\theta) F_{\min}(-\theta).$$

Explain the derivation carefully and state any branch, normalization, or sign conventions needed for the conclusion.
<|im_end|>

<|im_start|>assistant
<think>
{*model-generated reasoning begins here; no reference completion is stored*}

Case 2: Endpoint withheld and reconstructed

<|im_start|>system
You are a rigorous mathematical reasoning assistant. Reconstruct the strongest conclusion justified by the supplied derivation, explain which intermediate relations support each part of the result, and identify any remaining ambiguity or missing condition.
<|im_end|>

<|im_start|>user
Consider

$$S(\theta) = \epsilon e^{i\delta(\theta)}, \qquad \epsilon \in \{+1, -1\},$$

and the identities

$$\rho(i\pi - \theta) = \rho(i\pi + \theta),$$
$$\rho(\theta) - \rho(-\theta) = i\delta(\theta) + \frac{i\pi(1-\epsilon)}{2}\,\mathrm{sign}(\theta).$$

Let

$$\log F_{\min}(\theta) = \rho(\theta).$$

Using only these relations, reconstruct the functional equations satisfied by $F_{\min}$. For each equation, identify the intermediate relation from which it follows. Then determine whether the information above uniquely specifies $F_{\min}$, and state the minimal additional condition required if it does not.
<|im_end|>

<|im_start|>assistant
<think>
{*model-generated reasoning begins here; no reference completion is stored*}

Representative case studies. The assistant field marks only the generation boundary used during training and does not contain a stored reference response. Cases 1 and 3 illustrate the endpoint-exposed setting with different logical statuses of the proposed conclusion, whereas Case 2 illustrates endpoint reconstruction when the conclusion is withheld.

Taken together, the examples highlight the central objective of our formula-reasoning corpus: supervising the *logical relationship* between a derivation and its conclusion rather than only the surface correctness of the final formula. Endpoint-exposed prompts test whether a conclusion is entailed, compatible, or inconsistent with the premises, while endpoint-withheld prompts test reconstruction of the strongest justified conclusion. Both settings require tracking intermediate transformations, implicit assumptions, solution ambiguities, and convention-dependent conditions.

Case 3: Consistent but not entailed

<|im_start|>system
You are a rigorous mathematical reasoning assistant. Distinguish carefully between conclusions that are logically entailed by the derivation, conclusions that are merely compatible with it, and conclusions that contradict it. Explicitly identify missing assumptions.
<|im_end|>

<|im_start|>user
Consider the differential equation

$$\frac{dy}{dx} = 2xy.$$

For a nonzero solution, separation of variables gives

$$\frac{1}{y}\,dy = 2x\,dx,$$

and hence

$$\log|y| = x^2 + C.$$

A proposed endpoint is

$$y(x) = e^{x^2}.$$

Determine whether this endpoint is entailed by the derivation, merely consistent with it, or inconsistent with it. Give the strongest solution family supported by the available information, identify any additional condition needed to recover the proposed endpoint, and discuss whether the derivation omits any special solution.
<|im_end|>

<|im_start|>assistant
<think>
{*model-generated reasoning begins here; no reference completion is stored*}

A.5 Benchmarks and Evaluation Protocol

We evaluate mathematical reasoning, code generation, and scientific reasoning on fixed benchmark sets. The problem sets and evaluation protocols are held constant across all models being compared.

AIME 2024, AIME 2025, and AIME 2026. (Balunović et al., 2026) Each AIME split contains 30 olympiad-style problems with reference answers given as integers in $[0, 999]$. We sample eight independent responses per problem. A response is considered correct if the integer extracted from its final answer matches the reference answer.

HMMT February 2026. (Balunović et al., 2026) HMMT contains 33 contest problems. We sample eight independent responses per problem. Unlike AIME, reference answers are not restricted to integers; the answer checker therefore accepts numerically or symbolically equivalent forms, including fractions, radicals, and other closed-form expressions.

LiveCodeBench. (Jain et al., 2024) We use a fixed January–April 2025 slice containing 182 programming problems and sample two responses per problem. From each response, we extract the last syntactically valid Python program enclosed in a fenced code block and execute it against the official test cases in an isolated filesystem jail. A sample is considered correct only if it passes all required tests within the benchmark resource limits.

TACO. (Li et al., 2023) We use the test split at revision `d593ed0a`. Starting from 400 easy- and medium-difficulty candidates, we remove 75 special-judge problems, 21 problems with image dependencies or empty test suites, and 21 problems overlapping with the training pool, leaving 283 problems (149 easy and 134 medium). We execute the model-generated Python programs in an isolated environment. Each problem uses at most 40 paired input–output tests, and a solution is considered correct only if its outputs exactly match the expected outputs on all applicable tests.

FrontierScience Olympiad-100. FrontierScience (Wang et al., 2026b) evaluates scientific reasoning across physics, chemistry, and biology. We use its 100-problem Olympiad gold set, consisting of 50 physics, 40 chemistry, and 10 biology problems, and exclude the Research track. We sample one response per problem and use DeepSeek-V4 Pro Preview with the benchmark's reference-answer grading prompt to determine whether each response matches or is equivalent to the reference answer. Each response receives a binary correctness label.

PHYRD-40. PHYRD-40 comprises 40 physics problems developed by scientists invited by our team, covering quantum field theory and symmetries, gravity and holography, string theory and brane dynamics, celestial scattering amplitudes, cosmological large-scale structure, and tensor-network methods. These problems require sustained analytical reasoning and detailed derivations. We sample one response per problem and use DeepSeek-V4 Pro Preview to evaluate it against the reference solution and problem-specific rubric. Each response receives a score from 0 to 100, and we report the arithmetic mean over 40 problems. We will publicly release the PHYRD-40 problem set, reference solutions, scoring rubrics, and evaluation pipeline.

A.6 Common Generation Protocol

Within each experimental setting, we apply the same decoding settings and sampling budget to the base Student, the Teacher, and all distilled models. Each problem is formatted with a benchmark-specific instruction and rendered using the evaluated model's native chat template with thinking enabled. We use temperature 0.6 for mathematics and code, and 1.0 for scientific reasoning; top-$p = 0.95$ and top-$k = 20$ are shared across all benchmarks.

Table 9 summarizes the generation settings. For AIME, HMMT, LiveCodeBench, and TACO, we use a context limit of $32{,}768$ tokens and an output limit of $27{,}648$ tokens, sampling eight responses per mathematics problem and two per code problem. For FrontierScience Olympiad and PHYRD-40, we use a context limit of $262{,}144$ tokens and an output limit of $256{,}000$ tokens, sampling one response per problem. Token limits are measured using each model's native tokenizer, and the available output budget is also constrained by the remaining context capacity. Within each benchmark, all compared models use the same problem order and grading procedure.

Table 9: Generation settings for the controlled and large-scale experiments. Context and output limits are measured in tokens; the context limit includes both the input prompt and generated response.

Evaluation setting	Context	Max output	Temp.	Top-p	Top-k	Samples/problem
AIME24/25/26	32,768	27,648	0.6	0.95	20	8
HMMT-26	32,768	27,648	0.6	0.95	20	8
LiveCodeBench	32,768	27,648	0.6	0.95	20	2
TACO	32,768	27,648	0.6	0.95	20	2
FrontierScience Olympiad	262,144	256,000	1.0	0.95	20	1
PHYRD-40	262,144	256,000	1.0	0.95	20	1

Mathematical reasoning. The prompt requests a step-by-step solution ending with an `Answer:` line. The grader extracts the last such answer or, if absent, the last `\boxed{}` expression, checking numeric and symbolic equivalence.

Mathematical reasoning prompt (used for training and evaluation)

<|im_start|>user
Solve the following math problem step by step. The last line of your response should be of the form Answer: $Answer (without quotes), where $Answer is the answer to the problem.

{problem}

Remember to put your answer on its own line after Answer:.
<|im_end|>

<|im_start|>assistant
<think>
{*model-generated reasoning begins here; no reference completion is stored*}

Code generation. The prompt requests a complete Python program in a fenced code block, following the provided function signature or otherwise using standard input and output.

Code generation prompt (used for training and evaluation)

<|im_start|>user
You are given a competitive programming problem. Think step by step, then provide a single complete Python program in a `python` code block. If starter code or a function signature is given, implement it; otherwise, read from standard input and write to standard output.

{problem}
<|im_end|>

<|im_start|>assistant
<think>
{*model-generated reasoning begins here; no reference completion is stored*}

FrontierScience Olympiad. The system message requests a rigorous, self-contained scientific solution with supporting reasoning. The user message supplies the subject and problem statement.

FrontierScience Olympiad prompt (used for evaluation)

<|im_start|>system
You are an expert scientist. Solve the problem rigorously and return a complete, self-contained answer. Show the reasoning needed to support the final answer.
<|im_end|>

<|im_start|>user
Frontier Science Problem

Subject: {subject}

Problem:

{problem}

Provide your complete solution and final answer directly in this response.
<|im_end|>

<|im_start|>assistant
<think>
{*model-generated reasoning begins here; no reference completion is stored*}

PHYRD-40. The system message requests a complete derivation with explicit conventions and consistency checks, without external tools or sources. The user message supplies the problem identifier and statement.

PHYRD-40 prompt (used for evaluation)

<|im_start|>system
You are an expert mathematical physicist solving a research-level problem. Produce a rigorous, self-contained solution using LaTeX notation. Address every numbered request, show the essential derivations, define conventions, and check signs, limits, or consistency where relevant. You have only the problem statement: do not assume or claim access to a reference answer. Do not use external tools, web sources, or unstated numerical experiments.
<|im_end|>

<|im_start|>user
Solve the following problem completely. Preserve the problem's conventions.

<problem_id>{problem_id}</problem_id>

<problem_latex>
{problem}
</problem_latex>
<|im_end|>

<|im_start|>assistant
<think>
{*model-generated reasoning begins here; no reference completion is stored*}

B Cross-Tokenizer Distillation Under a Unified View

This section presents a unified view of cross-tokenizer on-policy distillation. We first define shared and non-shared token events and quantify direct vocabulary correspondence; we then characterize representative cross-tokenizer objectives by what they supervise and where supervision is applied; finally, we introduce Event-Set Completion Distillation (ESCD) and its training algorithm.

This section is organized as follows:

1. **Basic definitions (B.1):** we define byte realizations, shared and non-shared tokens, and the root and visited child states used throughout this section.
2. **Direct vocabulary overlap (B.2):** we quantify how much of the Teacher and Student vocabularies can be matched through direct one-to-one correspondence.
3. **Existing methods under a unified view (B.3):** we compare representative methods by asking what event they supervise and at which Student state that supervision is applied.
4. **Event-Set Completion Distillation (B.4):** we define the residual completion set and the corresponding visited-child objective.
5. **A worked example of cross-tokenizer supervision (B.5):** we compare SimCT, BPM, and ESCD using the same teacher and student probabilities, illustrating their target construction, loss calculations, and treatment of residual completion.
6. **Training algorithm (B.6):** we summarize how ESCD augments the standard on-policy pipeline and highlight the additional operations introduced by our method.

B.1 Basic Definitions

On-policy predictions. Let T be a frozen Teacher and S_θ a trainable Student with vocabularies V_T and V_S. At a response prefix x visited by the Student rollout, they define next-token distributions

$$q_T(\cdot \mid x) \in \Delta(V_T), \qquad p_\theta(\cdot \mid x) \in \Delta(V_S). \tag{12}$$

When the two models use the same tokenizer, these distributions can be compared directly at Student-visited states. Under tokenizer mismatch, however, token IDs are tokenizer-local coordinates and therefore do not define cross-model correspondence.

Shared and non-shared token events. Tokenizer-local IDs cannot be compared across models. We therefore describe each token in two different ways, each serving a distinct purpose.

First, let

$$b_T : V_T \to \mathcal{B}^*, \qquad b_S : V_S \to \mathcal{B}^*, \tag{13}$$

where $b_T(v)$ and $b_S(u)$ are the exact byte strings produced by Teacher token v and Student token u, respectively. These byte realizations are used throughout ESCD to define prefix relations, residual events, and completion sets. Separately, we define tokenizer-independent matching keys

$$\kappa_T : V_T \to \mathcal{K}, \qquad \kappa_S : V_S \to \mathcal{K}. \tag{14}$$

The matching key is used only to identify tokens that can be transferred one-to-one across the two vocabularies. It removes a small number of explicitly specified tokenizer conventions, such as equivalent whitespace markers, and includes designated correspondences between semantically identical special tokens such as the primary EOS token. It is *not* used to define ESCD byte events.

Using these keys, we construct a deterministic one-to-one shared-token mapping

$$\mathcal{P}_{\text{sh}} = \text{Select}_{1:1}\left\{(v,u) \in V_T \times V_S : \kappa_T(v) = \kappa_S(u)\right\}. \tag{15}$$

A Teacher token is **shared** if it appears in this mapping, and **non-shared** otherwise:

$$V_{T,\mathrm{sh}} = \{v \in V_T : \exists u,\, (v,u) \in \mathcal{P}_{\mathrm{sh}}\}, \qquad V_{T,\mathrm{ns}} = V_T \setminus V_{T,\mathrm{sh}}. \tag{16}$$

Shared tokens admit direct coordinate remapping through the selected pairs. Non-shared tokens are those outside this mapping; this designation does not necessarily imply the absence of an exact single-token byte counterpart. ESCD therefore uses a separate byte-based criterion to select child-event candidates, retaining content tokens whose complete byte strings have no single-token Student counterpart, as defined in Section 3.2.

For example, Qwen3 and Qwen3.5 both contain the single-token string `['apple']`, although their local token IDs differ. These tokens form a shared pair. In contrast, Qwen3 token 382 realizes `['.\n\n']`, whereas Qwen3.5 realizes the same bytes with the two-token sequence `[13, 271]`. This example illustrates how the same bytes can be realized through different token boundaries.

Root and visited child states. Let x denote the full response prefix generated so far by the Student rollout, i.e., the context on which the next-token prediction is conditioned. We call x the *root state*, as it serves as the common starting point for the continuation paths considered below. At this state, consider a non-shared Teacher token v whose byte realization is

$$y = b_T(v). \tag{17}$$

The Student then samples its next token $a \sim p_\theta(\cdot \mid x)$. If the bytes produced by this actually sampled token form a strict prefix of the Teacher event,

$$b_S(a) \prec y, \tag{18}$$

then the rollout naturally reaches the next state after appending a to x. We denote this *visited child state* by (x, a). The part of the Teacher event that remains unfinished at this state is

$$r = y[|b_S(a)| :]. \tag{19}$$

For example, suppose the Teacher assigns probability to a non-shared event whose byte surface is `Bread`. At the current Student-generated prefix x, the Student actually emits the token `Br`. Then

$$\underbrace{x}_{\text{root state}} \xrightarrow{a=\texttt{Br}} \underbrace{(x, \texttt{Br})}_{\text{visited child state}}, \qquad \underbrace{y = \texttt{Bread}}_{\text{Teacher event}} = \underbrace{\texttt{Br}}_{b_S(a)} \,\|\, \underbrace{\texttt{ead}}_{r}. \tag{20}$$

Here, x is the entire Student-generated response prefix before the current action, rather than a single token. The variable a is the Student token actually selected at that state, and r is the remaining part of the Teacher byte event after removing the bytes already realized by a.

This distinction is central to ESCD. The Teacher event is initially defined at the root state x. Once the Student naturally takes a compatible action a, ESCD can continue supervising the remaining event r at the visited child state (x, a), without constructing a counterfactual Student branch.

For the remainder of this section, we compare methods along two questions: *how is a non-shared Teacher event represented for the Student, and at which Student state is it supervised?*

B.2 How Much of the Vocabulary Is Directly Shared?

Before considering non-shared events, we quantify how much direct one-to-one correspondence already exists between the two vocabularies. Let

$$M = |\mathcal{P}_{\mathrm{sh}}| \tag{21}$$

denote the number of selected shared pairs. We report Teacher-side coverage, Student-side coverage, and Jaccard overlap:

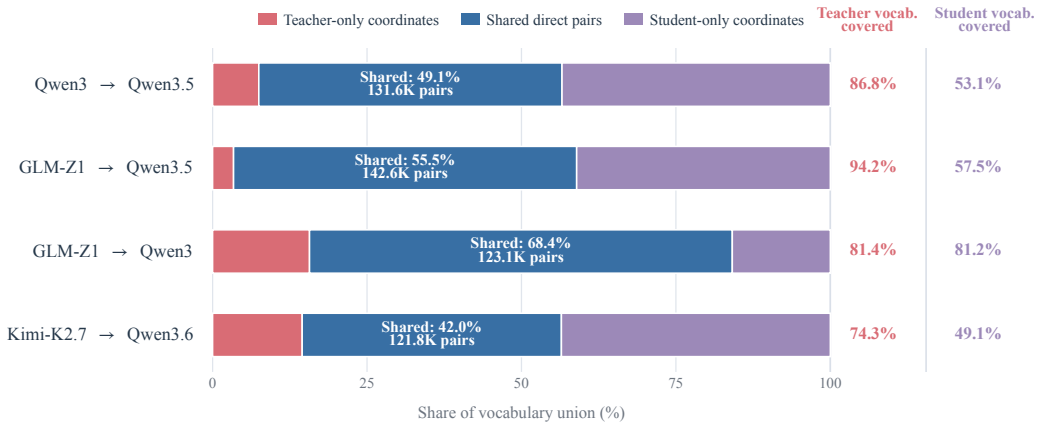


Figure 5: Static Teacher–Student vocabulary overlap. Bars show Teacher-only, shared, and Student-only shares of the vocabulary union; right-hand columns report directional coverage.

Table 10: Static shared-vocabulary overlap under the deterministic one-to-one canonical-surface mapping. Vocabulary sizes exclude checkpoint padding.

Tokenizer pair	$\lvert\mathbf{V_T}\rvert$	$\lvert\mathbf{V_S}\rvert$	**M**	$\mathbf{C_T}$	$\mathbf{C_S}$	**J**
Qwen3 → Qwen3.5	151,669	248,077	131,612	86.78%	53.05%	49.08%
Qwen3.5 → Qwen3	248,077	151,669	131,612	53.05%	86.78%	49.08%
GLM-Z1 → Qwen3.5	151,343	248,077	142,628	94.24%	57.49%	55.54%
GLM-Z1 → Qwen3	151,343	151,669	123,118	81.35%	81.18%	68.44%
Kimi-K2.7 → Qwen3.6	163,840	248,077	121,757	74.31%	49.08%	41.96%

$$C_T = \frac{M}{|V_T|}, \qquad C_S = \frac{M}{|V_S|}, \qquad J = \frac{M}{|V_T| + |V_S| - M}. \tag{22}$$

Here C_T measures the fraction of Teacher coordinates that can be transferred directly, C_S gives the corresponding Student-side coverage, and J measures overlap relative to the vocabulary union.

Table 10 shows that shared coordinates cover 53.05–94.24% of the Teacher vocabulary across evaluated directions. Direct remapping handles these coordinates, while our method targets the substantial non-shared subset requiring explicit cross-tokenizer treatment.

B.3 Existing Methods Under the Unified View

We compare representative methods by their cross-tokenizer alignment mechanisms and supervision objects, including ranked probability profiles, mapped vocabulary coordinates, and aligned continuation units. This distinguishes how methods construct comparable predictions from how they transfer supervision. Because position- and span-alignment conventions differ, prediction vectors need not share an identical conditioning prefix. The formulations below summarize the relevant distillation components and their relation to ESCD's completion-set supervision.

ULD: rank-based distribution matching. ULD compares probability profiles without constructing token-identity correspondences (Boizard et al., 2025). Let q_i and p_i denote the Teacher and Student probability vectors at the positions being compared. After padding the smaller vocabulary distribution with zeros to length $K = \max(|V_T|, |V_S|)$, define

$$q_i^{\downarrow}(k) = k\text{-th largest entry of the padded } q_i, \qquad p_i^{\downarrow}(k) = k\text{-th largest entry of the padded } p_i. \tag{23}$$

The rank-matching term is

$$\ell_{\mathrm{ULD}}(i) = \sum_{k=1}^{K} \left| q_i^{\downarrow}(k) - p_i^{\downarrow}(k) \right|. \tag{24}$$

Rank k identifies a probability magnitude rather than a shared token, so this term transfers distributional shape without preserving textual correspondence between coordinates. The original training objective also includes cross-entropy supervision and compares sequence positions up to the shorter tokenized length. Rank matching itself does not resolve mismatched text boundaries; applying it after an additional alignment procedure is a separate implementation choice.

GOLD: span alignment and hybrid matching. GOLD combines cross-tokenizer sequence alignment and probability merging with configurable distribution-matching losses. In its hybrid formulation, directly matched coordinates retain token identity, while unmatched coordinates are compared through sorted probability profiles. Let $\widehat{q}_T^{(k)}$ and $\widehat{p}_\theta^{(k)}$ denote the Teacher and Student probability representations associated with aligned unit k after merging. A schematic hybrid loss is

$$\ell_{\mathrm{GOLD}}(k) = \lambda_{\mathrm{sh}} D_{\mathrm{sh}} \left(\widehat{q}_{T,\mathrm{sh}}^{(k)}, \widehat{p}_{\theta,\mathrm{sh}}^{(k)} \right) + \lambda_{\mathrm{ns}} \left\| \mathrm{sort}_{\downarrow} \widehat{q}_{T,\mathrm{ns}}^{(k)} - \mathrm{sort}_{\downarrow} \widehat{p}_{\theta,\mathrm{ns}}^{(k)} \right\|_1, \tag{25}$$

where $\mathrm{sort}_{\downarrow}$ sorts entries in descending order, $\|\mathbf{z}\|_1 = \sum_j |z_j|$ denotes the L_1 norm, and unmatched vectors are zero-padded to equal dimension before comparison. Here, D_{sh} denotes the configured shared-coordinate discrepancy, and λ_{sh} and λ_{ns} weight the two terms. The shared term preserves explicit coordinate correspondences, whereas the unmatched term compares probabilities by rank rather than token identity. Sequence merging can incorporate conditional probabilities from multiple token positions, so GOLD operates on aligned, merged representations rather than necessarily comparing raw next-token distributions at a common boundary. Its hybrid matching objective is distinct from an auxiliary loss on the valid completions of a specified residual byte event.

X-Token: aligned-chunk vocabulary projection. X-Token combines span alignment, chain-rule probability merging, and vocabulary projection (Sreenivas et al., 2026). Aligned spans provide text-consistent comparison units, while a sparse matrix $W \in \mathbb{R}^{|V_S| \times |V_T|}$ maps Student coordinates into Teacher vocabulary space. The mapping is initialized from canonicalized token matches and multi-token decompositions, with row normalization; it can optionally be refined during training.

For the P-KL formulation, let $\widehat{p}_\theta^{(k)}$ and $\widehat{q}_T^{(k)}$ denote the aligned chunk distributions. The projected Student distribution is

$$\widetilde{p}_{\theta,W}^{(k)}(v) = \sum_{u \in V_S} W_{u,v} \widehat{p}_\theta^{(k)}(u), \qquad v \in V_T. \tag{26}$$

The corresponding distillation term is

$$\ell_{\text{XToken-P}}(k) = \mathrm{KL}\left(\widehat{q}_T^{(k)} \,\middle\|\, \widetilde{p}_{\theta,W}^{(k)} \right). \tag{27}$$

X-Token also introduces H-KL, which uses high-confidence mappings to expand the matched set within a hybrid objective. Thus, the projection above describes P-KL rather than every X-Token variant. Although the initialization of W depends on tokenizer structure, the complete method also uses sequence-level alignment and merging. Its central operation is distribution matching over aligned chunks, rather than constructing a completion set for an unfinished Teacher event.

SimCT: scoring minimal aligned continuation units. SimCT constructs a common supervision space of shared tokens and minimal aligned units realizable by both tokenizers (Sun et al., 2026). These units may have different token counts in each model. For $M \in \{T, S\}$, let $\tau_M(c) = (v_1, \ldots, v_{L_M(c)})$ denote the tokenization of candidate unit c. SimCT computes length-normalized continuation scores and normalizes them over the candidate space $\mathcal{U}(x)$:

$$s_M(c \mid x) = \frac{1}{L_M(c)} \sum_{j=1}^{L_M(c)} \log p_M(v_j \mid x, v_{<j}),$$
$$\pi_M(c \mid x) = \frac{\exp s_M(c \mid x)}{\sum_{c' \in \mathcal{U}(x)} \exp s_M(c' \mid x)}. \tag{28}$$

Here p_M denotes the native autoregressive probabilities of either model. An OPD divergence is then applied to the induced distributions:

$$\mathcal{L}_{\mathrm{SimCT}}(x) = D_{\mathrm{OPD}}\left(\pi_S(\cdot \mid x), \pi_T(\cdot \mid x)\right). \tag{29}$$

These are normalized continuation-score distributions, not mass-preserving marginals of the original next-token distributions. Candidate continuations are scored through autoregressive factors from the conditioning prefix; their intermediate states need not all be visited by the actual rollout.

BPM: Teacher-induced byte-prefix targets. BPM maps Teacher probability mass into Student-token targets through byte-prefix relations (Wang et al., 2026a). In the basic refinement case, define the longest compatible Student-token prefix of Teacher content token v:

$$\phi(v) = \operatorname*{arg\,max}_{u \in V_S:\, b_S(u) \preceq b_T(v)} |b_S(u)|, \tag{30}$$

with $\phi(v) = \perp$ when no eligible prefix exists. At aligned position i, the target aggregates Teacher probabilities:

$$t_i(u) = \sum_{v:\phi(v)=u} q_i(v), \qquad t_i(\varnothing) = \sum_{v:\phi(v)=\perp} q_i(v). \tag{31}$$

BPM also supervises positions inside a Teacher token. If c denotes the bytes already emitted since the Teacher boundary, let $\phi_c(v)$ map the remaining bytes to their longest Student-token prefix. For $\pi_i(c) = \sum_{v:\, c \preceq b_T(v)} q_i(v) > 0$, the conditional target is

$$t_i(u \mid c) = \frac{\sum_{v:\, c \preceq b_T(v),\, \phi_c(v)=u} q_i(v)}{\pi_i(c)}, \qquad t_i(\varnothing \mid c) = 1 - \sum_{u \in V_S} t_i(u \mid c). \tag{32}$$

The original method additionally handles Student tokens spanning Teacher boundaries and model-specific stopping events. Its targets are matched to Student probabilities through a distributional distillation loss. BPM therefore cannot be characterized as root-only supervision or as summing probabilities over all Student tokenization paths. The relevant distinction is that ESCD adds a Teacher-mass-weighted loss on the aggregate probability of valid residual completions, without assigning an individual target probability to each completion.

Relation to ESCD. These methods establish supervision through ranked coordinates, matched tokens, aligned spans, or Teacher-induced token targets. ESCD introduces a complementary supervision unit: the set of native Student actions that complete a residual byte constraint at a naturally visited child state. Existing distribution-matching objectives can also constrain aggregate probabilities indirectly; ESCD makes this particular completion event explicit through an auxiliary set-probability loss. Its distinction therefore concerns the construction and granularity of the completion target, rather than the presence of supervision at later Student positions alone.

B.4 ESCD: Event-Set Completion Distillation

From event entry to event completion. A Student action may realize only a prefix of a Teacher byte event, leaving a residual constraint at the next visited state. Existing cross-tokenizer methods can supervise intermediate Student positions; ESCD instead makes the residual completion set an

Table 11: Cross-tokenizer supervision mechanisms. Entries summarize distillation components, not full training recipes. ESCD explicitly supervises residual completion sets.

Method	Alignment or mapping mechanism	Supervision form
ULD	Independent probability sorting	Rank-matched probability profiles
GOLD	Span alignment and probability merging	Shared-coordinate and unmatched-profile matching
X-Token	Span alignment and vocabulary projection	Projected or hybrid chunk-level distribution matching
SimCT	Minimal aligned continuation units	Normalized continuation-score distribution matching
BPM	Byte-prefix mapping and conditional targets	Teacher-induced Student-token distribution matching
ESCD	**Prefix aggregation and visited residual construction**	**Auxiliary loss on completion-set probability**

explicit supervision object. It combines BPM-style root projection with a loss on the aggregate probability of valid completions at visited child states.

Consider an illustrative Teacher token with byte realization `Bread`, which the Student can realize through `Br` followed by `ead`:

$$\underbrace{\texttt{Bread}}_{\text{Teacher event}} \quad \Longleftrightarrow \quad \underbrace{\texttt{Br}}_{\text{Student action } a} \;\|\; \underbrace{\texttt{ead}}_{\text{residual bytes}}, \tag{33}$$

where $\|$ denotes byte-string concatenation. After sampling $a = \texttt{Br}$ at state x, the Student reaches

$$x \xrightarrow{\texttt{Br}} x' = (x, \texttt{Br}), \qquad \texttt{Bread} = \texttt{Br} \,\|\, \texttt{ead}. \tag{34}$$

At x', ESCD supervises the total probability of next-token actions that complete the remaining bytes, potentially extending beyond the event boundary. It neither selects a canonical completion nor assigns individual target probabilities to valid actions.

Root projection loss. ESCD reuses BPM's byte alignment and root projection, but not its full interior- and spanning-position supervision. At an eligible aligned root x, let $\mathcal{K}_T(x)$ denote the selected Teacher candidates, $U(x) \subseteq V_S$ the explicit Student-token coordinates, and ϕ_x the root byte mapping, with $\phi_x(v) = \perp$ for candidates not assigned to an explicit coordinate. The mapped Teacher mass is

$$q_{\text{map}}(u \mid x) = \sum_{\substack{v \in \mathcal{K}_T(x) \\ \phi_x(v) = u}} q_T(v \mid x), \qquad u \in U(x). \tag{35}$$

We construct Teacher and Student distributions over the same coordinates $U(x) \cup \{\perp\}$:

$$\begin{aligned} \bar{q}_x(u) &= q_{\text{map}}(u \mid x), & \bar{p}_{\theta,x}(u) &= p_\theta(u \mid x), \quad u \in U(x), \\ \bar{q}_x(\perp) &= 1 - \sum_{u \in U(x)} q_{\text{map}}(u \mid x), & \bar{p}_{\theta,x}(\perp) &= 1 - \sum_{u \in U(x)} p_\theta(u \mid x). \end{aligned} \tag{36}$$

The complement $\perp$ collects mass outside the explicit coordinates and is not a generatable token. Teacher probabilities retain their original values after candidate selection; omitted and unmapped mass remains in the complement. We apply forward KL:

$$\ell_{\text{root}}(x) = \text{KL}(\bar{q}_x \,\|\, \bar{p}_{\theta,x}). \tag{37}$$

Teacher-side prefix aggregation. Child candidates are selected by exact byte realizability. Let V_T^{cont} and V_S^{cont} denote the Teacher and Student content-token vocabularies. We retain Teacher candidates whose complete byte strings have no single-token Student counterpart:

$$\mathcal{V}(x) = \left\{ v \in \mathcal{K}_T(x) \cap V_T^{\text{cont}} : \nexists u \in V_S^{\text{cont}},\ b_T(v) = b_S(u) \right\}. \tag{38}$$

This eligibility rule excludes special tokens and exact single-token byte matches. It is distinct from merely excluding tokens selected by the one-to-one shared-coordinate mapping.

Distinct Teacher tokens are mutually exclusive outcomes, but their byte-prefix constraints may be nested. For example, satisfying the prefix `researches` also satisfies `resear`. ESCD aggregates such constraints into a representative prefix. We connect $v, w \in \mathcal{V}(x)$ whenever $b_T(v) \preceq b_T(w)$ or $b_T(w) \preceq b_T(v)$ and use the connected components as aggregation groups. For group g with members $\mathcal{V}_g$, define

$$M_g = \sum_{v \in \mathcal{V}_g} q_T(v \mid x). \tag{39}$$

The representative is a shortest byte realization in the component:

$$v_g^\star \in \arg\min_{v \in \mathcal{V}_g} |b_T(v)|, \qquad y_g = b_T(v_g^\star), \tag{40}$$

with deterministic tie breaking. The representative y_g prefixes every member, and representatives of distinct groups are prefix-incomparable. Aggregation preserves the included Teacher mass while retaining only the shared representative constraint, not each longer member's full byte requirement.

Completion sets at visited Student states. Suppose the Student samples a content token a whose bytes form a nonempty strict prefix of y_g:

$$0 < |b_S(a)| < |y_g|, \qquad b_S(a) \prec y_g. \tag{41}$$

At the visited child state $x' = (x, a)$, the residual is

$$r_g = y_g[|b_S(a)| :]. \tag{42}$$

The valid one-step completion set is

$$\mathcal{C}(r_g) = \left\{ u \in V_S^{\text{cont}} : r_g \preceq b_S(u) \right\}. \tag{43}$$

A valid token begins with the entire residual and may extend beyond its endpoint. For $r_g =$ `Memory`, both `Memory` and `MemoryWarning` qualify if present in the Student vocabulary. The latter satisfies the residual constraint without implying Teacher supervision on the additional suffix `Warning`. The completion probability sums native full-vocabulary probabilities:

$$P_\theta(\mathcal{C}(r_g) \mid x, a) = \sum_{u \in \mathcal{C}(r_g)} p_\theta(u \mid x, a). \tag{44}$$

All probabilities are evaluated at the same visited child state, without renormalization over content tokens or the completion set. No alternative continuations are sampled, and the actual next token need not belong to $\mathcal{C}(r_g)$ for the loss to apply.

Local child multiplicity. For a Teacher group g, suppose the sampled Student action a satisfies $b_S(a) \prec y_g$, leaving part of the representative event unfinished. To distinguish Teacher-side aggregation from Student-side completion, define

$$m_g = |V_g|, \qquad n_g(a) = |\mathcal{C}(r_g)|, \qquad r_g = y_g[|b_S(a)| :]. \tag{45}$$

Here, m_g counts Teacher candidates in the group, while $n_g(a)$ counts Student tokens that complete the residual in one additional step. The latter depends on the sampled action and may be zero; only groups with $n_g(a) > 0$ contribute to the child loss. These quantities describe local supervision structure, not token counts in an aligned span. In particular, $n_g(a)$ differs from the root-compatible token count b_g used in Table 15.

The following examples illustrate different combinations of Teacher group size and Student completion count, assuming eligible Teacher candidates and exactly the listed completion sets.

- **Single-member group, single completion ($m_g = 1$, $n_g(a) = 1$).** For $y_g = \texttt{Bread}$ and $a = \texttt{Br}$, the residual is $\texttt{ead}$. If V_g contains one Teacher token and $\mathcal{C}(r_g) = \{\texttt{ead}\}$, the child loss reduces to $-M_g \log p_\theta(\texttt{ead} \mid x, a)$.
- **Single-member group, multiple completions ($m_g = 1$, $n_g(a) > 1$).** For $y_g = \texttt{(Memory}$ and $a = \texttt{(}$, the residual is $\texttt{Memory}$. Suppose V_g contains one Teacher token and $\mathcal{C}(r_g) = \{\texttt{Memory}, \texttt{MemoryWarning}\}$. The child loss is

$$-M_g \log \left[p_\theta(\texttt{Memory} \mid x, a) + p_\theta(\texttt{MemoryWarning} \mid x, a)\right]. \tag{46}$$

- **Multi-member group, single completion ($m_g > 1$, $n_g(a) = 1$).** Suppose a group consists of $\texttt{[space]resear}$ and $\texttt{[space]researches}$, where $\texttt{[space]}$ denotes a literal leading space. Its mass is

$$M_g = q_T(\texttt{[space]resear} \mid x) + q_T(\texttt{[space]researches} \mid x). \tag{47}$$

  After the Student emits the space, the representative residual is $\texttt{resear}$. If $\mathcal{C}(r_g) = \{\texttt{research}\}$, then $m_g = 2$ and $n_g(a) = 1$. This completion satisfies the representative constraint but leaves the longer event $\texttt{researches}$ unfinished, illustrating the coarsening introduced by aggregation.
- **Multi-member group, multiple completions ($m_g > 1$, $n_g(a) > 1$).** Suppose a group consists of $\texttt{VEH}$ and $\texttt{VEHICLE}$, with representative $\texttt{VEH}$. After $a = \texttt{V}$, the residual is $\texttt{EH}$. If $\mathcal{C}(r_g) = \{\texttt{EH}, \texttt{EHICLE}\}$, then $m_g = 2$ and $n_g(a) = 2$: Teacher mass is aggregated across group members, and Student probability is summed across valid completions.

Child objective. Let $\mathcal{G}(x, a)$ contain groups for which the sampled action a partially realizes the representative event, as specified in Equation 41, and the residual admits at least one valid one-step completion ($n_g(a) > 0$). At the visited child state, the loss is

$$\ell_{\text{child}}(x, a) = -\sum_{g \in \mathcal{G}(x,a)} M_g \log P_\theta(\mathcal{C}(r_g) \mid x, a). \tag{48}$$

The weights M_g retain their root Teacher mass without renormalization over eligible groups. They weight residual constraints rather than define a Teacher posterior conditioned on a. No additional Teacher query is required. The loss is zero when $\mathcal{G}(x, a)$ is empty.

Combined objective. Root terms are assigned to their aligned prediction positions, and child terms to the immediately following visited prediction positions. Their per-position losses are added before applying the training mask and reduction. Writing $\mathcal{L}_{\text{root}}$ and $\mathcal{L}_{\text{child}}$ for the two contributions under this common reduction, the full objective is

$$\mathcal{L}_{\text{ESCD}} = \mathcal{L}_{\text{root}} + \mathcal{L}_{\text{child}}. \tag{49}$$

Both terms use the same reduction denominator, with zero contributions at ineligible positions; they are not separately averaged over their eligible positions. ESCD thus combines root projection and child completion rather than adding the child term to the complete BPM objective.

Why supervise a completion set? Selecting one valid completion introduces a preference within the set. For any $u^\star \in \mathcal{C}(r_g)$, the single-completion loss decomposes as

$$\begin{aligned} -M_g \log p_\theta(u^\star \mid x, a) = &-M_g \log P_\theta(\mathcal{C}(r_g) \mid x, a) \\ &- M_g \log \frac{p_\theta(u^\star \mid x, a)}{P_\theta(\mathcal{C}(r_g) \mid x, a)}. \end{aligned} \tag{50}$$

The first term encourages residual completion, while the second favors the selected token within the completion set. ESCD uses only the first term, without specifying a within-set target distribution. The objectives coincide when $n_g(a) = 1$; when $n_g(a) > 1$, the single-completion loss introduces an additional preference not specified by the residual constraint.

Scope. ESCD supervises a representative residual byte constraint, not the full byte requirement of every original Teacher token. Its current child objective covers completion by one additional Student token; residuals without a valid one-step completion contribute no child loss. Completion probabilities reuse Student logits at visited states, without canonical continuations or counterfactual rollouts. Teacher candidates may be selected from the full vocabulary or a sparse subset. With sparse selection, groups, representatives, and weights are constructed from the retained candidates using their original probabilities. Candidate restriction can therefore change the supervision itself, not merely its computational cost.

B.5 A Worked Example of Cross-Tokenizer Supervision

The preceding sections characterize the supervision mechanisms of SimCT, BPM, and ESCD. To make their differences concrete, we work through a numerical example in which teacher tokens are realized by multiple student tokens. Using the same hypothetical predictions, we trace how each method constructs its target and computes its loss, clarifying how subsequent student actions enter the objective and whether individual completions are distinguished. All probabilities are illustrative, logarithms are natural, and losses are reported before training-mask reduction.

A natural question is: if `Bread` receives the highest teacher probability, why do BPM and ESCD still retain other teacher candidates such as `Breads` and `Break`? Does including lower-probability candidates introduce undesirable supervision? The answer is no, because these candidates are alternative next-token predictions from the teacher distribution, rather than correctness labels. Their probabilities represent the teacher's preference over possible next-token events under the same context. After the student samples `Br`, BPM and ESCD preserve this teacher information at different granularities. BPM maintains token-level distinctions and assigns separate targets to the residual tokens `ead`, `eads`, and `eak`. ESCD instead aggregates teacher mass over residual byte constraints: `Bread` and `Breads` are grouped because they share the same representative prefix constraint, while `Break` remains a separate event with its own teacher mass. Therefore, ESCD does not treat `Breads` as equivalent to `Bread`; it only removes the need to allocate probability among tokenizer-dependent realizations that satisfy the same residual constraint.

Table 12 gives the teacher distribution at root context x. For simplicity, the four candidates carry all teacher probability mass. Assume the student has no single-token representation of `Bread`, `Breads`, or `Break`, and `Br` is their longest student-token prefix. The stated student tokenizations are assumed throughout; `Cat` is shared directly.

Table 12: Illustrative teacher predictions and student realizations. Residuals are shown for individual teacher tokens after `Br`, before ESCD aggregation; `Cat` is incompatible with this action.

Teacher token	$q_T(v \mid x)$	Student tokenization	Residual after `Br`
`Bread`	0.4	`Br`, `ead`	`ead`
`Breads`	0.2	`Br`, `eads`	`eads`
`Break`	0.3	`Br`, `eak`	`eak`
`Cat`	0.1	`Cat`	—

At the root, let

$$p_\theta(\texttt{Br} \mid x) = 0.5, \qquad p_\theta(\texttt{Cat} \mid x) = 0.2, \tag{51}$$

with probability 0.3 assigned to all remaining tokens. Suppose the student actually samples $a = \texttt{Br}$ and reaches $x' = (x, a)$. Its next-token distribution at this visited child is

$$\begin{aligned} &p_\theta(\texttt{ead} \mid x') = 0.2, && p_\theta(\texttt{eads} \mid x') = 0.3, \\ &p_\theta(\texttt{eak} \mid x') = 0.1, && \sum_{u \notin \{\texttt{ead},\texttt{eads},\texttt{eak}\}} p_\theta(u \mid x') = 0.4. \end{aligned} \tag{52}$$

These are native full-vocabulary probabilities, without renormalization over the displayed tokens.

SimCT compares distributions over aligned continuation units. Here, the candidate space is

$$\mathcal{U} = \{\texttt{Bread}, \texttt{Breads}, \texttt{Break}, \texttt{Cat}\}. \tag{53}$$

Each teacher realization contains one token, whereas the first three student realizations contain two. SimCT averages the log-probabilities along each realization. For `Bread`,

$$\begin{aligned} s_T(\texttt{Bread} \mid x) &= \log 0.4, \\ s_S(\texttt{Bread} \mid x) &= \frac{1}{2}\left[\log 0.5 + \log 0.2\right]. \end{aligned} \tag{54}$$

The path probability is $0.5 \times 0.2 = 0.1$, but the exponentiated, length-normalized score is $\sqrt{0.1}$.

In the listed order, the exponentiated scores are

$$\begin{aligned} w_T &= (0.4, 0.2, 0.3, 0.1), \\ w_S &= (\sqrt{0.1}, \sqrt{0.15}, \sqrt{0.05}, 0.2). \end{aligned} \tag{55}$$

The scores sum to 1 for the teacher and approximately 1.1271 for the student. Normalizing each vector over $\mathcal{U}$ yields

$$\begin{aligned} \pi_T &= (0.4, 0.2, 0.3, 0.1), \\ \pi_S &\approx (0.2806, 0.3436, 0.1984, 0.1774). \end{aligned} \tag{56}$$

The reverse-KL formulation then gives

$$\mathcal{L}_{\mathrm{SimCT}} = \mathrm{KL}(\pi_S \| \pi_T) \approx 0.1062. \tag{57}$$

SimCT supervises subsequent predictions through complete-unit scores, keeping `Bread` and `Breads` as separate coordinates. However, candidate normalization can obscure changes in native completion probability. To see this, keep the teacher distribution and $\mathcal{U}$ fixed and halve the displayed student probabilities at both x and x', reallocating the removed mass to other tokens. The root probabilities of `Br` and `Cat` become $(0.25, 0.1)$, and the child probabilities of `ead`, `eads`, and `eak` become $(0.1, 0.15, 0.05)$. Every exponentiated unit score is then halved, leaving the normalized distribution and local SimCT loss unchanged. Yet the probability of completing `ead` after `Br` falls from 0.5 to 0.25. Thus, this objective on the fixed candidate space does not uniquely determine completion probability at x'.

BPM constructs teacher-induced student-token distributions. We use its basic refinement targets with forward KL; spanning-token corrections and stopping cases are outside this example. At the root, the longest-prefix map sends `Bread`, `Breads`, and `Break` to `Br`, giving

$$t_x(\texttt{Br}) = 0.4 + 0.2 + 0.3 = 0.9, \qquad t_x(\texttt{Cat}) = 0.1. \tag{58}$$

Over coordinates $(\texttt{Br}, \texttt{Cat}, \perp)$, where $\perp$ collects the remaining mass, the projected distributions and root loss are

$$t_x = (0.9, 0.1, 0), \qquad \bar{p}_{\theta,x} = (0.5, 0.2, 0.3). \tag{59}$$

$$\ell_{\mathrm{BPM,root}} = \mathrm{KL}(t_x \| \bar{p}_{\theta,x}) = 0.9 \log \frac{0.9}{0.5} + 0.1 \log \frac{0.1}{0.2} \approx 0.4597. \tag{60}$$

The complement is an aggregate coordinate, not a generatable token; neither distribution is renormalized over only `Br` and `Cat`.

After `Br`, the compatible teacher mass is $0.4 + 0.2 + 0.3 = 0.9$. BPM conditions these candidates on the emitted prefix and maps each residual to its longest student-token prefix:

$$\begin{aligned} \texttt{Bread} &\longmapsto \texttt{ead}, & t_{x'}(\texttt{ead}) &= 0.4/0.9 = 4/9, \\ \texttt{Breads} &\longmapsto \texttt{eads}, & t_{x'}(\texttt{eads}) &= 0.2/0.9 = 2/9, \\ \texttt{Break} &\longmapsto \texttt{eak}, & t_{x'}(\texttt{eak}) &= 0.3/0.9 = 3/9. \end{aligned} \tag{61}$$

Here $t_{x'}$ denotes the conditional target derived from the root teacher prediction, rather than a new teacher query at x'. The incompatible `Cat` candidate is excluded. The denominator 0.9 is teacher mass, not the student's action probability 0.5.

In the coordinate order $(\texttt{ead}, \texttt{eads}, \texttt{eak}, \perp)$, the distributions are $t_{x'} = (4/9, 2/9, 3/9, 0)$ and $\bar{p}_{\theta,x'} = (0.2, 0.3, 0.1, 0.4)$. Hence

$$\ell_{\mathrm{BPM,interior}} = \frac{4}{9} \log \frac{4/9}{0.2} + \frac{2}{9} \log \frac{2/9}{0.3} + \frac{3}{9} \log \frac{3/9}{0.1} \approx 0.6895. \tag{62}$$

The root and interior losses apply at distinct prediction positions and sum to approximately 1.1492 before the common response-token reduction. They are not averaged separately by position type.

For the representative residual `ead` used by ESCD below, both `ead` and `eads` are valid one-step completions. They remain distinct full continuations, however, and BPM preserves the teacher's $2{:}1$ preference. Keeping the root predictions fixed, changing their child probabilities from $(0.2, 0.3)$ to $(0.1, 0.5)$, with `eak` fixed at 0.1 and the remaining mass adjusted, increases their total completion probability from 0.5 to 0.6. Nevertheless, the interior loss rises from 0.6895 to 0.8841 because the allocation further deviates from the teacher's preference. Thus, increasing completion of the representative constraint need not improve matching to the teacher's separate token targets.

ESCD retains the same root projection here and directly supervises native completion-set probabilities at the visited child. Its child objective weights residual completion by teacher source mass without prescribing allocation among valid completions. The construction proceeds as follows.

First, exclude teacher content tokens with an exact single-token student counterpart. This removes `Cat` from child-event construction while retaining its root supervision. Grouping the remaining candidates by byte-prefix comparability gives

$$\begin{aligned} V_1 &= \{\texttt{Bread}, \texttt{Breads}\}, \quad y_1 = \texttt{Bread}, \quad M_1 = 0.4 + 0.2 = 0.6, \\ V_2 &= \{\texttt{Break}\}, \qquad\qquad\ y_2 = \texttt{Break}, \quad M_2 = 0.3. \end{aligned} \tag{63}$$

The representative is the shortest member of each group. Although `Bread` and `Break` share `Br`, neither complete string is a prefix of the other, so they remain separate. Sharing a compatible first action is insufficient for aggregation.

Second, check whether the sampled action partially realizes each representative. Since `Br` is a strict prefix of both y_1 and y_2, the groups leave different residuals at the same visited child $x' = (x, \texttt{Br})$:

$$r_1 = y_1[|\texttt{Br}| :] = \texttt{ead}, \qquad r_2 = y_2[|\texttt{Br}| :] = \texttt{eak}. \tag{64}$$

The first residual is defined by the representative `Bread`; `eads` is not retained as a separate teacher residual for that group.

Third, collect every student content token whose bytes begin with the entire residual. Assume the complete sets in this toy vocabulary are

$$\begin{aligned} \mathcal{C}(r_1) &= \{\texttt{ead}, \texttt{eads}\}, \\ \mathcal{C}(r_2) &= \{\texttt{eak}\}. \end{aligned} \tag{65}$$

Both `ead` and `eads` complete r_1: the former ends at the representative boundary, while the latter extends beyond it. Membership depends on student token bytes, without requiring a one-to-one correspondence with teacher group members.

Using the existing logits at x' gives

$$\begin{aligned} P_1 &= P_\theta(\mathcal{C}(r_1) \mid x') = 0.2 + 0.3 = 0.5, \\ P_2 &= P_\theta(\mathcal{C}(r_2) \mid x') = 0.1. \end{aligned} \tag{66}$$

These probabilities are not renormalized over either completion set or their union; the remaining probability 0.4 stays outside both events. Completion candidates require no separate sampling, and the next sampled token need not belong to either set for the child loss to apply.

Finally, weight each completion event's negative log-probability by its root teacher mass:

$$\begin{aligned} \ell_{\text{child}} &= -0.6 \log[p_\theta(\texttt{ead} \mid x') + p_\theta(\texttt{eads} \mid x')] \\ &\quad - 0.3 \log p_\theta(\texttt{eak} \mid x') \\ &= -0.6 \log 0.5 - 0.3 \log 0.1 \approx 1.1067. \end{aligned} \tag{67}$$

The weights remain 0.6 and 0.3, without normalization by the compatible teacher mass 0.9 or division by the student's action probability 0.5. ESCD combines this child loss with the root projection loss, giving $0.4597 + 1.1067 \approx 1.5664$ before common training reduction.

In the probability-halving example used for SimCT, the completion probabilities fall from $(P_1, P_2) = (0.5, 0.1)$ to $(0.25, 0.05)$. With teacher weights unchanged, the ESCD child loss increases from 1.1067 to 1.7305, while the local SimCT loss remains unchanged. Because ESCD

uses full-vocabulary probabilities without candidate-set renormalization, the decrease in completion probability directly increases its child loss.

In the redistribution example used for BPM, changing the probabilities of `ead` and `eads` from $(0.2, 0.3)$ to $(0.1, 0.5)$ raises P_1 from 0.5 to 0.6, with $P_2 = 0.1$ unchanged. The ESCD child loss decreases from 1.1067 to 0.9973, although BPM's loss at x' increases. ESCD therefore assigns a lower loss to improved completion of the representative constraint, even though the allocation moves further from the teacher's token-level preference. Any redistribution preserving P_1 and P_2 leaves its child loss unchanged. This flexibility comes from coarsening the supervision: the mass of `Breads` supports completion of the representative `Bread`, without separately requiring its final `s`. The completion set groups actions satisfying the same residual constraint, which need not represent equivalent full continuations. ESCD remains restricted to one additional-token completion, with no child loss when the set is empty.

B.6 Training Algorithm

Algorithm 1 summarizes ESCD within the standard on-policy distillation pipeline. ESCD reuses the Student rollout, Student training forward pass, and aligned Teacher outputs, and adds completion supervision only at child states already visited by the rollout. The ESCD-specific computation consists only of sparse event operations on quantities already available from root-level distillation. Teacher events are aggregated by prefix relation, matched against the Student action actually taken, and converted into one-step completion sets at the corresponding visited child state. The resulting loss accesses only the selected Student log-probabilities, avoiding both a dense cross-vocabulary child target and any additional model execution.

Algorithm 1 Event-Set Completion Distillation on one student response

Require: Student rollout $A = (a_1, \ldots, a_L)$, aligned teacher hidden states H_T, student logits Z_S, tokenizer byte artifact $\mathcal{A}$
Require: Teacher candidate selector SelectTeacherEvents, $\lambda_{\text{ESCD}} = 1$
Ensure: Response-aligned loss vector $\ell_{1:L}$

$\ell_{1:L} \leftarrow 0$; $L_S \leftarrow \text{LogSoftmax}(Z_S)$
$R \leftarrow \text{AlignedRootRows}(A, \mathcal{A})$
[Root distillation]
for all $r \in R$ **do**
  $(v_r, q_r) \leftarrow \text{SelectTeacherEvents}(H_T[r])$ ▷ full vocabulary in our experiments
  $Q_{\text{root}}[r] \leftarrow \text{CompileRootTarget}(v_r, q_r, \mathcal{A})$
  $\ell[r] \leftarrow \text{RootDistillationLoss}(Q_{\text{root}}[r], L_S[r])$
end for
[ESCD] Visited-child completion
$I_S \leftarrow \text{BuildOrLoadBytePrefixIndex}(\mathcal{A})$
$(\widetilde{v}, \widetilde{q}) \leftarrow \text{GatherCandidateEventsAcrossCPRanks}(v, q)$
for all $r \in R$ with a valid next prediction row in the same response **do**
  $a \leftarrow A[r]$; $c \leftarrow r + 1$ ▷ logical response coordinates
  $\mathcal{G}_r \leftarrow \text{PrefixQuotient}\big(\text{EligibleNonSharedEvents}(\widetilde{v}[r], \widetilde{q}[r], \mathcal{A})\big)$
  for all $(y_g, M_g) \in \mathcal{G}_r$ such that $b_S(a) \prec y_g$ **do**
    $C_g \leftarrow I_S.\text{CompletionTokens}\big(y_g[|b_S(a)| :]\big)$
    if $C_g \neq \varnothing$ **then**
      $\ell[c] \leftarrow \ell[c] - \lambda_{\text{ESCD}} M_g \,\text{LogSumExp}_{u \in C_g} L_S[c, u]$
    end if
  end for
end for
return $\ell_{1:L}$

C Additional Analysis of ESCD

Section 5 reports two main observations: preserving the completion set yields higher local update fidelity than selecting a single student token, and child supervision is frequently available along student trajectories. This appendix analyzes conditioning states, event structures, boundary effects, and completion depth, alongside reasoning patterns, OPD degradation, and SFT-associated changes.

Specifically, we examine the following questions:

1. **Can the effect of the conditioning state be isolated?** (C.1) We examine the available support for comparing naturally visited and unvisited compatible child states while controlling event identity and completion depth.
2. **Which event structures dominate in practice?** (C.2) We classify events by teacher group size and the number of compatible student actions at the root, comparing their teacher-mass shares and conditional one-step completion coverage across tokenizer pairs.
3. **How much do boundary-crossing completions affect supervision?** (C.3) We measure the probability assigned to tokens extending beyond the residual event boundary and examine how removing them changes the local update.
4. **How often is child supervision available, and is one step enough?**(C.4) We quantify the teacher probability mass reached by compatible on-policy actions and the fraction requiring more than one additional student token to complete.
5. **How do rollout-level reasoning patterns vary across models?** (C.5) We use PCA to describe the overlap and variation of response-level reasoning features across models within each benchmark.
6. **How does OPD degradation manifest under large Teacher–Student scale disparity?** (C.6) We document repetitive continuations and premature termination through rollout statistics and saved responses, showing that termination alone does not imply task completion.
7. **What changes in reasoning performance and behavior accompany SFT?** (C.7) We compare aggregate scores and matched-prompt responses from the teacher, base student, and SFT student, focusing on derivation structure and explicit consistency checks .

C.1 Does the Choice of Child State Matter?

When a student action partially realizes a teacher event, ESCD supervises the residual at the resulting child state. Other compatible actions can lead to different child states and residuals. We ask whether the visited state provides better completion supervision than these alternatives.

Comparison setup. We compare the child reached by the sampled action with a child that would have been reached under another compatible action. We call the latter a counterfactual child because the rollout did not actually take that action. Both branches are evaluated for completion of the same teacher event, with root supervision held fixed. The alternative action is selected without using COUF scores or downstream results. To isolate the effect of the child state, both branches should also use the same completion depth. For the one-step objective studied here, this requires both residuals to admit completion by one additional student token. Otherwise, a difference in supervision could reflect the number of continuation steps rather than the choice of state.

Available comparisons. The current data provide few examples meeting these requirements. We identify 32 Qwen and 37 GLM rollout contexts where the visited child state and a counterfactual child state can be matched for the same teacher event. These pairs cover little teacher probability mass, and none allows both branches to complete the same event in one additional step (Table 13). Thus, the available data do not provide a comparison that simultaneously controls event identity and one-step completion depth.

Comparison allowing deeper completion. To evaluate the available alternatives, we allow the counterfactual branch to continue for additional steps when one-step completion is unavailable. The visited branch retains ESCD's one-step objective. Table 14 reports agreement with the reference

Table 13: Available comparisons between visited and alternative child states. The last column counts contexts where both branches admit one-step completion of the same teacher event.

Teacher → Student	Contexts	Matched mass	Both 1-step
Qwen3 → Qwen3.5	32	0.8745%	0
GLM-Z1 → Qwen3.5	37	0.0662%	0

Table 14: State-space COUF when the counterfactual branch may use deeper completion. The teacher event and root supervision are matched, but completion depth is not controlled.

Teacher → Student	Root	Counterfactual	Visited
Qwen3 → Qwen3.5	0.999031	0.999982	0.999032
GLM-Z1 → Qwen3.5	0.998486	0.999997	0.998489

logit gradient using the COUF metric defined in Section 5. Under this relaxed comparison, the counterfactual branch achieves higher COUF than the visited branch in both model pairs.

These results do not establish which child state is preferable: the branches differ in both conditioning state and allowed completion depth, and the comparison covers little teacher mass. ESCD uses visited states because their logits are already available from the student training forward pass, allowing completion supervision without additional rollouts. The present analysis does not establish an independent supervision advantage of these states over other compatible states.

C.2 What Event Structures Does ESCD Encounter?

An event may admit several compatible student actions at the root. Some produce only a prefix of the event, while others complete it immediately. When the sampled action produces a strict prefix, the student reaches a child state with a residual byte constraint. Different prefix actions can leave different residuals, so compatibility at the root does not guarantee that the event can be completed in one additional step. We therefore distinguish the number of compatible first actions from the number of valid completions after the sampled action. The analysis below characterizes the former and measures how often the latter is nonzero along student trajectories.

Event structure at the root. For a teacher group g, let $m_g = |V_g|$ denote its size and y_g its representative byte string. We partition compatible student first actions into those that partially realize y_g and those that complete it:

$$\begin{aligned} \mathcal{A}(y_g) &= \{u \in V_S^{\text{cont}} : b_S(u) \prec y_g\}, \\ \mathcal{T}(y_g) &= \{u \in V_S^{\text{cont}} : y_g \preceq b_S(u)\}, \\ b_g &= |\mathcal{A}(y_g)| + |\mathcal{T}(y_g)|. \end{aligned} \tag{68}$$

Actions in $\mathcal{A}(y_g)$ leave a nonempty residual, whereas those in $\mathcal{T}(y_g)$ complete the event exactly or extend beyond its boundary. Their combined count b_g describes the choices available at the root.

After sampling $a \in \mathcal{A}(y_g)$, $n_g(a) = |\mathcal{C}(r_g)|$ counts valid one-step completions at the child state (Appendix B.4). The counts describe different stages: a single compatible first action ($b_g = 1$) may leave a residual with zero, one, or multiple completions.

Coverage along student trajectories. We analyze the full teacher vocabulary at aligned, non-whitespace positions in 16 frozen student trajectories per pair. After removing invalid tokens and exact byte matches to student content tokens, we aggregate candidates by prefix compatibility. Table 15 groups event occurrences by m_g and b_g, restricting attention to $\mathcal{A}(y_g) \neq \varnothing$. Each occurrence is counted at its rollout position and weighted by teacher group mass M_g.

Mass share gives each class's proportion of analyzed teacher mass. *Child visited* is the within-class mass fraction with a sampled action in $\mathcal{A}(y_g)$. *One-step given visit* is the fraction of visited mass admitting at least one valid one-step completion, measuring availability rather than sampled success. Overall values use pooled masses across classes.

Table 15: Root event structure and completion availability from the full teacher vocabulary after exact-shared and invalid-token filtering. Statistics use aligned, non-whitespace positions in 16 frozen student trajectories per pair, grouped by m_g and b_g.

Pair	m_g	b_g	Mass share	Child visited	One-step given visit
Qwen3 → Qwen3.5	1	1	0.0026%	34.5351%	96.0012%
	1	>1	0.1089%	70.6500%	71.9885%
	>1	1	**88.2125%**	80.9535%	99.9991%
	>1	>1	11.6760%	88.6045%	92.4118%
	Overall		100.0000%	**81.8344%**	**99.0135%**
GLM-Z1 → Qwen3.5	1	1	2.3143%	88.8103%	99.9993%
	1	>1	0.0917%	51.3026%	80.4281%
	>1	1	**90.1026%**	82.6607%	100.0000%
	>1	>1	7.4914%	89.5888%	93.0875%
	Overall		100.0000%	**83.2933%**	**99.4319%**

Table 16: Boundary-crossing ambiguity on materialized natural-child events. "Set size" is weighted by teacher source mass; "Crossing prob." is the fraction of completion-event probability assigned to tokens that extend beyond the residual boundary.

teacher → Student	Events	Set size	Full prob.	Crossing prob.
Qwen3 → Qwen3.5	48	35.28	0.8846	**2.10%**
GLM-Z1 → Qwen3.5	287	14.05	0.9412	**0.63%**

Groups with multiple teacher candidates and one compatible student first token ($m_g > 1$, $b_g = 1$) account for 88.21% of analyzed mass for Qwen and 90.10% for GLM. Nearly all their visited mass admits one-step completion. Lower-coverage classes contribute much less visited mass, yielding overall conditional coverage of 99.01% and 99.43%, respectively. This describes supervision availability, without attributing downstream gains to individual classes.

C.3 Does the Completion Set Become Too Coarse at the Event Boundary?

The event-set construction in ESCD deliberately avoids choosing a single student realization. This raises a potential ambiguity when a student token completes the residual teacher event but also extends beyond its boundary. For example, if the residual bytes correspond to `Memory`, both `Memory` and `MemoryWarning` satisfy the current prefix-compatible completion rule. Importantly, this does not mean that the two tokens are treated as semantically equivalent. ESCD supervises the byte-prefix event that the continuation completes the residual, rather than requiring the next student token to terminate exactly at the same boundary. A longer token can therefore satisfy the event while additionally committing bytes that are not specified by the current teacher event.

How much probability crosses the boundary? To quantify whether this ambiguity is substantial in practice, we separate exact completions from tokens that strictly extend beyond the residual. We report the fraction of the full completion-event probability assigned to the latter:

$$\rho_{\text{cross}} = \frac{\sum_{u:\, r \prec b_S(u)} p_\theta(u \mid x, a)}{\sum_{u:\, r \preceq b_S(u)} p_\theta(u \mid x, a)}. \tag{69}$$

A large completion set does not necessarily imply a large ρ_{cross}: many prefix-compatible tokens may exist in the vocabulary while receiving negligible student probability. We therefore evaluate ρ_{cross} on naturally visited strict-prefix events whose child logits are available for computation, resulting in 48 Qwen and 287 GLM events.

Table 16 reveals an important distinction between tokenizer topology and realized student probability. Qwen has an average completion-set size above 35, yet boundary-crossing tokens carry only 2.10% of the completion probability; for GLM, the corresponding fraction is only 0.63%. Thus, a large prefix-compatible set does not imply that probability is broadly distributed across its members.

Table 17: Local target-gradient sensitivity to removing boundary-crossing completions.

Teacher → Student	Projection	Energy ratio	Cosine
Qwen3 → Qwen3.5	1.0566	1.1550	**0.9832**
GLM-Z1 → Qwen3.5	1.0302	1.0857	**0.9887**

For Qwen, this large set size is primarily driven by very short residuals: 45 of the 48 materialized events contain a one-byte residual, with newline being the dominant case. Short residuals naturally induce large prefix cylinders, but the student probability remains strongly concentrated near the exact completion.

What does the event loss leave unidentified? Although the realized crossing probability is small, the full event-set objective has a genuine identifiability limitation. The child loss depends only on the total probability assigned to the completion set:

$$\mathcal{L}_{\text{child}} = -M \log P_\theta(\mathcal{C}_{\text{full}}(r) \mid x, a). \tag{70}$$

Consequently, redistributing probability among tokens inside the same completion set while keeping their total probability fixed leaves this loss unchanged. For example, assigning completion probability $(0.8, 0.1)$ to an exact and a crossing token produces the same child loss as $(0.1, 0.8)$ whenever the total event probability is unchanged. This is a property of event marginalization rather than evidence of an observed failure. It means that ESCD identifies the probability of completing the teacher byte event, but does not locally identify how this probability should be allocated among different student realizations that already satisfy the event.

Does removing crossing completions materially change the local update? We next compare the current **Full** target with a stricter **Exact** target that retains only student tokens ending exactly at the residual boundary. We also consider a **Minimal** target that keeps the shortest prefix-compatible completions. In the current materialized subsets, every evaluated event has an exact one-step completion, and the shortest compatible completion is always exact. Therefore, **Minimal and Exact are identical on all evaluated events**; the present diagnostic can distinguish only Exact/Minimal from Full. We compute the local logit gradient induced by each target on the same student child logits and compare the Exact/Minimal gradient with the Full gradient.

Removing boundary-crossing tokens increases the local gradient magnitude slightly, but the resulting direction remains highly aligned with the Full target: the gradient cosine is 0.9832 for Qwen and 0.9887 for GLM. In the current frozen natural-child subsets, boundary crossing therefore changes the local target quantitatively but does not induce a substantially different update direction.

Takeaway. The analysis leads to three conclusions. First, boundary-crossing tokens are consistent with ESCD's byte-prefix event semantics and should not be interpreted as an implementation error or as semantically equivalent tokens. Second, the full event-set objective does leave probability allocation within the completion set under-specified, but the realized crossing probability is small in the current Qwen and GLM subsets. Third, removing crossing tokens changes gradient magnitude but preserves a highly similar local update direction. We therefore view boundary crossing as a genuine source of local target coarsening, but not as an observed failure mode in the current frozen analysis. Determining whether stricter boundary-aware targets improve training would require native Exact/Minimal/Full runs under a common training protocol; in particular, events without an exact one-step completion are needed to distinguish Minimal from Exact.

C.4 How Often Is Child Supervision Available, and Is One Step Enough?

We further examine the availability of child supervision and the scope of one-step completion. The analysis uses the full teacher vocabulary at aligned, non-whitespace positions in 16 frozen student trajectories per pair: 374809 positions for Qwen and 365153 for GLM. All statistics follow the candidate filtering and prefix aggregation described in the main text. We pool teacher-mass numerators and denominators across positions and trajectories before computing percentages.

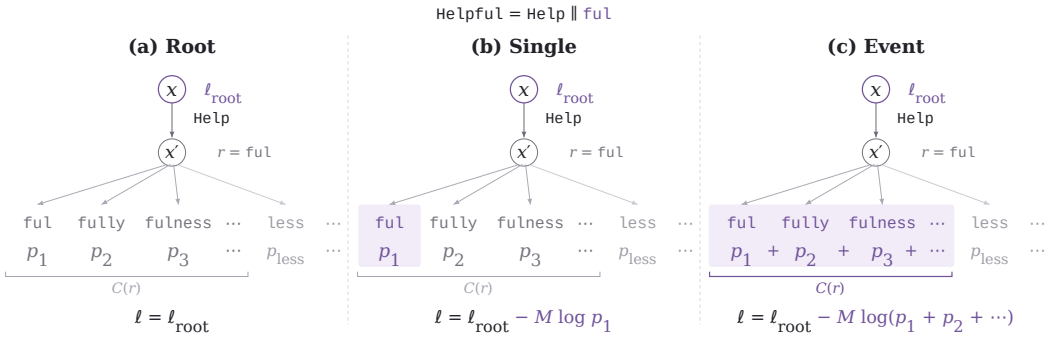


Figure 6: **One-step completion and supervision choices.** Sampling `Help` leaves residual $r =$ `ful` for the representative event `Helpful`. Valid one-token completions include `ful`, `fully`, and `fulness`; `less` lies outside $\mathcal{C}(r)$. All three objectives share the same root loss and student predictions. **(a) Root** adds no child loss. **(b) Single** supervises one selected completion. **(c) Event** supervises the completion set's total probability. Here, p_i denotes a candidate's native student probability at x', and M is the representative group's teacher mass. Ellipses indicate additional candidates. Tokenizations and the Single selection are illustrative; losses are shown for one group before training-mask reduction.

A representative event y_g is branchable if some student action produces a nonempty strict prefix of its bytes. At each position, a continuing student action is scored by the total teacher mass of compatible branchable representative events. The three visitation statistics in Table 18 share a denominator: total branchable teacher mass across all analyzed positions.

Position upper bound counts all branchable mass at a position whenever the sampled action matches any continuing branch. **Observed** counts only the event mass compatible with the sampled action. **Top-1 observed** counts the highest-scoring branch's mass only when that branch is sampled. Branches are ranked by teacher source mass, not student action probability, with ties broken by ascending student token ID. These are mass fractions, not position hit rates; the position upper bound can include events incompatible with the sampled action.

Table 18: Compatible-child visitation under full-vocabulary analysis. All columns are relative to the same pooled branchable teacher mass. The position upper bound counts all branchable mass at positions with any compatible sampled action; Top-1 ranks branches by teacher source mass.

Teacher → Student	Position upper bound	Observed	Top-1 observed
Qwen3 → Qwen3.5	93.47%	**81.83%**	71.55%
GLM-Z1 → Qwen3.5	95.49%	**83.29%**	76.23%

Compatible children are naturally visited for 81.83% and 83.29% of branchable teacher mass for Qwen and GLM, respectively. Thus, over 80% of branchable mass is compatible with sampled student actions. This measures visitation, not completion by the next sampled token. Visitation does not guarantee that the residual admits one-step completion.

We measure completion availability conditional on this visited teacher mass. For each visited representative event, the sampled action leaves a residual r_g. **1-step** is the mass fraction for which the completion set $\mathcal{C}(r_g)$ is nonempty, and **Deeper** is the remaining fraction. These quantities measure whether a next-token completion exists in the student vocabulary, rather than how often the student samples one. The completion criterion applies to the representative residual after prefix aggregation.

One-step completion may admit multiple valid tokens. Figure 6 illustrates how Root, Single, and Event use the same visited state: Root adds no child loss, Single supervises one selected valid completion, and Event supervises the total probability of the completion set. Single and Event therefore differ in their supervision targets, not in completion depth.

Table 19: One-step completion availability for representative residual constraints, weighted by teacher mass and conditional on a compatible child being visited.

Teacher → Student	1-step	Deeper
Qwen3 → Qwen3.5	**99.01%**	0.99%
GLM-Z1 → Qwen3.5	**99.43%**	0.57%

Table 19 shows that representative residuals covering 99.01% of visited Qwen mass and 99.43% of visited GLM mass admit completion by one additional student token. The remaining 0.99% and 0.57% lack one-step completion and contribute no auxiliary child loss under ESCD.

These measurements support the availability of ESCD's local child supervision in the studied tokenizer pairs and trajectories. Coverage depends on the filtering, aggregation, and visitation criteria above; it does not establish one-step completion of every original teacher token or satisfaction of longer group members' full byte requirements. Residuals without one-step completion remain a small but nonzero share of observed mass, motivating deeper completion.

C.5 Rollout-Level Reasoning Geometry

Before examining individual training regimes, we use PCA to visualize variation in response-level reasoning signatures across models and benchmarks (Figure 7). The projections cover Math, Code, PHYRD-40, and FrontierScience, providing a qualitative overview of the overlap and variation among teacher and student responses within each benchmark and model group.

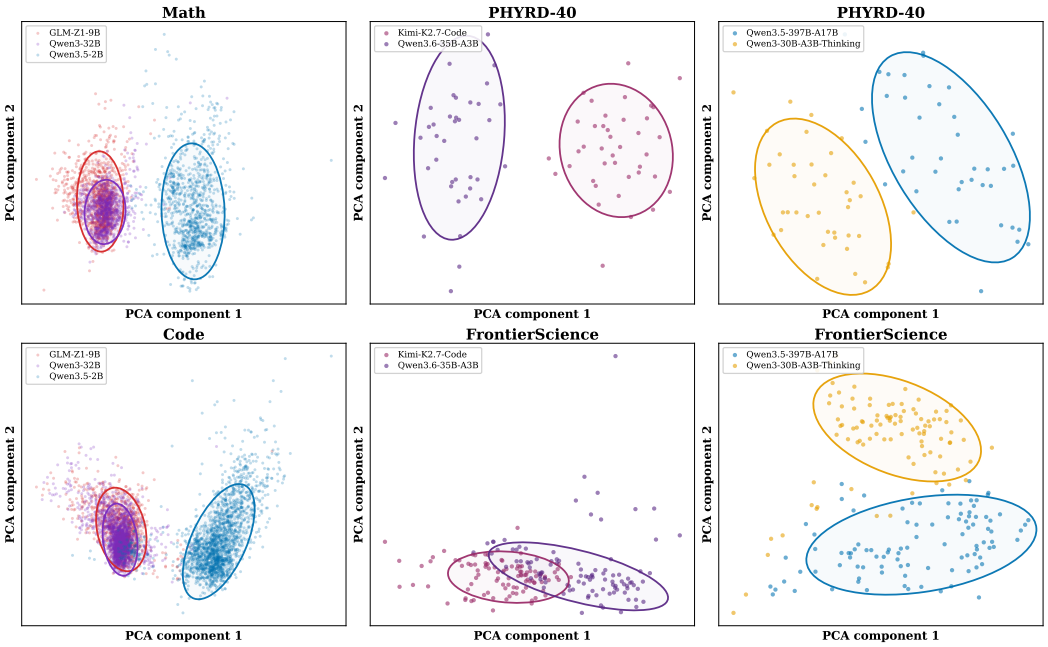


Figure 7: PCA projections of response-level reasoning features across math, code, and scientific reasoning benchmarks. Each point represents a generated response, with colors identifying models; ellipses summarize response distributions after robust trimming. PCA is fitted separately in each panel, so distances and directions are comparable only within panels.

C.6 OPD Degradation under Large Teacher–Student Scale Disparity

We observe severe generation degradation during direct BPM-based OPD from Kimi-K2.7-Code to Qwen3.6-35B-A3B, a pairing with a large disparity in total parameter count. Without additional SFT initialization, the student exhibits two failure modes: repetitive continuations that exhaust the de-

coding budget and premature termination before substantive reasoning. We document these failures using training statistics and saved responses.

Repetitive continuation and premature termination. At rollout steps 1–19, no responses are length-truncated or flagged by the native repetition detector. At step 20, one of 16 responses is both repetition-flagged and length-truncated; at step 21, two of 16 responses meet both criteria. All three reach the 80,000-token limit, with repetitive passages consisting of the symbol $\mathbb{Z}$, the phrase *"classifies by Weyl-Equivalence"*, or malformed LaTeX fragments. Premature termination also occurs at step 20: one response ends after only 12 tokens, briefly restating the topic without beginning substantive reasoning. These observations show that excessive repetition and premature termination can coexist at the same training stage.

Response-length collapse and premature termination. In the truncation-filtered run, retained responses become markedly shorter during training (Table 20), with mean length falling from 10,587 tokens at step 16 to 8 at step 32. All 16 responses at step 32 contain at most 20 tokens, consisting of unfinished preambles, task fragments, or special tokens followed by termination. Outputs remain short at step 38. The saved text confirms that this shortening reflects termination before substantive reasoning rather than more concise solutions.

Table 20: Response-length collapse in the truncation-filtered Kimi-to-Qwen BPM run. Statistics describe the 16 responses retained at each listed step, rounded to the nearest token.

Step	Mean tokens	Median tokens	Max tokens
1	7,698	7,550	14,470
16	10,587	7,605	50,590
20	3,022	2,518	8,099
26	247	24	1,818
30	56	8	566
32	8	6	20
38	20	19	52

Representative responses. Together, these cases reveal a stability challenge for direct OPD in this Teacher–student pairing: responses may exhaust the decoding budget through repetition or terminate before substantive reasoning. These failures motivate examining SFT as a preparatory stage for subsequent OPD. We next analyze the SFT checkpoint's reasoning performance and response structure to characterize the initialization used in our large-scale distillation experiments.

Repetitive continuation and premature termination

`Repetitive continuation`

Unfiltered run, step 20. The prompt asks for an analysis connecting linear angle evolution to a Fourier representation of late-time density. The response initially discusses this connection but later repeatedly emits:

$$\mathbb{Z}\mathbb{Z}\mathbb{Z}\mathbb{Z}\mathbb{Z}\mathbb{Z}\mathbb{Z}\mathbb{Z}\cdots$$

The repeated passage is abbreviated. The response reaches the 80,000-token limit and is recorded as length-truncated.

`Premature termination`

Filtered run, step 32. Three saved responses to distinct mathematical-physics prompts are shown below:

Example 1: `We need<|im_end|>`

Example 2: `Here<|im_end|>`

Example 3: `</think><|im_end|>`

All 16 retained responses at this step contain at most 20 tokens and fail to provide substantive solutions. No length-truncated responses are discarded.

C.7 Reasoning Performance and Structure after SFT

We examine the Qwen3.6-35B-A3B checkpoint after supervised fine-tuning, which improves aggregate long-form reasoning performance over the base student. Matched-prompt examples also show a recurring organizational difference: normalization choices, boundary conditions, limiting

cases, and local consistency checks are more often explicit before the derivation concludes. For SFT initialization, we use Kimi-K2.7-Code to generate approximately 1,800 trajectories from the OPD training prompts and fine-tune the student on these trajectories for three epochs.

Evidence and scope. On PHYRD-40, the Qwen3.6-35B-A3B base model scores 59.2, the SFT checkpoint 66.2, and Kimi-K2.7-Code 83.6. Their FrontierScience Olympiad accuracies are 61.0%, 68.0%, and 75.0%, respectively. SFT improves the student's mean rubric score by 7.0 points and Olympiad accuracy by 7.0 percentage points, yielding a more capable starting checkpoint for subsequent OPD. Matched-prompt examples complement these results by comparing how the teacher, base student, and SFT student handle assumptions, intermediate derivations, and consistency checks, characterizing the SFT student's reasoning behavior before OPD.

C.7.1 Matched-Prompt Reasoning Patterns

We compare lightly trimmed final-answer fragments from three shared PHYRD-40 prompts. The excerpts are drawn only from the models' visible responses, and all selected generations terminate normally. Three examples are not sufficient for a model ranking; they are used here to inspect how the teacher, base student, and SFT student structure long derivations, represent intermediate quantities, perform local checks, and close the requested argument.

A033: Linear displacement kernel. The task requires the progression $\Delta\mathbf{s} \to C_{ij}(q) \to \{A(q), B(q)\} \to K(\mathbf{q}, \mathbf{k})$, together with consistency checks such as $q \to 0$ and $\mathbf{k} = 0$.

Kimi-K2.7-Code

`Final-answer fragment`

One finds

$$C_{ij}(\mathbf{q}) = \frac{1}{\pi^2}\int_0^\infty dp\, P(p)\left[\frac{1 - j_0(pq) - j_2(pq)}{3}\,\delta_{ij} + j_2(pq)\,\hat{q}_i\hat{q}_j\right].$$

Thus

$$A(q) = \frac{1}{\pi^2}\int_0^\infty dp\, P(p)\frac{1 - j_0(pq) - j_2(pq)}{3}, \qquad B(q) = \frac{1}{\pi^2}\int_0^\infty dp\, P(p)\, j_2(pq).$$

At zero separation, $j_0(0) = 1$ and $j_2(0) = 0$, so $A(0) = B(0) = 0$. Substitution gives

$$K(\mathbf{k}, \mathbf{q}) = \exp\left[-\frac{1}{2}\left(A(q)k^2 + B(q)(\mathbf{k}\cdot\hat{\mathbf{q}})^2\right)\right].$$

Qwen3.6-35B-A3B Base

`Final-answer fragment`

The linear displacement is a zero-mean Gaussian random field. Therefore

$$W(\mathbf{q}, \mathbf{k}) = \exp\left[-\frac{1}{2}k_i k_j \langle \Delta s_i^{(1)} \Delta s_j^{(1)} \rangle\right].$$

Decompose

$$C_{ij}(q) = \alpha(q)\delta_{ij} + \beta(q)\hat{q}_i\hat{q}_j.$$

Defining transverse and longitudinal contractions $T(q)$ and $L(q)$ gives

$$3\alpha + \beta = 2T, \qquad \alpha + \beta = 2L,$$

and hence

$$\alpha = T - L, \qquad \beta = 3L - T.$$

Substitution yields the final rotationally invariant kernel.

Qwen3.6-35B-A3B + SFT

Final-answer fragment

Carrying out the angular integrals gives explicit radial forms for $A(q)$ and $B(q)$. The kernel is

$$K(\mathbf{q}, \mathbf{k}; t) = \exp\left[-\frac{1}{2}\left(A(q)k^2 + B(q)(\mathbf{k}\cdot\hat{\mathbf{q}})^2\right)\right].$$

For $\mathbf{k} = 0$, the exponent vanishes and $K = 1$. As $q \to 0$, $\Delta\mathbf{s} \to 0$, so $A(q), B(q) \sim q^2$ and $K \to 1$. The covariance is positive semidefinite because it is constructed from the squared displacement difference.

Observed pattern. All three responses reach the same kernel structure. The clearest difference appears after the main derivation: the SFT response explicitly checks $\mathbf{k} = 0$, the $q \to 0$ limit, and positive semidefiniteness, whereas the base response stops after the compact T/L construction. Kimi also checks the zero-separation limit, but organizes the derivation directly around the covariance functions.

A065: Coherent-state MPO exponential. The task requires iterating a normalized Gaussian source identity into a coherent-state MPO while preserving signs, auxiliary contractions, operator support, and boundary conditions.

Kimi-K2.7-Code

Final-answer fragment

The local tensor is

$$W_i(\phi_{i-1}, \bar{\phi}_i) = \exp\left[\phi^{\alpha}_{i-1}\hat{A}^{\alpha\beta}_i\bar{\phi}^{\beta}_i + \phi^{\alpha}_{i-1}\hat{B}^{\alpha}_i + \hat{C}^{\beta}_i\bar{\phi}^{\beta}_i + \hat{D}_i\right].$$

If interactions vanish, the Gaussian integrals decouple and recover the on-site exponential. For two sites, the remaining integral produces $e^{h_{L_1}\cdot h_{R_1}}$. For a three-site term $O_1O_2O_3$, choosing $\hat{C}_1 = O_1$, $\hat{A}_2 = O_2$, and $\hat{B}_3 = O_3$ reproduces the expected exponent after two Gaussian contractions. Thus $\hat{A}$ propagates the auxiliary channel responsible for longer-range operators.

Qwen3.6-35B-A3B Base

Final-answer fragment

Collecting the fields at one cut gives

$$-\bar{\phi}_i\cdot\phi_i + h_{L_i}\cdot\bar{\phi}_i + \phi_i\cdot\left(\hat{B}_{i+1} + \hat{A}_{i+1}h_{R_{i+1}}\right).$$

Applying the source identity yields

$$\mathcal{W}_{i+1}(u, v) = \exp\left[u\cdot\hat{B}_{i+1} + u\cdot\hat{A}_{i+1}\cdot v\right]\exp\left[\hat{C}_{i+1}\cdot v + \hat{D}_{i+1}\right].$$

Iterating over all cuts contracts neighboring auxiliary states and yields the full MPO exponential.

Qwen3.6-35B-A3B + SFT

Final-answer fragment

The full representation is

$$e^{H} = e^{H_{L_1}}\int\prod_{i=1}^{N-1}\mathcal{D}[\phi_i, \bar{\phi}_i]\prod_{i=1}^{N-1}\mathcal{W}_i(\phi_i, \bar{\phi}_i; \phi_{i+1}, \bar{\phi}_{i+1})e^{H_{R_N}}.$$

The Gaussian normalization is recovered when all sources vanish. Every quadratic term has sign $-\bar{\phi}_i\cdot\phi_i$, while the source identity produces $+J\cdot\bar{J}$. For $N = 2$, the remaining Gaussian integral reproduces $e^{h_{L_1}\cdot h_{R_1}}$, recovering the original cut decomposition.

Observed pattern. The base response proceeds directly from the cut identity to the iterated representation. The SFT response adds explicit checks of the Gaussian normalization, the sign convention, and the $N = 2$ reduction before stopping. Kimi performs a related set of checks through low-order recovery examples, including the two-site and three-site cases.

A075: AdS scalar bulk-to-boundary convolution. The task requires connecting the bulk solution, source normalization, distributional boundary limit, and CFT scaling law without conflating pointwise decay with source recovery.

Kimi-K2.7-Code

`Final-answer fragment`

With the source normalization fixed,

$$\lim_{x_0\to 0} x_0^{-(d-\Delta)}\Phi(x_0,\mathbf{x}) = \phi_0(\mathbf{x}).$$

Away from the source, the leading behavior is the faster x_0^{Δ} tail. The near-boundary expansion is

$$\Phi = x_0^{d-\Delta}\phi_0 + x_0^{\Delta}\psi + \cdots, \qquad \psi(\mathbf{x}) = C_\Delta \int d^d\mathbf{x}' \frac{\phi_0(\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|^{2\Delta}}.$$

The coefficient ψ is the linear response, so

$$\langle \mathcal{O}(\mathbf{x})\mathcal{O}(\mathbf{x}')\rangle \propto |\mathbf{x}-\mathbf{x}'|^{-2\Delta}.$$

Qwen3.6-35B-A3B Base

`Final-answer fragment`

Set $\mathbf{y} = x_0\mathbf{z}$ in the boundary convolution. Then

$$\phi(x_0,\mathbf{x}) = \mathcal{N}_\Delta \int_{\mathbb{R}^d} \frac{d^d\mathbf{z}}{(1+|\mathbf{z}|^2)^\Delta}\phi_0(\mathbf{x}+x_0\mathbf{z}).$$

Since

$$\int_{\mathbb{R}^d} \frac{d^d\mathbf{z}}{(1+|\mathbf{z}|^2)^\Delta} = \frac{\pi^{d/2}\Gamma(\Delta-d/2)}{\Gamma(\Delta)},$$

the Dirichlet condition fixes

$$\mathcal{N}_\Delta = \frac{\Gamma(\Delta)}{\pi^{d/2}\Gamma(\Delta-d/2)}.$$

Conformal invariance then fixes the boundary two-point power to 2Δ.

Qwen3.6-35B-A3B + SFT

`Final-answer fragment`

The kernel vanishes pointwise for $\mathbf{x}\neq\mathbf{x}'$ but is singular at the source. With

$$C_\Delta = \frac{\Gamma(\Delta)}{\pi^{d/2}\Gamma(\Delta-d/2)},$$

its rescaled distributional limit is

$$\lim_{x_0\to 0} x_0^{\Delta-d}K_\Delta(x_0,\mathbf{x}-\mathbf{x}') = \delta^d(\mathbf{x}-\mathbf{x}').$$

The dual boundary dimension and two-point function are

$$\Delta = \frac{d}{2} + \sqrt{\frac{d^2}{4}+m^2}, \qquad \langle O(\mathbf{x})O(\mathbf{x}')\rangle = \frac{C_{OO}}{|\mathbf{x}-\mathbf{x}'|^{2\Delta}}.$$

Observed pattern. The main distinction in this example is the treatment of the boundary limit. The SFT response explicitly separates pointwise decay away from the source from distributional recovery of the boundary source. The base response instead concentrates on the change of variables that fixes the normalization constant, while Kimi connects the normalized bulk solution directly to the normalizable coefficient and boundary response.

Cross-case pattern. Across the three prompts, the recurring SFT-associated difference lies primarily in the organization of the derivation rather than in a uniform change of the mathematical endpoint. The SFT responses more consistently expose normalization choices, boundary conditions, limiting cases, and local consistency checks as explicit steps before termination. The base student generally follows a shorter prompt-aligned progression, while Kimi often organizes the answer around the central theoretical structure and uses selected structural or low-order checks. These examples are qualitative, but they show a consistent separation between deriving the main result and checking the conditions under which that result is valid.

Relation to OPD degradation. The direct-OPD diagnostics in Appendix C.6 and the SFT comparisons here address distinct questions. The former document repetitive continuations and premature termination during training; the latter show improved benchmark performance and illustrate the organization of derivations in SFT responses. Together with the main results, these observations motivate using SFT as an initialization stage and show that subsequent OPD with ESCD can provide further gains. They do not, however, establish that SFT initialization prevents the observed degradation; assessing this effect requires matched OPD runs with and without SFT initialization under otherwise identical training and evaluation conditions.

C.8 SUMMARY

These analyses clarify the mechanisms and scope of ESCD. The local gradient comparisons favor the event-set target over a single-completion target, while the coverage analysis shows that one additional student action can complete representative residual constraints accounting for over 99% of the observed compatible teacher probability mass in the studied tokenizer pairs. Together, these results support the completion-set design and the practical availability of visited-child supervision. Additional OPD diagnostics characterize repetitive continuation and premature termination in the Kimi-to-Qwen configuration, while the SFT comparisons provide complementary evidence on benchmark performance and derivation structure.