This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

SelecTKD: Selective Token-Weighted Knowledge Distillation for LLMs

Haiduo Huang, Jiangcheng Song, Yadong Zhang, Pengju Ren*
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
huanghd,enone,3488928835@stu.xjtu.edu.cn, pengjuren@xjtu.edu.cn

Abstract

Knowledge distillation (KD) is a standard route to compress Large Language Models (LLMs) into compact students, yet most pipelines uniformly apply token-wise loss regardless of teacher confidence. This indiscriminate supervision amplifies noisy, high-entropy signals and is especially harmful under large teacher-student capacity gaps. We introduce SelecTKD, a plug-and-play Selective Token-Weighted *distillation framework that shifts the focus from "how to measure divergence" to "where to apply learning". At each step, the student proposes tokens that are verified by the teacher through a robust* propose-and-verify *procedure with two variants:* greedy Top-k *and* non-greedy Spec-k*. Accepted tokens receive full loss, while rejected tokens are masked or down-weighted. This objective-agnostic design works with on- and off-policy data, induces an implicit curriculum quantified by Token Acceptance Rate (TAR), and stabilizes optimization. Across instruction following, mathematical reasoning, code generation, and a VLM setting, SelecTKD consistently improves strong baselines and achieves state-of-the-art results for small models without architectural changes or extra reference models.*

1. Introduction

Large Language Models (LLMs) have made rapid progress through scaling compute, parameters, and data [23, 27], yet their inference-time latency and memory footprint hinder broad deployment [54]. Compressing LLMs into compact students without sacrificing capability is thus a central practical challenge. Knowledge distillation (KD) [18] is the de facto approach for transferring competency from a powerful teacher to a smaller student and has enabled strong Small Language Models (SLMs), as illustrated by recent open-source families [48, 55]. However, most KD pipelines apply a uniform token-wise loss, even when the teacher is uncertain, forcing the student to imitate high-entropy predictions. This indiscriminate supervision injects noise, especially under large capacity gaps, and can destabilize optimization.

The literature on LLM distillation mainly spans three axes. (1) Data regimes: dynamic selection and on-/off-policy pipelines mitigate exposure bias [2, 3, 29, 34, 63]. (2) Objectives: beyond vanilla KL, reverse/adaptive/skewed divergences and other f-divergences are explored [15, 26, 29, 50, 51]. (3) Transferred knowledge: multi-granularity semantics and adaptive interpolation address what and when to transfer [36, 38, 45, 49]. DistiLLM-2 [30] further argues that matching losses to data type (e.g., SKL for teacher data, SRKL for student data) is crucial. Yet we ask: is loss geometry the dominant factor?

Motivated by this question, we revisit the role of loss geometry from both empirical and theoretical perspectives. Our preliminary experiments (Figure 1) compare multiple KL-family objectives and loss/data pairings under identical training budgets and data curation, and observe surprisingly similar end performance across choices, even when DistiLLM-2-style asymmetric pairings are used; additional results on other evaluation platforms are provided in the Appendix. This trend echoes the analysis of [51]: while forward/reverse/skewed KLs can yield different optimization *paths*, they share the same fixed point in the limit (see also our Appendix proofs). In other words, much of the observed difference is attributable to training dynamics rather than the final solution.

This reframes the central question: rather than optimizing only *how* to measure divergence, we should ask **which tokens are worth learning from**. When the teacher-student capacity gap is large [45], uniformly forcing the student to mimic all teacher predictions, including high-entropy or uncertain ones, injects noise and can degrade generalization. We therefore propose to control *where and when* the learning signal is applied at the token level. Inspired by speculative decoding [6, 31], prior works have explored related ideas: SLIM [44] reweights losses but still optimizes logits and may lose dark knowledge [59]; SKD [53] improves data quality by interleaved sampling yet modifies the dataset rather than the supervision signal; AdaSPEC [21] filters difficult tokens via a reference model at extra cost.

In contrast, our **SelecTKD** uses a simple *propose-*

*Corresponding author.

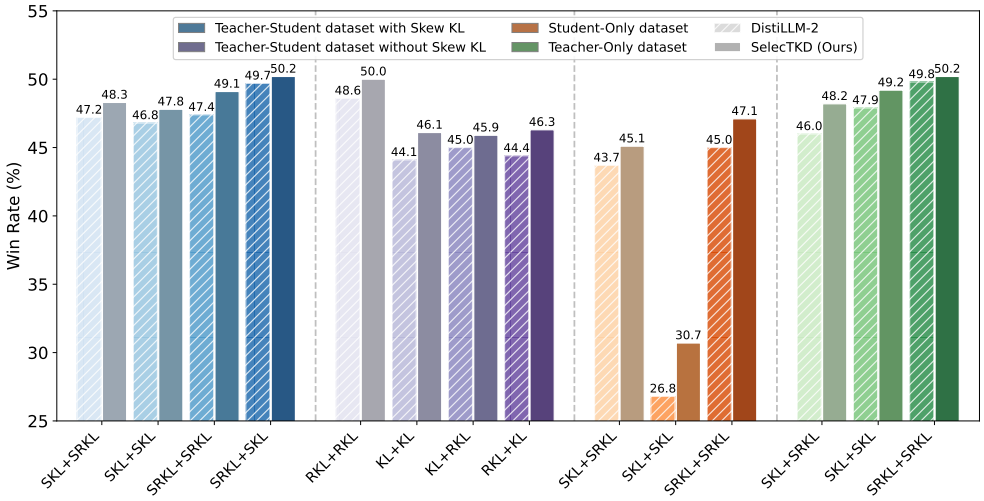


Figure 1. Performance comparison of different loss functions and training datasets on DistiLLM-2 [30] and DistiLLM-2 with SelecTKD, using Qwen2-7B-Inst as the teacher and Qwen2-1.5B as the student. The chart shows Win Rate on the Evol-Instruct benchmark evaluated with GPT-4o. The baseline is gpt-3.5-turbo. This comprehensive comparison highlights the impact of various loss/data combinations and demonstrates the consistent advantage of our SelecTKD method.

and-verify mechanism with token-weighted loss to focus on high-confidence teacher signals. It is plug-and-play, objective-agnostic (KL/RKL/SKL/SRKL), compatible with on- and off-policy regimes, and induces a stable implicit curriculum quantified by Token Acceptance Rate (TAR). Concretely, the student proposes tokens that the teacher verifies via two variants *greedy Top-k* and *non-greedy Spec-k*; accepted tokens receive full loss, while rejected tokens are masked or weakly weighted.

Our main **contributions** include: (i) We reframe KD from "how to measure divergence" to *"where to apply supervision"*, introducing a selective token-weighted mechanism that emphasizes teacher-consistent tokens. (ii) We present a robust verification scheme *greedy Top-k* and *non-greedy Spec-k* with a single intuitive hyperparameter k and a rejected-token weight β. (iii) We provide theoretical support showing monotonic improvement of TAR and discuss its link to smoother loss landscapes and better generalization. (iv) We demonstrate consistent gains across instruction following, math, code, and a VLM setting achieving SOTA results for small models while remaining plug-and-play and compatible with KL, RKL, SKL, and SRKL.

2. Related Work

2.1. Knowledge Distillation for LLMs

KD [18] compresses LLMs by transferring a strong teacher to a smaller student. On the *objective* side, MiniLLM [15] uses reverse KL, AKL [51] adaptively blends forward/reverse KL, and DistiLLM [29]/DistiLLM-2 [30] introduce skew KL with contrastive designs. On *teaching strategies*, ImitKD [34] mitigates exposure bias via on-policy imitation and ATKD [62] adjusts supervision for easy vs. hard tokens. These works optimize how divergence is measured or how data is curated; SelecTKD instead decides *where* to apply supervision and composes with any of them.

2.2. Token Filtering and Selective Distillation

Not all tokens are equally informative. RHO-1 [35] formalizes selective language modeling by reweighting token losses, while ENT [33] truncates high-error gradients under noisy supervision. Token Cleaning [43] and T-SHIRT [13] apply token-level data selection for fine-tuning; AdaSPEC [21] selectively aligns draft and target models for speculative decoding. SelecTKD is complementary: it applies a token-weighted loss directly during KD via a simple verification mechanism, preserving compatibility with existing objectives and pipelines (see Appendix A for an extended discussion).

3. Preliminaries

We formalize knowledge distillation (KD) for autoregressive LLMs and set up token-level divergences used throughout. An autoregressive model generates an output sequence $\boldsymbol{y} = [y_1, \ldots, y_{|\boldsymbol{y}|}]$ conditioned on an input $\boldsymbol{x}$. At step t, it predicts a token from a finite vocabulary $\mathcal{V} = \{v_1, \ldots, v_{|\mathcal{V}|}\}$. The teacher and student define conditional distributions $p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$ and $q_\theta(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$, respectively.

Token-level divergences. At each time step t and for each vocabulary token $v_i \in \mathcal{V}$, we write $p_i := p(v_i \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$ and $q_i := q_\theta(v_i \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$. The forward and reverse KL contribu-

tions at (t, i) are

$$D_{\mathrm{FKL}}^{(t,i)}(p, q_\theta) = p_i \log \frac{p_i}{q_i}, \qquad D_{\mathrm{RKL}}^{(t,i)}(p, q_\theta) = q_i \log \frac{q_i}{p_i}. \quad (1)$$

Sequence-level objectives. Aggregating over steps and vocabulary yields the total forward and reverse KL losses:

$$\mathcal{L}_{\mathrm{FKL}} = \sum_{t=1}^{|\boldsymbol{y}|} \sum_{i=1}^{|\mathcal{V}|} D_{\mathrm{FKL}}^{(t,i)}(p, q_\theta), \; \mathcal{L}_{\mathrm{RKL}} = \sum_{t=1}^{|\boldsymbol{y}|} \sum_{i=1}^{|\mathcal{V}|} D_{\mathrm{RKL}}^{(t,i)}(p, q_\theta). \quad (2)$$

However, forward KL encourages mass-covering, whereas reverse KL favors mode-seeking [15]. To balance these, DistiLLM [29] proposed skewed variants, skew KL (SKL) and skew reverse KL (SRKL), which interpolate between teacher and student distributions.

$$D_{\mathrm{SKL}}^{(\alpha)}(\boldsymbol{x}, \boldsymbol{y}; p \| q_\theta) = D_{\mathrm{KL}}(\boldsymbol{x}, \boldsymbol{y}; p \| \alpha p + (1-\alpha) q_\theta), \quad (3)$$

$$D_{\mathrm{SRKL}}^{(\alpha)}(\boldsymbol{x}, \boldsymbol{y}; p \| q_\theta) = D_{\mathrm{KL}}(\boldsymbol{x}, \boldsymbol{y}; q_\theta \| (1-\alpha) p + \alpha q_\theta). \quad (4)$$

where $\alpha \in [0, 1]$ controls the mixing ratio. Building on this, DistiLLM-2 [30] applies SKL to teacher-generated data and SRKL to student-generated data with a curriculum over skew coefficients. Let $D^{(t)}$ and $D^{(s)}$ denote batches of teacher and student responses. The per-step objective is

$$\begin{aligned} \mathcal{L}_{\text{DistiLLM-2}} = (1-\mu)\, \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \in D^{(t)}} \big[D_{\mathrm{SKL}}^{(\alpha_t)}(\boldsymbol{x}, \boldsymbol{y}; p \,\|\, q_\theta) \big] \\ + \; \mu\, \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \in D^{(s)}} \big[D_{\mathrm{SRKL}}^{(\alpha_s)}(\boldsymbol{x}, \boldsymbol{y}; p \,\|\, q_\theta) \big], \end{aligned} \quad (5)$$

where $\mu \in [0, 1]$ trades off teacher vs. student distributions and α_t, α_s are scheduled over training. All objectives are implemented token-wise.

4. Method

In this section, we present **SelecTKD**, our proposed framework designed to address the limitations of uniform, indiscriminate loss application in conventional knowledge distillation, as shown in Figure 2. Drawing inspiration from speculative decoding, SelecTKD introduces a dynamic, token-level verification mechanism that selectively filters the distillation signal, focusing learning on tokens where the student and teacher are sufficiently aligned.

Formally, the SelecTKD loss is defined with a generic token-wise divergence $D(\cdot \| \cdot)$ (e.g., KL/RKL/SKL/SRKL):

$$\mathcal{L}_{\text{SelecTKD}} = \sum_{t=1}^{|\boldsymbol{y}|} V_t \, D\big(p_t \| q_t\big) \quad (6)$$

where $p_t = p(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$ and $q_t = q_\theta(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$ are the teacher and student distributions at position t, and $V_t \in \{0, \beta, 1\}$ is a token-level verification weight determined by a verifier $V(\cdot)$[1]. The core innovation of SelecTKD lies in V_t, which adaptively decides whether to apply the distillation loss at each token based on student-teacher alignment.

[1]Note, $V(\cdot)$ will pass a detach operation to implement the stop-gradient operator.

4.1. Verification Mechanisms in SelecTKD

4.1.1. Hellinger-Based Verification

An intuitive starting point is to compare the teacher and student probability distributions with the Hellinger distance [14]. For a step t, let p_t and q_t denote the teacher and student distributions on $\mathcal{V}$. The Hellinger distance is

$$H(p_t, q_t) = \frac{1}{\sqrt{2}} \big\| \sqrt{p_t} - \sqrt{q_t} \big\|_2, \quad 0 \le H \le 1. \quad (7)$$

It is symmetric and bounded, offering numerical stability. A soft instantiation sets $V_t{=}H(p_t, q_t)$ ("more similar, less learning"), but Hellinger is tail-sensitive and rank-agnostic small low-probability shifts can change H despite agreement on Top-k, misaligning supervision. We therefore adopt a **discrete, rank-based** Top-k verifier that focuses on high-confidence tokens, ignores tail noise and calibration bias, and is computationally simpler. This choice also enables the **Spec-k** variant and aligns with our observations of rising TAR and flatter loss landscapes.

4.1.2. Greedy Top-k Token Verification

To enhance robustness, we propose a discrete, rank-based verification mechanism. At each step t, the student proposes its most likely token:

$$\hat{y}_t = \arg\max_y \, q_\theta(y \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) \quad (8)$$

We then check whether this token is among the teacher's top-k candidates:

$$V_t \;=\; \beta \;+\; (1-\beta)\, \mathbb{I}(\hat{y}_t \in \mathrm{Top}_k(p_t)) \quad (9)$$

where $\mathrm{Top}_k(p_t)$ denotes the set of k tokens with the highest probabilities under the teacher's distribution at position t. When $\beta = 0$, this reduces to hard gating; small $\beta > 0$ provides gentle regularization. This method is robust to the absolute probability values and provides an intuitive knob (k) to control the strictness of verification.

4.1.3. Non-Greedy Spec-k Verification

To further mitigate exposure bias and align with speculative sampling [31], we verify *multiple* student proposals per step with a rejection-sampling style test. At each position t, the student draws k i.i.d. candidates from its own distribution:

$$\tilde{\mathcal{Y}}_t = \big\{ y_t^{(1)}, \ldots, y_t^{(k)} \big\} \sim \text{i.i.d. } q_\theta(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t}). \quad (10)$$

For each sampled token $y_t^{(j)}$, we only query the token and compute the standard speculative-acceptance probability

$$a_t^{(j)} \;=\; \min\left(1, \; \frac{p_t\big(y_t^{(j)}\big)}{q_t\big(y_t^{(j)}\big)}\right) \quad (11)$$

Draw $r^{(j)} \sim \mathrm{U}[0, 1]$ and accept the candidate if $r^{(j)} < a_t^{(j)}$. Let $\mathcal{A}_t = \{ j \in [1..k] \mid r^{(j)} < a_t^{(j)} \}$ be the set of accepted samples. We then define the verification weight:

$$V_t = \beta + (1-\beta)\, \mathbb{I}(|\mathcal{A}_t| \ge 1) \quad (12)$$

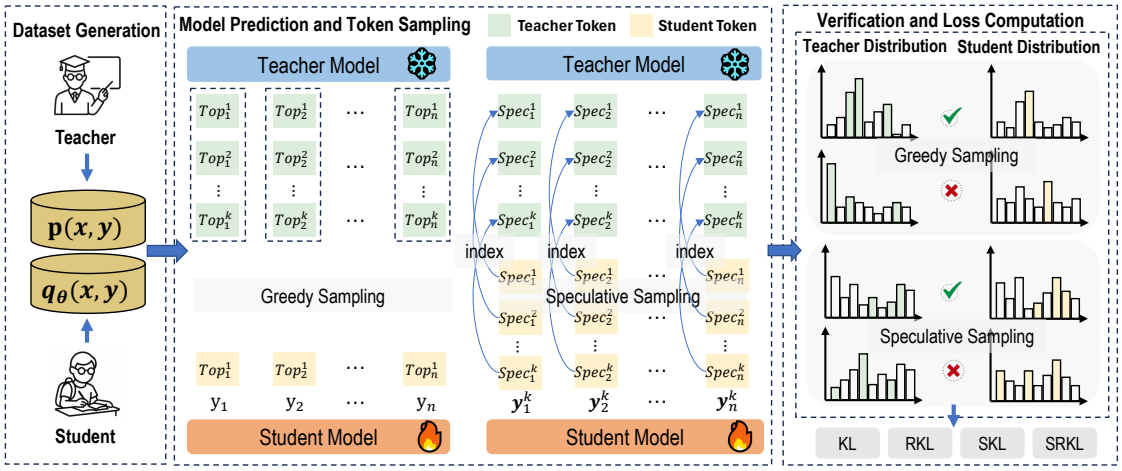


Figure 2. Overview of SelecTKD, a "propose-and-verify" framework for selective token-level knowledge distillation. Left: **Dataset Generation**. Both teacher $p(\boldsymbol{x}, \boldsymbol{y})$ and student $q_\theta(\boldsymbol{x}, \boldsymbol{y})$ can produce responses, enabling on- and off-policy data. Middle: **Model Prediction and Token Sampling**. Two variants are supported: (a) *Greedy Top-k*: the student proposes a greedy token $\hat{y}_t$ (argmax) and the teacher returns its Top-k set; (b) *Non-greedy Spec-k*: the student samples k candidate tokens and the teacher is queried only on these candidates to compute acceptance indices following speculative sampling [6, 31]. Right: **Verification and Loss Computation**. A token is accepted if it is in the teacher's Top-k (greedy) or passes the speculative acceptance test (non-greedy). Accepted tokens receive full loss; rejected tokens are masked or down-weighted by β. The loss is objective-agnostic and works with KL, RKL, SKL, and SRKL. This design focuses learning on reliable, teacher-aligned tokens and yields a stable, curriculum-like training dynamic.

Intuitively, if any of the k proposals are accepted by the teacher under the speculative test, the teacher's local signal is informative and the student should proceed with normal learning; if all are rejected, supervision at this step is likely noisy or misaligned and is therefore down-weighted. This criterion inherits the robustness of speculative sampling, requires k probability lookups at the teacher to score the student's proposals, curriculum-like [39] behavior.

4.2. Token Acceptance Rate (TAR)

A key property of SelecTKD is its stable and convergent training dynamics, which can be quantified by the *Token Acceptance Rate* (TAR):

$$\mathrm{TAR} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\left[\frac{1}{|\boldsymbol{y}|}\sum_{t=1}^{|\boldsymbol{y}|}\mathbb{I}(V_t = 1)\right] \tag{13}$$

TAR measures the expected fraction of tokens for which the student's proposal is accepted by the teacher's verifier (i.e., full-weight tokens).

Theorem 1 (Monotonic TAR Improvement)**.** *Assume (i) the student model satisfies standard Lipschitz continuity conditions [8], ensuring bounded output changes under small parameter updates, and (ii) the learning rate $\eta > 0$ is sufficiently small so that accepted tokens do not revert to rejected status. Then, for both Greedy Top-k and Non-Greedy Spec-k, each gradient step satisfies $\mathrm{TAR}_{t+1} - \mathrm{TAR}_t \geq \eta\,\kappa\,(1-\mathrm{TAR}_t)$, where $\kappa > 0$ depends on the teacher's confidence margin at the Top-k boundary, the model's Lipschitz constant, and the gradient magnitude for tokens near the acceptance threshold (full proof in Appendix E).*

The bound implies fastest improvement when TAR_t is low (many misaligned tokens) and natural saturation as $\mathrm{TAR}_t \to 1$, formalizing the implicit curriculum effect.

The SelecTKD Algorithm. We summarize the full SelecTKD training procedure in Algorithm 1. The core is the dynamic, token-level verification that determines whether to apply the distillation loss at each position. Note that the validation operation for each token position can be batched, and the actual implementation does not require a for loop.

5. Experiments

5.1. Experiment Setup in LLMs

All experiments use the Hugging Face TRL framework with LoRA [20] on 8 NVIDIA A100 80GB GPUs, following the training recipe of DistiLLM-2 [30]. Unless otherwise noted, SelecTKD uses $k = 5$ and $\beta = 0.01$. We run 2 epochs for instruction/code and 1 epoch for math, with early stopping on a held-out validation set. LoRA ranks and target modules are fixed across methods for fairness.

General Instruction-Following. We randomly sample 50k prompts from UltraChat200k [11] to construct the training set. Three teacher-student pairs are evaluated: Qwen2-7B-Instruct [22] → Qwen2-1.5B-Instruct, Mistral-7B-Instruct [41] → Danube2-1.8B [46], and Gemma-2-9B-IT [47] → Gemma-2-2B-IT. Performance is assessed on AlpacaEval [12], Evol-Instruct [52], and UltraFeedback [10],

Algorithm 1 The training process of SelecTKD

Input: Teacher p, Student q_θ (params θ), Top-k or Spec-k value k, rejected-token weight β, verification **mode** $\in$ {`greedy`, `non-greedy`}
Output: Updated student $q_{\hat{\theta}}$
for each batch $(\boldsymbol{x}, \boldsymbol{y})$ in dataset $\mathcal{D}$ **do**
  Initialize total loss $\mathcal{L}_{\text{total}} = 0$
  for each token position $t = 1, \ldots, |\boldsymbol{y}|$ **do**
    $p_t \leftarrow p(\cdot|\boldsymbol{x}, \boldsymbol{y}_{<t})$, $q_t \leftarrow q_\theta(\cdot|\boldsymbol{x}, \boldsymbol{y}_{<t})$
    if mode == `greedy` **then**
      $\hat{y}_t \leftarrow \arg\max_y q_t(y)$
      $V_t \leftarrow \beta + (1-\beta) \cdot \mathbb{I}(\hat{y}_t \in \text{Top}_k(p_t))$
    else if mode == `non-greedy` **then**
      Sample k candidates $\{y_t^{(j)}\}_{j=1}^k \sim$ i.i.d. q_t
      For each j: $a_t^{(j)} \leftarrow \min\left(1, \frac{p_t(y_t^{(j)})}{q_t(y_t^{(j)})}\right)$, draw $r^{(j)} \sim \text{U}[0,1]$
      $\mathcal{A}_t \leftarrow \{j \in [1..k] \mid r^{(j)} < a_t^{(j)}\}$
      $V_t \leftarrow \beta + (1-\beta) \cdot \mathbb{I}(|\mathcal{A}_t| \geq 1)$
    end if
    $L_t \leftarrow V_t \cdot D(p_t \| q_t)$ $\{D \in \{\text{KL}, \text{RKL}, \text{SKL}, \text{SRKL}\}\}$
    $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{total}} + L_t$
  end for
  $\mathcal{L}_{\text{SelecTKD}} \leftarrow \mathcal{L}_{\text{total}} / |\boldsymbol{y}|$
  Update θ using $\nabla_\theta \mathcal{L}_{\text{SelecTKD}}$
end for

following the LLM-as-a-Judge [60] protocol with GPT-4o or GPT-4o-mini as judges to report win rates. We reserve 1k held-out prompts from the 50k for validation and use the remainder for training. For evaluation sets, we follow DistiLLM-2 [30] splits and apply de-duplication against our training sources by prompt hashing.

Mathematical Reasoning. We evaluate on GSM8K [9] and MATH [17] benchmarks using Qwen2-Math-7B-Inst and Qwen2.5-Math-7B-Inst as teachers, and Qwen2-Math-1.5B and Qwen2.5-Math-1.5B as students. Each student is first supervised fine-tuned on the full MetaMathQA dataset for one epoch, then distilled with 50k samples from MetaMathQA [57]. Pass@1 accuracy is used for evaluation.

Code Generation. For code generation, we use WizardCoder prompts (Evol-Instruct) for training. Teacher-student pairs include Qwen2.5-Coder-7B-Inst $\rightarrow$ Qwen2.5-Coder-1.5B and DeepSeek-Coder-6.7B-Inst $\rightarrow$ DeepSeek-Coder-1.3B. Students are trained for 2 epochs and evaluated on HumanEval and MBPP using Pass@1 accuracy.

Reporting. Following DistiLLM-2 [30], all win rates and accuracy scores are averaged over 3 seeds. We report 95% confidence intervals from 10,000 bootstrap resamples stratified by task. For LLM-as-a-Judge, we use randomized, pairwise side-by-side comparisons and tie-aware scoring, excluding ties from win rate calculation. Judge prompts and system messages are fixed across all methods.

Baselines. We compare SelecTKD with a range of representative knowledge distillation methods: **Vanilla KD:** Standard supervised distillation on teacher token distributions. **SeqKD:** Supervised training on full teacher-generated sequences. **ImitKD:** Hybrid on-policy distillation using a mix of ground-truth and student outputs. **GKD:** On-policy distillation with student-generated data and alternative divergences. **DistiLLM:** Framework with skew KLD loss and adaptive off-policy reuse of student data. **SKD:** Speculative decoding-inspired method where the teacher corrects low-quality student tokens to build a better training set. **DistiLLM-2:** Contrastive method applying asymmetric losses (SKL/SRKL) to teacher and student data. **ATKD:** Adaptive teaching-mode distillation that adjusts supervision for easy vs. hard tokens under uncertainty [62]. **AdaSPEC:** Selective alignment via a reference model to filter difficult tokens for speculative decoding [21]. The training strategy for LLMs is consistent with DistiLLM-2 [30].

5.2. Experiment Setup in VLMs

For SelecTKD-VLM-2B (InternVL2-2B+SelecTKD), we trained on the Mantis-Instruct dataset [25]. The training process needs only 3 epochs, using the AdamW (β_1=0.9, β_2=0.95) optimizer with a cosine learning rate scheduler, batch size 128, no warmup. The initial learning rate was set to 8e$-$5. In this vision-language model distillation task, we employed InternVL2-8B [7] as the teacher model and InternVL2-2B as the student model.

5.3. Main Results

5.3.1. General Instruction-Following in LLMs

Table 1 presents the main results on three instruction-following benchmarks. Our proposed SelecTKD framework consistently achieves the highest win rates across all tasks and model pairs, substantially outperforming Vanilla KD and other strong baselines. The improvements are especially pronounced in challenging, open-ended generation scenarios, underscoring the effectiveness of SelecTKD's selective loss filtering in mitigating noisy or uninformative learning signals. Notably, SelecTKD delivers clear gains even when applied on top of advanced distillation methods such as DistiLLM-2 and SKD, demonstrating its generality and robustness for aligning smaller student models with nuanced human preferences. We also compare with two recent token-selective baselines on Evol-Instruct (Qwen2-7B-Inst $\rightarrow$ 1.5B): AdaSPEC [21] obtains 40.21% WR and ATKD [62] obtains 45.13%, while SelecTKD achieves 48.27%, outperforming both without requiring extra reference models.

5.3.2. Mathematical Reasoning in LLMs

As shown in Table 2 (left), SelecTKD delivers consistent and significant improvements on mathematical reasoning benchmarks across all teacher-student setups. Specif-

Table 1. Win rates (WR) on three instruction-following benchmarks. The baseline is `text-davinci-003` for AlpacaEval and `gpt-3.5-turbo` for Evol-Instruct and UltraFeedback. Judging is performed by GPT-4o (AlpacaEval, Evol-Instruct) and GPT-4o-mini (UltraFeedback). '+Ours' denotes SelecTKD.

	AlpacaEval WR(%)	Evol-Instruct WR(%)	UltraFeedback WR(%)	AVG. WR(%)
Qwen2-7B-Inst ($\mathcal{M}_T$) → Qwen2-1.5B ($\mathcal{M}_S$)				
$\mathcal{M}_T$	88.41	70.70	69.25	76.12
$\mathcal{M}_S$	51.06	18.00	21.93	30.33
KD	57.49	28.23	37.86	41.19
+Ours	59.72 (+2.23)	30.35 (+2.12)	39.93 (+2.07)	43.33 (+2.14)
SeqKD	58.02	29.11	38.35	41.83
+Ours	60.16 (+2.14)	31.18 (+2.07)	40.77 (+2.42)	44.04 (+2.21)
ImitKD	59.37	30.58	39.92	43.29
+Ours	61.16 (+1.79)	32.30 (+1.72)	41.33 (+1.41)	44.93 (+1.64)
GKD	66.07	44.61	57.74	56.14
+Ours	66.95 (+0.88)	45.73 (+1.12)	58.61 (+0.87)	57.10 (+0.96)
DistiLLM	66.30	44.61	58.18	56.36
+Ours	67.21 (+0.91)	45.72 (+1.11)	58.62 (+0.44)	57.18 (+0.82)
SKD	61.52	44.95	56.82	54.43
+Ours	63.19 (+1.67)	45.97 (+1.02)	57.96 (+1.14)	55.71 (+1.28)
DistiLLM-2	69.88	47.13	59.05	58.69
+Ours	**70.60** (+0.72)	**48.27** (+1.14)	**59.53** (+0.48)	**59.47** (+0.78)
Mistral-7B-Inst ($\mathcal{M}_T$) → Danube2-1.8B ($\mathcal{M}_S$)				
$\mathcal{M}_T$	91.92	73.51	83.59	83.01
$\mathcal{M}_S$	48.17	12.84	20.06	27.02
KD	60.21	18.23	41.56	40.00
+Ours	62.56 (+2.35)	20.21 (+1.98)	43.60 (+2.04)	42.12 (+2.12)
SeqKD	59.76	18.45	42.11	40.11
+Ours	62.31 (+2.55)	20.44 (+1.99)	43.89 (+1.78)	42.21 (+2.10)
ImitKD	58.34	17.89	40.87	39.03
+Ours	60.90 (+2.56)	20.54 (+2.65)	43.45 (+2.58)	41.63 (+2.60)
GKD	69.75	24.54	57.74	50.68
+Ours	70.40 (+0.65)	25.82 (+1.28)	58.33 (+0.59)	51.52 (+0.84)
DistiLLM	70.16	28.78	58.18	52.37
+Ours	71.05 (+0.89)	29.95 (+1.17)	59.11 (+0.93)	53.37 (+1.00)
SKD	64.58	38.87	60.04	54.50
+Ours	66.56 (+1.98)	**38.96** (+0.09)	60.95 (+0.91)	55.49 (+0.99)
DistiLLM-2	74.04	32.84	62.46	56.45
+Ours	**74.69** (+0.65)	33.78 (+0.94)	**63.03** (+0.57)	**57.17** (+0.72)
Gemma-2-9B-Inst ($\mathcal{M}_T$) → Gemma-2-2B ($\mathcal{M}_S$)				
$\mathcal{M}_T$	95.78	88.76	85.90	90.15
$\mathcal{M}_S$	42.51	16.74	26.60	28.62
KD	61.78	32.45	54.37	49.53
+Ours	64.49 (+2.71)	35.90 (+3.45)	56.79 (+2.42)	52.39 (+2.86)
SeqKD	62.43	33.21	55.18	50.27
+Ours	65.14 (+2.71)	36.58 (+3.37)	57.76 (+2.58)	53.16 (+2.89)
ImitKD	63.12	31.89	53.92	49.64
+Ours	65.48 (+2.36)	35.32 (+3.43)	57.02 (+3.10)	52.61 (+2.97)
GKD	81.43	50.57	77.20	69.73
+Ours	82.37 (+0.94)	51.91 (+1.34)	78.91 (+1.71)	71.06 (+1.33)
DistiLLM	82.95	51.26	76.68	70.30
+Ours	84.08 (+1.13)	52.44 (+1.18)	77.83 (+1.15)	71.45 (+1.15)
SKD	78.45	57.11	72.21	69.26
+Ours	79.93 (+1.48)	58.53 (+1.42)	73.70 (+1.49)	70.72 (+1.46)
DistiLLM-2	85.97	59.53	78.99	74.83
+Ours	**86.96** (+0.99)	**60.40** (+0.87)	**79.82** (+0.83)	**75.73** (+0.90)

ically, with Qwen2-Math-7B-Inst as teacher and 1.5B as student, SelecTKD improves Pass@1 on GSM8K from 76.27 to 77.82 (+1.55), and on MATH from 35.58 to 36.89 (+1.31), resulting in an overall average gain from 55.93 to 57.36 (+1.43) compared to DistiLLM-2. Similar trends are observed with Qwen2.5-Math-7B-Inst, where SelecTKD raises the GSM8K Pass@1 from 81.20 to 82.45 (+1.25) and MATH from 42.94 to 43.92 (+0.98), achieving an average improvement of +1.12. Notably, this consistent gain is observed under multiple baselines (e.g., GKD, DistiLLM), highlighting that the implicit curriculum and token filtering of SelecTKD allow the student to focus on high-confidence, learnable tokens. This targeted supervision not only prevents the student from overfitting to noisy signals but also leads to improved multi-step reasoning capabilities, especially on challenging tasks like MATH.

5.3.3. Code Generation in LLMs

Table 2 (right) summarizes results on code generation benchmarks. SelecTKD achieves the highest Pass@1 accuracy on both HumanEval and MBPP for all considered model pairs. For example, using DS-Coder-6.9B-Inst and Qwen2.5-Coder-7B-Inst as teachers, SelecTKD brings HumanEval Pass@1 from 59.92 to 61.72 (+1.80) and MBPP from 75.66 to 76.27 (+0.61) for DistiLLM-2, with an overall average increase from 67.79 to 69.00 (+1.21). Similarly, on Qwen2.5-Coder-7B-Inst, HumanEval improves from 42.24 to 43.63 (+1.39), and MBPP from 62.70 to 64.32 (+1.62). These results demonstrate the superiority of selective loss masking for code tasks, where precise syntax and logical accuracy are critical. By focusing optimization on confident teacher predictions, SelecTKD facilitates more robust and stable acquisition of programming structures, surpassing all previous baselines on Pass@1.

5.3.4. Comprehensive Evaluation in VLMs

To further validate SelecTKD, we extend it to vision-language models by distilling InternVL2-2B [7] and evaluating the resulting SelecTKD-VLM-2B on the Open VLM Leaderboard [42]. As summarized in Table 3, SelecTKD-VLM-2B surpasses all existing models up to 4B parameters, outperforming larger models like Phi-3-Vision (4.2B) and TAID-VLM-2B. Notably, it achieves top or near-top accuracy on nearly all benchmarks, especially excelling on MMStar and MMBench_V11. These results demonstrate the scalability and effectiveness of SelecTKD for compact, high-performing VLMs, highlighting its strong multimodal knowledge transfer capabilities.

5.4. Ablation Studies

5.4.1. Effect of the Verification Mechanisms

We conduct a comprehensive ablation to evaluate the effect of different verification mechanisms within the SelecTKD framework, as summarized in Table 4. Compared to the non-selective Vanilla KD baseline, all selective schemes yield superior average win rates, demonstrating the crucial role of targeted loss application. Among the studied approaches, token-based selection strategies provide more robust improvements over the Hellinger-based criterion. Notably, the Non-Greedy Spec-k strategy consistently achieves the highest win rates, suggesting that allowing for flexible acceptance rather than strict selection can further

Table 2. Combined results on Math and Code benchmarks. The best **Pass@1** score is highlighted in bold. '+Ours' denotes SelecTKD.

Method	Math Benchmarks						Code Benchmarks					
	Qwen2-Math-7B-Inst → 1.5B			Qwen2.5-Math-7B-Inst → 1.5B			DS-Coder-6.9B-Inst → 1.3B			Qwen2.5-Coder-7B-Inst → 1.5B		
	GSM8K Pass@1	MATH Pass@1	AVG. Pass@1	GSM8K Pass@1	MATH Pass@1	AVG. Pass@1	HEval Pass@1	MBPP Pass@1	AVG. Pass@1	HEval Pass@1	MBPP Pass@1	AVG. Pass@1
$\mathcal{M}_T$	83.93	41.28	62.61	89.31	44.82	67.07	85.37	82.54	83.96	75.61	74.60	75.11
$\mathcal{M}_S$	74.53	25.56	50.05	77.33	27.14	52.24	50.61	72.22	61.42	30.73	60.84	45.79
GKD	75.44	34.16	54.80	80.21	40.54	60.38	54.88	74.34	64.61	40.85	61.90	51.38
+Ours	76.86	35.43	56.15	82.10	42.31	62.21	56.23	76.08	66.16	42.32	63.67	53.00
DistiLLM	75.59	34.54	55.07	81.05	41.14	61.10	53.65	74.34	64.00	39.63	62.17	50.90
+Ours	77.38	35.92	56.65	82.23	42.20	62.22	55.18	75.92	65.55	41.61	63.89	52.75
DistiLLM-2	76.27	35.58	55.93	81.20	42.94	62.07	59.92	75.66	67.79	42.24	62.70	52.47
+Ours	**77.82**	**36.89**	**57.36**	**82.45**	**43.92**	**63.19**	**61.72**	**76.27**	**69.00**	**43.63**	**64.32**	**53.98**

Table 3. Performance of SelecTKD-VLM-2B, our new state-of-the-art VLM for models up to 4B parameters.

Model	MMBench_V11	MMStar	MMMU_VAL	MathVista	OCRBench	AI2D	HallusionBench	MMVet	Average
PaliGemma-3B-mix-448 [4]	65.6	48.3	34.9	28.7	61.4	68.3	32.2	33.1	46.6
MiniCPM-V-2 [56]	65.8	39.1	38.2	39.8	60.5	62.9	36.1	41.0	47.9
Phi-3-Vision [1]	65.2	47.7	**46.1**	44.6	63.7	**78.4**	39.0	44.1	53.6
InternVL2-2B [7]	69.6	49.8	36.3	46.0	78.1	74.1	38.0	39.7	54.0
TAID-VLM-2B [45]	70.7	49.5	35.1	51.6	78.6	74.0	**56.8**	35.1	56.4
SelecTKD-VLM-2B(Ours)	**71.3**	**50.2**	39.8	**52.6**	**79.5**	77.3	46.5	**45.8**	**57.9**

enhance student learning by embracing a broader range of valid learning signals.

5.4.2. Impact of Hyperparameters k and β

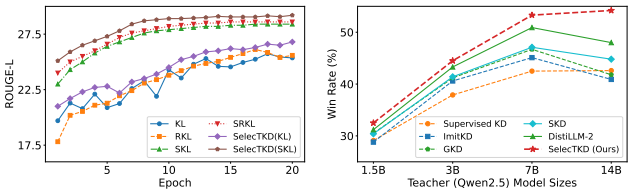
We further analyze the influence of the key hyperparameters k (the number of candidates) and β (the weighting for rejected tokens). As reported in Table 4, performance is not highly sensitive within a reasonable range of k, but peaks at $k = 5$. Adjusting β reveals that while hard gating ($\beta = 0$) is already effective, introducing a small non-zero value ($\beta = 0.01$) consistently improves results. This indicates that weakly training on rejected tokens serves as a beneficial regularizer, likely providing implicit exposure to valuable negative information. Conversely, increasing β further erodes the advantages of selective learning and causes performance to approach that of Vanilla KD. Overall, the best results are obtained with $k = 5$ and $\beta = 0.01$.

5.4.3. On the Role of Loss Function Geometry

To investigate whether loss function geometry is the dominant factor in distillation, we experiment with various commonly used distillation losses within the SelecTKD framework. All models are trained on the same mixed dataset generated offline by both teacher and student (see Figure 3 (a)). We observe that while different losses may converge at different rates, their final performance is similar when training is sufficiently long. This supports our hypothesis that, with selective loss application, the choice of loss function is less critical than the control over where and when the learning signal is applied. Losses incorporating Skew KL consistently outperform others, in line with prior findings.

Table 4. Comprehensive ablation on verification mechanisms, selection types, and key hyperparameters k and β for SelecTKD. Results are reported as average win rate (%) on instruction-following benchmarks (Qwen2-7B-Inst → Qwen2-1.5B).

Method	Verification	k	β	Avg. Win Rate (%)
Vanilla KD	None	–	1	41.19
SelecTKD (H)	Hellinger-based	–	–	41.70
SelecTKD (G)	Greedy Top-k	3	0.01	42.98
SelecTKD (G)	Greedy Top-k	5	0.01	43.09
SelecTKD (G)	Greedy Top-k	7	0.01	43.04
SelecTKD (G)	Greedy Top-k	5	0	42.27
SelecTKD (NG)	Non-Greedy Spec-k	3	0.01	43.22
SelecTKD (NG)	Non-Greedy Spec-k	5	0.01	**43.33**
SelecTKD (NG)	Non-Greedy Spec-k	7	0.01	43.26
SelecTKD (NG)	Non-Greedy Spec-k	10	0.01	43.13
SelecTKD (NG)	Non-Greedy Spec-k	5	0.1	43.20
SelecTKD (NG)	Non-Greedy Spec-k	5	0.05	43.18
SelecTKD (NG)	Non-Greedy Spec-k	5	0	42.83



(a) (b)

Figure 3. Performance analysis of SelecTKD in Qwen2-1.5B: (a) Comparison of different distillation methods during training; (b) Robustness of SelecTKD to increasingly larger teachers.

5.4.4. Mitigating the Curse of the Powerful Teacher

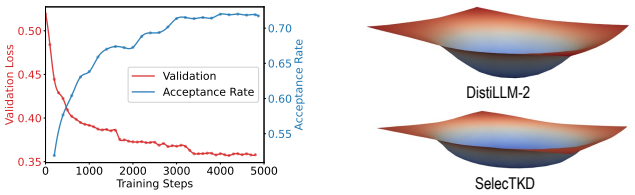
A well-known challenge in knowledge distillation is that using a much larger teacher can sometimes degrade student

performance, as noted by [62]. We hypothesize this is due to the higher-entropy, more complex distributions produced by large teachers, which can overwhelm a capacity-limited student. SelecTKD addresses this by filtering out such noisy signals. To validate this, we fix the student (Qwen2-1.5B) and distill from teachers of increasing size. As shown in Figure 3 (b), Vanilla KD performance plateaus and then drops as teacher size increases, confirming the "curse of the powerful teacher." In contrast, SelecTKD enables the student to benefit monotonically from stronger teachers, highlighting the importance of selective loss for robust knowledge transfer from frontier models.

5.5. Discussion of SelecTKD Optimization

5.5.1. Connection to Implicit Curriculum Learning

SelecTKD naturally induces an implicit, adaptive curriculum [39]. As training progresses, the Token Acceptance Rate (TAR) increases quasi-monotonically, see Figure 4 (a), indicating that the student is mastering "easy" tokens first. As the model improves, more challenging tokens are gradually included in the loss, automatically adjusting the learning difficulty to the student's current capability. This self-paced curriculum eliminates the need for manual scheduling and contributes to stable and efficient training.



(a) (b)

Figure 4. Analysis of SelecTKD dynamics: (a) validation loss generally decreases as the Token Acceptance Rate (TAR) increases, indicating an implicit curriculum; (b) SelecTKD yields a flatter loss landscape [32] than strong baselines (e.g., DistiLLM-2), which correlates with improved generalization.

5.5.2. Impact on Loss Landscape and Generalization

By masking loss contributions from tokens with high teacher-student divergence, SelecTKD smooths the loss landscape and mitigates the effect of sharp, noisy gradients. As shown in Figure 4 (b), SelecTKD yields a flatter loss surface compared to strong baselines such as DistiLLM-2. A smoother loss landscape is empirically linked to better generalization [24], which aligns with the robust performance improvements observed in our experiments.

5.5.3. Inference Speedup of Speculative Decoding

We further assess SelecTKD by measuring its impact on speculative decoding, where effective alignment between drafter and verifier models is essential for fast and efficient generation [64]. Table 5 summarizes results using Llama-68m [40] as drafter and Phi-3-medium or Phi-3.5-mini [1] as verifiers. All draft models share the same training data but are distilled via SFT, GKD, DistiLLM, or SelecTKD. We report both average speedup and token acceptance rate as key metrics. Across both verifier settings, SelecTKD consistently achieves the best speedup and acceptance rates—for example, with Phi-3-medium, SelecTKD yields $\times$2.05 speedup and 0.492 acceptance, outperforming other strong baselines, and similar gains are seen with Phi-3.5-mini. These improvements demonstrate that SelecTKD's selective token alignment produces better-matched draft-verifier distributions, directly leading to higher speculative decoding efficiency without compromising acceptance. Notably, the higher acceptance rates attained with SelecTKD indicate improved token-level agreement, implying a narrower drafter-verifier distribution gap compared to prior methods [64]. Thus, targeted token selection and weighting in distillation enable both theoretical and practical advances for inference with modern language models.

Table 5. Inference speedup (Spd.) and token acceptance rate (Acpt.) in the speculative decoding setting. Results compare different draft models trained via various KD methods for both Phi-3-medium and Phi-3.5-mini verifiers.

Phi-		SFT	GKD	DistiLLM	DistiLLM-2	SelecTKD
3-medium	Spd. (↑)	$\times$1.32	$\times$1.64	$\times$1.71	$\times$1.97	$\times$**2.05**
	Acpt. (↑)	0.412	0.464	0.469	0.487	**0.492**
3.5-mini	Spd. (↑)	$\times$1.24	$\times$1.58	$\times$1.65	$\times$1.84	$\times$**1.91**
	Acpt. (↑)	0.397	0.443	0.452	0.522	**0.534**

6. Conclusion

We identify a central limitation of conventional LLM distillation: uniform, indiscriminate token-wise supervision that forces students to imitate high-entropy or noisy teacher predictions, which is especially detrimental when there are large capacity gaps. We introduce SelecTKD, a plug-and-play framework using a simple Top-k propose-and-verify token mechanism via greedy and non-greedy Spec-k variants to determine where to apply learning. Accepted tokens receive full loss, while rejected tokens are masked or weakly weighted. SelecTKD is objective-agnostic (KL, RKL, SKL, SRKL) and compatible with on- and off-policy data. Across instruction following, mathematical reasoning, code generation, and a VLM task, SelecTKD consistently improves strong baselines without architectural changes or extra reference models. Analyses of Token Acceptance Rate (TAR) and the resulting smoother loss landscape corroborate an implicit curriculum and stable optimization.

Acknowledgement

This work was supported in part by Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry

of Education of China under Grant JYB2025XDXM504, and National Natural Science Foundation of China No.62302381, No.52441602. The Authors are with the National Key Laboratory of Human-Machine Hybrid Augmented Intelligence and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi, China.

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 7, 8

[2] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[3] Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. *arXiv preprint arXiv:2204.01171*, 2022. 1

[4] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 7

[5] Di Chai, Pengbo Li, Feiyuan Zhang, Yilun Jin, Han Tian, Junxue Zhang, and Kai Chen. Enhancing token filtering efficiency in large language model training with collider. *arXiv preprint arXiv:2502.00340*, 2025. 1

[6] Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023. 1, 4

[7] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 5, 6, 7

[8] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, pages 854–863. PMLR, 2017. 4, 2

[9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 5

[10] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback. In *International Conference on Machine Learning*, pages 9722–9744, 2024. 4

[11] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. 4

[12] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024. 4

[13] Yanjun Fu, Faisal Hamman, and Sanghamitra Dutta. T-shirt: Token-selective hierarchical data selection for instruction tuning. *arXiv preprint arXiv:2506.01317*, 2025. 2, 1

[14] Víctor González-Castro, Rocío Alaiz-Rodríguez, and Enrique Alegre. Class distribution estimation based on the hellinger distance. *Information Sciences*, 218:146–164, 2013. 3

[15] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. 1, 2, 3

[16] Jitai Hao, Qiang Huang, Hao Liu, Xinyan Xiao, Zhaochun Ren, and Jun Yu. A token is worth over 1,000 tokens: Efficient knowledge distillation through low-rank clone. In *Advances in Neural Information Processing Systems*, 2025. 1

[17] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 5

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2

[19] Le Hou, Richard Yuanzhe Pang, Tianyi Zhou, Yuexin Wu, Xinying Song, Xiaodan Song, and Denny Zhou. Token dropping for efficient BERT pretraining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 3774–3784, 2022. 1

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 4

[21] Yuezhou Hu, Jiaxin Guo, Xinyu Feng, and Tuo Zhao. Adaspec: Selective knowledge distillation for efficient speculative decoders. In *Advances in Neural Information Processing Systems*, 2025. 1, 2, 5

[22] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024. 4

[23] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 1

[24] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights

leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 8

[25] Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhu Chen. Mantis: Interleaved multi-image instruction tuning. *arXiv preprint arXiv:2405.01483*, 2024. 5

[26] Seongryong Jung, Suwan Yoon, DongGeon Kim, and Hwanhee Lee. Todi: Token-wise distillation via fine-grained divergence control. *arXiv preprint arXiv:2505.16297*, 2025. 1

[27] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

[28] Gyeongman Kim, Doohyuk Jang, and Eunho Yang. Promptkd: Distilling student-friendly knowledge for generative language models via prompt tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. 1

[29] Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. In *International Conference on Machine Learning*, pages 24872–24895, 2024. 1, 2, 3, 4

[30] 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. *arXiv preprint arXiv:2503.07067*, 2025. 1, 2, 3, 4, 5

[31] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR, 2023. 1, 3, 4

[32] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018. 8

[33] Tianjian Li, Haoran Xu, Philipp Koehn, Daniel Khashabi, and Kenton Murray. Error norm truncation: Robust training in the presence of data noise for text generation models. *arXiv preprint arXiv:2310.00840*, 2023. 2, 1

[34] Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6121–6133, 2020. 1, 2

[35] Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, et al. Rho-1: Not all tokens are what you need. In *Advances in Neural Information Processing Systems*, 2024. 2, 1

[36] Chang Liu, Chongyang Tao, Jiazhan Feng, and Dongyan Zhao. Multi-granularity structural knowledge distillation for language model compression. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1001–1011, 2022. 1

[37] Chengyuan Liu, Fubang Zhao, Kun Kuang, Yangyang Kang, Zhuoren Jiang, Changlong Sun, and Fei Wu. Evolving knowledge distillation with large language models and active learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6717–6731, 2024. 1

[38] Xiaoyu Liu, Yun Zhang, Wei Li, Simiao Li, Xudong Huang, Hanting Chen, Yehui Tang, Jie Hu, Zhiwei Xiong, and Yunhe Wang. Multi-granularity semantic revision for large language model distillation. *arXiv preprint arXiv:2407.10068*, 2024. 1

[39] Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, Yong Huang, and Wei Lu. Let's learn step by step: Enhancing in-context learning ability with curriculum learning. *arXiv preprint arXiv:2402.10738*, 2024. 4, 8

[40] Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, et al. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pages 932–949, 2024. 8

[41] Mistral AI. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 4

[42] OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023. 6

[43] Jinlong Pang, Na Di, Zhaowei Zhu, Jiaheng Wei, Hao Cheng, Chen Qian, and Yang Liu. Token cleaning: Fine-grained data selection for llm supervised fine-tuning. *arXiv preprint arXiv:2502.01968*, 2025. 2, 1

[44] Mrigank Raman, Pranav Mani, Davis Liang, and Zachary Lipton. For distillation, tokens are not all you need. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023. 1

[45] Makoto Shing, Kou Misaki, Han Bao, Sho Yokoi, and Takuya Akiba. Taid: Temporally adaptive interpolated distillation for efficient knowledge transfer in language models. In *International Conference on Learning Representations*, 2025. 1, 7

[46] Philipp Singer, Pascal Pfeiffer, Yauhen Babakhin, Maximilian Jeblick, Nischay Dhankhar, Gabor Fodor, and Sri Satish Ambati. H2o-danube-1.8 b technical report. *arXiv preprint arXiv:2401.16818*, 2024. 4

[47] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024. 4

[48] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. 1

[49] Jingxuan Wei, Linzhuang Sun, Yichong Leng, Xu Tan, Bihui Yu, and Ruifeng Guo. Sentence-level or token-level? a com-

prehensive study on knowledge distillation. *arXiv preprint arXiv:2404.14827*, 2024. 1

[50] Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 10817–10834, 2023. 1

[51] Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking kullback-leibler divergence in knowledge distillation for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, 2025. 1, 2

[52] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*, 2024. 4

[53] Wenda Xu, Rujun Han, Zifeng Wang, Long T Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, 2025. 1

[54] Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*, 2024. 1

[55] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 1

[56] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 7

[57] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, 2024. 5

[58] Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. Dual-space knowledge distillation for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. 1

[59] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11953–11962, 2022. 1

[60] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023. 5

[61] Qihuang Zhong, Liang Ding, Juhua Liu, Xuebo Liu, Min Zhang, Bo Du, and Dacheng Tao. Revisiting token dropping strategy in efficient BERT pretraining. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023. 1

[62] Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. Revisiting knowledge distillation for autoregressive language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 10900–10913, 2024. 2, 5, 8, 1

[63] Qinhong Zhou, Peng Li, Yang Liu, Yuyang Guan, Qizhou Xing, Ming Chen, and Maosong Sun. Adads: Adaptive data selection for accelerating pre-trained language model knowledge distillation. *AI Open*, 4:56–63, 2023. 1

[64] Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. Distillspec: Improving speculative decoding via knowledge distillation. *arXiv preprint arXiv:2310.08461*, 2023. 8