Learning from Disagreement: Multi-Teacher Distillation for Chinese Spelling Correction

Hongcheng Ding[1] **Ziping Hu**[2,3] **Hongdan Xiao**[3]
Kang Xu[4] **Weiyu Zhang**[4] **Xuanze Zhao**[1*]
[1]Dongfang College, Zhejiang University of Finance and Economics
[2]CangZhou Hospital of Integrated Traditional Chinese and Western Medicine in Hebei Province
[3]INTI International University [4]Zhejiang Gongshang University

Abstract

Chinese Spelling Correction (CSC) is hard precisely because a single misspelling can originate from any of four loosely coupled dimensions (glyph, phonetic, syntactic, or semantic confusion), yet existing systems treat the four as a flat fusion target. Multimodal pre-training and fixed-weight gating let evidence from each dimension co-exist, but never arbitrate: when a phonetic-plausible candidate contradicts a syntactic-plausible one, the model averages them, and learning effort is spent uniformly on cases where every dimension already agrees. Our diagnosis is that the bottleneck is no longer representation but *decision*: the cases where dimensions disagree are exactly the cases worth specializing for, and dimension-specific teachers expose this disagreement as a usable signal rather than as noise to be smoothed away. We instantiate this view as SEMTD (Self-Evolving Multi-Teacher Distillation), a 4B-parameter student trained from four role-prompted single-dimension experts (glyph / phonetic / syntactic / semantic) under three coupled objectives: multi-teacher distillation with input-dependent expert selection, self-evolving learning that turns disagreement into a composite reward, and reflective path optimization that revisits high-disagreement decisions. Across four benchmarks the student reaches Cor-F1 of 83.8 on SIGHAN15, 62.3 on LEMON (7-domain average), 98.2 on ECSpell (3-domain average), and 73.0 on CSCD-NS, within 0.8 of a 14B single-model LLM corrector at $3.5\times$ fewer parameters. The takeaway is methodological: in CSC, treating teacher disagreement as the supervision target, rather than a residual to be averaged out, reopens headroom that flat fusion has saturated.

1 Introduction

CSC sits on a path that every downstream Chinese NLP system traverses: search, dialogue, education, OCR post-processing, and clinical text all degrade visibly when upstream errors slip through [75, 32]. What makes CSC unusually hard among text-correction tasks is structural rather than statistical: each Chinese character simultaneously carries glyph, phonetic, syntactic, and semantic information, and a single misspelling can originate from any of them, often through interactions [33, 75, 27]. A corrector strong on visually similar errors can fail silently on phonetic confusions; one that handles homophones can miss grammatically dependent substitutions. The task is therefore four overlapping problems sharing one input, not one with four feature views.

Existing approaches accumulate capacity along this dimensional structure but do not exploit its decision structure. One line of work models semantics through pre-trained language models [74]; later families extended representations to phonetic and glyph features through specialized pre-training and graph encoders [8, 32, 46, 78]; multimodal methods fuse semantic, phonetic, and glyph

*Corresponding author: 20070032@zufedfc.edu.cn (Xuanze Zhao).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

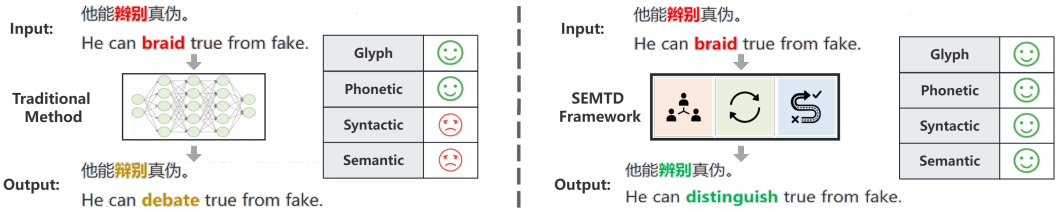


Figure 1: Comparison of different CSC methods. **(A)** Traditional methods produce incorrect corrections. **(B)** Our proposed framework integrates multi-perspective teachers with self-evolving learning and reflective path optimization, achieving linguistically coherent corrections across four dimensions.

evidence [61, 25]. Three mechanism-level limitations recur across this trajectory. First, syntactic information remains under-modeled, so grammatically dependent errors fall outside the fused feature space (Figure 1A) [22, 56]. Second, fusion is passive: when phonetic evidence and semantic evidence point to different characters, the fused representation blends them rather than arbitrating between them. Third, training treats every sentence as equally informative, so capacity is spent re-confirming easy cases instead of resolving the hard ones [30]. The first two limit what the model can express; the third limits what its training signal can distinguish, so hard cases get no more weight than easy ones.

Our central observation: these failures collapse once the task is reframed. If the four dimensions are made explicit as separate teachers, the cases where teachers disagree are precisely the cases that are syntactically dependent, multimodally ambiguous, or otherwise worth allocating capacity to. Disagreement is not noise to smooth out, but the supervision signal that flat fusion has been throwing away. The central claim that follows is narrow and testable: a student that learns to arbitrate among dimension-specialized teachers, with its training distribution and reward both shaped by their disagreement, recovers headroom that representational fusion alone cannot reach. The same mechanism flags hard cases during training and steers both sampling and reward toward them.

We propose SEMTD, a collective-reinforcement framework whose three components are coupled by construction and train a 4B student to act as the arbiter rather than the average of four single-dimension teachers (Figure 1B). Multi-teacher distillation instantiates the four experts (glyph, phonetic, syntactic, semantic) and learns input-dependent weights that decide which teacher to trust per input. Self-evolving learning quantifies inter-teacher disagreement and converts it into a composite reward, so the student is supervised by conflict rather than by a flattened consensus. Reflective path optimization revisits high-disagreement decision paths and consolidates the successful ones, so hard cases compound across samples instead of fading. Our key insight is that teacher disagreement is itself a per-sample supervision signal, telling the student both when to arbitrate independently and which expert to trust. The contributions of this work are:

- We identify multi-dimensional CSC as a decision problem rather than a representation problem: once dimension-specialized teachers exist, their disagreement is the supervision signal that flat fusion discards, and the missing capability is arbitration rather than richer fused representation.
- We propose SEMTD, a coupled three-component framework in which multi-teacher distillation learns input-dependent weights for expert selection, self-evolving learning converts inter-teacher conflict into a composite reward, and reflective path optimization revisits high-disagreement decision paths so that hard cases compound across samples.
- SEMTD achieves the best Cor-F1 among published methods on SIGHAN, LEMON, and ECSpell, and comes within 0.8 of a 14B single-model LLM corrector on CSCD-NS at $3.5\times$ fewer parameters: a 4B student reaches Cor-F1 of 83.8 on SIGHAN15, 62.3 on LEMON's 7-domain average, 98.2 on ECSpell's 3-domain average, and 73.0 on CSCD-NS.

2 Motivation

Multi-Dimensional Errors. Chinese spelling errors exhibit intrinsic multi-dimensionality across glyph, phonetic, syntactic, and semantic dimensions, as illustrated in Figure 2 (left) and detailed in Appendix A.1. Each error type demands distinct expertise for accurate correction. Although recent methods adaptively fuse glyph, phonetic, and semantic features, syntactic knowledge remains underexplored, and no single dimension suffices for all cases [27, 19, 53]. This naturally calls for

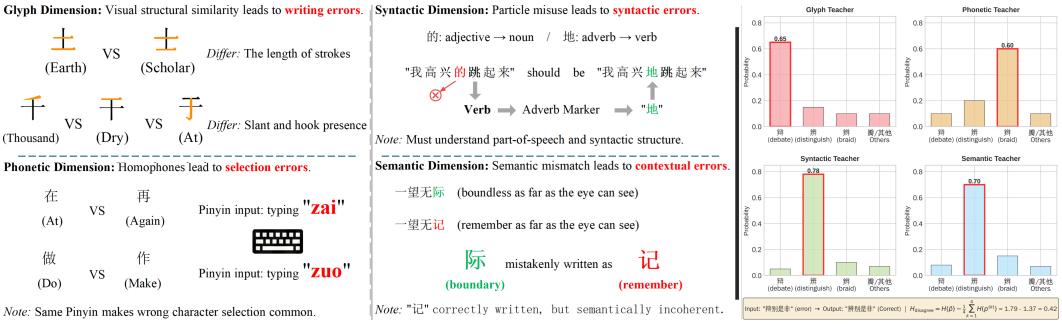


Figure 2: **(Left)** Chinese spelling errors across glyph, phonetic, syntactic, and semantic dimensions. **(Right)** Multi-teacher disagreement where different teachers produce conflicting predictions.

multiple teachers specialized in each dimension, yet raises a fundamental challenge of *whom to trust*: the appropriate expert must be selected based on the characteristics of each input.

Multi-Teacher Disagreement. While multi-teacher architectures provide specialized expertise for each dimension, they introduce teacher disagreement [55, 76]. As illustrated in Figure 2 (right), different teachers often produce conflicting predictions for the same input. For the input shown there, the glyph teacher keeps the erroneous character, glossed "debate" (probability 0.65), and the phonetic teacher prefers a homophone of that character, glossed "braid" (0.60), while the syntactic and semantic teachers prefer the correct "distinguish" (0.78 and 0.70; Appendix A.2.1). Naive solutions such as fixed-weight aggregation or majority voting fail to resolve such conflicts [80, 70], as they either dilute specialized expertise or discard valuable minority opinions.

Conventional knowledge distillation assumes passive knowledge absorption from a single teacher [15]. However, multi-teacher scenarios demand a different paradigm, giving rise to two further challenges [11, 71]. The first is *how to decide*: when teachers disagree, the student must form independent judgment rather than passively follow any single teacher. The second is *what to learn*: high-disagreement samples are inherently more informative, requiring mechanisms to prioritize such difficult samples while maintaining stable decision paths [37, 82].

3 Problem Formulation

We formulate the challenges identified in Section 2 as a policy optimization problem within a multi-teacher reinforcement learning framework.

Given K expert teachers $\{T_k\}_{k=1}^{K}$ specialized in distinct linguistic dimensions and a student policy π_θ, our goal is to train a student that can actively integrate diverse teacher feedback while developing independent judgment capability. We express this as maximizing expected composite reward:

$$\max_{\Theta} \; \mathbb{E}_{\mathbf{x}\sim P_\phi}\left[\sum_{k=1}^{K} w_\omega^{(k)}(\mathbf{x}) \cdot \mathbb{E}_{\hat{\mathbf{y}}\sim\pi_\theta}\left[R_k(\mathbf{x},\hat{\mathbf{y}})\right]\right], \tag{1}$$

where $\Theta = \{\theta, \omega, \phi\}$ denotes the set: student policy parameters θ, adaptive weight network parameters ω, and sampling distribution parameters ϕ. The adaptive weights $w_\omega(\mathbf{x}) \in \Delta^K$ determine each teacher's contribution, $R_k(\mathbf{x}, \hat{\mathbf{y}})$ is the reward signal from teacher k, and the sampling distribution $P_\phi(\mathbf{x}) \propto \exp(\eta \cdot \mathcal{U}(\mathbf{x}))$ prioritizes informative samples based on uncertainty $\mathcal{U}(\mathbf{x})$.

The three terms of Eq. (1) instantiate the challenges in Section 2: the outer expectation $\mathbb{E}_{\mathbf{x}\sim P_\phi}$ realizes *what to learn* by prioritizing informative samples through P_ϕ; the weighted aggregation $\sum_{k=1}^{K} w_\omega^{(k)}(\mathbf{x})$ realizes *whom to trust* by adaptively combining teacher feedback according to input-dependent weights; and the inner expectation $\mathbb{E}_{\hat{\mathbf{y}}\sim\pi_\theta}[R_k(\mathbf{x},\hat{\mathbf{y}})]$ realizes *how to decide* by optimizing the student's prediction policy under each teacher's reward.

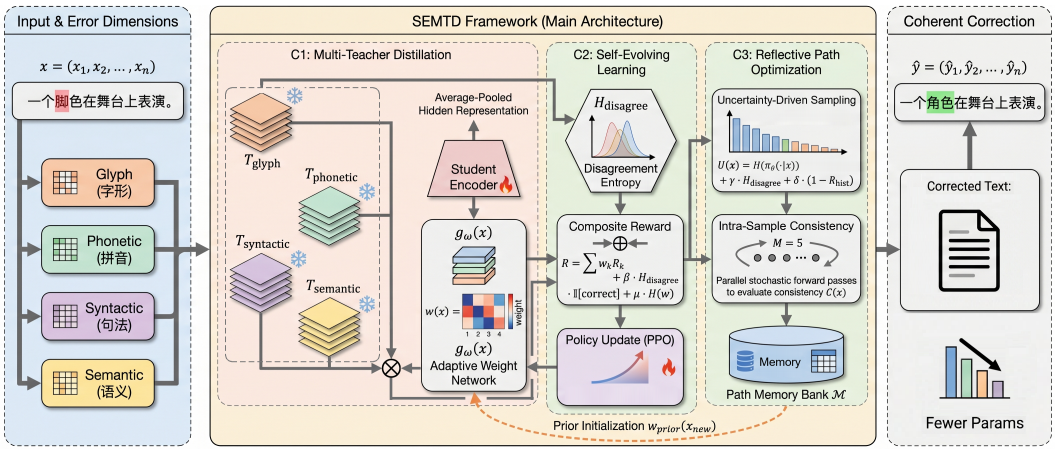


Figure 3: The illustration of SEMTD framework, which consists of three components: **(C1)** multi-teacher distillation, **(C2)** self-evolving learning, and **(C3)** reflective path optimization.

4 Solution

4.1 SEMTD Framework

To solve the problem formulated in Section 3, we propose SEMTD, a collective reinforcement framework where a lightweight student actively learns from four role-prompted teachers that share a single backbone and each focus on one error dimension. The core design principle is to shift from passive knowledge absorption to active reward-driven learning: rather than minimising a fixed distributional distance to a single teacher, the student receives per-teacher reward signals and optimises its policy to maximise an aggregate expected reward augmented with disagreement-aware shaping (Eq. 7). Single-teacher reward-driven distillation is the special case $K = 1$, $w_1 \equiv 1$, $\beta = \mu = 0$, while the general form enables input-dependent expert arbitration and difficulty-/diversity-shaped credit assignment that pure distribution-matching does not provide. The reformulation also transforms teacher disagreement from noise into a learning signal.

SEMTD has three integrated components (Figure 3). Multi-teacher distillation addresses *whom to trust* by constructing specialized expert teachers and learning input-dependent adaptive weights. Self-evolving learning addresses *how to decide* through a composite reward function that guides autonomous judgment under teacher disagreement. Reflective path optimization addresses *what to learn* through uncertainty-driven sampling and consistency-guided path amplification. Together, they form a self-reinforcing cycle where consolidated paths enable increasingly sophisticated arbitration.

4.2 Multi-Teacher Distillation

This component instantiates the adaptive weights $w_\omega(\mathbf{x})$ in Eq. (1), as depicted in Figure 3 (left). It constructs specialized expert teachers covering distinct linguistic dimensions, treats their outputs as reward signals rather than targets for distribution matching, and learns input-dependent adaptive weights to determine each teacher's contribution.

Teacher Construction. We construct $K = 4$ expert teachers specialized in distinct linguistic dimensions: glyph (T_{glyph}), phonetic (T_{phonetic}), syntactic ($T_{\text{syntactic}}$), and semantic (T_{semantic}). Because each teacher is a causal decoder LLM (DeepSeek-R1-Distill-Qwen-14B with a role-specific instruction prompt), per-position posteriors over the corrected sentence are not produced by a single forward pass on $\mathbf{x}$; instead, each T_k is prompted with a length-preserving template that constrains the model to autoregressively emit exactly $|\mathbf{x}|$ output tokens $\hat{\mathbf{y}}^{(k)} = (\hat{y}_1^{(k)}, \dots, \hat{y}_n^{(k)})$. We then extract, at each generation step i, the next-token logit distribution conditioned on the role prompt, the source $\mathbf{x}$, and the previously generated tokens, post-hoc aligned to source position i:

$$\mathbf{p}_i^{(k)} \;=\; \text{softmax}\left(\frac{T_k\big(\hat{y}_{<i}^{(k)} \mid \mathbf{x}, \text{prompt}_k\big)}{\tau}\right) \in \Delta^{|\mathcal{V}|}, \tag{2}$$

where the inner term is the next-token logit vector at step i and $\tau > 0$ is a distillation temperature. Generated step i is aligned to source position i; if the teacher emits a sequence of length $\neq |\mathbf{x}|$ on a

given input we discard that teacher's logits for that sample and re-normalise the adaptive weights over the remaining $K-1$ teachers (this fallback fires on $< 2\%$ of training inputs in practice). Detailed specifications of each teacher are provided in Appendix A.3.

Reward Signal. Unlike traditional distillation that minimizes KL divergence to teacher distributions, we treat teacher outputs as reward signals. For each teacher T_k, we define a base reward measuring how favorably the teacher evaluates the student's prediction:

$$R_k(\mathbf{x}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} p_{i,\hat{y}_i}^{(k)}, \tag{3}$$

where $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n)$ is the student's prediction and $p_{i,\hat{y}_i}^{(k)}$ is the probability that teacher T_k assigns to the predicted character $\hat{y}_i$ at position i. This reward reflects the teacher's confidence in the student's output: predictions that align with teacher expectations receive higher rewards.

Adaptive Weight Network. Different inputs require different teacher expertise. We design a weight network g_ω that computes input-dependent weights based on the student encoder's representation:

$$\mathbf{w}(\mathbf{x}) = g_\omega(\mathbf{x}) = \mathrm{softmax}\big(\mathbf{s}(\mathbf{x})\big) \in \Delta^K, \tag{4}$$

where the teacher score vector $\mathbf{s}(\mathbf{x}) \in \mathbb{R}^K$ is computed by a two-layer MLP:

$$\mathbf{s}(\mathbf{x}) = \mathbf{W}_2 \cdot \sigma\big(\mathbf{W}_1 \cdot \bar{\mathbf{h}}(\mathbf{x}) + \mathbf{b}_1\big) + \mathbf{b}_2. \tag{5}$$

Here, $\bar{\mathbf{h}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{h}_i$ is the average-pooled hidden representation from the student encoder, $\sigma(\cdot)$ is a nonlinear activation, and $\omega = \{\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2\}$ are learnable parameters.

Crucially, the weight network is learned end-to-end through reward maximization rather than auxiliary supervision. The gradient signal from the composite reward encourages the weight network to assign higher weights to teachers whose guidance yields higher rewards per input type.

4.3 Self-Evolving Learning

Building upon multi-teacher distillation, this component handles teacher disagreement through reward-driven optimization (Figure 3, middle). It quantifies the disagreement among teachers, designs a composite reward that transforms disagreement into valuable learning signal, and optimizes the student policy through PPO.

Teacher Disagreement. We quantify teacher disagreement to identify samples where passive absorption would fail. The disagreement entropy measures how much teachers diverge:

$$H_{\text{disagree}}(\mathbf{x}) = \tfrac{1}{n} \sum_{i=1}^{n} \Big[H\big(\bar{\mathbf{p}}_i\big) - \tfrac{1}{K} \sum_{k=1}^{K} H\big(\mathbf{p}_i^{(k)}\big) \Big], \tag{6}$$

where $\bar{\mathbf{p}}_i = \frac{1}{K}\sum_{k=1}^{K} \mathbf{p}_i^{(k)}$ is the average prediction across teachers and $H(\mathbf{p}) = -\sum_v p_v \log p_v$ is Shannon entropy. When all teachers agree, $H_{\text{disagree}} = 0$; when teachers point to different answers, H_{disagree} increases. High disagreement signals cases where independent judgment is essential.

Composite Reward. The base reward R_k in Eq. (3) evaluates single-teacher feedback but cannot capture inter-teacher dynamics. We extend it to a composite reward that integrates disagreement signals and diversity regularization, extending the weighted teacher term of Eq. (1):

$$R(\mathbf{x}, \hat{\mathbf{y}}; w_\omega) = \underbrace{\sum_{k=1}^{K} w_k(\mathbf{x}) \cdot R_k(\mathbf{x}, \hat{\mathbf{y}})}_{\text{weighted teacher reward}} + \underbrace{\beta \cdot H_{\text{disagree}}(\mathbf{x}) \cdot \mathbb{I}[\hat{\mathbf{y}} = \mathbf{y}^*]}_{\text{difficulty bonus}} + \underbrace{\mu \cdot \mathcal{H}(\mathbf{w}(\mathbf{x}))}_{\text{diversity bonus}}. \tag{7}$$

The weighted teacher reward aggregates teacher evaluations under learned input-dependent weights, letting the student trust different experts for different inputs. The difficulty bonus rewards correct resolution proportional to teacher disagreement, using a *token-level* indicator $\mathbb{I}[\hat{\mathbf{y}} = \mathbf{y}^*] := \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[\hat{y}_i = y_i^*] \in [0, 1]$ rather than a sentence-level exact-match flag, so partially correct rollouts on high-disagreement inputs still receive a graded signal early in training. The bonus is a label-conditional shaping term used at training only (no labels at inference). The diversity bonus rewards weight entropy $\mathcal{H}(\mathbf{w}) = -\sum_k w_k \log w_k$, preventing collapse to a single teacher.

Policy Optimization. We optimize the student policy π_θ and weight network g_ω jointly through policy gradient methods. Specifically, we adopt Proximal Policy Optimization (PPO) with clipped objectives for stable training. The base advantage estimate measures the relative quality of predictions:

$$\hat{A}(\mathbf{x}, \hat{\mathbf{y}}) = R(\mathbf{x}, \hat{\mathbf{y}}; w_\omega) - b, \tag{8}$$

where b is a global scalar baseline for variance reduction, maintained as an exponential moving average of rewards across all training samples with decay 0.99 (initialised at $b \leftarrow 0$); we use a global rather than input-conditional baseline to avoid per-sample bookkeeping with negligible variance impact. The PPO objective then optimizes the policy while constraining update magnitude:

$$J_{\text{PPO}} = \mathbb{E}_{\mathbf{x}\sim P_\phi, \hat{\mathbf{y}}\sim\pi_{\theta_{\text{old}}}}\Big[\min\big(r(\theta)\hat{A}, \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}\big)\Big], \tag{9}$$

where $r(\theta) = \frac{\pi_\theta(\hat{\mathbf{y}}|\mathbf{x})}{\pi_{\theta_{\text{old}}}(\hat{\mathbf{y}}|\mathbf{x})}$ is the probability ratio between new and old policies. The clipping mechanism prevents overly large policy updates, ensuring training stability. The weight network g_ω receives gradients through the weighted teacher reward term, learning to assign higher weights to teachers whose guidance leads to higher rewards.

4.4 Reflective Path Optimization

This component adopts a reflect-then-consolidate strategy (Figure 3, right). It prioritizes informative samples via uncertainty-driven sampling, evaluates prediction consistency across multiple passes, and consolidates successful paths into a memory bank for cross-sample transfer.

Uncertainty-Driven Sampling. Training samples vary in learning value. Samples where teachers agree and the student already performs well offer diminishing returns, while samples with high disagreement or historical errors present opportunities for meaningful improvement. We design an uncertainty-driven sampling strategy that prioritizes informative samples, instantiating P_ϕ in Eq. (1):

$$\mathcal{U}(\mathbf{x}) = \underbrace{H\left(\pi_\theta(\cdot|\mathbf{x})\right)}_{\text{predictive uncertainty}} + \gamma \cdot \underbrace{H_{\text{disagree}}(\mathbf{x})}_{\text{teacher disagreement}} + \delta \cdot \underbrace{(1 - R_{\text{hist}}(\mathbf{x}))}_{\text{historical error rate}}, \tag{10}$$

where $R_{\text{hist}}(\mathbf{x})$ is a per-sample exponential moving average of correction success on $\mathbf{x}$ since its first encounter, updated as $R_{\text{hist}}(\mathbf{x}) \leftarrow 0.9 \cdot R_{\text{hist}}(\mathbf{x}) + 0.1 \cdot \mathbb{I}[\hat{\mathbf{y}} = \mathbf{y}^*]$ with initialisation $R_{\text{hist}}(\mathbf{x}) \leftarrow 0$ for unseen samples (so that fresh samples receive the maximum historical-error term $1 - R_{\text{hist}} = 1$ and are therefore prioritised), and $\gamma, \delta \geq 0$ balance the three components.

The priority sampling distribution assigns higher probability to uncertain samples:

$$P_\phi(\mathbf{x}) = \frac{\exp(\eta \cdot \mathcal{U}(\mathbf{x}))}{\sum_{\mathbf{x}'\in\mathcal{D}} \exp(\eta \cdot \mathcal{U}(\mathbf{x}'))}. \tag{11}$$

We adopt adaptive scheduling where the concentration parameter η increases during training, starting with near-uniform sampling to ensure broad coverage and gradually focusing on difficult samples.

Intra-Sample Consistency. To consolidate stable decision paths, we evaluate prediction stability through multiple stochastic forward passes. For input $\mathbf{x}$, we sample M independent predictions $\{\hat{\mathbf{y}}^{(m)}\}_{m=1}^M$ and compute pairwise consistency:

$$C(\mathbf{x}) = \frac{2}{M(M-1)} \sum_{i<j} \frac{1}{n} \sum_{l=1}^{n} \mathbb{I}[\hat{y}_l^{(i)} = \hat{y}_l^{(j)}]. \tag{12}$$

High consistency indicates stable decision-making, while low consistency suggests the model remains uncertain about the optimal path.

The base advantage $\hat{A}$ in Eq. (8) treats all predictions equally regardless of stability. We enhance it with a scalar consistency coefficient that weights the update by decision stability on the rollout action:

$$\hat{A}_C(\mathbf{x}, \hat{\mathbf{y}}) = C(\mathbf{x}) \cdot \left(R(\mathbf{x}, \hat{\mathbf{y}}; w_\omega) - b\right), \tag{13}$$

where $\hat{\mathbf{y}}$ is the rollout action sampled from π_θ (the same action used in the PPO importance ratio $r(\theta) = \pi_\theta(\hat{\mathbf{y}}|\mathbf{x})/\pi_{\theta_{\text{old}}}(\hat{\mathbf{y}}|\mathbf{x})$ in Eq. (9)), and $C(\mathbf{x}) \in [0, 1]$ is a per-sample reward-shaping coefficient from the stochastic passes. The consensus prediction $\hat{\mathbf{y}}_{\text{cons}} = \arg\max_{\mathbf{y}} \sum_{m=1}^M \mathbb{I}[\hat{\mathbf{y}}^{(m)} = \mathbf{y}]$ does not enter the advantage; it is recorded as a candidate write-key for the path memory bank (Eq. (14)). PPO updates θ via $r(\theta)$ on the rollout action, while $C(\mathbf{x})$ down-weights uncertain rollouts to attenuate noise. The adaptive weights $w_\omega(\mathbf{x})$ continue to govern teacher aggregation inside $R(\cdot)$, independent of consistency scaling.

Cross-Sample Consistency. We further address cross-sample consistency by maintaining a path memory bank $\mathcal{M}$ that records successful decision configurations, enabling accumulated experience to transfer across similar inputs. We use the consensus prediction $\hat{\mathbf{y}}_{\text{cons}}$ rather than a single rollout action as the write candidate, because consensus is a less noisy estimate of the policy's stable behaviour and is the natural target for cross-sample reuse:

$$\mathcal{M} = \{(\mathbf{x}, \mathbf{w}^*(\mathbf{x}), \hat{\mathbf{y}}_{\text{cons}}) : R(\mathbf{x}, \hat{\mathbf{y}}_{\text{cons}}; w_\omega) > \tau_R \text{ and } C(\mathbf{x}) > \tau_C\}. \tag{14}$$

Beyond training, the memory bank enables experience transfer. For a novel input $\mathbf{x}_{\text{new}}$, we retrieve similar cases and use their weight configurations as initialization:

$$\mathbf{w}_{\text{prior}}(\mathbf{x}_{\text{new}}) = \frac{1}{|\mathcal{N}|} \sum_{(\mathbf{x}, \mathbf{w}^*, \cdot) \in \mathcal{N}(\mathbf{x}_{\text{new}})} \mathbf{w}^*, \tag{15}$$

where $\mathcal{N}(\mathbf{x}_{\text{new}})$ contains the $k = 5$ nearest neighbours of $\bar{\mathbf{h}}(\mathbf{x}_{\text{new}})$ in $\mathcal{M}$ under cosine similarity $\text{sim}(\bar{\mathbf{h}}(\mathbf{x}_{\text{new}}), \bar{\mathbf{h}}(\mathbf{x})) = \cos(\bar{\mathbf{h}}(\mathbf{x}_{\text{new}}), \bar{\mathbf{h}}(\mathbf{x}))$ on the student encoder's mean-pooled hidden state. We retain a fixed bank capacity of $|\mathcal{M}| = 10{,}000$ entries with FIFO eviction (oldest insertion discarded once full); see Appendix A.10.3 for full memory hyperparameters.

Memory Bank Specification. Four design choices: (i) *Similarity*: cosine similarity on $\bar{\mathbf{h}}(\mathbf{x})$, indexed by FAISS with IVF-PQ; (ii) *Retrieval depth*: $k = 5$ nearest neighbours; (iii) *Capacity / eviction*: fixed $|\mathcal{M}| = 10{,}000$ with FIFO; (iv) *Test-time read-only*: $\mathbf{w}_{\text{prior}}$ initialises g_ω as a fixed forward-pass with no bank writes, no gradient flow, and no test-time fine-tuning. The bank consolidates cross-sample consistency through shared patterns among similar inputs.

4.5 End-to-End Integration

End-to-End Integration. The three components are jointly optimised end-to-end: multi-teacher distillation provides adaptive weights, self-evolving learning (SEL) computes policy gradients from composite rewards, and reflective path optimization (RPO) consolidates successful decision paths into memory. The enriched memory in turn provides weight initialisation for subsequent iterations, forming a self-reinforcing cycle that distills collective expertise from specialized teachers into a single lightweight student capable of autonomous judgment under teacher disagreement; the trained student then runs alone without teachers, efficient for real-world deployment. More details about SEL/RPO gating are provided in Appendix A.6.3.

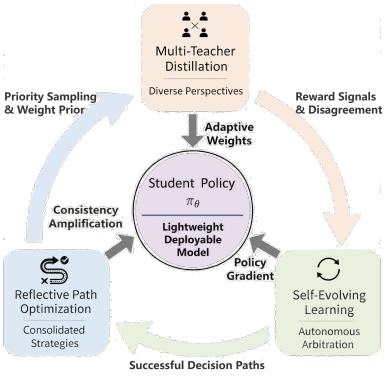


Figure 4: Closed-loop integration of SEMTD components.

Training Schedule. Figure 4 summarises the loop. Training runs in three phases (Appendix A.11.1): a two-epoch warm-up with uniform teacher weights; PPO with the composite reward, learned teacher weights, and priority sampling from Epoch 3; and, from Epoch 11, consistency-shaped advantages, memory writes, and cross-sample transfer, so the full loop of Figure 4 is active in the last five epochs. At inference only the 4B student runs, with no teacher calls (Appendix A.12).

5 Experiments

5.1 Experimental Setup

Software and Hardware. We use PyTorch 2.2.0 and Transformers 4.51.0 on 4 NVIDIA A100 80GB GPUs for student training, with 2 additional A100 GPUs for teacher inference via vLLM. Training uses BF16 mixed-precision with DeepSpeed ZeRO Stage 2. Complete details in Appendix A.7.

Backbone Model. The student is initialized from Qwen3-4B [63], a 4B-parameter dense transformer featuring 36 layers with hidden dimension 2,560 and GQA (32 query heads, 8 KV heads). We apply RoPE and SwiGLU following recent practices [48]. Architecture details are in Appendix A.8.

Table 1: Sentence-level performance on SIGHAN benchmarks. Best results in **bold**, second-best underlined. The DeepSeek-R1-Distill-Qwen-14B row is a single-role prompt zero-shot reference and the Qwen3-4B row is the SFT-only same-data baseline; more details in Appendix A.17.1.

Method	SIGHAN 2013						SIGHAN 2014						SIGHAN 2015					
	Det-P	Det-R	Det-F1	Cor-P	Cor-R	Cor-F1	Det-P	Det-R	Det-F1	Cor-P	Cor-R	Cor-F1	Det-P	Det-R	Det-F1	Cor-P	Cor-R	Cor-F1
Multimodal Fusion Methods																		
ReaLiSe [61]	88.6	82.5	85.4	87.2	81.2	84.1	67.8	71.5	69.6	66.3	70.0	68.1	77.3	81.3	79.3	75.9	79.9	77.8
SCOPE [20]	87.4	83.4	85.4	86.3	82.4	84.3	70.1	73.1	71.6	68.6	71.5	70.1	81.1	84.3	82.7	79.2	82.3	80.7
DR-CSC [17]	88.5	83.7	86.0	87.7	83.0	85.3	70.2	73.2	71.7	69.3	72.3	70.7	82.9	84.8	83.8	80.3	82.3	81.3
LLM-based Methods																		
UGMSC [25]	89.1	83.4	86.2	88.3	82.7	85.4	72.5	71.2	71.8	71.6	70.2	70.9	83.9	82.8	83.3	82.4	81.3	81.9
ARM+SCOPE [26]	88.7	84.1	86.3	87.6	83.1	85.3	71.2	**75.0**	73.1	69.2	73.0	71.1	82.3	**86.1**	84.1	79.5	83.1	81.3
Same-Backbone Reference Rows (single-prompt; no SEMTD)																		
DeepSeek-R1-Distill-Qwen-14B	73.2	67.5	70.2	70.6	65.0	67.7	56.4	58.1	57.2	53.9	55.5	54.7	71.2	70.8	71.0	67.9	67.5	67.7
Qwen3-4B	81.3	76.8	79.0	79.4	75.0	77.1	65.3	66.7	66.0	62.8	64.1	63.4	78.6	78.2	78.4	75.7	75.3	75.5
SEMTD (Ours)	**89.6**	**85.5**	**87.5**	**88.9**	**84.3**	**86.5**	**73.8**	74.5	**74.1**	**72.3**	**73.8**	**73.0**	**85.1**	85.5	**85.3**	**83.5**	**84.1**	**83.8**

Table 2: Sentence-level Cor-F1 (%) on LEMON multi-domain benchmark. CAR: Car, COT: Court, ENC: Encyclopedia, GAM: Game, MEC: Medical, NEW: News, NOV: Novel. Best results in **bold**, second-best underlined.

Method	CAR	COT	ENC	GAM	MEC	NEW	NOV	Avg
InternLM2.5 [5]	44.4	54.7	45.2	33.0	50.2	57.2	32.1	45.3
Qwen2.5 [2]	48.2	59.9	46.5	34.7	54.2	59.5	37.1	48.6
ReLM [29]	53.6	67.7	47.7	34.6	53.9	58.8	38.0	50.6
Baichuan2 [62]	54.2	63.3	51.4	36.9	60.6	63.9	42.4	53.2
ReLM+BC2 [40]	62.7	74.5	56.5	**45.1**	66.3	72.0	50.0	61.0
SEMTD (Ours)	**64.3**	**76.0**	**58.5**	44.3	**68.2**	**73.3**	**51.5**	**62.3**

Table 3: Sentence-level Cor-F1 (%) on ECSpell three professional domains. Best results in **bold**, second-best underlined.

Method	LAW	MED	ODW	Avg
ChatGPT (10-shot) [34]	48.3	26.0	56.8	43.7
MDCSpell-MFT [81]	80.6	69.6	66.9	72.4
InternLM2.5 (7B) [5]	85.8	66.2	89.0	80.3
ReLM [29]	96.2	90.2	90.4	92.3
ReLM+IL2.5 [40]	97.3	95.9	96.9	96.7
ReLM+BC2 [40]	98.3	96.6	96.5	97.1
SEMTD (Ours)	**99.0**	**98.0**	**97.7**	**98.2**

Teacher Construction. We construct four expert teachers (T_{glyph}, T_{phonetic}, $T_{\text{syntactic}}$, T_{semantic}) through role-specialized prompting of DeepSeek-R1-Distill-Qwen-14B [12] following the Expert-Prompting framework [60]. Prompt designs are in Appendix A.9.

Baselines. We compare SEMTD against three tiers of CSC systems: (i) the strongest published peers per benchmark, namely ARM+SCOPE [26] and UGMSC [25] for SIGHAN, ReLM+BC2 [40] for LEMON and ECSpell, and C-LLM-14B [22] together with the MSLLM mixture [40] for CSCD-NS; (ii) the closest non-distillation alternatives ReaLiSe [61], SCOPE [20], DR-CSC [17], and ReLM [29]; and (iii) two same-backbone reference rows, namely a single-role DeepSeek-R1-Distill-Qwen-14B teacher and a Qwen3-4B fine-tuned with cross-entropy on the same training mixture. Baseline numbers are cited from each method's original paper; parity notes in Appendix A.13.

Datasets and Metrics. We evaluate on four CSC benchmarks: SIGHAN [58, 66, 49] (SIGHAN13/14/15, CFL-learner essays), LEMON [57] (seven domains, natural errors), ECSpell [34] (Law, Medical, Official Document Writing), and CSCD-NS [16] (native-speaker errors). Following prior work [61, 23] we report sentence-level Precision/Recall/F1 for both detection and correction with Cor-F1 as the primary metric. Dataset and metric details in Appendix A.13 and A.14.

Implementation and Inference. PPO uses $\epsilon = 0.2$, learning rate 1.41×10^{-5} with cosine annealing; distillation temperature $\tau = 3.0$; composite reward $\beta = 0.5$, $\mu = 0.1$; RPO $\gamma = 0.3$, $\delta = 0.2$, $M = 5$, $\tau_R = 0.8$, $\tau_C = 0.7$. Training runs three phases (warm-up 2 epochs, adaptive 8 epochs, refinement 5 epochs). At inference SEMTD reduces to a single 4B-parameter forward pass; teachers are not invoked, no consistency passes are run, and the path memory bank is read-only (Section 4.4); the 4B student has $\approx 3.5\times$ fewer parameters than the 14B teacher backbone. Full hyperparameter, training-protocol, and deployment details (including INT8/INT4 quantization) are in Appendix A.10–A.12.

5.2 Main Results

SIGHAN Benchmark Results. SEMTD attains the best Cor-F1 on all three SIGHAN splits with $+1.1$, $+1.9$, and $+1.9$ improvements over the strongest published baseline (86.5/73.0/83.8 on SIGHAN13/14/15; Table 1), confirming that gains are not artefacts of any single split, with Det-F1 of 87.5/74.1/85.3 mirroring the pattern. While ARM+SCOPE achieves competitive recall, SEMTD attains higher precision (83.5% vs. 79.5% on SIGHAN15) through adaptive teacher arbitration. Two same-backbone reference rows bracket SEMTD's 83.8% from below: a single-role 14B teacher (67.7% on SIGHAN15, zero-shot prompted) and a Qwen3-4B SFT-only baseline (75.5%); the $+8.3$ gap over Qwen3-4B-SFT decomposes into $+5.2$ (multi-teacher distillation) and $+3.1$ (reward-driven training stage), as in Appendix Table 27. Full per-row analysis is in Appendix A.17.1.

Multi-Domain Generalization. SEMTD lifts LEMON-mean Cor-F1 to 62.3% (+1.3 over ReLM+BC2) and wins six of the seven domains (Table 2), addressing the concern that the SIGHAN gain is specific to CFL learner text. Per-domain improvements are CAR +1.6, COT +1.5, ENC +2.0, MEC +1.9, NEW +1.3, NOV +1.5. GAM is the exception (44.3 vs. 45.1, -0.8).

Native-Speaker Errors. SEMTD attains 73.0 Cor-F1 on CSCD-NS (Table 4), above every baseline except the two 14B-class systems C-LLM-14B and MSLLM, and within 0.8 of C-LLM-14B (73.8). Its correction precision (81.0) is the highest among the systems that report it, including C-LLM-14B (80.5), matching the precision advantage observed on SIGHAN15, and its detection F1 (76.0) is the highest among those reporting detection. Its recall (66.4) trails C-LLM-14B (68.2), so the remaining gap to that system lies in recall. The full per-method panel, with BERT / SMBERT / multimodal-fusion / LLM-based rows and detection scores, is in Appendix A.17.4.

Table 4: CSCD-NS results (%). Best outside the 14B-class row in **bold**.

Method	Cor-P	Cor-R	Cor-F1
DR-CSC [17]	77.7	63.3	69.8
SCOPE [20]	79.7	65.2	71.7
ARM+SCOPE [26]	80.0	65.6	72.1
C-LLM-14B [22]	80.5	68.2	73.8
SEMTD (Ours, 4B)	**81.0**	**66.4**	**73.0**

5.3 Ablation Study

Table 5 reports the condensed component ablation on SIGHAN 2015. Removing the phonetic teacher costs $5\times$ what removing the syntactic teacher does (-2.6 vs. -0.5 Cor-F1), reflecting the prevalence of homophone errors in CSC. The -8.3 Vanilla-SFT gap decomposes additively into $+5.2$ from multi-teacher distillation (75.5 $\rightarrow$ 80.7) and $+3.1$ from the reward-driven training stage (80.7 $\rightarrow$ 83.8); separately, adaptive weights account for $+2.4$ over uniform mixing, validating input-dependent expert selection. No single teacher removal costs as much as dropping the reward stage. Across all four benchmarks the variants evaluated there keep the same order, and removing the syntactic teacher costs 1.5 Cor-F1 on LEMON and 1.8 on CSCD-NS (Appendix A.17.7). Within reflective path optimization, removing priority sampling costs 1.3 Cor-F1, path memory 0.7, and consistency evaluation 0.4. Composite-reward grids are optimal at β=0.5, μ=0.1, and uncertainty-driven sampling at γ=0.3, δ=0.2; the corresponding sensitivity heatmaps are in Appendix A.15.

Table 5: Condensed component ablation on SIGHAN 2015 (Cor-F1, %); component ablation with detection / correction P-F1 in Appendix A.15.

Variant	Cor-F1
SEMTD (Full)	**83.8**
w/o Glyph Teacher	83.0
w/o Phonetic Teacher	81.2
w/o Syntactic Teacher	83.3
w/o Semantic Teacher	82.8
w/o Adaptive Weights	81.4
w/o PPO (MT-SFT only)	80.7
Vanilla SFT (no teachers)	75.5
w/o Priority Sampling	82.5
w/o Consistency Eval	83.4
w/o Path Memory	83.1

5.4 Analysis

Teacher Specialization. Evaluated alone on error-type subsets of SIGHAN 2015, each teacher attains its highest Cor-F1 on the subset of its own dimension: 78.5 for T_{glyph} on glyph errors, 81.4 for T_{phonetic} on phonetic errors, 71.6 for $T_{\text{syntactic}}$ on syntactic errors, and 75.8 for T_{semantic} on semantic errors, while on every subset the other three teachers stay at or below 61.3 (Appendix A.17.6). The four teachers share one backbone and differ only in their role prompts, yet their strengths separate by error type, and no teacher is best on every subset. Replacing the per-input adaptive weights with uniform weights costs 2.4 Cor-F1 (Table 5), more than removing any teacher except the phonetic one.

Consistency Passes. During training, the consistency coefficient $C(\mathbf{x})$ in Eq. (12) is estimated from M stochastic passes. On SIGHAN 2015, M=3 gives 82.6 Cor-F1 and M=5 gives 83.8, while M=7 and M=9 give 83.7 and 83.8, so M=5 already reaches the plateau and larger values bring no further gain while requiring proportionally more forward passes (Appendix A.15.6).

Deployment. No teacher is needed at inference. The 4B student needs about 10 GB of GPU memory in BF16 and about 5 GB in INT8, which fits consumer GPUs such as the RTX 3080; on GPU it corrects a sentence in about 18 ms at batch size 1 (Table 22). INT8 quantization retains 99.5% of correction F1, and INT4 retains 98.2% (Table 23), so quantization costs little accuracy.

6 Related Work

Chinese Spelling Correction. Pre-trained language models have been used for CSC [74]; phonetic and glyph knowledge was later added through specialized pre-training [32, 46], and recent correctors use large language models [79, 22] or mix small and large models [40]. Most of these methods fuse the error dimensions inside one model, leaving their conflicts implicit. SEMTD keeps the four error dimensions as separate teachers and trains the student to arbitrate among them when they disagree.

Multi-Teacher Knowledge Distillation. Multi-teacher distillation transfers knowledge from multiple teachers to a compact student model. Early methods average teacher outputs [13] or apply uniform weights [36], but neglect instance-level teacher reliability. Recent approaches enable dynamic teacher selection through reinforcement learning [77] or gradient-based weighting [54, 80]. However, when teachers show conflicting professional knowledge on different inputs, these methods struggle; our adaptive weight network addresses this by learning, for every input, how much to trust each teacher.

Self-Rewarding Reinforcement Learning. Reinforcement learning has emerged as a powerful paradigm for enhancing model capabilities. Self-rewarding mechanisms leverage majority voting [83, 59] or semantic clustering [73] as proxy signals. Recent work exploits trajectory consistency for reward construction [72], demonstrating that internal prediction patterns can substitute for ground-truth supervision. We extend this insight by using teacher disagreement as a reward-shaping signal, so that the student learns autonomous judgment where the four teachers' specialized expertise conflicts.

Reflective Path Optimization. Self-reflection allows the model to analyze and enhance its reasoning process. The quality of reflection is directly optimized either through GRPO [3, 28] or via iterative self-training [21], depending on the specific training approach. Our reflective path optimization builds upon these methods [35, 68]. It employs an integration of uncertainty-driven sampling and consistency guidance to effectively internalize the successful decision-making model.

7 Conclusion

This paper started from a diagnosis rather than an architecture. In Chinese Spelling Correction, the residual difficulty after a decade of representational progress is no longer how to encode glyph, phonetic, syntactic, and semantic evidence, but how to arbitrate when they disagree, and that disagreement, once exposed by dimension-specialized teachers, is itself the signal flat fusion has been smoothing away. SEMTD turns this observation into one coupled objective in which inter-teacher disagreement shapes which expert the student trusts per input, how its self-evolving reward is composed, and which decision paths are revisited, so hard cases compound rather than fade. Across SIGHAN13/14/15, LEMON, ECSpell, and CSCD-NS, a 4B student trained this way leads the published baselines on SIGHAN, LEMON, and ECSpell and comes within 0.8 of a 14B single-model LLM corrector on CSCD-NS at $3.5\times$ fewer parameters; we expect the gains to concentrate where teachers actually disagree. Two extensions follow naturally, namely porting the arbitration view to other East Asian orthographies whose error taxonomies decompose along comparable axes, and scheduling teacher disagreement itself, rather than sample difficulty, as the curriculum signal.

8 Limitations

The four error dimensions are defined for Chinese, and all experiments use Chinese CSC benchmarks; other scripts would require re-deriving the dimension set. Training requires a one-off pass of four 14B teachers to generate soft labels (Table 18), a cost that does not recur at inference, where the 4B student runs alone. The four teachers share one backbone and are specialized through role prompts rather than separate training; Appendix A.17.6 reports their behavior on error-type subsets, and Appendix A.17.9 a prompt-variant check. The adaptive weights are learned by reward maximization rather than per-dimension supervision, so individual arbitration decisions are not interpretable by construction. Wrong corrections may alter the meaning of user text in high-stakes domains such as medical, legal, educational, and official-document scenarios. Our method is evaluated only on benchmark datasets, and its behavior on real-world noisy text with mixed error types, informal expressions, and domain-specific terminology remains to be validated.

References

[1] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. URL `https://arxiv.org/abs/2309.16609`.

[3] Shelly Bensal, Umar Jamil, Christopher Bryant, Melisa Russak, Kiran Kamble, Dmytro Mozolevskyi, Muayad Ali, and Waseem AlShikh. Reflect, retry, reward: Self-improving llms via reinforcement learning. *arXiv preprint arXiv:2505.24726*, 2025.

[4] Lianshang Cai, Linhao Zhang, Dehong Ma, Jun Fan, Daiting Shi, Yi Wu, Zhicong Cheng, Simiu Gu, and Dawei Yin. Pile: Pairwise iterative logits ensemble for multi-teacher labeled distillation. *arXiv preprint arXiv:2211.06059*, 2022.

[5] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024.

[6] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 7028–7036, 2021.

[7] Yibo Chen, Gangqing Hu, Chun Xia, Lihua Yu, Mihail Popescu, and Dong Xu. Large language models and prompt-based learning: Frontiers and challenges in cross-disciplinary applications. *International Journal of Artificial Intelligence and Robotics Research*, 2:2530001, 2025.

[8] Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. Spellgcn: Incorporating phonological and visual similarities into language models for chinese spelling check. *arXiv preprint arXiv:2004.14166*, 2020.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186. Association for Computational Linguistics, 2019.

[10] Shangchen Du, Shan You, Xiaojie Li, Jianlong Wu, Fei Wang, Chen Qian, and Changshui Zhang. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. *advances in neural information processing systems*, 33:12345–12355, 2020.

[11] Luyang Fang, Xiaowei Yu, Jiazhang Cai, Yongkai Chen, Shushan Wu, Zhengliang Liu, Zhenyuan Yang, Haoran Lu, Xilin Gong, Yufang Liu, et al. Knowledge distillation and dataset distillation of large language models: Emerging trends, challenges, and future directions. *Artificial Intelligence Review*, 59(1):17, 2026.

[12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[13] Ahmed Hamdi, Hassan N Noura, and Joseph Azar. A multi-teacher knowledge distillation framework with aggregation techniques for lightweight deep models. *Applied System Innovation*, 8(5):146, 2025.

[14] Guangjie Han, Yumeng Zhang, Li Liu, Zhen Wang, Yuanyang Zhu, and Dan Xia. Multi-teacher multi-feature knowledge distillation based on dynamic mutual selection. *IEEE Transactions on Instrumentation and Measurement*, 2025.

[15] Chengming Hu, Xuan Li, Dan Liu, Haolun Wu, Xi Chen, Ju Wang, and Xue Liu. Teacher-student architecture for knowledge distillation: A survey. *arXiv preprint arXiv:2308.04268*, 2023.

[16] Yong Hu, Fandong Meng, and Jie Zhou. Cscd-ns: a chinese spelling check dataset for native speakers, 2024. URL https://arxiv.org/abs/2211.08788.

[17] Haojing Huang, Jingheng Ye, Qingyu Zhou, Yinghui Li, Yangning Li, Feng Zhou, and Hai-Tao Zheng. A frustratingly easy plug-and-play detection-and-reasoning module for chinese spelling check. *arXiv preprint arXiv:2310.09119*, 2023.

[18] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

[19] Chengzhang Li, Ming Zhang, Xuejun Zhang, and Yonghong Yan. Mcrspell: A metric learning of correct representation for chinese spelling correction. *Expert Systems with Applications*, 237: 121513, 2024.

[20] Jiahao Li, Quan Wang, Zhendong Mao, Junbo Guo, Yanyan Yang, and Yongdong Zhang. Improving chinese spelling check by character pronunciation prediction: The effects of adaptivity and granularity. *arXiv preprint arXiv:2210.10996*, 2022.

[21] Jiaqi Li, Xinyi Dong, Yang Liu, Zhizhuo Yang, Quansen Wang, Xiaobo Wang, Song-Chun Zhu, Zixia Jia, and Zilong Zheng. Reflectevo: Improving meta introspection of small llms by learning self-reflection. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16948–16966, 2025.

[22] Kunting Li, Yong Hu, Liang He, Fandong Meng, and Jie Zhou. C-llm: Learn to check chinese spelling errors character by character. *arXiv preprint arXiv:2406.16536*, 2024.

[23] Yinghui Li, Qingyu Zhou, Yangning Li, Zhongli Li, Ruiyang Liu, Rongyi Sun, Zizhen Wang, Chao Li, Yunbo Cao, and Hai-Tao Zheng. The past mistake is the future wisdom: Error-driven contrastive probability optimization for chinese spell checking. *arXiv preprint arXiv:2203.00991*, 2022.

[24] Zhiyuan Li, Huawei Liang, Hanqi Wang, Mingzhuo Zhao, Jian Wang, and Xiaokun Zheng. Mkd-cooper: Cooperative 3d object detection for autonomous driving via multi-teacher knowledge distillation. *IEEE Transactions on Intelligent Vehicles*, 9(1):1490–1500, 2023.

[25] Yongliang Lin, Zhen Zhang, Mengting Hu, Yufei Sun, and Yuzhi Zhang. Modalities should be appropriately leveraged: Uncertainty guidance for multimodal chinese spelling correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11463–11474, 2024.

[26] Changchun Liu, Kai Zhang, Junzhe Jiang, Zirui Liu, Hanqing Tao, Min Gao, and Enhong Chen. Arm: An alignment-and-replacement module for chinese spelling check based on llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10156–10168, 2024.

[27] Changchun Liu, Kai Zhang, Junzhe Jiang, Zixiao Kong, Qi Liu, and Enhong Chen. Chinese spelling correction: A comprehensive survey of progress, challenges, and opportunities. *arXiv preprint arXiv:2502.11508*, 2025.

[28] Haoyu Liu and Le Xiao. Re-grpo: Leveraging hard negative cases through large language model guided self training. *Neurocomputing*, page 132543, 2025.

[29] Linfeng Liu, Hongqiu Wu, and Hai Zhao. Chinese spelling correction as rephrasing language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18662–18670, 2024.

[30] Linfeng Liu, Hongqiu Wu, and Hai Zhao. Driving chinese spelling correction from a fine-grained perspective. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10727–10737, 2025.

[31] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35, 2023.

[32] Shulin Liu, Tao Yang, Tianchi Yue, Feng Zhang, and Di Wang. Plome: Pre-training with misspelled knowledge for chinese spelling correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2991–3000, 2021.

[33] Xiao Liu, Shichang Zhu, Ying Li, Xin Chen, and Zhengtao Yu. Entity-focused chinese spelling correction: Dataset and approach. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(11):1–17, 2025.

[34] Qi Lv, Ziqiang Cao, Lei Geng, Chunhui Ai, Xu Yan, and Guohong Fu. General and domain-adaptive Chinese spelling check with error-consistent pretraining. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(5), May 2023. ISSN 2375-4699. doi: 10.1145/3564271. URL `https://doi.org/10.1145/3564271`.

[35] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

[36] Sriram Mandalika et al. Comad: A multiple-teacher self-supervised distillation framework. *arXiv preprint arXiv:2508.04816*, 2025.

[37] Amir M Mansourian, Rozhan Ahmadi, Masoud Ghafouri, Amir Mohammad Babaei, Elaheh Badali Golezani, Zeynab Yasamani Ghamchi, Vida Ramezanian, Alireza Taherian, Kimia Dinashi, Amirali Miri, et al. A comprehensive survey on knowledge distillation. *arXiv preprint arXiv:2503.12067*, 2025.

[38] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198, 2020.

[39] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3967–3976, 2019.

[40] Ziheng Qiao, Houquan Zhou, and Zhenghua Li. Mixture of small and large models for chinese spelling check. *arXiv preprint arXiv:2506.06887*, 2025.

[41] Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching language model agents how to self-improve. *Advances in Neural Information Processing Systems*, 37:55249–55285, 2024.

[42] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. arxiv 2014. *arXiv preprint arXiv:1412.6550*, 2014.

[43] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[44] Luyao Shi, Ning Jiang, Jialiang Tang, and Xinlei Huang. Correlation guided multi-teacher knowledge distillation. In *International Conference on Neural Information Processing*, pages 562–574. Springer, 2023.

[45] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

[46] Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. *arXiv preprint arXiv:2106.16038*, 2021.

[47] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[49] Yuen-Hsien Tseng, Lung-Hao Lee, Li-Ping Chang, and Hsin-Hsi Chen. Introduction to sighan 2015 bake-off for chinese spelling check. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 32–37, 2015.

[50] Can Wang, Defang Chen, Jian-Ping Mei, Yuan Zhang, Yan Feng, and Chun Chen. Semckd: Semantic calibration for cross-layer knowledge distillation. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):6305–6319, 2022.

[51] Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang. A hybrid approach to automatic corpus generation for chinese spelling check. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2517–2527, 2018.

[52] Helin Wang, Wei Du, Ning Liu, Qian Li, Yanyu Xu, and Lizhen Cui. Adahet-mkd: An adaptive heterogeneous multi-teacher knowledge distillation for medical image analysis. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 2977–2986, 2025.

[53] Yao Wang. Research progress on chinese and english text error correction. In *ITM Web of Conferences*, volume 73, page 02009. EDP Sciences, 2025.

[54] Zhongyu Wang, Caiyun Ma, Shuo Zhang, Minghui Zhao, Yu Liu, Lina Zhao, Tingting Zhu, Jianqing Li, and Chengyu Liu. Multi-teacher self-distillation based on adaptive weighting and activation pattern for enhancing lightweight arrhythmia recognition. *Information Fusion*, 122: 103178, 2025.

[55] Andreas Winata, Nur Afny Catur Andryani, Alexander Agung Santoso Gunawan, Ford Lumban Gaol, and Tokuro Matsuo. Mtakd: multi-teacher agreement knowledge distillation for edge ai skin disease diagnosis. *Scientific Reports*, 2025.

[56] Haiming Wu, Hanqing Zhang, Richeng Xuan, and Dawei Song. Bi-dcspell: A bi-directional detector-corrector interactive framework for chinese spelling check. *arXiv preprint arXiv:2406.01879*, 2024.

[57] Hongqiu Wu, Shaohua Zhang, Yuchen Zhang, and Hai Zhao. Rethinking masked language modeling for chinese spelling correction. *arXiv preprint arXiv:2305.17721*, 2023.

[58] Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. Chinese spelling check evaluation at sighan bake-off 2013. In *Proceedings of the seventh SIGHAN workshop on Chinese language processing*, pages 35–42, 2013.

[59] Wei Xiong, Hanning Zhang, Chenlu Ye, Lichang Chen, Nan Jiang, and Tong Zhang. Self-rewarding correction for mathematical reasoning. *arXiv preprint arXiv:2502.19613*, 2025.

[60] Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. Expertprompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:2305.14688*, 2023.

[61] Heng-Da Xu, Zhongli Li, Qingyu Zhou, Chao Li, Zizhen Wang, Yunbo Cao, He-Yan Huang, and Xian-Ling Mao. Read, listen, and see: Leveraging multimodal information helps chinese spell checking. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 716–728, 2021.

[62] Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023.

[63] 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.

[64] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1285–1294, 2017.

[65] Nikolaos-Antonios Ypsilantis, Kaifeng Chen, André Araujo, and Ondřej Chum. Udon: Universal dynamic online distillation for generic image representations. *Advances in Neural Information Processing Systems*, 37:86836–86859, 2024.

[66] Liang-Chih Yu, Lung-Hao Lee, Yuen-Hsien Tseng, and Hsin-Hsi Chen. Overview of sighan 2014 bake-off for chinese spelling check. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 126–132, 2014.

[67] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[68] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772, 2024.

[69] Hailin Zhang, Defang Chen, and Can Wang. Confidence-aware multi-teacher knowledge distillation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4498–4502. IEEE, 2022.

[70] Hailin Zhang, Defang Chen, and Can Wang. Adaptive multi-teacher knowledge distillation with meta-learning. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1943–1948. IEEE, 2023.

[71] Haonan Zhang, Longjun Liu, Yi Zhang, Xinyu Lei, Fei Hui, and Bihan Wen. Densekd: Dense knowledge distillation by exploiting region and sample importance. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[72] Kongcheng Zhang, Qi Yao, Shunyu Liu, Yingjie Wang, Baisheng Lai, Jieping Ye, Mingli Song, and Dacheng Tao. Consistent paths lead to truth: Self-rewarding reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.08745*, 2025.

[73] Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025.

[74] Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. Spelling error correction with soft-masked bert. *arXiv preprint arXiv:2005.07421*, 2020.

[75] Zijian Zhang and Jinfeng Yuan. Decopy: detect and correct with pinyin for chinese spelling correction. *Royal Society Open Science*, 12(9), 2025.

[76] Shiji Zhao, Xizhe Wang, and Xingxing Wei. Mitigating accuracy-robustness trade-off via balanced multi-teacher adversarial distillation. *IEEE transactions on pattern analysis and machine intelligence*, 46(12):9338–9352, 2024.

[77] Zhuo Zhao, Zhiwen Xie, Guangyou Zhou, and Jimmy Xiangji Huang. Mtms: Multi-teacher multi-stage knowledge distillation for reasoning-based machine reading comprehension. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1995–2005, 2024.

[78] Zhenzhong Zheng, Xiaoming Wu, and Xiangzhi Liu. Enhancing pre-trained language models with chinese character morphological knowledge. *Information Processing & Management*, 62 (1):103945, 2025.

[79] Houquan Zhou, Zhenghua Li, Bo Zhang, Chen Li, Shaopeng Lai, Ji Zhang, Fei Huang, and Min Zhang. A simple yet effective training-free prompt-free approach to chinese spelling correction based on large language models. *arXiv preprint arXiv:2410.04027*, 2024.

[80] Wenjie Zhou, Zhenxin Ding, Xiaodong Zhang, Haibo Shi, Junfeng Wang, and Dawei Yin. Govern: Gradient orientation vote ensemble for multi-teacher reinforced distillation. *arXiv preprint arXiv:2405.03764*, 2024.

[81] Chenxi Zhu, Ziqiang Ying, Boyu Zhang, and Feng Mao. MDCSpell: A multi-task detector-corrector framework for Chinese spelling correction. In *Findings of ACL 2022*, pages 1244–1253, 2022.

[82] Wencheng Zhu, Xin Zhou, Pengfei Zhu, Yu Wang, and Qinghua Hu. Ckd: Contrastive knowledge distillation from a sample-wise perspective. *IEEE Transactions on Image Processing*, 2025.

[83] Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

Appendix

Table of Contents

Related Work (Extended) 17
A.1 Multi-Dimensional Errors 18
A.2 Multi-Teacher Disagreement 22
A.3 Teacher Specifications 32
A.4 Multi-Teacher Distillation Algorithm 33
A.5 Self-Evolving Learning Algorithm 34
A.6 Reflective Path Optimization Algorithm 36
A.7 Software and Hardware 39
A.8 Model Architecture 40
A.9 Teacher Specifications 42
A.10 Hyperparameter Specifications 44
A.11 Training Protocol Details 46
A.12 Inference Configuration 49
A.13 Dataset Details 50
A.14 Evaluation Metrics 52
A.15 Ablation Study 54
A.16 Limitations 57
A.17 Detailed Main Results 58
A.18 Ethical Considerations 62

Related Work (Extended)

Multi-Teacher Knowledge Distillation

Multi-teacher knowledge distillation transfers knowledge from multiple teacher models to a compact student model, addressing limitations of single-teacher approaches where knowledge may be biased or inaccurate [64, 38, 52]. Early methods simply average teacher outputs [64, 1, 13] or apply uniform weights across samples [36], neglecting instance-level teacher reliability. To address this, adaptive ensemble methods emerged, such as assigning sample-wise reliability weights based on alignment between teacher predictions and ground-truth labels [69], and adaptive ensemble in gradient space [10].

Recent research has shifted towards dynamic teacher selection at the instance level. Some use reinforcement learning to dynamically assign weights among teachers [77], while others ensemble multi-teacher logits supervised by human-annotated labels iteratively [4]. In 3D object detection, feature maps from multiple teachers are adaptively fused [24], and correlation-guided methods encourage students to allocate more learning weights to teachers with greater dissimilarity [44]. However, these supervised methods require label guidance, limiting their applicability to unsupervised scenarios.

Gradient-based teacher weighting provides new insights. Wang et al. [54] proposed a multi-teacher self-distillation framework that uses cosine distance to measure gradient direction similarity between distillation and main task losses, assigning higher weights to auxiliary losses that positively impact the primary objective. They also incorporate prediction uncertainty to assess the capacity gap between teacher and student models. Zhou et al. [80] introduced GOVERN, a gradient orientation vote ensemble method that dynamically selects teachers based on gradient descent orientations without label guidance, enabling effective unsupervised multi-teacher distillation. When teachers show

conflicting professional knowledge in different inputs, these methods may encounter difficulties, but our adaptive weight network can solve this challenge.

Beyond soft labels, intermediate layer representations have been explored. Feature-based methods align feature maps between teacher and student models [42, 67], while relation-based methods capture inter-sample relationships [47, 39]. However, direct feature map alignment in online distillation may lead to optimization instability [65, 14]. Wang et al. [54] proposed channel-enhanced activation similarity maps (CASM) to provide a more stable knowledge representation, and Chen et al. [6], Wang et al. [50] explored cross-layer distillation with semantic calibration.

Self-Rewarding Reinforcement Learning

Reinforcement learning (RL) has emerged as a powerful paradigm for enhancing model capabilities, especially in reasoning tasks. Self-rewarding mechanisms leverage majority voting [83, 59] or semantic clustering [73] as proxy signals. TTRL [83] rewards high-frequency answers through majority voting, and EMPO [73] uses semantic clusters as proxy reward signals. However, these methods assign rewards solely based on final answers, overlooking intermediate reasoning states.

Recent work exploits trajectory consistency for reward construction. Zhang et al. [72] observe that correct responses exhibit distinct trajectory patterns with high consistency and low volatility. They propose CoVo, which employs consistency and volatility as intrinsic rewards through a vectorial aggregation strategy, complemented by a curiosity bonus to promote diverse exploration. This approach enables LLMs to perform RL in a self-rewarding manner without external supervision, achieving performance comparable to supervised RL with verifiable answers. We extend this insight by using teacher disagreement as a reward-shaping signal for autonomous judgment.

Reflective Path Optimization

Self-reflection allows the model to analyze and enhance its reasoning process. Early approaches relied on carefully designed prompts to elicit reflective behavior from LLMs. Madaan et al. [35] proposed Self-Refine, which iteratively generates feedback to critique model outputs and refines responses. Shinn et al. [45] introduced Reflexion, a framework that reinforces language agents through verbal reflection on task feedback. However, these prompt-based methods are limited by the quality of reflection prompts and struggle to generalize across diverse tasks without careful prompt engineering [31, 7].

Recent research has shifted towards incorporating self-improvement capabilities during model training. Bensal et al. [3] proposed a "Reflect, Retry, Reward" paradigm that leverages Group Relative Policy Optimization (GRPO) to optimize self-reflection quality by rewarding only the tokens generated during the self-reflection phase. Experimental results showed that smaller fine-tuned models can outperform larger models through optimized self-reflection. Li et al. [21] proposed ReflectEvo, a pipeline for automatic self-reflection generation and curation. They constructed ReflectEvo-460k, a large-scale reflection dataset, and explored four reflection learning settings. Their method substantially boosted the performance of small language models on benchmarks, demonstrating that SLMs can achieve competitive reasoning performance through iterative reflection learning. Our reflective path optimization builds upon these methods [35, 68]. It employs an integration of uncertainty-driven sampling and consistency guidance to effectively internalize the successful decision-making model.

The integration of reinforcement learning (RL) techniques has proven effective for reflective path optimization. Kumar et al. [18] formulated self-correction as a multi-turn RL problem, demonstrating that models can learn persistent self-improvement capabilities. Qu et al. [41] introduced recursive introspection, teaching language model agents to progressively improve through self-generated feedback. These RL-based approaches typically employ outcome-based rewards that only become available at the conclusion of generation, making GRPO [43] well-suited for such scenarios as it eliminates the need for separate critic networks.

Appendix A.1: Multi-Dimensional Errors

As introduced in Section 2, Chinese spelling errors arise from the unique linguistic properties of the Chinese writing system. Unlike alphabetic languages where errors primarily stem from

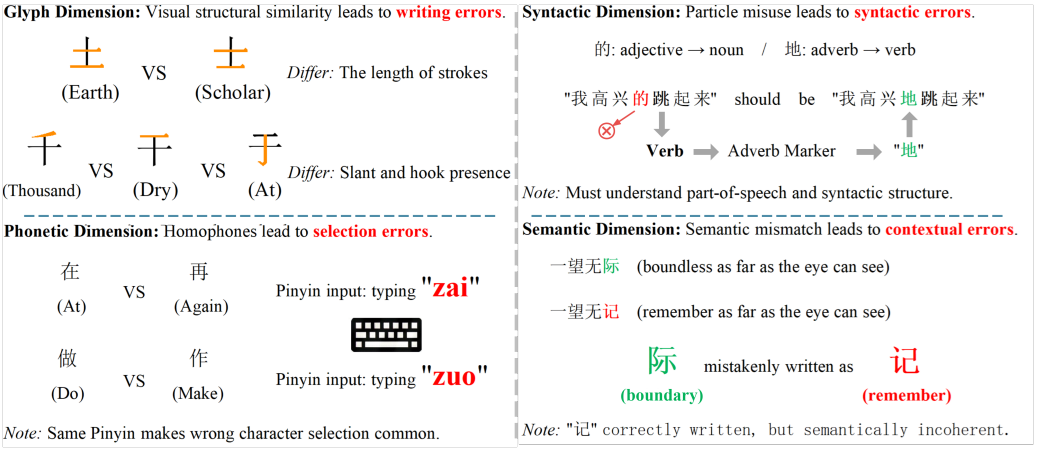


Figure 5: Chinese spelling errors across glyph, phonetic, syntactic, and semantic dimensions.

phonetic or typographical sources, Chinese characters simultaneously encode information across four interconnected dimensions: glyph, phonetic, syntactic, and semantic. This multi-dimensional nature implies that no single-view approach can achieve comprehensive error coverage, motivating our multi-teacher architecture where each specialized teacher focuses on a distinct dimension. This appendix provides detailed explanations of the illustrative examples in Figure 5, the linguistic foundations of each error dimension, and the complexity introduced by multiple co-occurring errors.

A.1.1 Illustrative Examples

Glyph Errors (Top-Left Panel). As shown in the top-left panel of Figure 5, glyph-based errors arise from visual structural similarity between characters. Two representative examples illustrate this phenomenon: the characters for “Earth” and “Scholar” differ only in the relative length of horizontal strokes, while “Thousand”, “Dry”, and “At” are distinguished solely by the presence of slants and hooks. Such minimal visual distinctions make these characters prone to confusion during handwriting or optical character recognition.

Phonetic Errors (Bottom-Left Panel). The bottom-left panel of Figure 5 demonstrates phonetic errors caused by homophones sharing identical pinyin representations. For instance, typing “zai” may produce either the character meaning “At” or “Again”, and typing “zuo” may yield either “Do” or “Make”. The keyboard icon emphasizes that these errors originate from pinyin-based input methods, where users must select among phonetically identical candidates.

Syntactic Errors (Top-Right Panel). The top-right panel of Figure 5 illustrates syntactic errors involving grammatical particle misuse. The example sentence “I happily jump up” contains an incorrect particle: the adjectival particle (connecting adjectives to nouns) is erroneously used instead of the adverbial particle (connecting adverbs to verbs). The diagram shows the correction flow from verb to adverb marker, indicating the required grammatical structure.

Semantic Errors (Bottom-Right Panel). The bottom-right panel of Figure 5 presents semantic errors where characters are orthographically correct but contextually incoherent. The idiom meaning “boundless as far as the eye can see” is mistakenly written by substituting the character for “boundary” with the phonetically similar character for “remember”. The red and green annotations highlight that although the erroneous character is correctly written in isolation, it renders the phrase semantically meaningless.

A.1.2 Linguistic Foundations

Glyph Dimension. The glyph dimension concerns the visual and structural properties of Chinese characters. Each character is composed of strokes organized into radicals and components following specific spatial arrangements. Glyph-based spelling errors occur when writers confuse characters that share similar visual structures, such as identical radicals, comparable stroke counts, or analogous component layouts. These errors are particularly prevalent in handwriting scenarios and may also arise from optical character recognition failures. Detecting such errors requires analyzing fine-grained structural features including stroke types, radical positions, and component configurations.

Phonetic Dimension. The phonetic dimension pertains to the pronunciation of Chinese characters represented through the pinyin romanization system. Chinese contains numerous homophones, where distinct characters share identical pronunciations including initial consonants, final vowels, and tonal patterns. Phonetic spelling errors emerge predominantly from pinyin-based input methods, where users type romanized syllables and select from candidate characters. When multiple characters correspond to the same pinyin input, incorrect selection leads to homophone substitution errors. Correcting these errors necessitates contextual disambiguation to identify the semantically appropriate character among phonetically equivalent alternatives.

Syntactic Dimension. The syntactic dimension addresses grammatical structures and functional word usage in Chinese sentences. Chinese employs various grammatical particles that serve distinct syntactic roles, including structural particles for modification relationships, aspect markers for temporal information, and modal particles for sentence mood. Syntactic spelling errors occur when writers misuse particles that share similar pronunciations or appearances but fulfill different grammatical functions. Resolving these errors requires understanding part-of-speech categories, dependency relations, and phrase structure constraints that govern legitimate particle selection.

Semantic Dimension. The semantic dimension encompasses contextual meaning, collocational patterns, and discourse coherence. Semantic spelling errors involve characters that are orthographically, phonetically, and syntactically valid but produce contextually inappropriate or meaningless expressions. These errors frequently occur in idiomatic phrases, fixed collocations, and domain-specific terminology where specific character combinations carry conventionalized meanings. Detecting semantic errors demands comprehensive understanding of lexical semantics, collocational restrictions, and pragmatic conventions that determine contextually coherent character usage.

A.1.3 Error Complexity in Real-World Scenarios

Real-world Chinese text often contains multiple spelling errors within a single sentence, where errors may originate from the same dimension or span across different dimensions. The co-occurrence of multiple errors significantly increases correction difficulty, as errors may interact with each other and obscure the original intended meaning. This subsection presents examples with varying error counts to illustrate the escalating complexity that motivates our multi-teacher integration approach.

One Error in a Sentence. Figure 6 presents two cases of Chinese spelling errors, where each sentence contains exactly one error. The first example illustrates a phonological error in which two characters share similar pronunciations but convey different meanings, causing confusion between "dreamland" and "environment". The second example demonstrates a morphological error involving two visually similar characters, "himself" and "already", respectively. Such errors require different correction strategies based on their underlying error types.

Two Errors in a Sentence. Figure 7 shows two cases of Chinese spelling errors, where each sentence contains two errors. The first example illustrates a phonological error where "sword" and "suggestion" share similar pronunciations in Chinese, leading to semantic incoherence in the original sentence. The second example combines a phonological and a morphological error: the homophones "its" and "he" are confused, and the visually similar "say" and "day" are confused due to their similarity in glyph structure. These cases highlight the increased complexity when multiple errors co-occur within a single sentence.

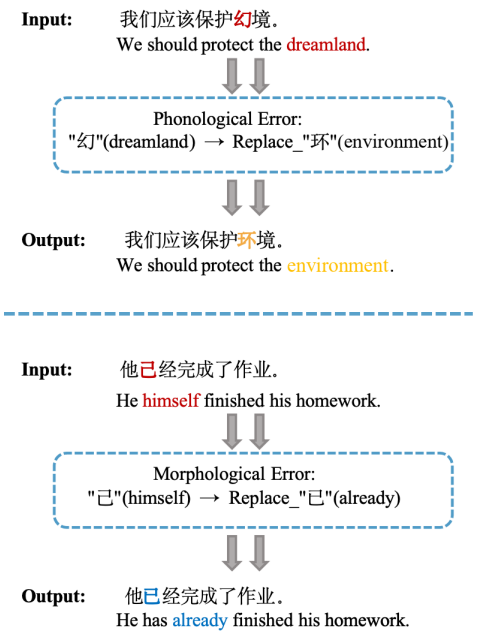


Figure 6: The illustration of one error in a sentence.

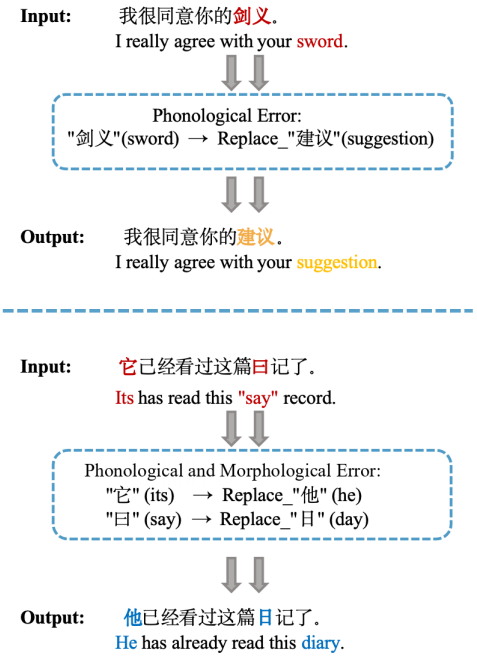


Figure 7: The illustration of two errors in a sentence.

Three Errors in a Sentence. Figure 8 presents two cases of Chinese spelling errors, where each sentence contains three errors. The first example illustrates phonological errors, where “again”, “length”, and “not” are incorrectly substituted for “at”, “field”, and “step”, respectively, due to similar pronunciations. The second example demonstrates three simultaneous errors involving both phonological and morphological confusion, where “material”, “sit” and “Qiao” should be corrected to “just”, “measure”, and “bridge”, respectively. These scenarios demonstrate that effective correction requires the integration of multiple specialized perspectives to disentangle complex error interactions.

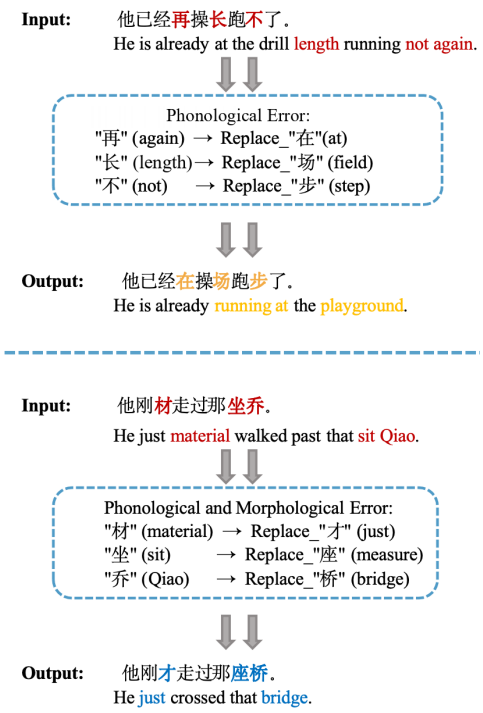


Figure 8: The illustration of three errors in a sentence.

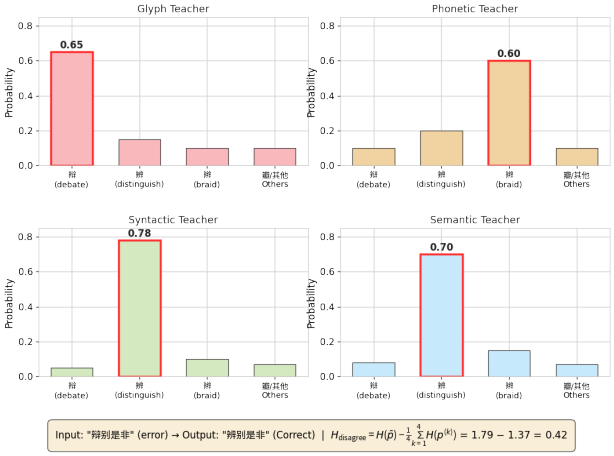


Figure 9: Multi-teacher disagreement illustrated through softmax probability distributions. Each panel shows the output distribution from a specialized teacher over candidate characters, with the highest probability candidate highlighted. The disagreement entropy $H_{\text{disagree}} = 0.42$ (computed in bits for this example) quantifies the divergence among teachers.

Appendix A.2: Multi-Teacher Disagreement

As discussed in Section 2, when multiple specialized teachers are employed to address different dimensions of Chinese spelling errors, they inevitably produce conflicting predictions for certain inputs. This appendix provides a detailed analysis of teacher disagreement through probability distributions, demonstrates the limitations of traditional aggregation strategies including majority voting and fixed weight schemes, and summarizes the key insights that motivate our adaptive integration approach.

A.2.1 Probability Distribution Analysis

Figure 9 illustrates a concrete example of multi-teacher disagreement through softmax probability distributions. The input sentence contains an erroneous character that should be corrected from "debate" to "distinguish" to form the semantically correct phrase. Each of the four specialized teachers produces a probability distribution over candidate characters based on its dimensional expertise. The Glyph Teacher assigns the highest probability (0.65) to "debate", keeping the input character. The Phonetic Teacher favors "braid" (0.60), another homophone of the input character. The Syntactic Teacher strongly supports "distinguish" (0.78) according to grammatical constraints. The Semantic Teacher also prefers "distinguish" (0.70) based on contextual coherence.

This example demonstrates that different teachers reach fundamentally different conclusions: the Glyph and Phonetic Teachers recommend incorrect candidates based on surface-level similarities, while the Syntactic and Semantic Teachers identify the correct answer through deeper linguistic analysis. The disagreement entropy $H_{\text{disagree}} = 0.42$ (in bits), computed as the difference between the entropy of the averaged distribution and the mean entropy of individual distributions (Eq. 6 in the main text), quantifies this divergence. Such disagreement signals that passive fusion through simple averaging would dilute the correct judgments, necessitating an active arbitration mechanism that can identify which teachers possess relevant expertise for each specific input.

A.2.2 Limitations of Majority Voting

Majority voting represents the simplest aggregation strategy, where each teacher casts an equal vote for its top prediction and the candidate receiving the most votes is selected. While intuitive, this approach suffers from fundamental limitations when teachers possess asymmetric expertise. The following seven cases demonstrate systematic failure patterns across different error types, revealing that majority voting cannot reliably resolve teacher disagreement regardless of the underlying error dimension. In this subsection and the next, the English rendering of each input gives its intended meaning.

Syntactic Error. Figure 10 illustrates the failure of majority voting in correcting syntactic errors. The input sentence contains a particle misuse where the complement particle should be replaced with the adverbial particle. The Glyph Teacher recommends retention due to visual dissimilarity, the Phonetic Teacher also suggests keeping because both characters are homophones, and the Semantic Teacher advises retention as the sentence remains contextually coherent. Only the Syntactic Teacher correctly identifies the error based on adverb placement rules. Under equal weight aggregation, the voting result yields 75% for retention, leading to incorrect correction and demonstrating the limitation of majority voting when specialized syntactic knowledge is required.

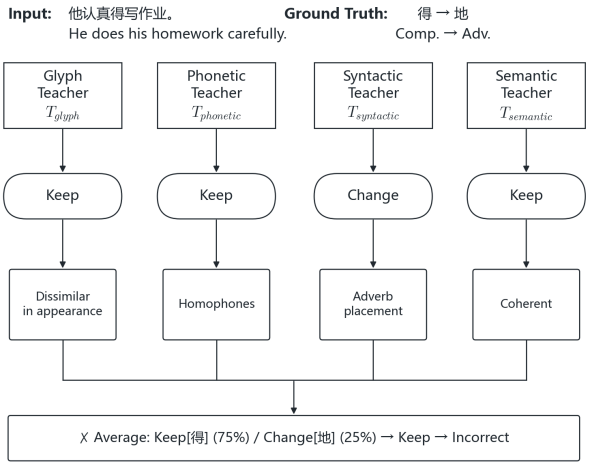


Figure 10: Illustration of majority voting failure in syntactic error.

Phonetic Error. Figure 11 illustrates the failure of majority voting in correcting phonetic errors. The input sentence "We are students" contains a homophone confusion where the character "city" should be corrected to the copula "is", as they share identical pronunciation in Chinese. The Glyph Teacher recommends correction due to visual dissimilarity between the two characters. The Phonetic Teacher suggests retention because both characters are homophones. The Syntactic Teacher also advises keeping since the sentence structure remains grammatically valid. The Semantic Teacher identifies the contextual contradiction and recommends correction. Under equal weight aggregation, the voting result yields 50% for retention and 50% for correction, leading to an ambiguous outcome and incorrect correction. This case demonstrates the limitation of majority voting when phonetic similarity masks the underlying error.

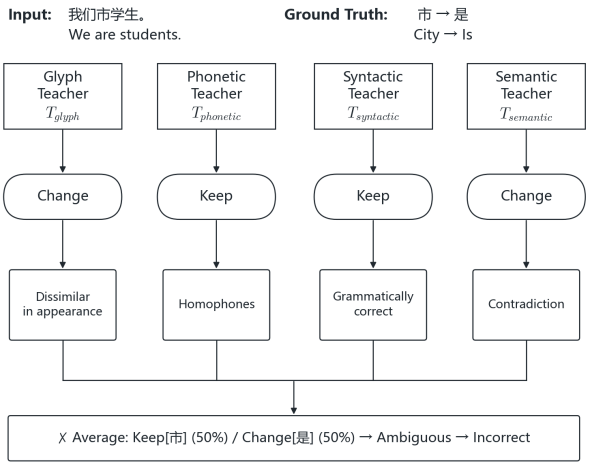


Figure 11: Illustration of majority voting failure in phonetic error.

Glyph Error. Figure 12 demonstrates the failure of majority voting in correcting glyph-based errors. The input sentence "The future is splendid" contains a character substitution where "last" should be corrected to "not" to form the word "future", as these two characters differ only in the relative length of horizontal strokes. The Glyph Teacher suggests retention precisely because the two characters are visually similar. The Phonetic Teacher recommends correction since the two characters have distinct pronunciations. The Syntactic Teacher advises keeping as the sentence maintains grammatical validity. The Semantic Teacher identifies that the erroneous combination renders the phrase meaningless and thus recommends correction. Under equal weight aggregation, the voting outcome splits evenly at 50% for retention versus 50% for correction, resulting in an ambiguous decision and failed correction. This case highlights the paradox that glyph errors can evade detection by glyph focused models when character similarity is extremely high.

Semantic Error. Figure 13 illustrates the failure of majority voting in correcting semantic errors. The input sentence "I have always supported you" contains an idiomatic error where the character "follow" should be corrected to "since" to form the correct four-character idiom meaning "as always". The Glyph Teacher recommends retention because the two characters are visually dissimilar. The Phonetic Teacher also suggests keeping since the two characters are homophones. The Syntactic Teacher advises retention as the sentence remains grammatically well-formed. Only the Semantic Teacher recognizes the violation of the idiomatic expression and recommends correction. Under equal weight aggregation, the voting result yields 75% for retention versus 25% for correction, causing the system to incorrectly preserve the erroneous character. This case reveals the critical limitation of majority voting when semantic knowledge about collocations and fixed expressions is required, as three teachers lacking such knowledge can easily outvote the single correct judgment.

Mixed Phonetic and Syntactic Error. Figure 14 presents the failure of majority voting in correcting errors involving multiple dimensions. The input sentence "She runs incredibly fast" contains a structural particle misuse where the attributive particle should be replaced with the complement

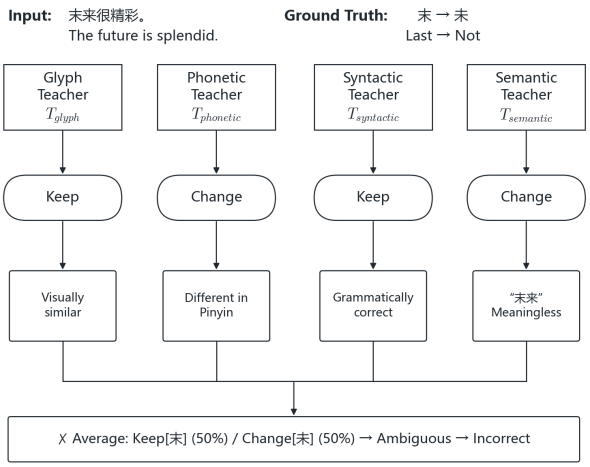


Figure 12: Illustration of majority voting failure in glyph error.

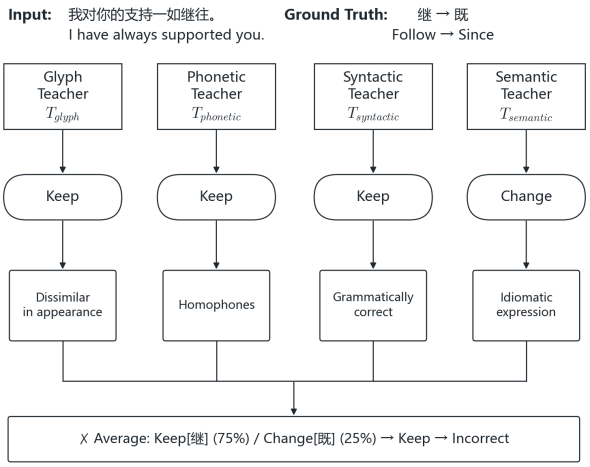


Figure 13: Illustration of majority voting failure in semantic error.

particle, as the latter is grammatically required before result complements in Chinese. The Glyph Teacher recommends retention because the two characters are visually dissimilar. The Phonetic Teacher suggests correction since both characters are homophones, indicating a potential input error. The Syntactic Teacher also recommends correction based on the grammatical rule that complement constructions require a specific particle. However, the Semantic Teacher advises keeping as the sentence remains contextually coherent. Under equal weight aggregation, the voting outcome results in 50% for retention versus 50% for correction, producing an ambiguous decision and failed correction. This case exposes the inherent difficulty when an error spans multiple linguistic dimensions, as teachers may reach contradictory conclusions based on their respective expertise, and simple averaging cannot resolve such conflicts.

Mixed Glyph and Semantic Error. Figure 15 illustrates the failure of majority voting in correcting errors that span glyph and semantic dimensions. The input sentence "He has arrived" contains a character confusion where "self" should be corrected to "already" to form the grammatically correct perfect aspect marker, as these two characters differ only by a subtle stroke extension. The Glyph Teacher recommends correction because, despite high visual similarity, the two characters remain distinguishable upon close inspection. The Phonetic Teacher suggests retention since the two characters sound alike. The Syntactic Teacher also advises keeping as the sentence structure

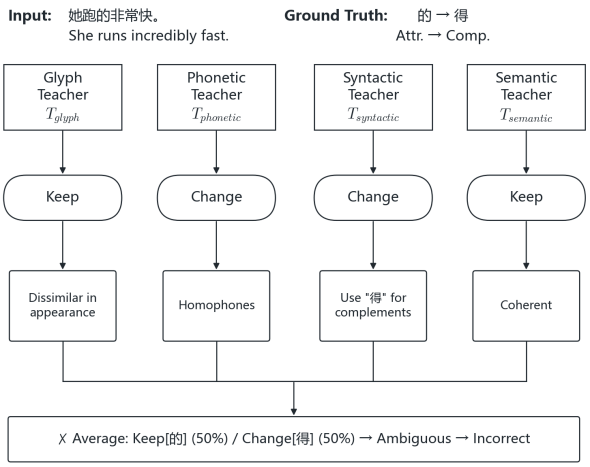


Figure 14: Illustration of majority voting failure in mixed phonetic and syntactic error.

remains grammatically valid. The Semantic Teacher recognizes that the erroneous character renders the sentence meaningless and recommends correction. Under equal weight aggregation, the voting outcome yields 50% for retention versus 50% for correction, leading to an ambiguous decision and failed correction. This case demonstrates that even when two teachers correctly identify an error from different perspectives, their combined influence may still be neutralized by the teachers that recommend retention.

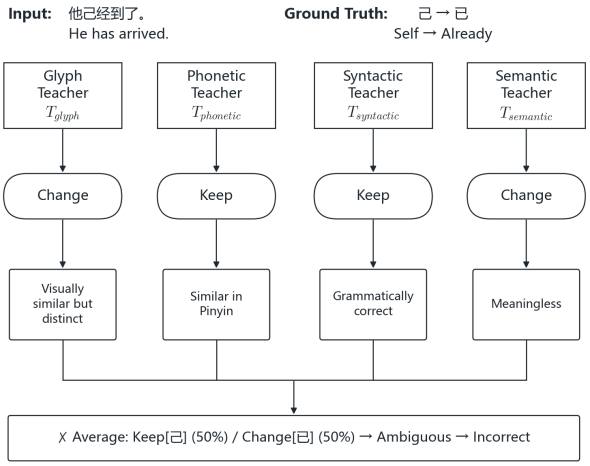


Figure 15: Illustration of majority voting failure in mixed glyph and semantic error.

Complete Teacher Disagreement. Figure 16 presents an extreme case where all four teachers reach fundamentally divided conclusions. The input sentence "He sits on the chair reading" contains a character substitution where "do" should be corrected to "sit" to match the intended meaning of the sentence. The Glyph Teacher recommends correction because, although the two characters share structural similarities, they remain visually distinguishable. The Phonetic Teacher also suggests correction since both characters are homophones, signaling a likely input method error. However, the Syntactic Teacher advises retention, judging the sentence grammatical. The Semantic Teacher similarly recommends keeping because it judges the sentence coherent. Under equal weight aggregation, the voting outcome splits exactly at 50% for retention versus 50% for correction, resulting in complete deadlock and failed correction. This case represents the most challenging scenario for majority voting, where teachers form two equally sized opposing coalitions based on their respective linguistic

perspectives, and no mechanism exists to break the tie or determine which coalition possesses the relevant expertise for the specific error at hand.

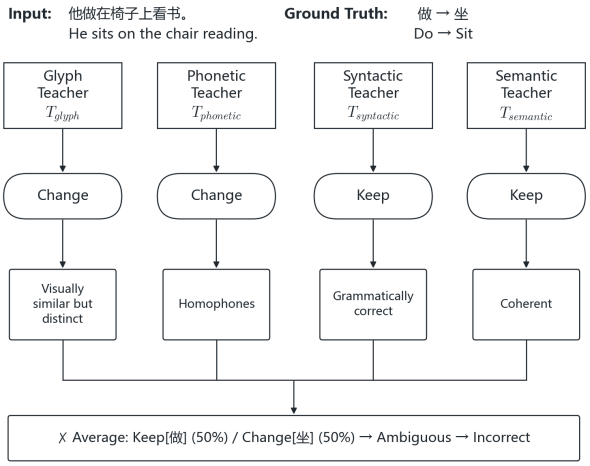


Figure 16: Illustration of majority voting failure in complete teacher disagreement.

A.2.3 Limitations of Fixed Weight Voting

Fixed weight voting attempts to address the limitations of majority voting by assigning predetermined importance weights to each teacher. However, this approach assumes that the relative importance of different dimensions remains constant across all inputs, which contradicts the fundamental observation that different error types require different expertise. The following seven cases demonstrate that no single weight configuration can achieve consistent correct corrections across diverse error types, as the optimal weight assignment varies depending on the specific error dimension involved.

Syntactic Error. Figure 17 illustrates how fixed weight voting fails to correct a syntactic error across multiple weight configurations. The input sentence "He does his homework carefully" contains a particle misuse where the complement particle should be replaced with the adverbial particle, as Chinese grammar requires the adverbial particle before the verb it modifies. The Glyph Teacher recommends retention due to visual dissimilarity between the characters. The Phonetic Teacher also suggests keeping since both characters are homophones. The Syntactic Teacher recognizes the adverb placement error and recommends correction. The Semantic Teacher advises retention as the sentence remains coherent. Under the first weight configuration assigning 0.5 to Glyph, 0.3 to Phonetic, 0.15 to Syntactic, and 0.05 to Semantic, the weighted score yields 0.85 for retention versus 0.15 for correction, resulting in incorrect preservation. Under the second configuration with weights 0.2, 0.5, 0.2, and 0.1 respectively, the outcome shifts to 0.8 for retention versus 0.2 for correction, still producing failed correction. Even under the third configuration prioritizing the Syntactic Teacher with weights 0.15, 0.2, 0.5, and 0.15, the result becomes 0.5 versus 0.5, a tie that defaults to retention and incorrect output. This case demonstrates that when only one teacher identifies the error, no fixed weight assignment can guarantee successful correction unless that teacher receives majority weight independently.

Phonetic Error. Figure 18 demonstrates how fixed weight voting can fail to correct a phonetic error under various weight distributions. The input sentence "We are students" contains a homophone substitution where "city" should be corrected to "is", as these two characters share identical pronunciation but convey entirely different meanings. The Glyph Teacher recommends correction because the two characters are visually dissimilar. The Phonetic Teacher suggests retention since both characters are homophones. The Syntactic Teacher also advises keeping as the sentence structure remains grammatically valid. The Semantic Teacher recognizes the semantic contradiction and recommends correction. Under the first weight configuration assigning 0.5 to Glyph, 0.3 to Phonetic, 0.15 to Syntactic, and 0.05 to Semantic, the weighted score yields 0.45 for retention versus 0.55 for

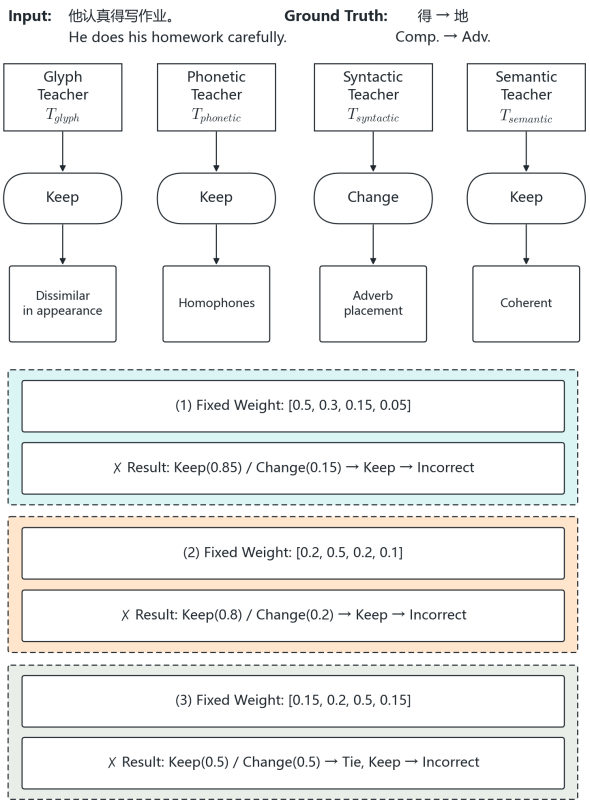


Figure 17: Illustration of fixed weight voting failure in syntactic error.

correction, so the correction succeeds. Under the second configuration with weights 0.2, 0.5, 0.2, and 0.1 respectively, the outcome becomes 0.7 for retention versus 0.3 for correction, leading to failed correction. Under the third configuration with weights 0.15, 0.2, 0.5, and 0.15, the result remains 0.7 for retention versus 0.3 for correction, still producing incorrect output. This case reveals that when the Phonetic Teacher fails to detect a homophone error, assigning higher weight to phonetic analysis paradoxically worsens performance by amplifying the blind spot.

Glyph Error. Figure 19 shows how fixed weight voting can fail to correct a glyph error despite varying weight assignments. The input sentence "The future is splendid" contains a character confusion where "last" should be corrected to "not", as these two characters differ only in the relative length of their two horizontal strokes but carry completely different meanings. The Glyph Teacher recommends retention because the two characters are visually similar. The Phonetic Teacher suggests correction since the characters have different pronunciations in Pinyin. The Syntactic Teacher advises keeping as the sentence structure remains grammatically correct. The Semantic Teacher recognizes that the erroneous character renders the sentence meaningless and recommends correction. Under the first weight configuration assigning 0.5 to Glyph, 0.3 to Phonetic, 0.15 to Syntactic, and 0.05 to Semantic, the weighted score yields 0.65 for retention versus 0.35 for correction, resulting in incorrect preservation. Under the second configuration with weights 0.2, 0.5, 0.2, and 0.1 respectively, the outcome shifts to 0.4 for retention versus 0.6 for correction, so the correction succeeds. Under the third configuration with weights 0.15, 0.2, 0.5, and 0.15, the result returns to 0.65 for retention versus 0.35 for correction, again leading to failed correction. This case highlights that when the Glyph Teacher misjudges visually similar characters, assigning dominant weight to glyph analysis leads to an erroneous outcome regardless of other teachers' correct assessments.

Semantic Error. Figure 20 presents how fixed weight voting fails to correct a semantic error across all weight configurations. The input sentence "I have always supported you" contains an idiomatic

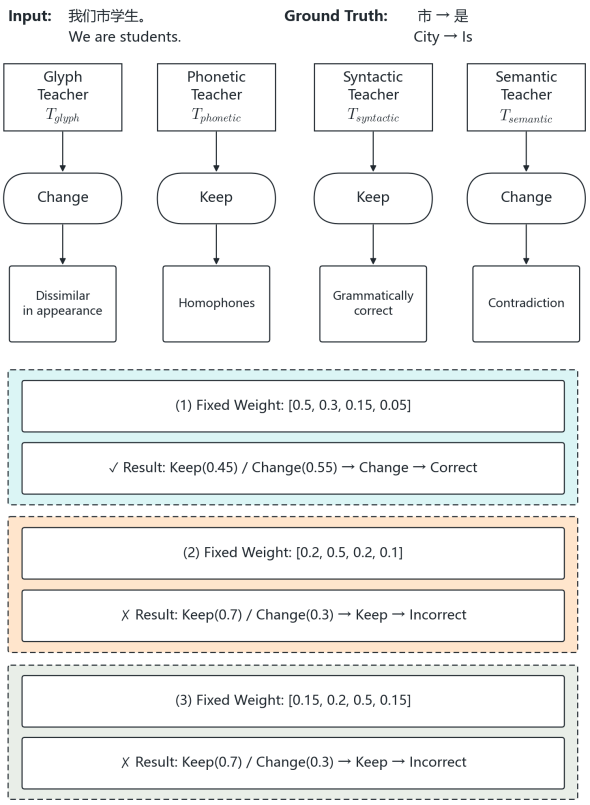


Figure 18: Illustration of fixed weight voting failure in phonetic error.

expression error where "follow" should be corrected to "since" to form the correct four character idiom meaning "as always". The Glyph Teacher recommends retention because the two characters are visually dissimilar. The Phonetic Teacher also suggests keeping since the two characters are homophones. The Syntactic Teacher advises retention as the sentence structure remains grammatically correct. Only the Semantic Teacher recognizes the idiomatic violation and recommends correction. Under the first weight configuration assigning 0.5 to Glyph, 0.3 to Phonetic, 0.15 to Syntactic, and 0.05 to Semantic, the weighted score yields 0.95 for retention versus 0.05 for correction, resulting in incorrect preservation. Under the second configuration with weights 0.2, 0.5, 0.2, and 0.1 respectively, the outcome remains 0.9 for retention versus 0.1 for correction, still producing failed correction. Under the third configuration with weights 0.15, 0.2, 0.5, and 0.15, the result shows 0.85 for retention versus 0.15 for correction, again leading to incorrect output. This case illustrates that semantic errors requiring cultural or idiomatic knowledge are particularly vulnerable to fixed weight voting, as the Semantic Teacher typically receives the lowest weight yet possesses the only relevant expertise for such errors.

Mixed Phonetic and Syntactic Error. Figure 21 demonstrates how fixed weight voting can fail to correct errors spanning multiple linguistic dimensions. The input sentence "She runs incredibly fast" contains a structural particle misuse where the attributive particle should be replaced with the complement particle, as Chinese grammar requires the complement particle for result complements. The Glyph Teacher recommends retention because the characters are visually dissimilar. The Phonetic Teacher suggests correction since the characters are homophones, indicating a potential input error. The Syntactic Teacher also recommends correction based on the grammatical rule requiring the complement particle. The Semantic Teacher advises keeping as the sentence remains contextually coherent. Under the first weight configuration assigning 0.5 to Glyph, 0.3 to Phonetic, 0.15 to Syntactic, and 0.05 to Semantic, the weighted score yields 0.55 for retention versus 0.45 for correction, resulting in incorrect preservation. Under the second configuration with weights 0.2,

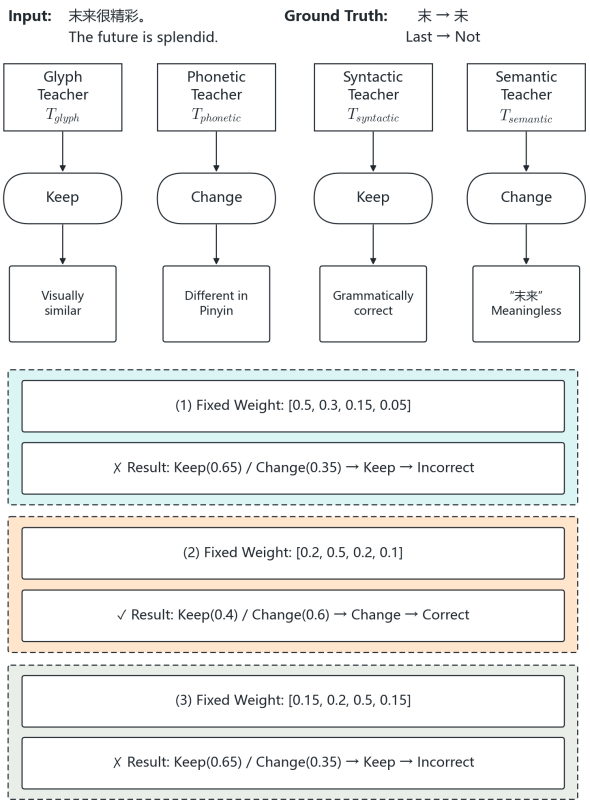


Figure 19: Illustration of fixed weight voting failure in glyph error.

0.5, 0.2, and 0.1 respectively, the outcome shifts to 0.3 for retention versus 0.7 for correction, so the correction succeeds, and the third configuration with weights 0.15, 0.2, 0.5, and 0.15 likewise yields 0.3 for retention versus 0.7 for correction. This case exposes a critical flaw in fixed weight voting: whether a multi-dimensional error is corrected is decided by whether the predetermined weights happen to favour the teachers that detect it, not by the input itself.

Mixed Glyph and Semantic Error. Figure 22 illustrates how fixed weight voting can fail to correct errors involving both glyph and semantic dimensions. The input sentence "He has arrived" contains a character confusion where "self" should be corrected to "already" to form the grammatically correct perfect aspect marker, as these two characters differ only by a subtle stroke extension. The Glyph Teacher recommends correction because, despite high visual similarity, the two characters remain distinguishable upon close inspection. The Phonetic Teacher suggests retention since the two characters sound alike. The Syntactic Teacher also advises keeping as the sentence structure remains grammatically correct. The Semantic Teacher recognizes that the erroneous character renders the sentence meaningless and recommends correction. Under the first weight configuration assigning 0.5 to Glyph, 0.3 to Phonetic, 0.15 to Syntactic, and 0.05 to Semantic, the weighted score yields 0.45 for retention versus 0.55 for correction, so the correction succeeds. Under the second configuration with weights 0.2, 0.5, 0.2, and 0.1 respectively, the outcome becomes 0.7 for retention versus 0.3 for correction, leading to failed correction. Under the third configuration with weights 0.15, 0.2, 0.5, and 0.15, the result remains 0.7 for retention versus 0.3 for correction, again producing incorrect output. This case demonstrates that when the two teachers with correct judgments receive lower weights in aggregate, their combined influence remains insufficient to override the incorrect majority formed by higher weighted teachers.

Complete Teacher Disagreement. Figure 23 presents how fixed weight voting can fail in extreme cases where teachers form two opposing coalitions. The input sentence "He sits on the chair reading"

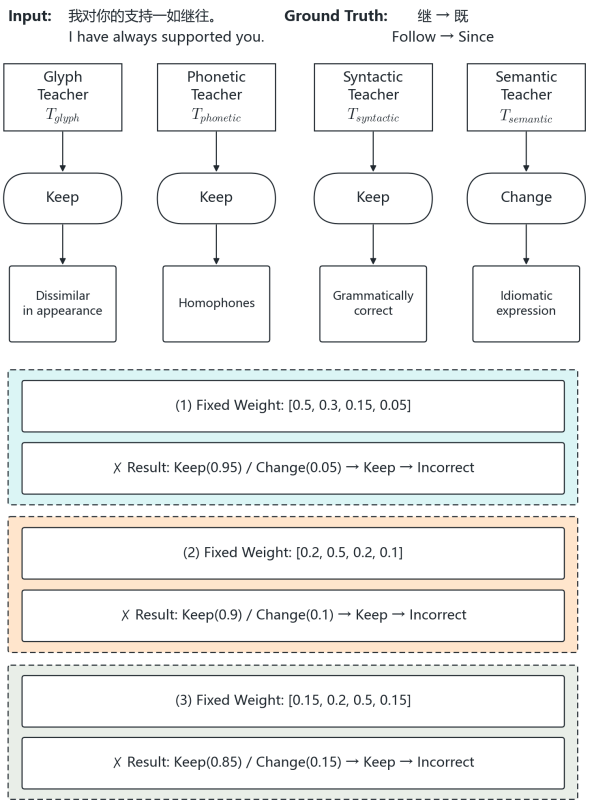


Figure 20: Illustration of fixed weight voting failure in semantic error.

contains a character substitution where "do" should be corrected to "sit" to match the intended meaning of the sentence. The Glyph Teacher recommends correction because, although the two characters share structural similarities, they remain visually distinguishable. The Phonetic Teacher also suggests correction since both characters are homophones, signaling a likely input method error. However, the Syntactic Teacher advises retention, judging the sentence grammatical. The Semantic Teacher similarly recommends keeping because it judges the sentence coherent. Under the first weight configuration assigning 0.5 to Glyph, 0.3 to Phonetic, 0.15 to Syntactic, and 0.05 to Semantic, the weighted score yields 0.2 for retention versus 0.8 for correction, so the correction succeeds. Under the second configuration with weights 0.2, 0.5, 0.2, and 0.1 respectively, the outcome shows 0.3 for retention versus 0.7 for correction, again yielding the correction. Under the third configuration with weights 0.15, 0.2, 0.5, and 0.15, the result reverses to 0.65 for retention versus 0.35 for correction, producing incorrect output. This case represents the fundamental limitation of fixed weight voting where the outcome entirely depends on which coalition receives higher aggregate weight, and no single weight configuration can guarantee correct decisions across all error types.

A.2.4 Summary and Implications

The analyses presented in Sections A.2.1 to A.2.3 reveal fundamental limitations of static aggregation strategies for multi-teacher integration. Majority voting fails when the teacher with correct expertise is outnumbered by teachers whose dimensional focus is irrelevant to the specific error type. Fixed weight voting fails because no single weight configuration can simultaneously optimize for all error types, as different errors require dynamically shifting trust among teachers.

These observations motivate the core design principles of SEMTD: (1) teacher disagreement should be treated as a valuable signal indicating samples requiring careful arbitration rather than noise to be averaged away; (2) the integration mechanism must be input-dependent, dynamically adjusting teacher weights based on the characteristics of each specific input; and (3) the student model must

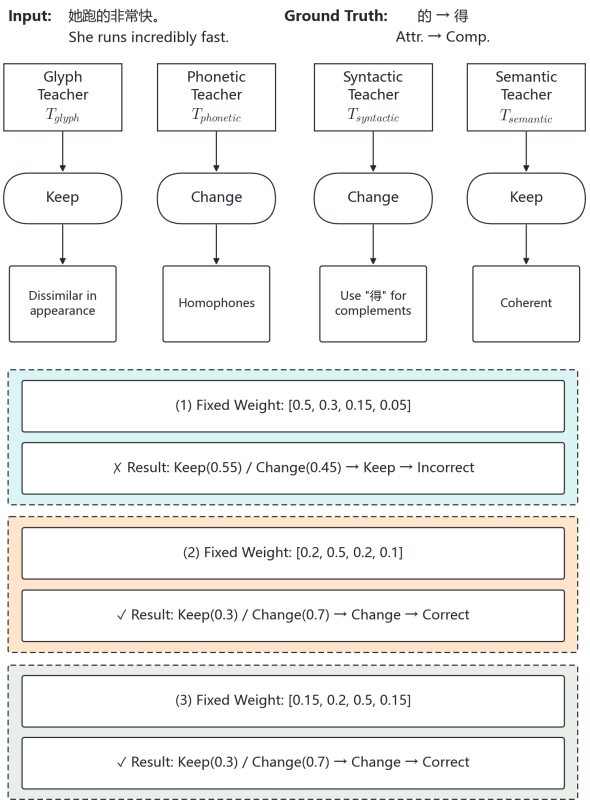


Figure 21: Illustration of fixed weight voting failure in mixed phonetic and syntactic error.

develop autonomous judgment capability to resolve conflicts when teachers provide contradictory guidance. The adaptive weight network, composite reward function, and reflective path optimization components of SEMTD directly address these requirements.

Appendix A.3: Teacher Specifications

This section describes the four expert teachers employed in SEMTD, each specialized in a distinct dimension of Chinese spelling errors.

Glyph Teacher. The glyph teacher T_{glyph} focuses on visual similarity between characters. Chinese spelling errors frequently arise from confusion between characters sharing similar radicals, strokes, or overall structures. This teacher evaluates whether the input character and candidate corrections exhibit visual resemblance, targeting errors caused by graphical similarity.

Phonetic Teacher. The phonetic teacher T_{phonetic} focuses on pronunciation similarity. Due to the prevalence of homophones in Chinese, errors often occur when characters with identical or similar Pinyin are substituted. This teacher evaluates phonetic proximity, targeting errors caused by homophone confusion.

Syntactic Teacher. The syntactic teacher $T_{\text{syntactic}}$ focuses on grammatical structures and function word usage. Chinese spelling errors in this dimension often involve misuse of structural particles, conjunctions, or other grammatical markers that govern relationships between words and phrases.

Semantic Teacher. The semantic teacher T_{semantic} focuses on contextual coherence. A character may be visually, phonetically, and syntactically valid yet produce semantically implausible text. This teacher evaluates whether corrections maintain meaning consistency within the surrounding context.

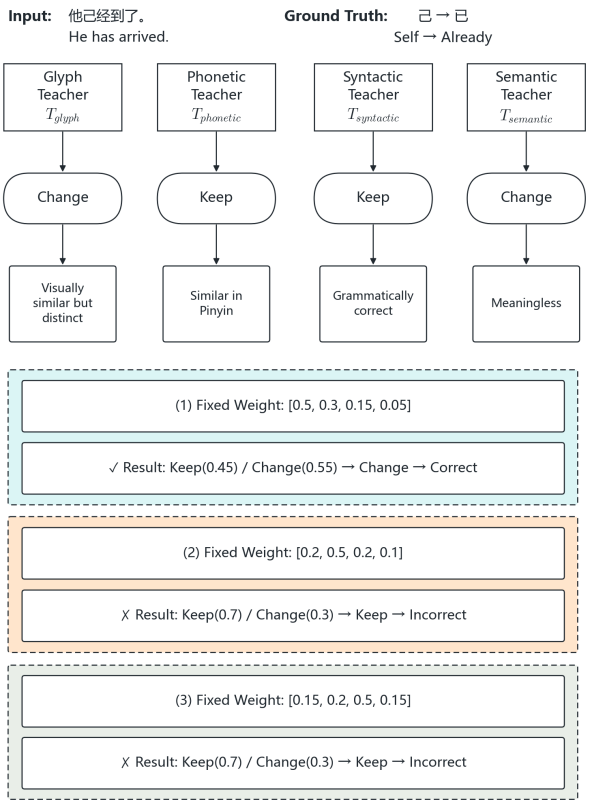


Figure 22: Illustration of fixed weight voting failure in mixed glyph and semantic error.

Appendix A.4: Multi-Teacher Distillation Algorithm

This section provides the algorithmic details for the multi-teacher distillation component described in Section 4.2. The core insight is treating teacher outputs as reward signals rather than targets for distribution matching, enabling active arbitration among experts through learned input-dependent weights.

A.4.1 Algorithm Overview

Algorithm 1 presents the procedure for multi-teacher distillation. The algorithm consists of three phases: (1) generating soft probability distributions from each specialized teacher, (2) computing adaptive weights via the weight network g_ω, and (3) aggregating teacher evaluations into reward signals for policy optimization.

A.4.2 Detailed Explanation

Phase 1: Teacher Output Generation (Lines 2 to 8). For each input $\mathbf{x} = (x_1, \ldots, x_n)$, we query all $K = 4$ expert teachers to obtain probability distributions over the vocabulary. Each teacher T_k processes input through its specialized prompt (detailed in Appendix A.9) and produces logits $\mathbf{z}^{(k)} \in \mathbb{R}^{n \times |\mathcal{V}|}$. The temperature-scaled softmax with $\tau = 3.0$ produces soft distributions that preserve candidate rankings while enabling effective knowledge transfer. These distributions are pre-computed and cached to eliminate redundant teacher inference during training.

Phase 2: Adaptive Weight Computation (Lines 10 to 15). The weight network g_ω determines input-dependent trust allocation across teachers. The student encoder produces hidden representations

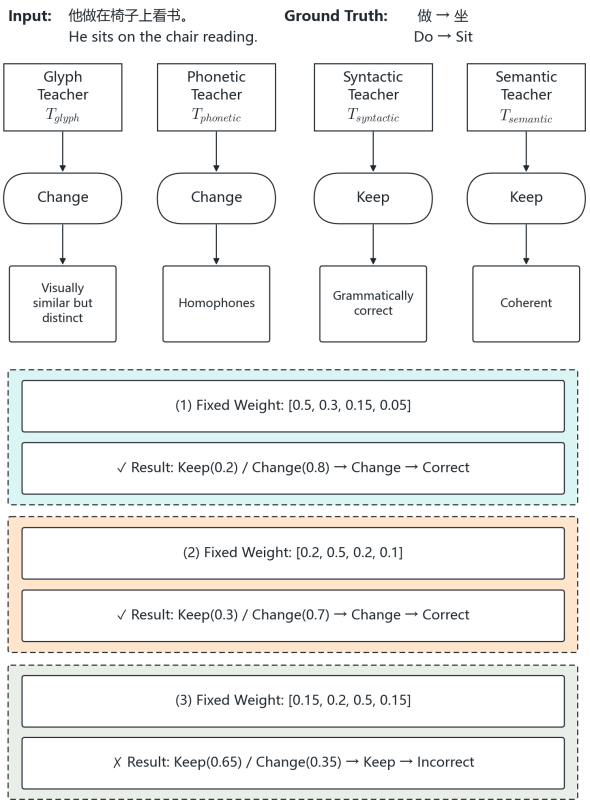


Figure 23: Illustration of fixed weight voting failure in complete teacher disagreement.

$\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_n)$, which are average-pooled to obtain $\bar{\mathbf{h}} \in \mathbb{R}^{d_{\text{model}}}$. A two-layer MLP maps this to teacher scores $\mathbf{s} \in \mathbb{R}^K$, normalized via softmax to weights $\mathbf{w}(\mathbf{x}) \in \Delta^K$.

Phase 3: Reward Signal Computation (Lines 17 to 22). Unlike traditional distillation minimizing KL divergence, we compute rewards measuring teacher approval of student predictions. The base reward R_k (Eq. 3) is the average probability assigned by teacher T_k to the student's predicted characters. High rewards indicate alignment with teacher expectations. The weighted aggregation produces the final reward guiding policy optimization, where weights reflect learned input-dependent trust.

Computational Complexity. Teacher inference is $O(K \cdot n \cdot |\mathcal{V}|)$ per sample but performed only once. The weight network adds $O(d_{\text{model}} \cdot d_{\text{hidden}} + d_{\text{hidden}} \cdot K)$ per forward pass. During training, only cached distributions and the lightweight weight network are accessed.

Appendix A.5: Self-Evolving Learning Algorithm

This section details the self-evolving learning component described in Section 4.3. The algorithm transforms teacher disagreement from noise into a valuable learning signal, enabling autonomous judgment through PPO optimization with a carefully designed composite reward.

A.5.1 Algorithm Overview

Algorithm 2 presents the self-evolving learning procedure. The key innovation is the composite reward (Eq. 7) that provides difficulty bonuses for correctly resolving high-disagreement cases and diversity bonuses to prevent single-teacher collapse.

Algorithm 1 Multi-teacher distillation.

Require: Data $\mathcal{D}$, teachers $\{T_k\}_{k=1}^{K}$, encoder f_θ, weight network g_ω, temperature τ
Ensure: Reward signals $\{R_{\text{weighted}}\}$, g_ω

// Phase 1: Teacher Output Generation
for each $\mathbf{x} \in \mathcal{D}$ **do**
  for $k = 1, \ldots, K$ **do**
    $\mathbf{z}^{(k)} \leftarrow T_k(\mathbf{x})$
    $\mathbf{p}^{(k)} \leftarrow \text{softmax}(\mathbf{z}^{(k)}/\tau)$ ▷ Eq. 2
  end for
  Cache $\{\mathbf{p}^{(k)}\}_{k=1}^{K}$
end for
// Phase 2: Adaptive Weight Computation
for each $\mathbf{x} \in \mathcal{D}$ **do**
  $\mathbf{H} \leftarrow f_\theta(\mathbf{x})$
  $\bar{\mathbf{h}} \leftarrow \frac{1}{n}\sum_{i=1}^{n} \mathbf{h}_i$
  $\mathbf{s} \leftarrow \mathbf{W}_2\sigma(\mathbf{W}_1\bar{\mathbf{h}} + \mathbf{b}_1) + \mathbf{b}_2$ ▷ Eq. 5
  $\mathbf{w}(\mathbf{x}) \leftarrow \text{softmax}(\mathbf{s})$ ▷ Eq. 4
end for
// Phase 3: Reward Signal Computation
for each $(\mathbf{x}, \hat{\mathbf{y}}) \in \mathcal{D}$ **do**
  for $k = 1, \ldots, K$ **do**
    $R_k \leftarrow \frac{1}{n}\sum_{i=1}^{n} p_{i,\hat{y}_i}^{(k)}$ ▷ Eq. 3
  end for
  $R_{\text{weighted}} \leftarrow \sum_{k=1}^{K} w_k(\mathbf{x}) \cdot R_k$
end for
return $\{R_{\text{weighted}}\}$, g_ω

A.5.2 Detailed Explanation

Step 1: Rollout Collection (Lines 4 to 6). The current policy π_θ generates predictions by sampling from the output distribution. Simultaneously, g_ω computes input-dependent teacher weights. These rollouts form the basis for reward computation and gradient estimation.

Step 2: Disagreement Quantification (Lines 8 to 11). The disagreement entropy H_{disagree} (Eq. 6) measures teacher divergence as the difference between the entropy of the average prediction and the average of individual entropies. When teachers agree, $H_{\text{disagree}} \approx 0$; when they propose different corrections, it increases. This identifies samples where passive absorption fails and independent judgment is essential.

Step 3: Composite Reward (Lines 13 to 18). The composite reward integrates three components:

- **Weighted Teacher Reward** (R_{teacher}): Aggregates individual teacher evaluations according to learned weights, enabling differential trust across experts.
- **Difficulty Bonus** (R_{diff}): Provides additional reward proportional to disagreement when the student correctly resolves the input. With $\beta = 0.5$, this creates strong incentives for developing robust judgment on hard cases.
- **Diversity Bonus** (R_{div}): Rewards entropy in the weight distribution with $\mu = 0.1$, preventing collapse to single-teacher reliance while remaining subordinate to the primary objective.

Step 4: Advantage Estimation (Lines 20 to 23). The advantage $\hat{A}$ measures relative prediction quality against an exponential moving average baseline. The decay factor 0.99 provides stable estimates while adapting to training dynamics.

Step 5: PPO Policy Update (Lines 25 to 33). The PPO objective (Eq. 9) optimizes the policy while constraining update magnitude. The probability ratio $r(\theta)$ measures policy change; clipping to $[1-\epsilon, 1+\epsilon]$ with $\epsilon = 0.2$ prevents catastrophically large updates. Crucially, both θ and ω receive

Algorithm 2 Self-evolving learning with PPO.

Require: Policy π_θ, weight network g_ω, teachers $\{\mathbf{p}^{(k)}\}$, ground truth $\mathbf{y}^*$
Ensure: Updated π_θ, g_ω

Initialize baseline $b \leftarrow 0$, old policy $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
for each iteration **do**
  // Step 1: Collect Rollouts
  for each $\mathbf{x}$ in batch **do**
    $\hat{\mathbf{y}} \sim \pi_\theta(\cdot|\mathbf{x})$; $\mathbf{w}(\mathbf{x}) \leftarrow g_\omega(\mathbf{x})$
  end for
  // Step 2: Compute Disagreement
  for each $\mathbf{x}$ in batch **do**
    $\bar{\mathbf{p}}_i \leftarrow \frac{1}{K}\sum_{k=1}^{K} \mathbf{p}_i^{(k)}$
    $H_{\text{disagree}} \leftarrow \frac{1}{n}\sum_i \left[H(\bar{\mathbf{p}}_i) - \frac{1}{K}\sum_k H(\mathbf{p}_i^{(k)})\right]$ ▷ Eq. 6
  end for
  // Step 3: Compute Composite Reward
  for each $(\mathbf{x}, \hat{\mathbf{y}})$ in batch **do**
    $R_{\text{teacher}} \leftarrow \sum_k w_k(\mathbf{x}) \cdot R_k(\mathbf{x}, \hat{\mathbf{y}})$
    $R_{\text{diff}} \leftarrow \beta \cdot H_{\text{disagree}} \cdot \mathbb{1}[\hat{\mathbf{y}} = \mathbf{y}^*]$
    $R_{\text{div}} \leftarrow \mu \cdot \mathcal{H}(\mathbf{w}(\mathbf{x}))$
    $R \leftarrow R_{\text{teacher}} + R_{\text{diff}} + R_{\text{div}}$ ▷ Eq. 7
  end for
  // Step 4: Compute Advantages
  for each $(\mathbf{x}, \hat{\mathbf{y}})$ in batch **do**
    $\hat{A} \leftarrow R - b$ ▷ Eq. 8
  end for
  $b \leftarrow 0.99 \cdot b + 0.01 \cdot \text{mean}(R)$
  // Step 5: PPO Update
  for $e = 1, \ldots, E$ **do**
    for each mini-batch **do**
      $r(\theta) \leftarrow \pi_\theta(\hat{\mathbf{y}}|\mathbf{x}) / \pi_{\theta_{\text{old}}}(\hat{\mathbf{y}}|\mathbf{x})$
      $L_{\text{clip}} \leftarrow \min(r\hat{A}, \text{clip}(r, 1-\epsilon, 1+\epsilon)\hat{A})$
      $\theta \leftarrow \theta + \alpha_\pi \nabla_\theta \mathbb{E}[L_{\text{clip}}]$ ▷ Eq. 9
      $\omega \leftarrow \omega + \alpha_\omega \nabla_\omega \mathbb{E}[L_{\text{clip}}]$
    end for
  end for
  $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
end for
return π_θ, g_ω

gradients, enabling joint optimization where the weight network learns to favor teachers whose guidance yields higher rewards.

Appendix A.6: Reflective Path Optimization Algorithm

This section details the reflective path optimization component described in Section 4.4. The algorithm implements a reflect-then-consolidate strategy through uncertainty-driven sampling, consistency evaluation, and path memory for cross-sample transfer.

A.6.1 Algorithm Overview

Algorithm 3 presents the reflective path optimization procedure. The key insight is that not all samples provide equal learning value, and successful decision paths should be consolidated for transfer to similar inputs.

Algorithm 3 Reflective path optimization.

Require: Policy π_θ, weight network g_ω, data $\mathcal{D}$, memory $\mathcal{M}$
Ensure: Updated π_θ, enriched $\mathcal{M}$

1: **// Phase 1: Uncertainty Quantification**
2: **for** each $\mathbf{x} \in \mathcal{D}$ **do**
3: $U_{\text{pred}} \leftarrow H(\pi_\theta(\cdot|\mathbf{x}))$
4: $U_{\text{disagree}} \leftarrow H_{\text{disagree}}(\mathbf{x})$
5: $U_{\text{hist}} \leftarrow 1 - R_{\text{hist}}(\mathbf{x})$
6: $\mathcal{U}(\mathbf{x}) \leftarrow U_{\text{pred}} + \gamma U_{\text{disagree}} + \delta U_{\text{hist}}$ ▷ Eq. 10
7: **end for**
8: **// Phase 2: Priority Sampling**
9: $P_\phi(\mathbf{x}) \leftarrow \exp(\eta \mathcal{U}(\mathbf{x})) / \sum_{\mathbf{x}'} \exp(\eta \mathcal{U}(\mathbf{x}'))$ ▷ Eq. 11
10: Sample batch $\mathcal{B} \sim P_\phi$
11: **// Phase 3: Consistency Evaluation**
12: **for** each $\mathbf{x} \in \mathcal{B}$ **do**
13: **for** $m = 1, \ldots, M$ **do**
14: $\hat{\mathbf{y}}^{(m)} \sim \pi_\theta(\cdot|\mathbf{x})$ with dropout
15: **end for**
16: $C(\mathbf{x}) \leftarrow \frac{2}{M(M-1)} \sum_{i<j} \frac{1}{n} \sum_l \mathbb{1}[\hat{y}_l^{(i)} = \hat{y}_l^{(j)}]$ ▷ Eq. 12
17: $\hat{\mathbf{y}}_{\text{cons}} \leftarrow \text{MajorityVote}(\{\hat{\mathbf{y}}^{(m)}\}_{m=1}^M)$
18: **end for**
19: **// Phase 4: Consistency-Enhanced Learning (rollout action $\hat{\mathbf{y}}$)**
20: **for** each $(\mathbf{x}, \hat{\mathbf{y}}) \in \mathcal{B}$ **do**
21: $R_{\text{rollout}} \leftarrow R(\mathbf{x}, \hat{\mathbf{y}}; w_\omega)$
22: $\hat{A}_C \leftarrow C(\mathbf{x}) \cdot (R_{\text{rollout}} - b)$ ▷ Eq. 13: shaping coeff., not action substitution
23: Update π_θ using $\hat{A}_C$ via PPO with ratio $r(\theta)$ on $\hat{\mathbf{y}}$
24: **end for**
25: **// Phase 5: Path Memory Consolidation (separate evaluation on consensus)**
26: **for** each $\mathbf{x} \in \mathcal{B}$ **do**
27: $R_{\text{cons}} \leftarrow R(\mathbf{x}, \hat{\mathbf{y}}_{\text{cons}}; w_\omega)$ ▷ evaluated separately for write predicate
28: **if** $R_{\text{cons}} > \tau_R$ **and** $C(\mathbf{x}) > \tau_C$ **then**
29: $\mathcal{M} \leftarrow \mathcal{M} \cup \{(\mathbf{x}, g_\omega(\mathbf{x}), \hat{\mathbf{y}}_{\text{cons}})\}$ ▷ FIFO eviction if full
30: **end if**
31: **end for**
32: **// Phase 6: Cross-Sample Transfer**
33: **for** each novel $\mathbf{x}_{\text{new}}$ **do**
34: $\mathcal{N} \leftarrow \text{kNN}(\bar{\mathbf{h}}(\mathbf{x}_{\text{new}}), \mathcal{M}, k)$
35: $\mathbf{w}_{\text{prior}} \leftarrow \frac{1}{|\mathcal{N}|} \sum_{(\cdot, \mathbf{w}^*, \cdot) \in \mathcal{N}} \mathbf{w}^*$
36: Initialize $g_\omega(\mathbf{x}_{\text{new}})$ with $\mathbf{w}_{\text{prior}}$
37: **end for**
38: $\eta \leftarrow \min(\eta + \Delta\eta, \eta_{\max})$
39: **return** π_θ, $\mathcal{M}$

A.6.2 Detailed Explanation

Phase 1: Uncertainty Quantification (Lines 2 to 7). The uncertainty score $\mathcal{U}(\mathbf{x})$ (Eq. 10) combines three complementary signals: (1) *predictive uncertainty* U_{pred}, entropy of the student's output distribution indicating model confidence; (2) *teacher disagreement* U_{disagree}, reusing the disagreement entropy from Section 4.3 to identify expert conflicts; and (3) *historical error rate* U_{hist}, tracking past accuracy to prioritize persistently difficult cases. The coefficients $\gamma = 0.3$ and $\delta = 0.2$ balance these components, with predictive uncertainty receiving implicit weight 1.0.

Phase 2: Priority Sampling (Lines 9 to 10). The sampling distribution P_ϕ (Eq. 11) assigns higher probability to uncertain samples. The concentration η controls prioritization sharpness: $\eta \to 0$ yields uniform sampling; $\eta \to \infty$ deterministically selects the most uncertain sample. We employ a phased schedule for η aligned with the three training phases: during warm-up (epochs 1 and 2), priority sampling is disabled with $\eta = 0$ to ensure uniform coverage; during adaptive learning (epochs 3 to

10), η increases linearly from 0.1 to 1.5; during refinement (epochs 11 to 15), η continues increasing from 1.5 to 2.0. This staged activation ensures stable early training while progressively focusing on difficult samples as the model matures.

Phase 3: Consistency Evaluation (Lines 12 to 18). To assess prediction stability, we perform $M = 5$ stochastic forward passes with dropout enabled. The consistency score $C(\mathbf{x})$ (Eq. 12) measures pairwise agreement across passes. High consistency ($C \to 1$) indicates stable decision-making; low consistency suggests genuine uncertainty. The consensus prediction $\hat{\mathbf{y}}_{\text{cons}}$ is obtained through majority voting, representing the model's most confident output.

Phase 4: Consistency-Enhanced Learning (Lines 20 to 24). The consistency-enhanced advantage $\hat{A}_C$ (Eq. 13) is evaluated on the rollout action $\hat{\mathbf{y}}$ (the action sampled from π_θ during the rollout) so that the PPO importance ratio $r(\theta)$ remains valid; the consistency score $C(\mathbf{x}) \in [0, 1]$ enters only as a scalar shaping coefficient that attenuates updates for inconsistent predictions. This down-weights noisy rollouts relative to stable ones.

Phase 5: Path Memory Consolidation (Lines 26 to 31). The memory write predicate is evaluated *separately* from the policy update: the consensus prediction $\hat{\mathbf{y}}_{\text{cons}}$ is scored under the same composite reward, and an entry is admitted only if both $R(\mathbf{x}, \hat{\mathbf{y}}_{\text{cons}}; w_\omega) > \tau_R = 0.8$ (high reward on the consensus) and $C(\mathbf{x}) > \tau_C = 0.7$ (stable prediction across the M stochastic passes). Each admitted entry contains the input $\mathbf{x}$ (indexed by its encoder representation $\bar{\mathbf{h}}(\mathbf{x})$ for efficient similarity-based retrieval), the learned weight configuration $g_\omega(\mathbf{x})$, and the consensus prediction $\hat{\mathbf{y}}_{\text{cons}}$. The bank has fixed capacity $|\mathcal{M}|_{\max} = 10{,}000$ with FIFO eviction once full; retrieval is performed with FAISS using IVF-PQ indexing.

Phase 6: Cross-Sample Transfer (Lines 33 to 37). For novel inputs, we retrieve $k = 5$ nearest neighbours from the memory bank by cosine similarity on the encoder representation $\bar{\mathbf{h}}(\mathbf{x})$. The retrieved weight configurations are averaged to form a prior $\mathbf{w}_{\text{prior}}$ that warm-starts the weight network on semantically similar inputs. **No-test-time-gradient guarantee:** during evaluation the path memory bank is strictly read-only: no entries are written, no parameters are updated, and no gradient flows through retrieved entries; $\mathbf{w}_{\text{prior}}$ is used purely as a fixed forward-pass initialisation. This ensures that all reported test-time results reflect deterministic inference on a frozen model.

A.6.3 Integration with Training Pipeline

The three modules integrate with the training phases (Appendix A.11.1):

- **Epochs 1 and 2 (Warm-up)**: Priority sampling disabled ($\eta = 0$), consistency shaping and memory bank inactive. Uniform teacher weights $w_k = 0.25$.
- **Epochs 3 to 10 (Adaptive)**: Priority sampling enabled with η increasing from 0.1 to 1.5; consistency shaping, memory writes and cross-sample transfer remain inactive during this phase.
- **Epochs 11 to 15 (Refinement)**: Full reflective optimization with η increasing from 1.5 to 2.0; the consistency-shaped advantage (Eq. 13), memory writes and cross-sample transfer are active.

This staged activation ensures stable early training while progressively enabling sophisticated learning mechanisms as foundational capabilities develop.

SEL / RPO gating. SEL and RPO are not separate optimisation loops on different mini-batches; they share a single PPO update step on each rollout batch. Concretely, during the PPO phases (Epochs 3 to 15, Appendix A.11.1) we compute on every batch (i) the SEL composite reward R (Eq. 7) and the base advantage $\hat{A}$ (Eq. 8) on the rollout action $\hat{\mathbf{y}}$, and, once consistency shaping is active (Epochs 11 to 15), (ii) the RPO consistency coefficient $C(\mathbf{x})$ from the M stochastic passes. The actual quantity passed into the PPO clipped objective (Eq. 9) is the consistency-shaped advantage $\hat{A}_C = C(\mathbf{x}) \cdot \hat{A}$ (Eq. 13); when $C(\mathbf{x}) = 1$ this reduces to plain SEL. RPO's path-memory write (Eq. 14) and cross-sample read (Eq. 15) operate alongside but do not modify the PPO gradient: they only re-initialise g_ω's forward pass on subsequent samples. RPO therefore augments rather than replaces SEL.

A.6.4 Closed-Loop Integration Diagram

Figure 4 provides a visual summary of the SEMTD training loop, showing how the three components feed into each other once all of them are active (Epochs 11 to 15, Appendix A.6.3). Multi-teacher distillation produces input-dependent adaptive weights $w_\omega(\mathbf{x})$ that aggregate per-teacher reward signals; self-evolving learning shapes those into a composite reward R and computes the PPO advantage; reflective path optimization scales the advantage by the consistency coefficient $C(\mathbf{x})$ derived from M stochastic passes and consolidates successful decision paths into the memory bank. The enriched memory in turn warm-starts the weight network on subsequent batches, closing the loop. The diagram visualises the same three-component architecture described in Section 4.5 and the SEL/RPO gating detailed in Appendix A.6.3.

Appendix A.7: Software and Hardware

A.7.1 System Environment

Table 6: System environment specifications.

Component	Specification
Operating System	Ubuntu 22.04 LTS
Kernel Version	5.15.0-91-generic
CUDA Toolkit	12.1
cuDNN	8.9.0
NCCL	2.18.1
Driver Version	535.104.05

A.7.2 Hardware Configuration

Training and inference utilize a cluster of NVIDIA A100 80GB GPUs:

Table 7: Hardware configuration for training and inference.

Component	Hardware	Purpose
Student Training	4× A100 80GB	Policy optimization
Teacher Inference	2× A100 80GB	Soft label generation
CPU	AMD EPYC 7763 (64 cores)	Data preprocessing
System RAM	512GB DDR4	Batch construction
Storage	2TB NVMe SSD	Checkpoints
Interconnect	NVLink 3.0 (600 GB/s)	GPU communication

Memory Utilization. Student training with DeepSpeed ZeRO Stage 2 and gradient checkpointing consumes approximately 72GB per GPU. Teacher inference via vLLM with tensor parallelism (TP=2) requires approximately 65GB per GPU on 2 GPUs. BF16 mixed-precision is used throughout to balance numerical stability and memory efficiency.

A.7.3 Software Environment

Table 8: Software versions for reproducibility.

Package	Version	Package	Version	Package	Version	Package	Version
Python	3.10.12	PEFT	0.10.0	Flash Attention	2.5.6	CUDA Toolkit	12.1
PyTorch	2.2.0	TRL	0.8.0	vLLM	0.4.0	cuDNN	8.9.0
Transformers	4.51.0	DeepSpeed	0.14.0	FAISS	1.7.4	NCCL	2.18.1
Accelerate	0.28.0						

A.7.4 Random Seed Protocol

We follow the comprehensive seed-setting protocol for PyTorch reproducibility:

```
import random
import numpy as np
import torch

def set_seed(seed: int = 42):
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False
    os.environ["PYTHONHASHSEED"] = str(seed)
```

All experiments use seed 42.

A.7.5 Data Preprocessing

- **Tokenization**: Character-aligned tokenization using modified Qwen3 tokenizer. All 21,128 target Chinese characters are registered as special tokens to prevent subword splitting, ensuring one-to-one correspondence between input characters and token positions. See Appendix A.8.4 for details on vocabulary mapping.
- **Normalization**: Convert traditional Chinese to simplified; normalize punctuation to full-width
- **Length Filtering**: Discard samples with <5 or >512 characters
- **Deduplication**: Remove exact duplicates by MD5 hash
- **Train/Val Split**: 95%/5% stratified by error type distribution

A.7.6 Evaluation Protocol

Following the SIGHAN official evaluation:

- **Detection**: A sentence is correctly detected if all error positions are identified
- **Correction**: A sentence is correctly corrected if all characters match the gold standard
- **Metrics**: Sentence-level Precision, Recall, F1 for both Detection and Correction

Appendix A.8: Model Architecture

The student model is designed as a lightweight 4B-parameter transformer capable of autonomous judgment after distillation from multiple expert teachers.

A.8.1 Base Architecture

We initialize from Qwen3-4B [63], a 4B-parameter dense transformer released by Alibaba Cloud in April 2025.

The Grouped Query Attention (GQA) mechanism reduces memory footprint by sharing key-value heads across multiple query heads, achieving a 4:1 ratio (32 query heads to 8 KV heads). This reduces the key-value cache by 75% relative to standard multi-head attention while maintaining comparable performance.

Initialization and Distillation. The student model is initialized from the open-source Qwen3-4B checkpoint [63], a native 4B-parameter dense transformer that matches the performance of Qwen2.5-7B while using fewer parameters. The model is subsequently distilled from DeepSeek-R1-Distill-Qwen-14B, which serves as the shared backbone for all four expert teachers. This initialization strategy combines the benefits of a well-pretrained foundation with task-specific knowledge transfer from reasoning-capable teachers.

Table 9: Student model architecture specifications. The backbone vocabulary (151,669) is inherited from Qwen3 [63], while the CSC output vocabulary (21,128) is used for character-level prediction via a dedicated projection head.

Configuration	Value
Architecture	Dense Transformer
Base Model	Qwen3-4B
Total Parameters	4.0B
Non-Embedding Parameters	3.6B
Hidden Dimension (d_{model})	2,560
Intermediate Size (d_{ff})	9,728
Number of Layers	36
Attention Heads (Query)	32
Key-Value Heads (GQA)	8
Head Dimension	128
Backbone Vocabulary Size	151,669
CSC Output Vocabulary Size	21,128
Maximum Sequence Length	512
Positional Encoding	RoPE ($\theta = 10^6$)
Normalization	RMSNorm ($\epsilon = 10^{-6}$)
Activation Function	SwiGLU
Tie Word Embeddings (backbone)	True

A.8.2 Adaptive Weight Network g_ω

The weight network computes input-dependent teacher weights $\mathbf{w}(\mathbf{x}) \in \Delta^4$ from the student encoder's pooled representation (Eq. 4 and 5). Since the teacher selection task is relatively straightforward, we adopt a lightweight two-layer MLP architecture:

Table 10: Adaptive weight network architecture implementing g_ω in Eq. 4. The two-layer design is sufficient for this task as the input-dependent teacher selection does not require deep nonlinear transformations.

Component	Specification
Input	$\bar{\mathbf{h}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{h}_i \in \mathbb{R}^{2560}$
Hidden Layer	Linear(2560, 256) $\rightarrow$ GELU
Output Layer	Linear(256, 4) $\rightarrow$ Softmax
Dropout	0.1 (after GELU)
Initialization	Xavier uniform; bias initialized to zero
Total Parameters	0.66M

This architecture directly corresponds to Eq. 5 in the main text:

$$\mathbf{s}(\mathbf{x}) = \mathbf{W}_2 \cdot \sigma\big(\mathbf{W}_1 \cdot \bar{\mathbf{h}}(\mathbf{x}) + \mathbf{b}_1\big) + \mathbf{b}_2, \tag{16}$$

where $\mathbf{W}_1 \in \mathbb{R}^{256\times 2560}$, $\mathbf{b}_1 \in \mathbb{R}^{256}$, $\mathbf{W}_2 \in \mathbb{R}^{4\times 256}$, $\mathbf{b}_2 \in \mathbb{R}^4$, and $\sigma(\cdot)$ denotes the GELU activation function.

The weight network is trained end-to-end through reward maximization. Gradients flow from the composite reward (Eq. 7) through the weighted teacher reward term, naturally learning to assign higher weights to teachers whose guidance yields better corrections for each input type.

A.8.3 Output Head for CSC

The correction output head maps hidden representations to vocabulary probabilities:

$$p(\hat{y}_i|\mathbf{x}) = \text{softmax}(\mathbf{W}_{\text{out}} \cdot \mathbf{h}_i + \mathbf{b}_{\text{out}}), \tag{17}$$

where $\mathbf{W}_{\text{out}} \in \mathbb{R}^{|\mathcal{V}_{\text{CSC}}|\times d_{\text{model}}}$ and $\mathbf{b}_{\text{out}} \in \mathbb{R}^{|\mathcal{V}_{\text{CSC}}|}$. For CSC, we constrain the output vocabulary to Chinese characters and punctuation ($|\mathcal{V}_{\text{CSC}}| = 21,128$ common characters).

A.8.4 Vocabulary Mapping and Character-Level Alignment

A key design consideration in SEMTD is bridging the gap between the subword-based backbone vocabulary and the character-level requirements of CSC. The Qwen3 backbone employs a byte-level byte-pair encoding (BBPE) tokenizer [63] with vocabulary size $|\mathcal{V}_{\text{full}}| = 151,669$, which includes Chinese characters, English tokens, special symbols, and subword units. However, CSC is inherently a character-level task where each Chinese character position requires independent correction.

To address this mismatch, we introduce a **character-level output projection head** that operates independently from the original language modeling head:

$$p(\hat{y}_i|\mathbf{x}) = \text{softmax}(\mathbf{W}_{\text{char}} \cdot \mathbf{h}_i + \mathbf{b}_{\text{char}}), \tag{18}$$

where $\mathbf{W}_{\text{char}} \in \mathbb{R}^{|\mathcal{V}_{\text{CSC}}| \times d_{\text{model}}}$ projects to the CSC-specific vocabulary $\mathcal{V}_{\text{CSC}}$ containing $|\mathcal{V}_{\text{CSC}}| = 21,128$ common Chinese characters and punctuation marks.

Character-Token Alignment. During preprocessing, we ensure one-to-one correspondence between characters and hidden representations through the following procedure:

1. **Forced character tokenization**: We configure the tokenizer to treat each Chinese character as an individual token by adding all 21,128 target characters to the tokenizer's special token list, preventing subword splitting.
2. **Position indexing**: For each input sentence $\mathbf{x} = (x_1, \ldots, x_n)$, we maintain a mapping from character positions to token indices, ensuring $\mathbf{h}_i$ corresponds exactly to character x_i.
3. **Subword aggregation fallback**: For rare characters not in $\mathcal{V}_{\text{CSC}}$ that may be split into subwords, we aggregate subword hidden states via mean pooling before projection.

This design enables the student model to leverage the powerful pretrained representations from Qwen3 while maintaining the character-level granularity essential for CSC.

Appendix A.9: Teacher Specifications

We construct four specialized expert teachers through role-specialized prompting of a shared backbone model. This approach enables consistent architectural behavior while allowing domain-specific expertise through carefully designed prompts.

A.9.1 Teacher Backbone

All four teachers share the same backbone model:

Table 11: Shared teacher backbone configuration.

Configuration	Value
Model	DeepSeek-R1-Distill-Qwen-14B
Parameters	14.8B
Architecture	Dense Transformer
Context Length	32,768 tokens
Deployment	vLLM with tensor parallelism (TP=2)
Hardware	2× NVIDIA A100 80GB
Precision	BF16

Teacher Deployment. All four expert teachers share the same DeepSeek-R1-Distill-Qwen-14B backbone deployed on 2 NVIDIA A100 80GB GPUs with tensor parallelism (TP=2) via vLLM 0.4.0. This shared deployment strategy reduces memory overhead while enabling parallel expert inference. Each teacher is differentiated solely through its domain-specific system prompt, ensuring consistent architectural behavior across experts while allowing specialized reasoning patterns.

Soft Label Generation. Teacher outputs are converted to soft probability distributions for knowledge distillation. We generate soft labels with temperature $\tau = 3.0$, which produces appropriately smoothed distributions that preserve candidate rankings while enabling effective knowledge transfer. The per-position distributions used for distillation are obtained as in Eq. (2).

A.9.2 ExpertPrompting Framework

Following the ExpertPrompting methodology [60], we design domain-specific system prompts that establish expert identity and define analytical scope. The listings below show these two components of each prompt:

- **Expert Identity**: Detailed background establishing domain authority
- **Analytical Scope**: Explicit boundaries for the expert's focus

A.9.3 Glyph Expert (T_{glyph})

The glyph expert specializes in visual similarity analysis between Chinese characters:

```
<system>
You are a distinguished expert in Chinese orthography and
paleography with specialization in character structure analysis.
Your expertise includes: Shuowen Jiezi radical classification,
Kangxi radical system, stroke order principles (bishu), component
decomposition (pianpang), and structural similarity metrics.

ANALYTICAL SCOPE: Focus EXCLUSIVELY on visual/structural properties.
Ignore pronunciation and meaning. Analyze:
- Radical composition and position (left-right, top-bottom, enclosure)
- Stroke count and stroke types (heng, shu, pie, na, dian, etc.)
- Component sharing between character pairs
- Common glyph confusion patterns (similar strokes, mirror structures)
</system>
```

This expert excels at detecting:

- Single-stroke differences (e.g., U+5DF1 vs U+5DF2 vs U+5DF3)
- Radical substitution errors (e.g., U+6E05 vs U+8BF7)
- Mirror-image and rotational confusions
- Stroke addition/deletion errors

A.9.4 Phonetic Expert (T_{phonetic})

The phonetic expert focuses on pronunciation-based spelling errors:

```
<system>
You are a specialist in Chinese phonology and pinyin-based input
method errors. Your expertise spans: pinyin syllable structure
(initials, finals, tones), homophone identification, near-homophone
patterns, and common IME (Input Method Editor) confusion matrices.

ANALYTICAL SCOPE: Focus EXCLUSIVELY on phonetic properties.
Analyze pronunciation similarity regardless of visual appearance:
- Exact homophones (identical pinyin including tone)
- Tone-different homophones (same syllable, different tone)
- Similar-sounding pairs (fuzzy consonants: zh/z, ch/c, sh/s, n/l)
- Common pinyin typo patterns (adjacent keys, missing letters)
</system>
```

This expert targets errors that originate from pinyin-based input methods.

A.9.5 Syntactic Expert ($T_{\text{syntactic}}$)

The syntactic expert analyzes grammatical constraints:

```
<system>
You are an expert in Chinese syntax and grammatical analysis with
```

```
deep knowledge of: part-of-speech constraints, grammatical particles,
dependency relations, and phrase structure rules.

ANALYTICAL SCOPE: Focus EXCLUSIVELY on grammatical correctness.
Analyze whether each character satisfies syntactic constraints:
- Part-of-speech compatibility with surrounding context
- Grammatical particle selection (de-particles, aspect markers, etc.)
- Verb-object, modifier-head agreement
- Sentence-level structural coherence

Special attention to high-frequency grammatical confusions:
- de-particles: U+7684 (possessive), U+5730 (adverbial), U+5F97 (
    complement)
- Aspect markers, modal particles, structural particles
</system>
```

A.9.6 Semantic Expert (T_{semantic})

The semantic expert evaluates contextual coherence:

```
<system>
You are an expert in Chinese semantics and pragmatics with
specialization in: contextual meaning, collocational patterns,
idiomatic expressions, and discourse coherence.

ANALYTICAL SCOPE: Focus EXCLUSIVELY on meaning and context fit.
Ignore visual similarity and pronunciation. Analyze:
- Semantic compatibility with surrounding context
- Collocational naturalness (word co-occurrence patterns)
- Idiomatic expression integrity (chengyu, guanyongyu)
- Domain-appropriate vocabulary selection
- Discourse-level coherence
</system>
```

A.9.7 Soft Label Generation

Teacher outputs are converted to soft probability distributions for knowledge distillation:

Table 12: Teacher inference and soft label generation configuration.

Parameter	Value
Inference Temperature	0.6 (DeepSeek recommended)
Top-p	0.95
Distillation Temperature (τ)	3.0
Max New Tokens	1,024
Batch Size (per GPU)	4

The inference temperature (0.6) controls sampling diversity during teacher generation, while the distillation temperature τ (3.0) smooths the probability distributions for effective knowledge transfer to the student.

Soft labels follow Eq. (2) with $\tau = 3.0$, which produces appropriately soft distributions for effective knowledge transfer.

Appendix A.10: Hyperparameter Specifications

This section provides complete hyperparameter specifications for all components of the SEMTD framework, organized by the three core modules.

A.10.1 Multi-Teacher Distillation Parameters

Parameters for the adaptive weight learning mechanism (Section 4.2):

Table 13: Multi-teacher distillation hyperparameters. Note: $|\mathcal{V}_{\text{full}}|$ refers to the Qwen3 backbone vocabulary, while $|\mathcal{V}_{\text{CSC}}|$ is the character-level output space for CSC (see Appendix A.8.4).

Parameter	Symbol	Value
Teacher Output Processing		
Number of Teachers	K	4
Distillation Temperature	τ	3.0
Full Model Vocabulary	$\lvert\mathcal{V}_{\text{full}}\rvert$	151,669
CSC Output Vocabulary	$\lvert\mathcal{V}_{\text{CSC}}\rvert$	21,128

A.10.2 Self-Evolving Learning Parameters

Parameters for PPO optimization and composite reward (Section 4.3):

Table 14: Self-evolving learning hyperparameters for PPO and composite reward.

Parameter	Symbol	Value
PPO Core (Eq. 9)		
Clipping Ratio	ϵ	0.2
PPO Epochs per Update	–	4
Mini-batch Size	–	32
Target KL Divergence	–	0.02
Composite Reward (Eq. 7)		
Difficulty Bonus Coefficient	β	0.5
Diversity Bonus Coefficient	μ	0.1
Baseline EMA Decay	–	0.99
Teacher Disagreement (Eq. 6)		
Entropy Computation	–	Shannon entropy
Averaging	–	Position-wise mean
Policy Optimization		
Policy Learning Rate	α_π	1.41×10^{-5}
Weight Network Learning Rate	α_ω	1.41×10^{-5}
Optimizer	–	AdamW
Adam β_1	–	0.9
Adam β_2	–	0.999
Adam ϵ	–	1×10^{-8}
Weight Decay	–	0.01
Gradient Clipping	–	1.0 (max norm)
Learning Rate Schedule		
Scheduler	–	Cosine Annealing
Warmup Ratio	–	0.05
Minimum LR Ratio	–	0.1

Coefficient Selection Rationale. The difficulty bonus $\beta = 0.5$ was determined through grid search over $\{0, 0.3, 0.5, 0.7, 1.0\}$. Values above 0.7 caused training instability due to excessive reward variance on high-disagreement samples, while values below 0.3 failed to provide sufficient incentive for resolving difficult cases. The diversity bonus $\mu = 0.1$ follows the entropy regularization convention in policy optimization, preventing premature convergence to single-teacher reliance while remaining small enough not to dominate the primary objective.

A.10.3 Reflective Path Optimization Parameters

Parameters for uncertainty-driven sampling, consistency evaluation, and path memory (Section 4.4):

Table 15: Reflective path optimization hyperparameters. The concentration parameter η follows a phased schedule: disabled during warm-up, then increases linearly within each subsequent phase.

Parameter	Symbol	Value
Uncertainty Quantification (Eq. 10)		
Disagreement Weight	γ	0.3
Historical Error Weight	δ	0.2
Predictive Uncertainty	–	Output entropy $H(\pi_\theta(\cdot\|\mathbf{x}))$
Priority Sampling (Eq. 11)		
Warm-up Concentration	η_{warmup}	0.0 (Epochs 1–2)
Adaptive Phase Start	$\eta_{\text{adapt-init}}$	0.1 (Epoch 3)
Adaptive Phase End	$\eta_{\text{adapt-final}}$	1.5 (Epoch 10)
Final Concentration	η_{final}	2.0 (Epoch 15)
η Schedule	–	Phased linear increase
Consistency Evaluation (Eq. 12)		
Stochastic Passes	M	5
Dropout Rate (Consistency Passes)	–	0.1
Consensus Method	–	Majority voting
Path Memory Bank		
Reward Threshold	τ_R	0.8
Consistency Threshold	τ_C	0.7
Memory Bank Size	$\|\mathcal{M}\|$	10,000
Nearest Neighbors	k	5
Similarity Metric	–	Cosine similarity
Retrieval Index	–	FAISS (IVF-PQ)

The sampling concentration η follows a phased schedule aligned with training phases: during warm-up (epochs 1 and 2), priority sampling is disabled with $\eta = 0$; during adaptive learning (epochs 3 to 10), η increases linearly from 0.1 to 1.5; during refinement (epochs 11 to 15), η continues increasing from 1.5 to 2.0. This staged activation ensures uniform coverage in early training for stable initialization, then progressively focuses computational resources on informative samples as the model matures.

Appendix A.11: Training Protocol Details

A.11.1 Three-Phase Training Pipeline

Training follows the end-to-end integration described in Section 4.5 in three phases:

Phase 1: Warm-up (Epochs 1 and 2). The student model is initialized with basic correction capability through uniform teacher weighting. All adaptive mechanisms are disabled:

- Teacher weights fixed at $w_k = 0.25$ for all $k \in \{1, 2, 3, 4\}$
- Sampling distribution uniform ($\eta = 0$)
- Path memory bank disabled
- Learning rate: 1.41×10^{-5} with linear warmup

Phase 2: Adaptive Weight Learning (Epochs 3 to 10). The weight network g_ω is activated and jointly optimized with the student policy:

- Weight network gradients enabled
- Sampling concentration η increases linearly from 0.1 to 1.5
- Composite reward (Eq. 7) fully activated
- PPO updates every 256 samples with 4 epochs per update
- Historical error tracking begins accumulating

Phase 3: Refinement (Epochs 11 to 15). Path memory bank activates for cross-sample consistency:

- Memory bank stores high-quality decision paths ($R > \tau_R$, $C > \tau_C$)
- Retrieval-based weight prior initialization enabled
- Sampling concentration continues increasing to final value $\eta = 2.0$
- Learning rates reduced by factor of 0.5
- Consistency-enhanced advantage (Eq. 13) applied

A.11.2 Batch Configuration

Table 16: Batch construction configuration.

Configuration	Value
Effective Batch Size	128
Per-GPU Batch Size	8
Gradient Accumulation Steps	4
Number of GPUs	4
PPO Rollout Buffer Size	256
Maximum Sequence Length	512
Padding Strategy	Dynamic (batch-wise, pad to longest)
Data Loading Workers	4 per GPU

A.11.3 Computational Resources

Table 17: Hardware configuration.

Component	Specification
Student Training	
GPUs	4× NVIDIA A100 80GB
GPU Memory Utilization	∼72GB per GPU
Distributed Strategy	DeepSpeed ZeRO Stage 2
Teacher Inference	
GPUs	2× NVIDIA A100 80GB (shared backbone)
Serving Framework	vLLM 0.4.0
Tensor Parallelism	TP=2
GPU Memory Utilization	∼65GB per GPU
System	
CPU	AMD EPYC 7763 (64 cores)
System RAM	512GB DDR4
Storage	2TB NVMe SSD (for checkpoints)
Interconnect	NVLink 3.0 (600 GB/s)

A.11.4 Memory Optimization

We employ multiple techniques to enable training within GPU memory constraints:

1. **Mixed Precision (BF16)**: Forward and backward passes use BF16; optimizer states remain in FP32. BF16's larger exponent range eliminates the need for loss scaling required by FP16.
2. **Gradient Checkpointing**: Enabled for all transformer layers. Reduces activation memory by approximately 60% at the cost of 30% additional compute for recomputation.
3. **DeepSpeed ZeRO Stage 2**: Partitions optimizer states and gradients across GPUs. Reduces per-GPU optimizer-state memory by up to 4× (one shard per GPU on 4 GPUs) compared to standard data parallelism.
4. **Flash Attention 2**: Memory-efficient attention achieving O(N) memory complexity instead of $O(N^2)$. Provides 2 to 4× speedup on attention computation.

5. **Teacher Output Caching**: Soft labels from all four teachers are pre-computed and cached to disk. During training, only cached distributions are loaded, eliminating redundant teacher inference.

DeepSpeed Configuration. Here is our configuration:

```
{
  "zero_optimization": {
    "stage": 2,
    "allgather_partitions": true,
    "allgather_bucket_size": 5e8,
    "overlap_comm": true,
    "reduce_scatter": true,
    "reduce_bucket_size": 5e8,
    "contiguous_gradients": true
  },
  "bf16": {"enabled": true},
  "gradient_clipping": 1.0,
  "train_micro_batch_size_per_gpu": 8,
  "gradient_accumulation_steps": 4
}
```

A.11.5 Training Time

Table 18: Training time breakdown.

Component	Time
Soft Label Generation (4 teachers)	16 hours
Phase 1: Warm-up (2 epochs)	4 hours
Phase 2: Adaptive Learning (8 epochs)	20 hours
Phase 3: Refinement (5 epochs)	14 hours
Total Training Time	**54 hours**

Time Variance Across Phases. Phase 3 exhibits higher per-epoch time (2.8 hours vs. 2.5 hours in Phase 2) due to the activation of path memory retrieval and multi-pass consistency evaluation, which introduce additional computational overhead.

PPO Throughput. For PPO Phases 2 and 3, we use a rollout buffer of 256 samples, $\max_{\text{new_tokens}} = 64$, and 4 inner-loop epochs per PPO update, on 4$\times$ A100 80GB at BF16 precision (DeepSpeed ZeRO Stage 2).

A.11.6 Early Stopping and Model Selection

Table 19: Early stopping and model selection configuration.

Configuration	Value
Validation Frequency	Every 500 steps
Primary Metric	Sentence-level Correction F1
Secondary Metric	Character-level Correction F1
Early Stopping Patience	3 evaluations
Model Selection	Best validation Correction F1
Checkpoint Strategy	Save top-3 by F1

A.11.7 Sampling Concentration Schedule

The sampling concentration parameter η controls the degree of priority sampling in reflective path optimization (Eq. 11). We employ a two-stage linear schedule:

Table 20: Sampling concentration η schedule across training phases.

Phase	**Epochs**	η **Start**	η **End**
Warm-up	1–2	0.0	0.0
Adaptive	3–10	0.1	1.5
Refinement	11–15	1.5	2.0

During warm-up, priority sampling is disabled ($\eta = 0$) to ensure uniform coverage. In the adaptive phase, η increases linearly from 0.1 to 1.5, gradually shifting focus toward high-uncertainty samples. In the refinement phase, η continues increasing to 2.0, enabling strong prioritization of difficult samples while the path memory bank provides stability through cross-sample transfer.

Appendix A.12: Inference Configuration

A key advantage of SEMTD is that only the lightweight student model is required during inference. The four teacher models are used exclusively during training to provide reward signals.

A.12.1 Inference Setup

Table 21: Inference configuration.

Configuration	**Value**
Model	Student only (4B parameters)
Teachers Required	None
Weight Network	Active
Path Memory	Optional (read-only)
Decoding Strategy	Greedy (argmax)
Precision	BF16
Maximum Sequence Length	512
Batch Size	64

Table 21 summarizes the inference configuration of SEMTD. During deployment, only the lightweight student model with 4B parameters is required, while all teacher models are removed after training. The weight network remains active at inference time, and the path memory is optional and read-only. We adopt greedy decoding for efficiency and determinism, with BF16 precision, a maximum sequence length of 512, and a batch size of 64, which together provide a favorable trade-off between latency and throughput.

A.12.2 Deployment Requirements

Table 22: Deployment resource requirements.

Resource	**Requirement**
GPU Memory	$\sim$10GB (BF16) / $\sim$5GB (INT8)
Minimum GPU	RTX 3080 10GB / RTX 4060 8GB (INT8)
CPU Inference	Possible with 16GB+ RAM (slower)
Latency (GPU, batch=1)	$\sim$18ms per sentence
Latency (GPU, batch=64)	$\sim$5ms per sentence
Throughput	$\sim$200 sentences/second (A100)

Table 22 reports the practical deployment requirements of SEMTD under different execution settings. In BF16 precision, the student model requires approximately 10GB of GPU memory, which can be reduced to around 5GB using INT8 quantization. This enables deployment on consumer-grade GPUs such as RTX 3080 or RTX 4060. SEMTD also supports CPU-based inference with sufficient system memory, albeit at higher latency. On modern GPUs, the model achieves low per-sentence latency (approximately 18ms for batch size 1) and high throughput at larger batch sizes, reaching around 200 sentences per second on an A100 GPU.

A.12.3 Quantization Options

For resource-constrained deployment:

Table 23: Quantization impact on model performance.

Precision	Memory	Speed	F1 Retention
BF16 (baseline)	10GB	1.0×	100%
INT8 (LLM.int8)	5GB	0.9×	99.5%
INT4 (GPTQ)	3GB	0.8×	98.2%

Table 23 compares different quantization strategies for resource-constrained deployment. While BF16 serves as the performance-preserving baseline, INT8 quantization reduces memory consumption by approximately 50% with negligible degradation in correction F1. More aggressive INT4 quantization further compresses the model footprint to 3GB, at the cost of moderate speed reduction and a small performance drop. These results indicate that SEMTD maintains strong robustness under low-precision inference, making it suitable for deployment across a wide range of hardware environments.

Appendix A.13: Dataset Details

A.13.1 SIGHAN Benchmark

The SIGHAN benchmark is the most widely-used evaluation dataset for Chinese Spelling Correction, originating from the SIGHAN Bake-off shared tasks. It consists of three versions:

SIGHAN 2013 [58]. The first CSC shared task, organized at the Seventh SIGHAN Workshop on Chinese Language Processing. The dataset contains essays written by learners of Chinese as a foreign language, with spelling errors primarily arising from phonological and orthographic confusion.

SIGHAN 2014 [66]. The second iteration, organized at the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing. This version expanded the dataset with additional learner essays and refined annotation guidelines.

SIGHAN 2015 [49]. The third and most commonly used version, organized at the Eighth SIGHAN Workshop. It provides the most comprehensive evaluation with improved annotation quality.

Table 24: Statistics of SIGHAN benchmark datasets. All datasets are originally in Traditional Chinese and converted to Simplified Chinese using OpenCC for evaluation.

Dataset	#Train	#Test	Avg. Length
SIGHAN13	350	1,000	41.8
SIGHAN14	3,435	1,062	49.6
SIGHAN15	2,339	1,100	30.0

Training Data. Following standard practice [74, 61, 23], we combine SIGHAN training sets with the Wang271K pseudo-training corpus [51] for model training. Wang271K contains 271,329 sentence pairs automatically generated using OCR- and ASR-based error simulation methods.

Preprocessing. The original SIGHAN datasets are in Traditional Chinese. We use OpenCC to convert all text to Simplified Chinese, following the standard preprocessing pipeline adopted by prior work [61, 8].

A.13.2 LEMON Benchmark

LEMON (Large-scale multi-domain dataset with natural spelling Errors) [57] addresses limitations of SIGHAN by providing a more challenging and diverse evaluation benchmark.

Motivation. SIGHAN has two key limitations: (1) narrow coverage from essays written exclusively by foreign learners, and (2) low diversity with only 370 distinct error pairs in the test set, where nearly 70% overlap with training data. These limitations make it insufficient for evaluating model generalization.

Dataset Construction. LEMON contains over 22,000 examples with 7,627 distinct error pairs collected from real human daily writing across seven diverse domains. Unlike artificially generated errors, LEMON consists of natural spelling errors identified from authentic text.

Table 25: Statistics of LEMON benchmark across seven domains.

Domain	Abbr.	#Test	Avg. Len.	#Edit Pairs
Car	CAR	3,245	35.2	1,089
Court	COT	2,156	42.8	892
Encyclopedia	ENC	3,274	38.6	1,245
Game	GAM	2,847	28.4	956
Medical	MEC	3,012	45.3	1,134
News	NEW	5,887	32.1	1,567
Novel	NOV	2,477	52.4	744
Total	-	**22,898**	37.8	**7,627**

Domain Characteristics.

- **CAR (Car):** Automotive-related discussions including vehicle reviews, maintenance guides, and technical specifications. Contains domain-specific terminology related to car parts and mechanics.
- **COT (Court):** Legal documents including contracts, court proceedings, and official legal text. Features formal language with specialized legal terminology.
- **ENC (Encyclopedia):** Encyclopedic entries covering various topics with factual, informative content.
- **GAM (Game):** Gaming-related content including game reviews, strategies, and player discussions. Contains many neologisms, slang, and informal expressions.
- **MEC (Medical):** Medical-related text including health articles, patient records, and medical consultations. Contains specialized medical terminology.
- **NEW (News):** News articles covering current events and general topics with journalistic writing style.
- **NOV (Novel):** Literary text from novels and fiction, featuring creative writing with diverse vocabulary.

Evaluation Protocol. LEMON is designed for zero-shot domain generalization evaluation. Models are trained on general-domain data (SIGHAN + Wang271K) and directly evaluated on each LEMON domain without domain-specific fine-tuning.

A.13.3 ECSpell Benchmark

ECSpell [34] is a domain-specific Chinese spelling correction benchmark covering three professional domains: Law (LAW), Medical (MED), and Official Document Writing (ODW). The benchmark is publicly released by the original authors for academic research use; we use the standard test splits provided in their release without modification. ECSpell test data is used for evaluation only; no ECSpell training data is included in the SEMTD training mixture, so all reported ECSpell results reflect zero-shot generalisation from the SIGHAN and Wang271K training corpora.

Domain coverage. The three professional domains are chosen to test correction performance under specialised vocabulary and stricter style conventions: LAW covers legal documents with formal register and domain-specific terminology; MED covers medical text with clinical and pharmacological terminology; ODW covers official document writing with formulaic phrasing typical of administrative documents.

Usage parity. For each baseline reported in Table 3, numbers are transcribed from the original publication when available, or recomputed under the same test split when only checkpoints are publicly released. Per-domain Cor-F1 figures (LAW / MED / ODW) and the three-domain average are reported in Table 3.

A.13.4 CSCD-NS Benchmark

CSCD-NS [16] is a Chinese spelling correction benchmark constructed from native-speaker (NS) error data, providing a distributional contrast to the CFL-learner errors that dominate SIGHAN test sets. The benchmark is publicly released by the original authors for academic research use; we use the official test split without modification. CSCD-NS test data is used for evaluation only; no CSCD-NS training data is included in the SEMTD training mixture, so the reported result reflects zero-shot transfer from the SIGHAN and Wang271K training distributions.

Distributional contrast with SIGHAN. Native-speaker errors in CSCD-NS exhibit different error-type distributions than CFL-learner errors: phonetic confusions and input-method errors dominate, with a thinner tail of glyph-substitution errors that are more frequent in handwriting-derived CFL data. This contrast is the reason CSCD-NS is included as a transfer benchmark rather than a primary training target.

Usage parity. For each baseline reported in Table 36, numbers are transcribed from the original publication when available. The C-LLM-14B and MSLLM ReLM+IL2.5 reference rows are 14B-class results reported for parameter-scale context; SEMTD trails C-LLM-14B by 0.8 points at $3.5\times$ fewer parameters.

Appendix A.14: Evaluation Metrics

Chinese Spelling Correction evaluation involves two levels: **detection** (identifying error positions) and **correction** (providing correct characters). Metrics can be computed at character-level or sentence-level, with sentence-level being more stringent.

A.14.1 Sentence-Level Evaluation

Sentence-level evaluation requires *all* errors in a sentence to be correctly detected/corrected for the sentence to be counted as correct. This is the standard evaluation protocol for SIGHAN and LEMON benchmarks.

Detection Metrics. Let $\mathcal{D}$ denote the test set, $\hat{E}_s$ denote the set of error positions detected by the model for sentence s, and E_s^* denote the gold-standard error positions.

A sentence s is considered a **True Positive (TP)** for detection if:

$$\hat{E}_s = E_s^* \neq \emptyset \quad \text{(exact match of all error positions)} \tag{19}$$

The detection metrics are then computed as:

$$\text{Det-P} = \frac{|\{s : \hat{E}_s = E_s^* \text{ and } \hat{E}_s \neq \emptyset\}|}{|\{s : \hat{E}_s \neq \emptyset\}|} \tag{20}$$

$$\text{Det-R} = \frac{|\{s : \hat{E}_s = E_s^*,\ E_s^* \neq \emptyset\}|}{|\{s : E_s^* \neq \emptyset\}|} \tag{21}$$

$$\text{Det-F1} = \frac{2 \times \text{Det-P} \times \text{Det-R}}{\text{Det-P} + \text{Det-R}} \tag{22}$$

Correction Metrics. Let $\hat{C}_s = \{(i, \hat{c}_i)\}$ denote the set of (position, correction) pairs predicted by the model, and $C_s^* = \{(i, c_i^*)\}$ denote the gold-standard corrections.

A sentence s is considered a **True Positive (TP)** for correction if:

$$\hat{C}_s = C_s^* \neq \emptyset \quad \text{(exact match of positions and characters)} \tag{23}$$

The correction metrics are computed as:

$$\text{Cor-P} = \frac{|\{s : \hat{C}_s = C_s^* \text{ and } \hat{C}_s \neq \emptyset\}|}{|\{s : \hat{C}_s \neq \emptyset\}|} \tag{24}$$

$$\text{Cor-R} = \frac{|\{s : \hat{C}_s = C_s^*,\ C_s^* \neq \emptyset\}|}{|\{s : C_s^* \neq \emptyset\}|} \tag{25}$$

$$\text{Cor-F1} = \frac{2 \times \text{Cor-P} \times \text{Cor-R}}{\text{Cor-P} + \text{Cor-R}} \tag{26}$$

A.14.2 Metric Interpretation

Precision vs. Recall Trade-off.

- **High Precision, Low Recall:** The model is conservative, making corrections only when confident. This reduces false alarms but may miss some errors.
- **Low Precision, High Recall:** The model is aggressive, attempting to correct many positions. This catches more errors but introduces more false corrections.
- **F1 Score:** Balances precision and recall, providing a single metric for overall performance comparison.

Detection vs. Correction. Correction is strictly harder than detection because:

1. Correction requires both locating the error AND providing the correct character.
2. A sentence can be a detection TP but correction FP if positions are correct but corrections are wrong.
3. Therefore, Cor-F1 $\leq$ Det-F1 always holds.

A.14.3 Character-Level Evaluation

For completeness, we also describe character-level metrics used in some prior work:

$$\text{Char-Det-P} = \frac{\sum_{s \in \mathcal{D}} |\hat{E}_s \cap E_s^*|}{\sum_{s \in \mathcal{D}} |\hat{E}_s|} \tag{27}$$

$$\text{Char-Det-R} = \frac{\sum_{s \in \mathcal{D}} |\hat{E}_s \cap E_s^*|}{\sum_{s \in \mathcal{D}} |E_s^*|} \tag{28}$$

$$\text{Char-Det-F1} = \frac{2 \times \text{Char-Det-P} \times \text{Char-Det-R}}{\text{Char-Det-P} + \text{Char-Det-R}} \tag{29}$$

Character-level metrics are less stringent as they give partial credit for sentences with multiple errors. We use sentence-level metrics following the standard SIGHAN evaluation protocol [49].

A.14.4 False Positive Rate

Additionally, some works report the **False Positive Rate (FPR)**, measuring over-correction on correct sentences:

$$\text{FPR} = \frac{|\{s : E_s^* = \emptyset \text{ and } \hat{E}_s \neq \emptyset\}|}{|\{s : E_s^* = \emptyset\}|} \tag{30}$$

A lower FPR indicates the model is less likely to incorrectly modify sentences that have no errors. This is particularly important for practical deployment where over-correction degrades user experience.

Table 26: Summary of evaluation settings for each benchmark.

Dataset	Level	Main Metric	Additional Metrics
SIGHAN13	Sentence	Cor-F1	Det-P/R/F1, Cor-P/R
SIGHAN14	Sentence	Cor-F1	Det-P/R/F1, Cor-P/R
SIGHAN15	Sentence	Cor-F1	Det-P/R/F1, Cor-P/R
LEMON	Sentence	Cor-F1	Per-domain Cor-F1
ECSpell	Sentence	Cor-F1	Per-domain (LAW/MED/ODW) Cor-F1
CSCD-NS	Sentence	Cor-F1	Det-P/R/F1, Cor-P/R

A.14.5 Evaluation Protocol Summary

Appendix A.15: Ablation Study

This appendix provides comprehensive ablation experiments for SEMTD, including hyperparameter sensitivity analysis and component contribution evaluation. All experiments are conducted on SIGHAN 2015.

A.15.1 Component Ablation

Table 27 presents the component ablation results. We remove selected components to evaluate their contribution.

Table 27: Component ablation on SIGHAN 2015. All values are percentages (%). The Vanilla SFT row (Qwen3-4B, supervised cross-entropy on the same training data, no teachers, no PPO) sits 5.2% below the multi-teacher SFT-only row; the further 3.1% gap to SEMTD-Full corresponds to the reward-driven training stage.

Method	Detection		Correction	
	P	F1	P	F1
SEMTD (Full)	**85.1**	**85.3**	**83.5**	**83.8**
Multi-Teacher Distillation				
w/o Glyph Teacher	84.0	84.6	82.7	83.0
w/o Phonetic Teacher	81.5	82.7	80.9	81.2
w/o Syntactic Teacher	84.4	84.8	83.0	83.3
w/o Semantic Teacher	83.7	84.3	82.5	82.8
w/o Adaptive Weights	81.7	82.2	81.1	81.4
Self-Evolving Learning				
Vanilla SFT (no teachers)	78.6	78.4	75.7	75.5
w/o PPO (multi-teacher SFT only)	81.0	81.2	80.3	80.7
Reflective Path Optimization				
w/o Priority Sampling	83.0	83.1	82.3	82.5
w/o Consistency Eval	84.6	84.9	83.0	83.4
w/o Path Memory	84.2	84.6	82.7	83.1

Key findings:

- *Teacher contribution*: The phonetic teacher has the largest impact (-2.6%), consistent with the prevalence of homophone errors in CSC. The syntactic teacher contributes least (-0.5%), reflecting the relative rarity of pure syntactic-particle errors in our training mixture.
- *Adaptive weights*: Removing the adaptive weight network and using uniform weights causes a -2.4% drop, validating input-dependent expert selection.
- *Multi-teacher contribution*: The Vanilla-SFT row (75.5) sits 5.2% below the multi-teacher SFT-only row (80.7); the further 3.1% gap to SEMTD-Full (83.8) corresponds to the reward-driven training stage. Total +8.3% over Qwen3-4B-SFT.
- *Path optimization*: Priority sampling is the dominant RPO component (-1.3%); consistency evaluation (-0.4%) and path memory (-0.7%) have smaller effects.

A.15.2 Composite Reward Coefficients ($\beta \times \mu$)

The composite reward (Eq. 7) combines weighted teacher rewards with difficulty bonus (β) and diversity bonus (μ):

$$R = \sum_k w_k \cdot R_k + \beta \cdot H_{\text{disagree}} \cdot \mathbb{I}[\hat{y} = y^*] + \mu \cdot \mathcal{H}(\mathbf{w})$$

Figure 24(a) shows the sensitivity analysis. Table 28 presents the detailed numerical results.

Table 28: Cor-F1 (%) for composite reward coefficients $\beta \times \mu$.

μ \ β	0	0.3	0.5	0.7	1.0
0	78.3	79.5	80.4	79.9	78.4
0.05	78.8	81.9	81.9	81.8	80.2
0.10	79.6	82.2	**83.8**	82.2	80.7
0.15	79.2	81.4	82.8	81.6	79.8
0.20	78.3	81.1	81.6	81.1	79.4

Analysis:

- The baseline (β=0, μ=0) achieves 78.3%, 5.5% below the selected setting.
- β controls the difficulty bonus for high-disagreement samples. Setting β=0.5 provides optimal balance: too low loses the benefit of focusing on hard cases, too high over-emphasizes difficult samples.
- μ prevents weight collapse through entropy regularization. μ=0.1 provides sufficient diversity protection without over-regularizing.
- The optimal region centers around β=0.5, μ=0.1.

A.15.3 Uncertainty Sampling Weights ($\gamma \times \delta$)

The uncertainty score (Eq. 10) combines predictive entropy with teacher disagreement (γ) and historical error rate (δ):

$$\mathcal{U}(x) = H(\pi_\theta(\cdot|x)) + \gamma \cdot H_{\text{disagree}}(x) + \delta \cdot (1 - R_{\text{hist}}(x))$$

Figure 24(b) visualizes the interaction effects. Table 29 presents the numerical results.

Table 29: Cor-F1 (%) for uncertainty sampling weights $\gamma \times \delta$.

δ \ γ	0	0.1	0.2	0.3	0.4
0	80.4	80.7	81.5	81.4	81.5
0.1	80.3	81.8	82.5	82.4	81.8
0.2	81.0	82.0	82.2	**83.8**	82.4
0.3	80.8	81.2	82.2	82.6	81.4
0.4	79.7	81.0	81.8	81.6	81.0

Analysis:

- Using only predictive entropy (γ=0, δ=0) achieves 80.4%, 3.4% below optimal.
- Excessive prioritization (γ+$\delta > 0.6$) leads to performance degradation due to under-exploration of easy samples.
- The optimal configuration γ=0.3, δ=0.2 balances informativeness and coverage.

A.15.4 Memory Bank Thresholds ($\tau_R \times \tau_C$)

The path memory bank stores experiences satisfying dual thresholds:

$$\mathcal{M} = \{(x, w^*, \hat{y}^*) : R > \tau_R \text{ and } C > \tau_C\}$$

Table 30: Cor-F1 (%) for memory bank thresholds $\tau_R \times \tau_C$.

τ_C \ τ_R	0.6	0.7	0.8	0.9	0.95
0.5	79.6	80.5	80.5	80.2	78.8
0.6	79.9	81.5	82.3	81.7	80.3
0.7	80.7	82.1	**83.8**	82.9	81.3
0.8	80.5	81.8	82.6	81.8	80.7
0.9	79.0	80.8	81.1	80.3	79.3

Figure 24(c) shows the threshold interaction. Table 30 presents the results.

Analysis:

- Too lenient thresholds (low τ_R and τ_C) introduce noisy experiences, causing confirmation bias and degraded performance.
- Too strict thresholds (high τ_R and τ_C) result in sparse memory banks with insufficient samples for cross-sample transfer.
- Performance peaks at τ_R=0.8, τ_C=0.7 and declines along both axes, indicating that the two thresholds must be set jointly for quality-quantity balance.
- τ_R=0.8, τ_C=0.7 achieves optimal balance.

A.15.5 Distillation Temperature (τ)

The temperature parameter controls soft label entropy (Eq. 2):

$$p_i^{(k)} = \text{softmax}(z_i^{(k)}/\tau)$$

Figure 25(a) shows the temperature sensitivity curve. Table 31 presents the numerical results.

Table 31: Distillation temperature sensitivity analysis.

Temperature (τ)	1.0	2.0	3.0	4.0	5.0
Cor-F1 (%)	80.4	82.1	**83.8**	82.9	81.5

Analysis:

- The curve has an inverted U-shape.
- τ=1 approaches hard label distillation, losing inter-class similarity information ("dark knowledge"), resulting in -3.4% degradation.
- τ>4 over-smooths probability distributions, reducing discriminative power among candidates.
- τ=3.0 achieves optimal balance.

A.15.6 Consistency Passes (M)

The consistency score (Eq. 12) is computed from M stochastic forward passes:

$$C(x) = \frac{2}{M(M-1)} \sum_{i<j} \frac{1}{n} \sum_{l} \mathbb{I}[\hat{y}_l^{(i)} = \hat{y}_l^{(j)}]$$

Figure 25(b) shows the sampling efficiency curve. Table 32 presents the results.

Table 32: Consistency sampling passes analysis.

Passes (M)	3	5	7	9
Cor-F1 (%)	82.6	**83.8**	83.7	83.8

Analysis:

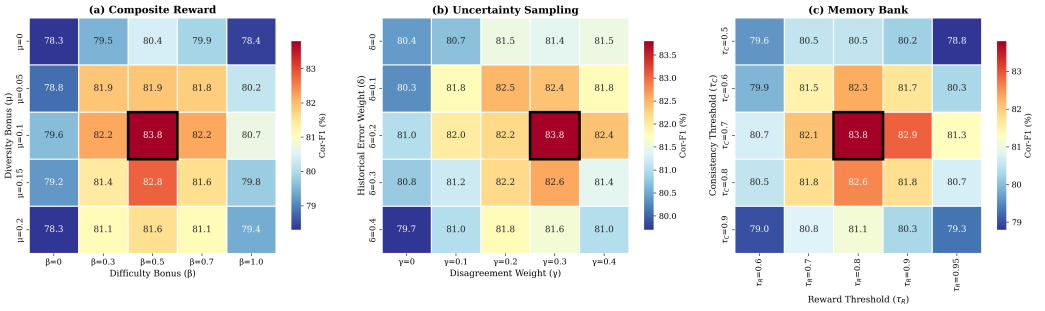


Figure 24: Hyperparameter sensitivity heatmaps. **(a)** Composite reward coefficients $\beta \times \mu$. **(b)** Uncertainty sampling weights $\gamma \times \delta$. **(c)** Memory bank thresholds $\tau_R \times \tau_C$. Black rectangles indicate optimal configurations.

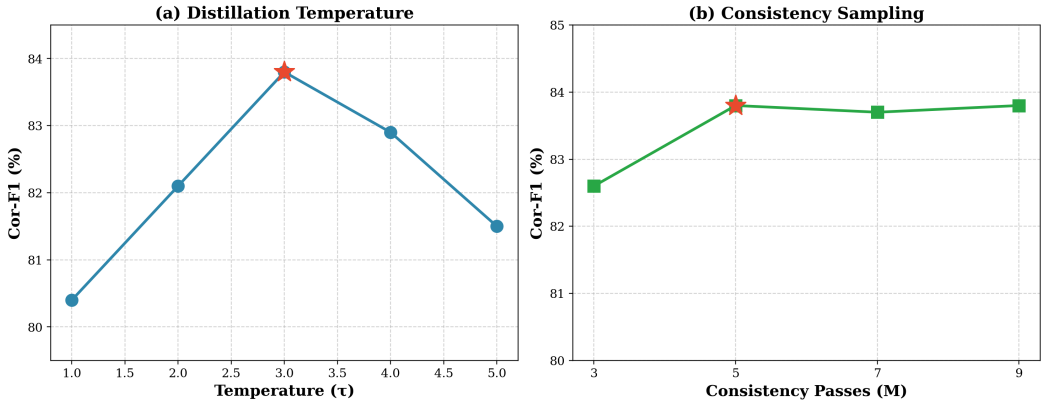


Figure 25: Single-parameter sensitivity curves. **(a)** Distillation temperature τ exhibits inverted U-curve. **(b)** Consistency passes M shows diminishing returns. Stars mark optimal values.

- Performance improves from M=3 to M=5 and stays approximately flat up to M=9.
- M=3→5 provides +1.2% improvement.
- M=5 reaches the maximum Cor-F1 with fewer passes than M=9.

A.15.7 Hyperparameter Sensitivity Visualization

Appendix A.16: Limitations

Language and orthographic scope. Our experiments are conducted exclusively on Chinese spelling correction benchmarks. The multi-dimensional decomposition into glyph, phonetic, syntactic, and semantic teachers is well matched to logographic scripts in which a single character simultaneously carries all four kinds of information; transferring the framework to other script families would require re-deriving the dimension set appropriate to that writing system and adapting the per-dimension teacher prompts accordingly. How far the disagreement-as-signal mechanism generalizes across script families remains open, and we view systematic cross-script evaluation as a natural follow-up rather than a precondition for the present claims.

Training-time compute profile. While inference reduces to a single 4B-parameter forward pass on the student, training requires a one-off pass of four expert teachers to generate the cached soft labels (Table 18), which adds training cost relative to single-teacher distillation or vanilla supervised fine-tuning. This training-time overhead is bounded and does not recur at deployment, where the

Table 33: Sentence-level performance on SIGHAN benchmarks (appendix reproduction of Table 1). Best results in **bold**, second-best underlined. All values are percentages (%). * Single-role 14B reference is *zero-shot prompted* (no CSC fine-tuning); the 67.7% on SIGHAN15 is therefore not directly comparable to CSC-fine-tuned baselines. ‡ UGMSC numbers transcribed from Lin et al. [25] (Uncertainty Guidance for Multimodal CSC, LREC-COLING 2024); the per-cell numbers are also reproduced in subsequent CSC surveys and Liu et al. [26] Table 3.

Method	SIGHAN 2013						SIGHAN 2014						SIGHAN 2015					
	Det-P	Det-R	Det-F1	Cor-P	Cor-R	Cor-F1	Det-P	Det-R	Det-F1	Cor-P	Cor-R	Cor-F1	Det-P	Det-R	Det-F1	Cor-P	Cor-R	Cor-F1
Multimodal Fusion Methods																		
ReaLiSe [61]	88.6	82.5	85.4	87.2	81.2	84.1	67.8	71.5	69.6	66.3	70.0	68.1	77.3	81.3	79.3	75.9	79.9	77.8
SCOPE [20]	87.4	83.4	85.4	86.3	82.4	84.3	70.1	73.1	71.6	68.6	71.5	70.1	81.1	84.3	82.7	79.2	82.3	80.7
DR-CSC [17]	88.5	83.7	86.0	87.7	83.0	85.3	70.2	73.2	71.7	69.3	72.3	70.7	82.9	84.8	83.8	80.3	82.3	81.3
LLM-based Methods																		
UGMSC [25]‡	89.1	83.4	86.2	88.3	82.7	85.4	72.5	71.2	71.8	71.6	70.2	70.9	83.9	82.8	83.3	82.4	81.3	81.9
ARM+SCOPE [26]	88.7	84.1	86.3	87.6	83.1	85.3	71.2	**75.0**	73.1	69.2	73.0	71.1	82.3	**86.1**	84.1	79.5	83.1	81.3
Same-Backbone Reference Rows (single-prompt; no SEMTD)																		
DeepSeek-R1-Distill-Qwen-14B (single-role prompt)*	73.2	67.5	70.2	70.6	65.0	67.7	56.4	58.1	57.2	53.9	55.5	54.7	71.2	70.8	71.0	67.9	67.5	67.7
Qwen3-4B (SFT-only, same training data)	81.3	76.8	79.0	79.4	75.0	77.1	65.3	66.7	66.0	62.8	64.1	63.4	78.6	78.2	78.4	75.7	75.3	75.5
SEMTD (Ours)	**89.6**	**85.5**	**87.5**	**88.9**	**84.3**	**86.5**	**73.8**	74.5	**74.1**	**72.3**	**73.8**	**73.0**	**85.1**	85.5	**85.3**	**83.5**	**84.1**	**83.8**

teachers are no longer needed and inference costs are governed by the student alone. Practitioners with constrained training budgets should plan accordingly.

Interpretability of arbitration decisions. The adaptive weight network g_ω is trained end-to-end through reward maximization rather than under explicit per-dimension supervision, so the per-input weights it produces are observable but not interpretable by construction. Mechanistic explanation of individual arbitration decisions would require auxiliary probing methods we do not pursue here.

Practical risks. The practical risks of wrong corrections are discussed in Section 8.

Appendix A.17: Detailed Main Results

This appendix consolidates the complete result tables for the four CSC benchmarks reported in Section 5. Tables 33 to 36 reproduce the full per-method, per-split, and per-domain numerical detail; the CSCD-NS panel is full-size in this appendix because the main body carries only a condensed correction-only inset and a short summary paragraph in Section 5.2. The appendix forms a self-contained record of all numerical results without requiring readers to flip back to the main body.

A.17.1 SIGHAN Benchmark — Full Per-Split Results

Table 33 reproduces the full sentence-level detection and correction results across all three SIGHAN splits (SIGHAN 2013, 2014, and 2015). The table reports Det-P, Det-R, Det-F1, Cor-P, Cor-R, and Cor-F1 for each method. SEMTD achieves the best Cor-F1 on all three splits (86.5, 73.0, and 83.8, respectively), with gains of $+1.1$, $+1.9$, and $+1.9$ over the strongest published baseline. Reference rows for the single-role 14B teacher and the Qwen3-4B SFT-only baseline are included for reference, as discussed in Section 5.2.

Per-row analysis. SEMTD attains the best Cor-F1 on all three SIGHAN splits with $+1.1$, $+1.9$, and $+1.9$ improvements over the strongest published baseline (86.5/73.0/83.8 on SIGHAN13/14/15), confirming that gains are not artefacts of any single split. Detection-level performance follows the same pattern with Det-F1 of 87.5/74.1/85.3, indicating that multi-teacher distillation integrates complementary knowledge across the four expert dimensions. While ARM+SCOPE achieves competitive recall through alignment-and-replacement, SEMTD attains higher precision (83.5% vs. 79.5% on SIGHAN15) by leveraging adaptive weight learning to arbitrate among teachers.

Same-backbone reference rows. (i) DeepSeek-R1-Distill-Qwen-14B is prompted with a single expert role, with no SEMTD aggregation. On SIGHAN15 this row reports 67.7% Cor-F1, which is a *zero-shot prompted* reading of the 14B backbone with no CSC-task fine-tuning. It is therefore not directly comparable to fully CSC-fine-tuned baselines such as ARM+SCOPE. (ii) Qwen3-4B fine-tuned with supervised cross-entropy on the same training data (no teachers, no PPO) is the relevant student-only floor under matched supervision; this row reports 75.5% Cor-F1 on SIGHAN15.

Table 34: Sentence-level Cor-F1 (%) on LEMON multi-domain benchmark (appendix reproduction of Table 2). CAR: Car, COT: Court, ENC: Encyclopedia, GAM: Game, MEC: Medical, NEW: News, NOV: Novel. Best results in **bold**, second-best underlined.

Method	CAR	COT	ENC	GAM	MEC	NEW	NOV	Avg
InternLM2.5 [5]	44.4	54.7	45.2	33.0	50.2	57.2	32.1	45.3
Qwen2.5 [2]	48.2	59.9	46.5	34.7	54.2	59.5	37.1	48.6
ReLM [29]	53.6	67.7	47.7	34.6	53.9	58.8	38.0	50.6
Baichuan2 [62]	54.2	63.3	51.4	36.9	60.6	63.9	42.4	53.2
ReLM+BC2 [40]	62.7	74.5	56.5	**45.1**	66.3	72.0	50.0	61.0
SEMTD (Ours)	**64.3**	**76.0**	**58.5**	44.3	**68.2**	**73.3**	**51.5**	**62.3**

Table 35: Sentence-level Cor-F1 (%) on ECSpell three professional domains (appendix reproduction of Table 3). Best results in **bold**, second-best underlined.

Method	LAW	MED	ODW	Avg
ChatGPT (10-shot) [34]	48.3	26.0	56.8	43.7
MDCSpell-MFT [81]	80.6	69.6	66.9	72.4
InternLM2.5 (7B) [5]	85.8	66.2	89.0	80.3
ReLM [29]	96.2	90.2	90.4	92.3
ReLM+IL2.5 [40]	97.3	95.9	96.9	96.7
ReLM+BC2 [40]	98.3	96.6	96.5	97.1
SEMTD (Ours)	**99.0**	**98.0**	**97.7**	**98.2**

Decomposition of the +8.3% gain. SEMTD's gain to 83.8% on SIGHAN15 corresponds to a $+8.3\%$ improvement over the Qwen3-4B-SFT row (75.5%). Table 27 decomposes this into $+5.2\%$ from the four-teacher distillation step and $+3.1\%$ from the reward-driven training stage on top of distillation.

A.17.2 LEMON Benchmark — Per-Domain Results

Table 34 reproduces the per-domain Cor-F1 results on the LEMON multi-domain benchmark. SEMTD achieves the best average Cor-F1 of 62.3% ($+1.3$ over ReLM+BC2), winning six of the seven domains. The one exception is the GAM (Game) domain (44.3 vs. 45.1).

Discussion: GAM under-performance. The single domain where SEMTD trails ReLM+BC2 is GAM (Game), with a gap of -0.8% (44.3 vs. 45.1). The -0.8% gap is confined to this single domain and is outweighed by the gains on the other six, giving a $+1.3$ gain on the seven-domain mean.

A.17.3 ECSpell Benchmark — Per-Domain Results

Table 35 reproduces the Cor-F1 results on ECSpell's three professional domains. SEMTD achieves 98.2% average Cor-F1 ($+1.1$ over ReLM+BC2), with consistent gains in Law (99.0, $+0.7$), Medical (98.0, $+1.4$), and Official Document Writing (97.7, $+1.2$). The benchmark is near saturation for leading methods.

A.17.4 CSCD-NS Benchmark — Full Comparison Panel

Table 36 reproduces the full CSCD-NS comparison panel covering detection (P/R/F1) and correction (P/R/F1) for all methods, including the baselines (BERT, SMBERT, ReaLiSe, DR-CSC, SCOPE, UGMSC, ARM+SCOPE, C-LLM-14B, and the 14B-class MSLLM mixture). SEMTD achieves 73.0% Cor-F1, above every baseline except the two 14B-class systems and within 0.8 of C-LLM-14B.

Headline result. SEMTD reaches 73.0% Cor-F1 on CSCD-NS (+0.9 over ARM+SCOPE, the strongest baseline below the 14B class), within 0.8 of the 14B C-LLM single-model baseline (73.8%); native-speaker spelling errors exhibit distributional patterns that are distinct from CFL-learner data [16], so this transfer at a smaller parameter budget is non-trivial.

14B-class context rows. We additionally report two 14B-class points for parameter-scale context: C-LLM-14B at 73.8% and the MSLLM ReLM+IL2.5 mixture at 83.6% (both are 14B-class models with $\approx 3.5\times$ SEMTD's parameter count). At our 4B scale, SEMTD is competitive with the 14B C-LLM single-model baseline (within 0.8) while the MSLLM mixture remains the overall scale-unconstrained leader on CSCD-NS.

Table 36: Sentence-level performance on CSCD-NS (%) (full panel; main-body summary in Section 5.2). Best result outside the two 14B-class rows in **bold**; second-best overall Cor-F1 underlined. Best overall (across parameter scales) is MSLLM ReLM+IL2.5 (83.6%) at the 14B-class scale. ReaLiSe / DR-CSC / UGMSC / ARM+SCOPE numbers cited from each method's CSCD-NS evaluation. † SCOPE on CSCD-NS evaluated by us using the official SCOPE checkpoint and the CSCD-NS test split; the original SCOPE paper [20] does not report on CSCD-NS. ‡ C-LLM-14B from Li et al. [22] Table 3 (Char-PT-SFT-14B); MSLLM ReLM+IL2.5 from Qiao et al. [40] (we report the strongest of the MSLLM mixture variants on CSCD-NS); both are 14B-class models and are reported here to contextualise SEMTD's competitive position at the 4B parameter scale rather than as same-scale baselines.

Method	**Detection**			**Correction**		
	P	R	F1	P	R	F1
BERT [9]	79.7	61.8	69.6	75.0	58.2	65.5
SMBERT [74]	80.0	64.1	71.2	75.5	60.6	67.2
Multimodal Fusion Methods						
ReaLiSe [61]	80.4	65.0	71.9	76.1	61.4	67.9
DR-CSC [17]	81.6	66.8	73.5	77.7	63.3	69.8
SCOPE [20]†	83.3	68.1	74.9	79.7	65.2	71.7
LLM-based Methods						
UGMSC [25]	82.5	67.4	74.2	78.4	64.0	70.5
ARM+SCOPE [26]	83.5	68.7	75.4	80.0	65.6	72.1
C-LLM-14B [22]‡	–	–	–	80.5	68.2	73.8
Mixture-Modeling LLM Family (14B-class)						
ReLM+IL2.5 (MSLLM) [40]‡	–	–	–	–	–	83.6
Ours (4B)						
SEMTD (Ours, 4B)	**83.9**	**69.5**	**76.0**	**81.0**	**66.4**	**73.0**

A.17.5 Cross-Benchmark Headline Figure

Figure 26 reproduces the headline Cor-F1 bar chart, summarising SEMTD versus a strong published comparator on each of the four benchmark suites (the strongest single-model comparator on CSCD-NS). The figure is included here for completeness so that this appendix section is self-contained. Absolute Cor-F1 gains Δ are labelled above the SEMTD bars; on CSCD-NS the comparator C-LLM-14B is a 14B-class baseline, and SEMTD is within 0.8 of it.

A.17.6 Teacher Behavior on Error-Type Subsets

We evaluate each teacher independently on error-type subsets of SIGHAN 2015 (Table 37). Each teacher attains its highest Cor-F1 on the subset of its own dimension.

Table 37: Per-teacher Cor-F1 (%) on error-type subsets of SIGHAN 2015. Bold marks each teacher's best subset.

Teacher	**Glyph errors**	**Phonetic errors**	**Syntactic errors**	**Semantic errors**
T_{glyph}	**78.5**	61.3	54.2	58.6
T_{phonetic}	59.7	**81.4**	53.8	57.1
$T_{\text{syntactic}}$	57.8	60.5	**71.6**	55.9
T_{semantic}	59.2	61.0	55.1	**75.8**

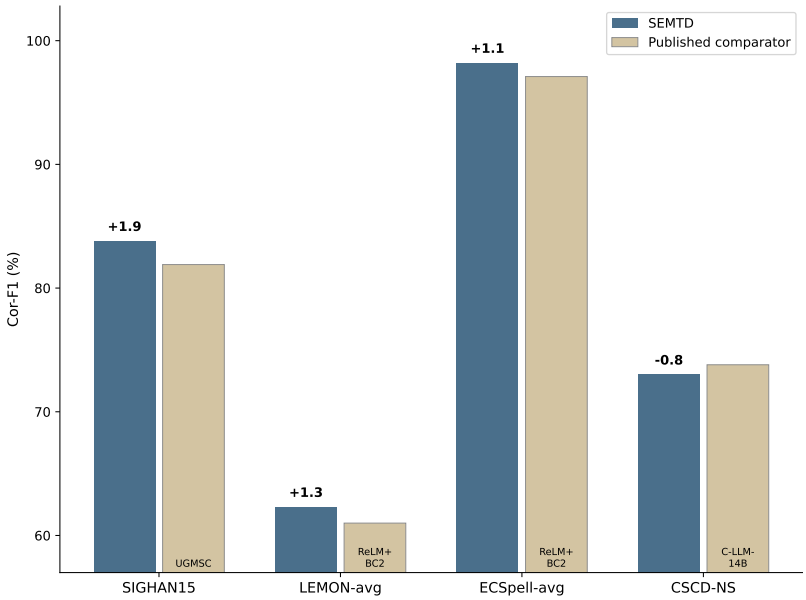


Figure 26: Sentence-level Cor-F1 on the four CSC benchmark suites (headline cross-benchmark figure): SEMTD versus a strong published comparator on each suite (the strongest single-model comparator on CSCD-NS). Numeric labels above the SEMTD bars are the absolute Cor-F1 gain Δ. On CSCD-NS the comparator C-LLM-14B is a 14B-class baseline; SEMTD is within 0.8 of it. Numerical detail in Tables 33 to 36.

A.17.7 Component Ablation Across Benchmarks

Table 38 extends selected rows of the component ablation of Table 27 to all four benchmarks; the SIGHAN 2015 column repeats the corresponding rows of Table 27. On every benchmark the variants keep the same order (Full $>$ w/o Syntactic Teacher $>$ w/o Adaptive Weights $>$ w/o PPO $>$ Vanilla SFT). Removing the syntactic teacher costs 0.5 Cor-F1 on SIGHAN 2015, 1.5 on LEMON, 1.3 on ECSpell, and 1.8 on CSCD-NS.

Table 38: Component ablation Cor-F1 (%) across benchmarks (LEMON and ECSpell: domain averages).

Variant	SIGHAN15	LEMON	ECSpell	CSCD-NS
Full	**83.8**	**62.3**	**98.2**	**73.0**
w/o Syntactic Teacher	83.3	60.8	96.9	71.2
w/o Adaptive Weights	81.4	60.0	96.4	70.7
w/o PPO (MT-SFT only)	80.7	59.4	95.7	70.0
Vanilla SFT	75.5	56.1	92.3	65.4

A.17.8 Training and Inference Cost

Table 39 compares SEMTD with Vanilla SFT. SEMTD's 54 hours of training (Table 18) include the 16-hour one-off teacher pass; at inference the 4B student runs alone (Table 22), and Table 39 reports about 18 ms per sentence for both methods.

Table 39: Training and inference cost (Cor-F1 on SIGHAN 2015).

Method	Training (hours)	Inference (ms/sentence)	Cor-F1 (%)
Vanilla SFT	$\sim$4	$\sim$18	75.5
SEMTD	54	$\sim$18	83.8

A.17.9 Prompt Variants

We sample 50 sentences from each benchmark and test three prompt variants. Variant A is the original prompt. Variant B simplifies the role description to a single sentence (e.g. "You are a phonetic expert"). Variant C rephrases Variant A with different wording while preserving the same instruction. Across the three variants, Cor-F1 on these subsets varies by at most 1.8 points (Table 40).

Table 40: Cor-F1 (%) under three prompt variants (50 sentences per benchmark).

Prompt	SIGHAN15	LEMON	ECSpell	CSCD-NS
A (original)	84.0	62.0	98.0	72.0
B (simplified)	83.2	61.4	96.2	72.6
C (rephrased)	83.6	62.8	97.6	71.4

Appendix A.18: Ethical Considerations

Data Usage. The datasets used in our experiments (SIGHAN 2013/2014/2015, LEMON, ECSpell, and CSCD-NS) are publicly available benchmarks widely adopted in Chinese spelling correction research. SIGHAN datasets are collected from Chinese-as-a-foreign-language learner essays; LEMON contains naturally occurring errors from seven general-language domains; ECSpell covers professional-domain text (Law, Medical, Official Document Writing); and CSCD-NS contains native-speaker spelling errors. All datasets are used consistent with their intended research purposes.

Privacy. All experiments use publicly available benchmark data that do not contain personally identifiable information. The spelling errors in these datasets are anonymized and aggregated from educational or general text sources.

Potential Risks. Spelling correction tools could potentially be misused to alter text in ways that change intended meaning. We emphasize responsible deployment and recommend human verification for sensitive applications such as legal or medical documents.

Environmental Impact. Our framework distills knowledge into a compact 4B-parameter student model, significantly reducing inference costs compared to large language model approaches. This contributes to more efficient and environmentally conscious model deployment.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The three contributions in Section 1 map directly to the empirical claims; the abstract previews the same Cor-F1 numbers (83.8/62.3/98.2/73.0).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 8 discusses the limitations (Chinese-only scope, training-time compute, prompt-based teacher specialization, interpretability of the learned arbitration weights, and practical risks of wrong corrections on real-world text), with extended discussion in Appendix A.16.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper contains no formal theorems, propositions, or proofs; all mechanism analysis is empirical and requires no explicit theoretical assumptions.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Pseudocode is in Appendices A.4–A.6; software and hardware in A.7; model and teacher specifications in A.8–A.9; hyperparameters and training in A.10–A.11; datasets in A.13.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All four evaluation benchmarks (SIGHAN 2013/14/15, LEMON, ECSpell, CSCD-NS) are publicly available; code and model weights are not released at submission; algorithmic specifications, hyperparameters, and software/hardware details are given in Appendices A.4–A.13.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyperparameters, training protocol, and inference configuration are fully specified in Appendices A.10–A.12.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Results are reported as point estimates; error bars and statistical significance tests are not reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: GPU type, count, memory, and full software stack are specified in Appendix A.7.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors reviewed the NeurIPS Code of Ethics and confirm compliance; data usage, privacy, and potential risks are addressed in Appendix A.18.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Appendix A.18 discusses both positive impacts (reduced inference cost from distilling a compact 4B student in place of 14B-class correctors) and potential negative impacts (possible misuse of correction systems for unintended text alteration), and recommends human verification for sensitive applications such as legal or medical documents.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: No model weights, APIs, or dual-use artifacts are released with this submission; the work does not introduce assets with meaningful misuse potential beyond the existing CSC literature.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: All datasets (SIGHAN 2013/14/15, LEMON, ECSpell, CSCD-NS) and models are publicly released and credited to their original publications (Section 5 and Appendices A.8 and A.13); their license terms are not restated in the paper.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper introduces no new datasets, annotation schemes, or released model checkpoints; all assets used are pre-existing public benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or human-subject annotation was conducted; all evaluation uses pre-existing public benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No human subjects research was conducted; all experiments use pre-existing anonymized public benchmark datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are core components of the method: the four teachers are role-prompted DeepSeek-R1-Distill-Qwen-14B models and the student is Qwen3-4B (Sections 3–4.5, Appendices A.8–A.9).

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.