WHY FREEZE THE TARGET MODEL? ALIGNING THE LLM TOWARD ITS SMALL DRAFTER FOR SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Speculative decoding accelerates large language model (LLM) inference by using a small language model (SLM) to draft several tokens that the LLM verifies in parallel. Existing methods typically focus on training the SLM towards LLM to fully imitate the LLM's distribution. However, we observe that, in open-ended generation, many rejected SLM outputs are semantically correct but differ from the LLM in wording or style. For a capacity-limited SLM, matching the LLM's wording and style in addition to generating correct content remains challenging. This raises a key question: can the LLM instead accommodate the SLM's expression style while retaining its own capabilities? Building on this idea, we propose SpecL2S, a two-stage mutual alignment framework. We first use content-aware distillation to train the SLM to prioritize content accuracy over stylistic imitation. We then freeze the trained SLM and optimize a lightweight soft prompt for the LLM, adapting its expression preferences to the SLM while preserving its content-critical behavior. Together, these two stages align the SLM toward the LLM's capabilities and the LLM toward the SLM's expression style. The resulting pair uses standard parallel verification without an additional semantic judge. Across five open-ended benchmarks, SpecL2S increases the mean number of accepted tokens per target call from 3.05 to 4.57 and achieves a 2.43× decoding speedup, while quality remains nearly unchanged.

1 INTRODUCTION

Autoregressive generation makes capable large language models (LLMs) expensive to serve because each new token requires another sequential pass through the model. Speculative decoding reduces this latency by asking a much smaller language model (SLM) to propose several future tokens and verifying the entire block with the LLM in parallel (Chen et al., 2023; Leviathan et al., 2023; Xia et al., 2023). Accepted tokens eliminate otherwise sequential LLM calls, while rejected tokens are replaced by the LLM. The practical speedup therefore depends not only on how cheaply the SLM drafts, but also on how often its proposals agree with the target.

Most learned speculative-decoding methods treat disagreement as a defect of the SLM. They distill the target distribution into the SLM, adapt the SLM to online queries, specialize it to a domain, or attach target-coupled prediction heads (Zhou et al., 2024; Liu et al., 2024; Yi et al., 2024; Cai et al., 2024; Li et al., 2024a). All of these strategies improve draft accuracy, but they retain the same direction of responsibility: the target LLM is fixed, and the smaller model must learn to reproduce its token-level behavior. This requirement is demanding because the SLM must simultaneously acquire the target's substantive ability and mimic its lexical, syntactic, formatting, and stylistic preferences. The latter consume modeling capacity even when they do not determine whether an answer is correct or useful.

Our starting observation is that token disagreement does not always imply content disagreement. This distinction is especially important in open-ended dialogue, where many expressions can convey the same valid answer. An SLM may select a synonym, connective, sentence structure, or formatting convention that preserves the intended content but differs from the LLM's preferred continuation. Token-level verification nevertheless rejects the draft at the first such mismatch and asks the LLM

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

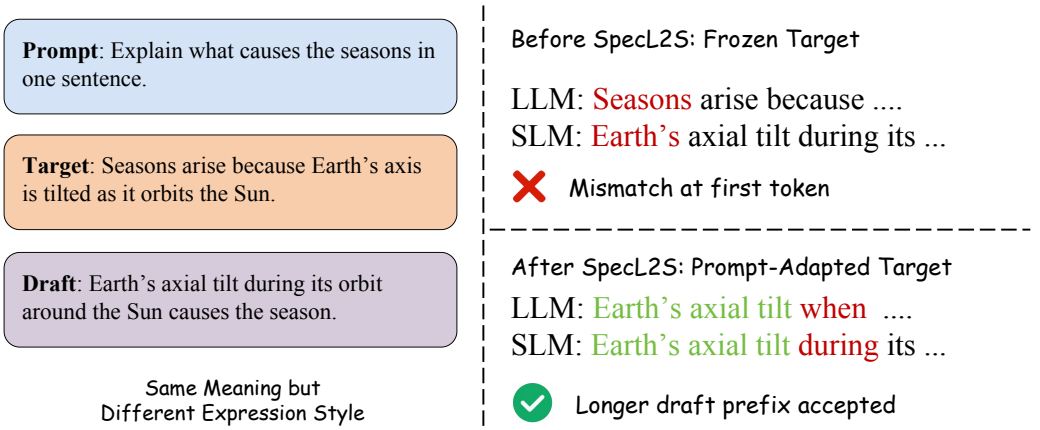

Figure 1: A toy example of avoidable surface-form rejection. Aligning the target's expression style with the draft moves the first mismatch later, allowing standard verification to accept a longer draft prefix.

to regenerate the remaining text. In effect, the system sometimes pays the cost of correction even though the SLM's content was already satisfactory. Moreover, requiring a small model to produce both the right content and the target's exact manner of saying it is a harder learning problem than requiring it to produce the right content alone. Figure 1 illustrates this failure. Although the target and draft give the same explanation with different wording and clause order, strict verification rejects the draft immediately; target-side expression alignment moves the first mismatch later and permits a longer accepted prefix under the same token-level verifier.

This observation motivates us to revisit a convention that is rarely questioned: *why must the target model remain fixed?* If the SLM already expresses correct content in a stable but different style, an alternative is to make the LLM compatible with that style. Once the target recognizes the SLM's surface realization, residual disagreement is more likely to reflect a substantive error that genuinely warrants correction. This reverses the usual logic of speculative-decoding training. We first decide that the LLM should move toward the SLM's way of expression; only under this premise do we ask what the SLM should still learn from the LLM. The answer is the part that cannot be safely delegated: content-bearing decisions that determine factuality, reasoning, instruction fulfillment, and the final answer.

Based on this division of labor, we propose SpecL2S, an asymmetric mutual-alignment framework. In the first stage, content-aware distillation concentrates the SLM's limited capacity on decisions whose alteration changes the utility of the response. In the second stage, the trained SLM is fixed and its remaining mismatch patterns supervise a soft prompt for the LLM. The prompt increases the LLM's compatibility with SLM continuations at surface-form mismatches, while distribution-preservation constraints prevent the LLM from following the SLM when content is at stake. At inference time, the SLM drafts normally and the prompt-adapted LLM performs standard parallel verification; the adaptation adds only a short learned prefix rather than a second copy of the target model (Lester et al., 2021; Li & Liang, 2021).

Our evaluation is designed to test whether this change of alignment direction can produce approximately $2.43\times$ speedup over autoregressive target decoding while keeping open-ended response quality nearly unchanged, and whether the gain comes from fewer surface-form rejections rather than shorter or lower-quality answers. Our contributions are:

- We identify a source of avoidable speculative-decoding failure: content-correct SLM continuations can be rejected because their surface realization differs from that of the target LLM.
- We introduce asymmetric mutual alignment, which makes the LLM accommodate the SLM's expression while reserving SLM-to-LLM learning for content-critical decisions.

- We give a parameter-efficient realization based on content-aware SLM distillation and constrained soft-prompt adaptation of the LLM, together with an evaluation protocol that separates efficiency gains from changes in answer utility.

2 Related Work

2.1 Speculative decoding

Speculative decoding replaces several serial target-model steps with a cheap draft followed by one parallel target-model verification pass. Classical rejection-sampling formulations are lossless with respect to the target distribution (Chen et al., 2023; Leviathan et al., 2023), and SpecDec trains an efficient task-specific drafter for sequence-to-sequence generation (Xia et al., 2023). Subsequent systems reduce proposal cost or expose more candidate branches: Draft & Verify skips selected target layers (Zhang et al., 2024), while Medusa trains multiple future-token heads on the target representation (Cai et al., 2024). In all cases, speed depends on both drafting overhead and agreement with the target verifier.

Training-based methods therefore concentrate on making proposals imitate a fixed target more accurately. DistillSpec studies on-policy knowledge distillation and divergence objectives for compact autoregressive drafters (Zhou et al., 2024); Online Speculative Decoding adapts the drafter to the observed query stream (Liu et al., 2024); and specialized drafters learn language- or domain-specific target behavior (Yi et al., 2024). The EAGLE family couples the drafter more tightly to the target: EAGLE autoregressively predicts second-to-top-layer features (Li et al., 2024a), EAGLE-2 allocates a context-dependent draft tree using calibrated draft confidence (Li et al., 2024b), and EAGLE-3 replaces feature regression with direct token prediction from fused multi-layer target features and simulates multi-step generation during training (Li et al., 2025). Despite their different architectures and training recipes, these methods keep the target fixed and assign residual draft–target mismatch to the proposal model. SpecL2S retains SLM training for content-bearing decisions but also adapts the target's surface preferences, dividing the alignment burden between the two models.

2.2 Relaxing verification constraints

An orthogonal line of work raises acceptance by weakening the requirement that decoding reproduce the original target distribution exactly. DistillSpec adds a lenience factor to rejection sampling (Zhou et al., 2024), and Fuzzy Speculative Decoding accepts candidates according to a tunable measure of draft–target distributional divergence (Holsman et al., 2025). Recent training-free methods make this relaxation context dependent. MARS uses the target's local logit margin to retain plausible alternatives when its decision is weak (Song et al., 2026); FLy combines an entropy gate with deferred verification to recover differently worded but semantically valid continuations (Li et al., 2026); and LinguaSpec uses linguistic categories and category-normalized uncertainty to apply different drafting and verification rigor to content-rich and structurally predictable tokens (Wang et al., 2026). These methods are plug-and-play, but they make the quality–speed trade-off online by deciding which mismatched draft tokens may bypass the standard acceptance rule.

SpecL2S addresses the same inefficiency from the model side rather than the verifier side. It learns a pair-specific soft prompt from observed mismatches, using parameter-efficient prompt adaptation (Lester et al., 2021; Li & Liang, 2021), and then runs the standard speculative verification algorithm against the adapted target. Thus, no permissive threshold or semantic judge is added to the serving path. The cost is that the method is not training-free and is lossless only with respect to the adapted target distribution; its preservation objective and direct quality evaluation are therefore essential parts of the method rather than optional safeguards.

3 METHOD

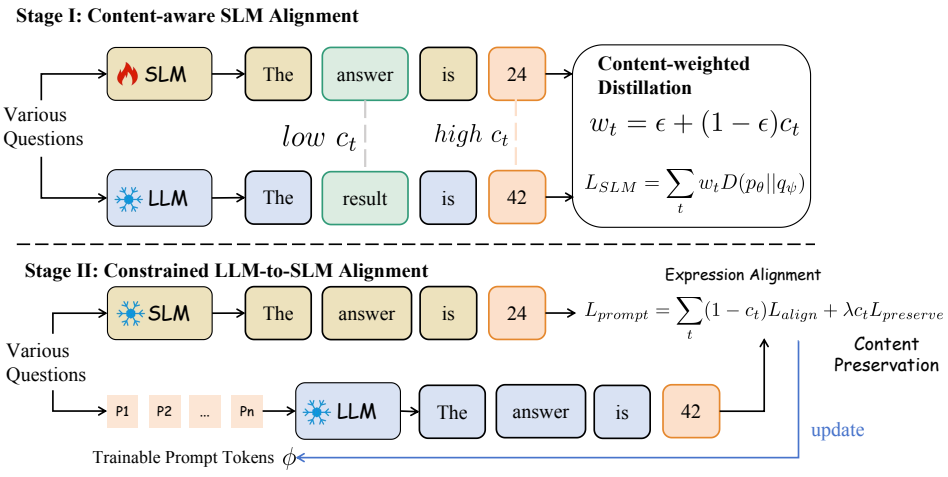

Figure 2: Overview of SPECL2S. Stage I trains the SLM by assigning larger distillation weights to content-critical decisions. Stage II freezes both model checkpoints and updates only the target soft prompt: low-criticality mismatches drive expression alignment, while high-criticality mismatches constrain the adapted target to preserve its original behavior.

SPECL2S divides draft–target alignment according to the effect of a token choice on response utility (Figure 2). Stage I uses the original target distribution to teach the SLM, but concentrates the SLM's limited capacity on content-critical decisions. Stage II fixes the resulting SLM and the target checkpoint, then learns a soft prompt that moves the target toward the SLM only where their disagreement is unlikely to change the answer. The criticality score therefore has complementary roles across the two stages: it strengthens SLM supervision at substantive decisions and restricts target adaptation to benign expression differences. The trained pair requires neither the criticality evaluator nor a modified verifier at inference time.

3.1 PROBLEM FORMULATION

Let q_ψ denote the SLM drafter and p_θ the original target LLM. Given a prefix $h_t = (x, y_{<t})$, the drafter proposes a K-token block, which the target verifies in one parallel forward pass. The verifier accepts the longest valid draft prefix; a disagreement at an early position discards that token and all later draft tokens in the block. We adapt the target by prepending a learned soft prompt ϕ:

$$p_{\theta,\phi}(\cdot \mid h_t) = p_\theta(\cdot \mid [\phi; h_t]), \tag{1}$$

while keeping the target parameters θ frozen. Our objective is to increase agreement between q_ψ and $p_{\theta,\phi}$ on interchangeable surface realizations without making the adapted target inherit factual, reasoning, or instruction-following errors from the drafter.

3.2 MOTIVATION

As a pilot, we freeze both model checkpoints and optimize only a short target soft prompt. This simple intervention increases accepted tokens per target call from 3.05 to 3.48, while MT-Bench Open remains nearly unchanged (8.40 vs. 8.39). Thus, target-side adaptation can recover agreement without a measurable quality loss in this setting. It must nevertheless be constrained: uniformly increasing the probability of SLM proposals would also reward factual, reasoning, and instruction-following errors.

3.3 STAGE I: CONTENT-AWARE SLM ALIGNMENT

Stage I directs SLM capacity toward disagreements that change answer utility. Starting from the initial drafter, we collect on-policy draft–target mismatches. At each mismatch position t, the SLM proposes $\hat{y}_t$ and the target prefers $y_t^\star$. From their common prefix, we construct two counterfactual continuations $r_t^S = \mathrm{Complete}_{p_\theta}(h_t, \hat{y}_t)$ and $r_t^T = \mathrm{Complete}_{p_\theta}(h_t, y_t^\star)$. Both branches are generated by the original target with deterministic decoding for at most 64 tokens or until a stop condition, so the token at position t is the only controlled difference.

We define c_t as a calibrated probability that replacing the target token with the SLM token changes the meaning or task utility of the continuation. A blind evaluator receives the instruction, common prefix, and an anonymized, randomly ordered continuation pair. Across three independently randomized presentations, it returns task-conditioned semantic substitutability $e_t \in [0,1]$ and a utility score $u_{t,k}^b \in [0,1]$ for each branch $b \in \{S,T\}$ and dimension $k \in \mathcal{K} = \{\text{factuality}, \text{reasoning}, \text{instruction fulfillment}, \text{answer correctness}\}$. Bars below denote the mean over the three evaluations. We first compute a conservative raw criticality score and then calibrate it:

$$a_t = \max\left\{1 - \bar{e}_t, \max_{k\in\mathcal{K}} \left|\bar{u}_{t,k}^S - \bar{u}_{t,k}^T\right|\right\}, \qquad c_t = g(a_t). \tag{2}$$

Here, $g : [0,1] \to [0,1]$ is an isotonic calibrator fitted on a disjoint, stratified human-labeled subset, where a positive label means that the token substitution changes meaning or at least one utility dimension. The maximum in Equation 2 prevents a serious factual or reasoning change from being diluted by agreement on other dimensions. Thus, interchangeable wording receives $c_t \approx 0$, whereas a changed fact, reasoning step, instruction outcome, or final answer receives $c_t \approx 1$. We assign $c_t = 0$ to non-mismatch and unscored positions.

On target responses and on-policy draft prefixes, we train the SLM with the content-weighted distillation objective shown in Figure 2:

$$w_t = \epsilon + (1-\epsilon)c_t, \qquad \mathcal{L}_{\mathrm{SLM}} = \sum_t w_t D(p_\theta(\cdot \mid h_t) \,\|\, q_\psi(\cdot \mid h_t)), \tag{3}$$

where D is the chosen distillation divergence and $\epsilon \in (0,1)$ retains a learning signal at low-criticality and unscored positions. Thus, high-criticality decisions receive full-strength target supervision, whereas style and wording preferences no longer dominate the SLM's training budget. This asymmetry is intentional: Stage II will absorb part of the remaining surface mismatch from the target side, but it cannot safely repair missing SLM content capability.

3.4 STAGE II: CONSTRAINED LLM-TO-SLM ALIGNMENT

After Stage I, we freeze the trained SLM and collect a fresh set $\mathcal{M}$ of on-policy mismatches. We recompute c_t on these examples because SLM training changes both the frequency and type of residual errors. The SLM and target checkpoints remain frozen throughout Stage II; only the soft prompt ϕ is updated. For each $t \in \mathcal{M}$, the expression-alignment term increases the adapted target probability of the SLM token, whereas the content-preservation term anchors the full adapted distribution to the original target:

$$\begin{aligned} \mathcal{L}_{\mathrm{align},t} &= -\log p_{\theta,\phi}(\hat{y}_t \mid h_t), \\ \mathcal{L}_{\mathrm{preserve},t} &= \mathrm{KL}(p_\theta(\cdot \mid h_t) \,\|\, p_{\theta,\phi}(\cdot \mid h_t)), \\ \mathcal{L}_{\mathrm{prompt}} &= \sum_{t\in\mathcal{M}} [(1-c_t)\mathcal{L}_{\mathrm{align},t} + \lambda c_t \mathcal{L}_{\mathrm{preserve},t}] + \beta\|\phi\|_2^2. \end{aligned} \tag{4}$$

For low c_t, expression alignment dominates and teaches the target the SLM's recurrent lexical, syntactic, and formatting preferences. For high c_t, the alignment weight vanishes while the preservation constraint becomes strong, preventing the soft prompt from legitimizing substantive SLM errors. The coefficient λ controls this quality–acceptance trade-off, and the norm penalty limits the magnitude of the intervention. Only ϕ is stored for a model pair, so the adaptation is reversible and the original target checkpoint remains intact.

3.5 Training and inference procedure

The training order is deliberately asymmetric. We (1) collect mismatches from the initial SLM and assign criticality labels; (2) train the SLM with Equation 3; (3) freeze the selected SLM, collect fresh on-policy mismatches, and recompute their criticality; and (4) optimize the target soft prompt with Equation 4. Recollecting data after Stage I ensures that target adaptation matches the final drafter. Reversing the order would make the prompt chase an SLM distribution that subsequently changes.

At inference time, the fixed SLM proposes K tokens and $p_{\theta,\phi}$ verifies the block with the standard speculative-decoding algorithm. The criticality evaluator is used only to construct training data, so no semantic judge or auxiliary classifier enters the latency-critical path. Decoding preserves the distribution of the adapted target $p_{\theta,\phi}$, not necessarily that of the original p_θ. Accordingly, we evaluate response quality, length, factuality, and instruction following together with acceptance length and wall-clock speedup, and isolate the two alignment directions through equal-budget ablations.

4 Experiments

We test whether SpecL2S improves end-to-end decoding over standard and trained speculative decoding, whether the gain comes from recovering benign surface variants, whether both alignment directions are necessary, and whether the adapted target retains utility across open-ended tasks.

4.1 Models and training data

We evaluate four SLM–LLM pairs from the Qwen2.5 and Llama families. The primary pair uses Qwen2.5-1.5B-Instruct as q_ψ and Qwen2.5-7B-Instruct as p_θ. The other pairs vary drafter scale, model family, and target scale while retaining tokenizer compatibility within each pair; full configurations appear in Appendix A.

Stage I uses a 120,000-prompt mixture of UltraChat and OpenAssistant together with on-policy prefixes. Stage II uses a disjoint set of 50,000 draft–target mismatch pairs. The evaluator labels a mismatch by comparing completions obtained after the SLM and target alternatives. To prevent contamination, none of the benchmark questions used for evaluation appears in either training split. We split a stratified set of 6,000 human-labeled mismatches equally into calibration and audit partitions; the former fits g in Equation 2, and the latter evaluates the resulting criticality scores.

For every trainable baseline, the drafter sees the same number of prompts and target tokens. The SLM-side-only and full methods receive the same Stage-I budget; the LLM-side-only and full methods receive the same number of mismatch examples for prompt tuning. Hyperparameters are selected on a held-out validation set by a joint criterion that maximizes accepted length subject to a quality-retention constraint.

4.2 Benchmarks

All quality experiments use open-ended prompts that remain within the practical capability range of the small drafter. **AlpacaEval 2.0** provides simple single-turn instruction-following prompts and reports length-controlled win rate (Dubois et al., 2024). **Koala-180** contains 180 conversational queries posted by real users; its original construction already removes coding prompts (Geng et al., 2023). We further define three transparent easy/open subsets. **Vicuna-Bench Open** retains the generic, knowledge, role-play, common-sense, counterfactual, and writing categories while excluding coding, mathematics, and Fermi-estimation questions. **MT-Bench Open** retains writing, role-play, extraction, and humanities prompts from MT-Bench (Zheng et al., 2023). **IFEval-Simple** retains prompts containing a single verifiable instruction, allowing free-form generation while testing basic constraint following (Zhou et al., 2023).

These five settings cover ordinary assistance, explanation, role-play, rewriting, and lightweight instruction following without making advanced mathematical or programming ability a prerequisite. Subset membership is fixed before generation and released with the evaluation scripts. For every setting, we report generated length and use identical stopping, sampling, and judge parameters across comparable systems. We study serving robustness separately with concurrency and sequence-length sweeps, using only prompts from the same open-ended evaluation pool.

4.3 BASELINES

We compare the following systems under matched serving conditions:

- **Target AR**: conventional autoregressive decoding with the original target LLM.
- **Standard SD**: speculative decoding with the unadapted SLM and original target.
- **DistillSpec** and **Online SD**: representative offline and deployment-aware SLM alignment methods (Zhou et al., 2024; Liu et al., 2024).
- **EAGLE-2** and **Medusa**: target-coupled multi-token proposal systems included as broader systems baselines (Li et al., 2024b; Cai et al., 2024).
- **Relaxed verification**: a quality–latency baseline that changes the acceptance rule using a tuned divergence or margin threshold (Holsman et al., 2025; Song et al., 2026).
- **SLM-side only**: our Stage-I content-aware distillation paired with the original target.
- **LLM-side only**: target soft-prompt alignment learned toward the raw, untrained SLM.
- **SpecL2S**: the complete two-stage method.

All speculative systems use the same draft length K unless a method requires its own search structure. We additionally sweep K and the preservation weight λ in Appendix D.

4.4 METRICS AND SERVING PROTOCOL

Efficiency is measured by wall-clock latency, output tokens per second, speedup over Target AR, and accepted tokens per target call. Timings include drafting, prompt-prefix overhead, verification, rejection recovery, and host synchronization. We warm up each model, run at batch size 1, and repeat each setting with five seeds on one NVIDIA A100 80GB GPU using a PyTorch 2.5/CUDA 12.4 stack. Confidence intervals use paired bootstrap resampling over prompts.

Appendix D reports concurrency, sequence-length, latency, and memory stress tests.

Quality is measured by benchmark-specific scores and pairwise judgments against the original target response. Judges are blind to method identity and response order is randomized. We separately report response length, factual consistency, instruction compliance, coherence, and style quality. For IFEval-Simple we also report exact constraint satisfaction, but the response content itself remains open-ended. These checks prevent apparent speedups caused by systematically shorter outputs or a more permissive adapted model.

5 RESULTS AND ANALYSIS

5.1 MAIN RESULTS

On the primary pair, SPECL2S reaches $2.43\times$ speedup and 4.57 accepted tokens per target call while retaining an MT-Bench Open score of 8.39 (Table 1). It improves quality by 0.09 over relaxed verification while remaining exact with respect to the adapted target.

Strict baselines retain original-target quality by construction; the relevant test is whether target adaptation reduces mismatch without changing utility.

5.2 GENERALIZATION ACROSS BENCHMARKS AND MODEL PAIRS

Across five benchmarks, quality changes by at most 0.40 in the native metric while first-token mismatch falls by 12.60%–15.20% absolute (Table 2). Speedup ranges from $2.41\times$ on IFEval-Simple to $2.55\times$ on Koala-180. Across four model pairs, it ranges from $2.01\times$ to $2.56\times$, with normalized quality changes between -0.08 and -0.01; the full pair table is in Appendix D.

5.3 WHERE DO REJECTIONS COME FROM?

We manually audit a stratified sample of 1,200 mismatches before and after Stage II. Table 3 decomposes them by consequence rather than by token identity. Before adaptation, surface-form

Table 1: Primary-pair results under identical serving settings.

Method	MT-B. Open ↑	Accepted ↑	Speedup ↑
Target AR	8.40	–	1.00×
Standard SD	8.40	3.05	1.90×
DistillSpec	8.40	3.39	2.05×
Online SD	8.40	3.55	2.12×
EAGLE-2	8.40	3.88	2.26×
Medusa	8.40	3.61	2.18×
Relaxed verification	8.30	4.10	2.26×
SpecL2S	**8.39**	**4.57**	**2.43**×

Table 2: Cross-benchmark quality and efficiency. "Target" and "Ours" use the benchmark's native quality metric. Mismatch is the first-rejection rate.

Benchmark	Target	Ours	SD mismatch	Ours mismatch	Speedup
MT-Bench Open	8.40	8.39	36.80%	21.60%	2.43×
AlpacaEval 2.0	50.00	49.70	35.10%	20.90%	2.48×
Koala-180	50.00	49.80	33.60%	19.10%	2.55×
Vicuna-B. Open	50.00	49.60	34.40%	19.70%	2.52×
IFEval-Simple	85.40	85.10	32.80%	20.20%	2.41×

differences include lexical choice, discourse structure, and stylistic formatting. Together, these categories account for 56.30% of first rejections. Target adaptation recovers 69.80% of them while recovering only 3.40% of content-critical rejections. This is the central mechanistic prediction of our method: the target becomes more compatible in how it says an acceptable answer, without becoming indiscriminate about what the answer says.

We also pair each rejected SLM continuation with the target-corrected response and ask human annotators whether both are acceptable before seeing token-level agreement. The SLM response is independently acceptable in 91.70% of surface-labeled cases but only 8.60% of content-labeled cases. This gap supports the use of continuation utility, rather than raw target probability, as the routing criterion.

5.4 Ablation study

Removing either alignment direction reduces speedup, whereas removing the preservation term raises acceptance but lowers mean quality retention to 95.40% (Table 4). Reversing the training order also drops speedup to $2.20\times$, supporting prompt fitting after the SLM stabilizes. Additional optimization ablations appear in Appendix D.

5.5 Router reliability and quality safeguards

Against human continuation labels, the utility router reaches 89.70% agreement and limits false accommodation of content errors to 3.40%; a human-label upper bound improves speedup by only $0.02\times$. The adapted target retains 99.21% normalized quality, and its output-length ratio to the original target is 1.01. Adapted-target autoregressive decoding and SpecL2S obtain the same benchmark scores, confirming that speculative execution reproduces the adapted distribution. Detailed routing, quality, latency, sensitivity, and serving-stress results are reported in Appendix D; notably, speedup falls to $1.22\times$ at concurrency 32 as batching saturates the target device.

6 Limitations

SpecL2S deliberately changes the served target distribution. It is lossless with respect to the prompt-adapted target, not the original checkpoint, so quality must be measured directly. The criticality router is costly to train and may miss subtle factual or safety consequences; conservative thresholds reduce this risk at the cost of leaving some benign rejections unresolved.

Table 3: Mismatch diagnosis. "Share" is the fraction of first rejections in each category; "recovered" is the fraction no longer rejected after target adaptation. Human agreement validates the automatic router.

Mismatch category	Share ↓	Recovered ↑	Human agreement ↑
Lexical / paraphrase	24.30%	71.20%	89.60%
Syntax / discourse order	17.80%	64.80%	85.30%
Benign formatting / style	14.20%	76.50%	91.20%
Factual content	18.10%	3.10%	93.40%
Reasoning transition	15.40%	4.70%	87.90%
Instruction / exact format	10.20%	2.40%	94.10%

Table 4: Ablations on the primary pair. Quality retention averages normalized performance across the five small-model-friendly open-ended benchmarks.

Variant	MT-B. Open ↑	Retention ↑	Accepted ↑	Speedup ↑
Full SPECL2S	8.39	99.21%	4.57	2.43×
− content weighting	8.39	99.00%	4.24	2.32×
− target-side alignment	8.40	100.00%	3.69	2.20×
− preservation loss	8.12	95.40%	4.86	2.55×
Reverse training order	8.37	98.00%	3.82	2.20×

The learned prompt is pair- and domain-specific and may require retraining after the drafter or traffic distribution changes. We do not evaluate exact-match mathematics or executable code, and gains shrink under high concurrency, very short generation, or small target models because batching and prompt-prefill costs dominate.

7 CONCLUSION

Speculative decoding need not make a small drafter imitate every preference of a fixed target. SPECL2S instead teaches the SLM content-critical decisions and adapts the target only to benign expression differences. It achieves $2.43\times$ speedup, recovers 69.80% of surface-form rejections, and retains 99.21% normalized quality across five open-ended benchmarks. Alignment can therefore be mutual in direction while remaining asymmetric in responsibility.

REPRODUCIBILITY STATEMENT

The appendix specifies all training and serving settings. We will release code, evaluation prompts, mismatch labels, learned prompts, anonymized judgment pairs, and per-example timing logs with hardware traces.

AI USE STATEMENT

Generative AI assisted with language editing, document organization, and the architecture figure. The authors remain responsible for all research and claims.

REFERENCES

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5209–5235. PMLR, 2024. URL https://proceedings.mlr.press/v235/cai24b.html.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023. URL https://arxiv.org/abs/2302.01318.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024. URL https://arxiv.org/abs/2404.04475.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. Koala: A dialogue model for academic research. Berkeley Artificial Intelligence Research Blog, 2023. URL https://bair.berkeley.edu/blog/2023/04/03/koala/.

Maximilian Holsman, Yukun Huang, and Bhuwan Dhingra. Fuzzy speculative decoding for a tunable accuracy–runtime tradeoff. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 26257–26273. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1346.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Jinze Li, Yixing Xu, Guanchen Li, Shuo Yang, Jinfeng Xu, Xuanwu Yin, Dong Li, Edith C. H. Ngai, and Emad Barsoum. Training-free loosely speculative decoding: Accepting semantically correct drafts beyond exact match. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/acb94e709f02895fd98b5867f0b184f3-Abstract-Conference.html.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, 2024a. URL https://arxiv.org/abs/2401.15077.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-2: Faster inference of language models with dynamic draft trees. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7421–7432. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.emnlp-main.422.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-3: Scaling up inference acceleration of large language models via training-time test. *arXiv preprint arXiv:2503.01840*, 2025. URL https://arxiv.org/abs/2503.01840.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Alvin Cheung, Zhijie Deng, Ion Stoica, and Hao Zhang. Online speculative decoding. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 31131–31146. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24y.html.

Jingwei Song, Xinyu Wang, Hanbin Wang, Xiaoxuan Lei, Tianyu Shi, Shixin Han, Eric Yang, Xiao-Wen Chang, and Lynn Ai. MARS: Unleashing the power of speculative decoding via margin-aware verification. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 8652–8663. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.421.

Jingyi Wang, Jiaqi Huang, Zunnan Xu, Jun Zhou, Kehong Yuan, and Xiang Qian. Training-free adaptive speculative decoding via linguistic priors. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 21184–21194. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1065.

Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3909–3925. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.257.

Euiin Yi, Taehyeon Kim, Hongseok Jeung, Du-Seong Chang, and Se-Young Yun. Towards fast multilingual LLM inference: Speculative decoding and specialized drafters. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10789–10802. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.602.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. Draft & verify: Lossless large language model acceleration via self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 11263–11282. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.607.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023. URL https://arxiv.org/abs/2306.05685.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. Distillspec: Improving speculative decoding via knowledge distillation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rsY6J3ZaTF.

A Implementation Details

Model configuration. The primary pair is Qwen2.5-1.5B-Instruct drafting for Qwen2.5-7B-Instruct. Additional pairs are Qwen2.5-0.5B-Instruct $\rightarrow$ Qwen2.5-7B-Instruct, Llama-3.2-1B-Instruct $\rightarrow$ Llama-3.1-8B-Instruct, and Qwen2.5-1.5B-Instruct $\rightarrow$ Qwen2.5-14B-Instruct. Each pair uses its native shared tokenizer, bfloat16 weights, and a 32,768-token context window. The learned soft prompt contains 32 virtual tokens initialized from the mean target-token embedding plus Gaussian noise with standard deviation 0.01. Both stages use AdamW; learning rates are 2×10^{-5} for the SLM and 5×10^{-3} for the prompt, with effective batch sizes of 128 and 256 and training lengths of 3,000 and 1,500 steps, respectively. The loss weights are $\epsilon = 0.10$, $\lambda = 2.00$, and $\beta = 10^{-4}$.

Serving configuration. We run inference on one NVIDIA A100 80GB GPU with PyTorch 2.5, CUDA 12.4, and a custom speculative-decoding runner using FlashAttention-2. Unless otherwise stated, all methods use draft length $K = 8$, temperature 0.7, top-p 0.9, a maximum of 512 new tokens, and the benchmark's standard stop strings. Timing begins after prompt tokenization and ends after the final device synchronization; prefill and decoding are reported separately.

Criticality labels. For each mismatch, we branch from the common prefix and complete both the SLM token and target token alternatives using the original target, greedy decoding, and a 64-token cap. Each pair is evaluated three times with independently randomized order and without model identities. The evaluator returns semantic substitutability and branch-level factuality, reasoning, instruction-fulfillment, and answer-correctness scores; we aggregate them with Equation 2. Two annotators independently assign the binary criticality labels used for calibration and audit, and a third resolves disagreements. We fit the isotonic calibrator only on the calibration partition and report agreement, Brier score, expected calibration error, and category-wise false-positive and false-negative rates on the untouched audit partition.

B TRAINING ALGORITHM

1. Sample prompts from the Stage-I corpus and obtain target distributions and counterfactual continuation labels from the frozen original LLM.
2. Update the SLM with Equation 3, emphasizing decisions with high continuation-utility difference. Select the Stage-I checkpoint on the held-out content and acceptance criteria.
3. Freeze the selected SLM. Generate fresh on-policy draft blocks and record the first mismatch and both continuation alternatives.
4. Score each mismatch to obtain c_t. Construct a surface set from low-c_t examples and a preservation set from high-c_t examples.
5. Freeze both model checkpoints and optimize only the target soft prompt with Equation 4. Select the prompt on the constrained quality–acceptance criterion.
6. At inference, prepend the selected soft prompt to the target context and run the unchanged speculative draft-and-verify algorithm.

C QUALITATIVE CASE PROTOCOL

Each released case contains the user prompt, common prefix, SLM draft, original target continuation, adapted-target continuation, first rejection position, criticality score, and anonymized human judgments. We select cases before viewing method outcomes and balance them across lexical, syntactic, formatting, factual, reasoning, and constraint-following categories. Successful examples must show that the adapted target accepts a semantically equivalent realization; failure examples include both false accommodation of a content error and false rejection of a benign surface variant. This protocol prevents cherry-picking fluent examples that do not explain the measured speedup.

D EXTENDED RESULTS

The following tables provide detailed efficiency, sensitivity, routing, data-efficiency, and serving-stress results. Per-example quality judgments and raw serving logs support the aggregate values.

Table 5: Category-level results over the five evaluation sets. Retention normalizes the adapted-target score by the original target.

Task type	Primary sources	Retention ↑	Mismatch ↓	Speedup ↑
Everyday advice	Alpaca / Koala	99.60%	18.80%	2.58×
Knowledge explanation	Koala / Vicuna	99.40%	21.50%	2.44×
Role-play / dialogue	MT / Vicuna	99.20%	18.90%	2.57×
Writing / rewriting	MT / Vicuna	99.50%	20.10%	2.51×
Single constraint	IFEval-Simple	99.60%	20.20%	2.41×

Table 6: Generalization across model pairs. ΔQ is the mean normalized quality difference from original-target autoregressive decoding.

Model pair	ΔQ ↑	Accept. ↑	Tokens/s ↑	Speedup ↑
Qwen2.5-1.5B → 7B	-0.01	78.40%	68.90	2.43×
Qwen2.5-0.5B → 7B	-0.02	75.20%	65.40	2.32×
Llama-3.2-1B → 3.1-8B	-0.08	68.60%	57.10	2.01×
Qwen2.5-1.5B → 14B	-0.02	80.10%	48.80	2.56×

Table 7: Per-benchmark efficiency. Latency is mean end-to-end seconds per response. Speedup is computed per prompt against Target AR before aggregation.

Benchmark	Latency (s)	Tokens/s	Accepted / call	Speedup
MT-Bench Open	3.84	68.90	4.57	2.43×
AlpacaEval 2.0	4.92	70.30	4.71	2.48×
Koala-180	4.05	71.60	4.82	2.55×
Vicuna-B. Open	4.18	70.80	4.76	2.52×
IFEval-Simple	3.12	66.40	4.43	2.41×

Table 8: Sensitivity to draft length K and preservation strength λ. The selected operating point is $K = 8, \lambda = 2.00$.

K	λ	MT-B. Open ↑	Accepted ↑	Speedup ↑
4	2.00	8.39	3.26	2.18×
6	2.00	8.39	4.03	2.36×
8	2.00	8.39	4.57	2.43×
12	2.00	8.39	4.83	2.38×
8	0.50	8.26	4.83	2.51×
8	1.00	8.35	4.70	2.47×
8	2.00	8.39	4.57	2.43×
8	4.00	8.40	4.22	2.34×

Table 9: Routing accuracy and safety against human continuation-level labels. False accommodation is the fraction of content-critical errors that the adapted target would accept.

Routing signal	Human agreement	False accommodation ↓	Speedup
Token-class heuristic	71.40%	9.80%	2.36×
Target logit margin	76.80%	7.40%	2.39×
LLM pointwise judge	86.90%	4.80%	2.42×
Continuation utility	89.70%	3.40%	2.43×
Human-label upper bound	100.00%	2.90%	2.45×

Table 10: Additional optimization ablations omitted from the main text.

Variant	MT-B. Open ↑	Retention ↑	Accepted ↑	Speedup ↑
Full SpecL2S	8.39	99.21%	4.57	2.43×
Uniform mismatch alignment	8.21	96.10%	4.71	2.49×
Jointly update SLM and prompt	8.35	97.60%	4.01	2.27×

Table 11: Latency and resource breakdown per response under a matched generated-token budget.

Component	Standard SD	SLM-side only	SpecL2S
SLM drafting time (s)	0.62	0.66	0.69
Target verification time (s)	2.14	1.86	1.52
Rejection recovery time (s)	0.78	0.52	0.31
Target calls / response	72.40	60.80	48.20
Peak memory (GB)	47.20	47.30	47.60
Output tokens / second	53.90	62.50	68.90

Table 12: Quality preservation under autoregressive and speculative execution.

System	MT-B. Open	LC win	Koala	Vicuna	IFEval-S	Retain.
Original target AR	8.40	50.00	50.00	50.00	85.40	100.00%
Adapted target AR	8.39	49.70	49.80	49.60	85.10	99.21%
SpecL2S	8.39	49.70	49.80	49.60	85.10	99.21%

Table 13: Data and prompt-length efficiency. Each block changes one factor while keeping the selected operating point fixed.

Factor	Setting	MT-B. Open	Mismatch ↓	Speedup
Mismatch pairs	1k	8.40	34.10%	1.99×
	5k	8.40	29.80%	2.17×
	10k	8.39	26.20%	2.29×
	25k	8.39	22.30%	2.40×
	50k	8.39	21.60%	2.43×
Soft-prompt tokens	8	8.40	27.10%	2.22×
	16	8.39	24.40%	2.34×
	32	8.39	21.60%	2.43×
	64	8.37	21.30%	2.44×
	128	8.34	21.20%	2.43×

Table 14: System stress tests. Both concurrency and generation-length sweeps use matched prompts from the five open-ended evaluation sets.

Sweep	Setting	Requests/s	Tokens/s	Speedup
Concurrency	1	0.31	68.90	2.43×
	4	0.93	151.20	2.08×
	8	1.41	219.60	1.78×
	16	1.88	287.40	1.46×
	32	2.05	318.10	1.22×
Generation tokens	64	0.74	44.80	1.58×
	256	0.43	61.70	2.18×
	512	0.27	68.90	2.43×
	1,024	0.15	74.00	2.61×