DRELAY: GLOBAL DRAFT CONTEXT FOR PREFIX-AWARE PARALLEL SPECULATIVE DECODING REPAIR

Anonymous authors
Paper under double-blind review

ABSTRACT

Parallel drafting reduces the drafting overhead of speculative decoding for large language models (LLMs), but its gains remain limited by the accepted prefix length. Even when the correct token is present in the candidate pool, a single early selection error prevents subsequent predictions from being used. We propose DRelay, which uses global information from the entire draft block to perform prefix-aware selective repair of candidate selections before target-model verification. DRelay bases its decisions on both candidate correlations and the selected path: a global reader extracts predictive information across positions for each candidate. While a causal selector combines candidate-level information extracted by the global read with the tokens selected at preceding positions to determine whether the native choice at the current position is consistent with the global evidence and the selected prefix. It then decides whether to retain or replace the token, thereby repairing early errors and extending the accepted prefix. We further jointly train the draft backbone and the selector, combining candidate-support learning with a repair objective, while weighting the repair loss according to each block position's potential contribution to the consecutive accepted prefix. Across eight diverse benchmarks on an H800 GPU, DRelay consistently improves both average acceptance length and end-to-end decoding performance over DFlash, Domino, and DSpark. Under SGLang serving, DRelay improves average end-to-end speedup over DFlash, Domino, and DSpark by 14.7%–16.8%, 8.7%–9.3%, and 8.1%–9.3%, respectively.

1 INTRODUCTION

Speculative decoding reduces the overhead of token-by-token autoregressive inference in large language models through inexpensive drafting and batched verification. Parallel drafting further reduces drafting overhead by predicting multiple positions in a single forward pass (Leviathan et al., 2023; Chen et al., 2023) . DFlash (Chen et al., 2026) adopts this approach to predict an entire masked draft block in parallel within a single forward pass. Building on this approach, Domino and DSpark (Huang et al., 2026; Cheng et al., 2026) model dependencies among draft tokens through a GRU state and transitions conditioned on preceding tokens, respectively, allowing subsequent predictions to condition on the selected prefix and thereby increasing acceptance length.

However, speculative decoding imposes a key constraint: each verification round can accept only a contiguous correct prefix starting at the first position of the draft block. Consequently, an error at an early position prevents subsequent tokens from being committed in the current round, even if those tokens have been predicted correctly. Moreover, when subsequent predictions are conditioned on an incorrect token, their conditioning context deviates from the correct prefix, so the early error may further affect suffix predictions. This creates an inherent limitation for methods such as Domino and DSpark that introduce additional autoregressive modules: they rely on forward causal dependencies to improve subsequent choices, but information in later predictions cannot directly revise choices already made at earlier positions.

In the base draft model, the correct token is often already present in the candidate set when an early prediction error occurs. Correcting an early selection can therefore enable acceptance of the current token and potentially a subsequent run of correctly predicted tokens. An underutilized opportunity lies in backward information flow within the draft block: global information can guide candidate selection

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.

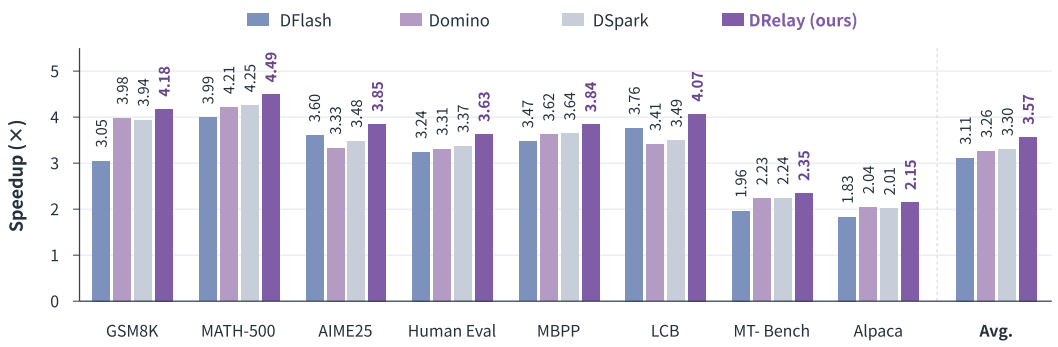


Figure 1: End-to-end decoding speedup on Qwen3-4B. DFlash (Chen et al., 2026), Domino (Huang et al., 2026), DSpark (Cheng et al., 2026), and DRelay are compared against autoregressive decoding.

at earlier positions. After candidate generation, we focus on explicitly revisiting the distributions produced by that forward pass and using global information to repair early selections.

Our central idea is for the entire draft block to serve two roles simultaneously: proposing candidates at each position and providing context for candidate selection. Because predictions at later positions are already available before verification, we can identify the most suitable candidate at an earlier position by comparing candidates' compatibility with global information from the entire draft block and with subsequent predictions. Once a candidate is selected, that specific token should condition subsequent decisions, allowing later choices to follow the actual selected or repaired prefix.

We introduce DRelay, which implements this candidate repair process through the following key components and techniques: (1) A global reader first integrates candidate distributions across positions and then performs reads across positions for each candidate to construct a global context tailored to each alternative token. (2) A causal selector combines global context, the verified target-model history, and an addressable representation of the selected prefix to sequentially decide whether to retain the original draft token or select an alternative candidate. (3) We jointly train the backbone and the selector to learn candidate distributions and repair actions, respectively, and use these two factors to weight the repair loss, focusing training on choices that contribute more to the accepted prefix. As shown in Figure 1, DRelay outperforms DFlash, Domino, and DSpark across all displayed benchmarks on Qwen3-4B, achieving up to a 4.49× end-to-end speedup over autoregressive decoding. Our main contributions are:

- We propose DRelay, which uses the entire draft block to guide repairs to early selections and allows subsequent candidates to directly query the actual repaired prefix, completing candidate selection sequentially before verification.
- We design a joint training objective that simultaneously trains the draft model to propose correct candidates and the selector to make repair decisions, while weighting the repair loss according to prefix value.
- We evaluate DRelay on Qwen3-4B and Qwen3-8B across eight math, code, dialogue, and instruction-following benchmarks. Across target models and temperatures, its average end-to-end speedup exceeds that of the best-performing baseline among DFlash, Domino, and DSpark by 7.8–9.9%, while concurrent-serving throughput reaches up to 4.38× that of autoregressive decoding at matched concurrency.

2 Related Work

Parallel drafting and causal selection. Speculative decoding amortizes target computation by verifying several proposals together (Leviathan et al., 2023; Chen et al., 2023). DFlash obtains parallel proposals through noncausal attention over target-conditioned masked slots (Chen et al., 2026); DFlare improves how deeper draft layers use target features (Zhang et al., 2026a). Domino restores within-block dependence with a GRU-conditioned vocabulary residual (Huang et al., 2026).

DSpark uses a preceding-token Markov head by default, offers a recurrent variant, and schedules verification length according to confidence and load (Cheng et al., 2026). These works establish the parallel-backbone/causal-head split. DRelay adds an explicit read of the resulting candidate distributions, then conditions each decision on addressable selected-prefix entries. The distinction is where selection obtains evidence, rather than whether the backbone has noncausal attention.

Draft-model training and distillation. Target-aligned drafters are trained with supervised or knowledge-distillation objectives. DistillSpec uses on-policy trajectories and a decoding-aware divergence (Zhou et al., 2024); EAGLE and EAGLE-3 learn target features through autoregressive feature prediction and training-time test (Li et al., 2024; 2025). For parallel drafters, DFlash predicts a masked block in one pass (Chen et al., 2026). Draft-OPD replays verification-exposed errors under target-assisted rollouts, while D-PACE derives position weights from a differentiable accepted-length surrogate (Lei et al., 2026; Wu et al., 2026). DRelay jointly trains candidate support and a causal repair selector, weighting repair by prefix value.

Verifier feedback and prefix-conditioned repair. TAPS scores path reach to select a compact prefix-closed verification tree (Wang et al., 2026a). Verification-Aware Training predicts cumulative acceptance and begins a decaying loss schedule at the first rejection (Gu et al., 2026). AdaFlash and online draft co-training adapt drafters to target-generated data (Qian et al., 2026; Wang et al., 2026b). ReTrace and DFlow reuse rejected-trajectory or verifier information across rounds (Lin et al., 2026; Zhang et al., 2026b). DRelay instead reads predicted evidence within the current block, alongside strictly verified target history, and uses that evidence to make addressable repair decisions before verification.

3 Losses from Candidate Selection and Repair Opportunities in Parallel Drafting

Parallel drafting generates an entire block of predictions at once, but verification can accept only a contiguous prefix starting at the beginning of the block. We investigate an opportunity enabled by parallel drafting: once the entire block has been predicted, can its candidate distributions and content at later positions help correct early selections that truncate the accepted prefix? Using native greedy decoding traces from DFlash and Domino with Qwen3-4B (Yang et al., 2025) as the target model, we analyze this opportunity from three perspectives: candidate availability, information from later predictions, and gains in accepted prefix length.

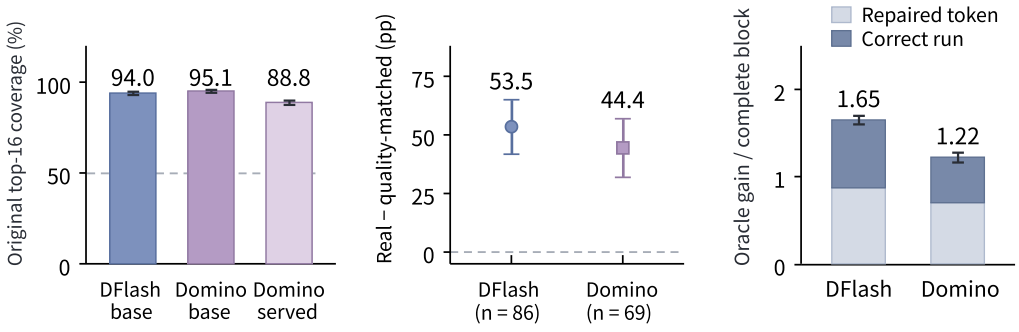


Figure 2: **Candidate availability, continuation evidence, and repair value. Left:** original top-16 coverage at first base/served errors ($n = 7{,}787/5{,}348/4{,}825$). **Middle:** real-minus-correctness-matched best-alternative accuracy at covered early served errors ($n = 86/69$), using served pools (post-head for Domino). Receiver and donor each have four correct subsequent predictions, an oracle condition. Intervals are 97.5% per model for two primary comparisons. **Right:** original-pool one-edit gain over 8,057/5,896 complete blocks, separating the repaired token from the recovered consecutive run. Left/right intervals are 95%; all intervals resample prompts within task. These panels measure repair opportunities, not trained DRelay performance.

3.1 The Correct Candidate Usually Remains Available at the First Error

An early prediction error does not necessarily imply that the correct candidate is absent. As shown in Figure 2, at the first error in the base predictions, the original top-16 candidate pool contains the target token in 94.0% of DFlash cases and 95.1% of Domino cases. We separately analyze the first rejection in Domino's actual draft sequences and find that the original candidate pool still contains the target token in 88.8% of cases. These results show that suitable alternatives for repair are already available at many error positions.

3.2 Later Predictions Provide Discriminative Signals for Early Candidate Selection

Candidate coverage only indicates whether the correct token is present in the candidate pool; it does not determine which candidate should be selected when multiple alternatives are available. We therefore ask whether predictions at later positions in the draft block can provide additional evidence for an earlier candidate. Intuitively, a correct early candidate should not only be consistent with the prefix at its own position, but should also support a plausible continuation of the later predictions.

To test this hypothesis, we select samples with an early error for which the correct token is present in the candidate pool and the next four draft tokens are all correct. For each candidate at the current position, we treat that candidate as the selected token and use a frozen target model to compute the total conditional log-probability of the four subsequent draft tokens given the preceding prefix and the candidate. The candidate with the highest score is considered the candidate supported by the subsequent predictions. As a control, we replace the four subsequent tokens with a suffix from another prompt, thereby changing the semantics of the sequence, and compare the resulting token-repair accuracy. As shown in the middle panel of Figure 2, the real suffix selects the correct candidate in 46/86 DFlash cases and 31/69 Domino cases. Relative to the matched control suffixes, the accuracy improves by 53.5 and 44.4 percentage points, respectively. Thus, subsequent predictions contain context-dependent discriminative information that can help repair an early error, rather than merely providing a suffix unrelated to the preceding candidates.

The entire draft block can therefore serve not only as a source of candidates, but also as global evidence for candidate selection. DRelay accordingly combines each candidate's native confidence with global information from the draft block and the consistency of subsequent predictions. After a candidate is selected, its token is fed into the subsequent decisions through the selected prefix.

3.3 Early Repair Can Recover Subsequent Correct Predictions

The value of a repair is not limited to the position that it corrects. Let R denote the number of draft tokens immediately following the first error that are correct. If all other predictions remain unchanged, correctly repairing this error increases the accepted prefix length by $1 + R$: one token comes from the repaired position and R tokens come from the subsequently recovered consecutive predictions.

To quantify this potential gain, we perform a one-edit oracle experiment. If the target token at the first rejected position is contained in the original top-16 candidate pool, we replace the erroneous token with the target token and leave every other position unchanged. As shown in the right panel of Figure 2, when this gain is averaged over all complete draft blocks and blocks for which the replacement is unavailable contribute zero, the mean accepted length increases by 1.65 and 1.22 tokens for DFlash and Domino, respectively.

This result shows that the value of a repair should be measured by how much accepted prefix it enables. DRelay is not restricted to the first error: it evaluates every position and selectively repairs a position when the global continuation and the selected prefix indicate that the native draft may be inconsistent. Together, the three observations motivate a selector that reads global information, performs repairs selectively, and propagates each selected repair to subsequent decisions. During training, the repair loss should assign greater weight to decisions according to their potential contribution to the consecutive accepted prefix, improving the prefix length realized at inference time.

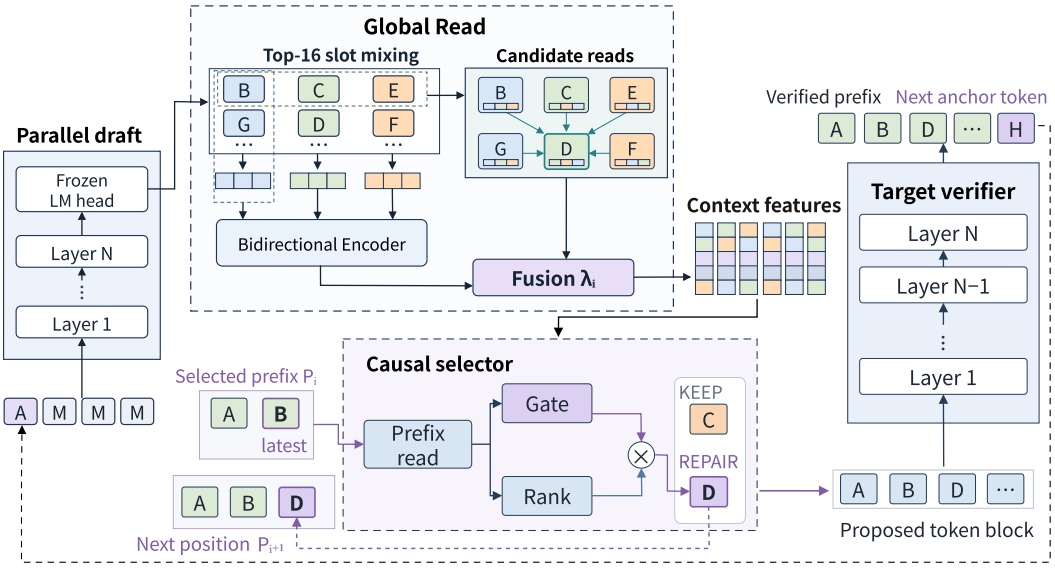


Figure 3: **DRelay reads globally and decides causally.** The parallel draft predicts candidate pools from anchor A and masked slots M. Global Read combines a top-16 slot mixture with candidate-specific reads; Fusion λ_i supplies contextual features to the Causal selector. Prefix read accesses $P_i = [x_0, \ldots, x_{i-1}]$, the selected prefix. Gate and Rank jointly choose KEEP or REPAIR: replacing C with D extends the prefix to $[\mathrm{A}, \mathrm{B}, \mathrm{D}]$, so the next decision uses D. The target verifies the proposed path and supplies the next anchor. Anchor-feature and verified-history projections are omitted for clarity.

4 DRelay: Global Candidate Reads and Causal Selection

DRelay constructs a candidate path after parallel drafting and before target-model verification. As shown in Figure 3, DRelay organizes path repair into two modules. Global Read first aggregates candidate distributions across positions to construct a global context for each candidate. The Causal Selector then combines this context with the verified history and the selected prefix to decide, at each position, whether to retain the base prediction or choose an alternative candidate, producing a path for target-model verification.

4.1 Overall Architecture and Execution Process

Figure 3 illustrates a single round of DRelay execution. At the start of each round, the parallel drafter receives an anchor token confirmed by the target model and historical context features, then generates hidden states and candidate distributions for subsequent positions in a single forward pass.In Figure 3, A denotes the anchor token, and M denotes positions to be predicted. The drafter retains the top-16 candidates at each position. The Global Read module then processes all candidates in the block. The Top-16 slot mixing stage first aggregates the candidate distributions across positions, and the Candidate reads stage then retrieves information across positions relevant to each candidate.

The contexts produced by these two stages are fused and passed to the Causal Selector. At each position, the Causal Selector combines the global context, verified history, and currently selected prefix to decide whether to retain the base prediction or choose an alternative candidate. After completing candidate selection for the entire block, DRelay submits the resulting single path to the target model for verification. The target model accepts a contiguous prefix of this path and produces the anchor token for the next round, after which drafting resumes.

4.2 Global Read: Compare Candidates with the Predicted Block

A single summary can obscure competing continuations. DRelay therefore reads the block at two resolutions. First, two bidirectional attention–FFN blocks mix slot features with soft candidate

summaries $\sum_k \pi_{ik} e_{ik}$, where $\pi_{ik} = q_i(v_{ik}) / \sum_l q_i(v_{il})$. This gives each slot a shared interpretation u_i of the predicted block. Second, candidate-specific attention preserves individual alternatives. With $n_{ik} = \mathrm{LN}(u_i + W_n e_{ik})$, each of four attention heads computes

$$G_{ik} = \sum_{j,l} A_{ik,jl} W_V n_{jl}, \qquad A_{ik,jl} \propto \exp\left(\frac{(W_Q n_{ik})^\top W_K n_{jl}}{\sqrt{d_g}} + \log q_j(v_{jl}) + b_{j-i} \right). \tag{1}$$

Here d_g is the head width and b a relative-position bias. The draft prior weights plausible evidence, while the query lets competing current candidates attend to different continuations.

Uncertain context should not force an edit. Attention is normalized jointly with an additional zero-valued null key/value, permitting a content-free read, and a learned slotwise fusion scalar λ_i scales the global slot and candidate residuals. The scalar is shared across candidates; their reads G_{ik} remain distinct. This contextual fusion is separate from the later KEEP/REPAIR decision. Neither is supervised with oracle future correctness. Global Read uses only the hidden representations and candidate distributions produced during parallel drafting; it does not read future tokens that have not been verified by the target model. The noncausal backbone has already performed an initial information fusion among the masked positions. Global Read therefore does not establish the first interaction across positions; instead, it explicitly reads the backbone's output candidate distributions across positions and extracts global information relevant to each candidate.

4.3 Causal selector: make each repair part of the context

After replacing a token, later choices must use the replacement. An addressable prefix memory stores the anchor and each selected entry, whose key and value encode $(h_j, z_j^{\mathrm{global}}, E(x_j), g_a, j)$. At position i, a candidate can retrieve only entries $j < i$; the latest chosen token $E(x_{i-1})$ also conditions its query directly. This exposes earlier choices individually instead of compressing them into one recurrent state. A separate candidate-specific read of the last 64 strictly verified target features provides direct access to established context instead of relying only on its compression into draft states. Because this history does not change within the block, all its reads are computed before selection.

The Rank branch combines native log-probabilities with a shared hidden residual and compact candidate-specific corrections from global, history, and prefix evidence. Its scores S_{ik} rank nonbase alternatives. The Gate branch asks a different question: whether the best available repair justifies leaving the native choice. This conservative gate is intended to correct likely error positions while minimizing changes to tokens that are already consistent with the global context and selected prefix. Together they define

$$r_{ik} = \mathrm{softmax}_{k>0}(S_{ik}), \qquad p_i^{\mathrm{act}}(0) = \sigma(-g_i), \quad p_i^{\mathrm{act}}(k) = \sigma(g_i) r_{ik} \quad (k > 0). \tag{2}$$

The best alternative k^* replaces the base only if $g_i + \log r_{ik^*} > 0$, with numerical ties resolved to KEEP. Thus a positive edit tendency alone is insufficient when alternatives are ambiguous. Native pool-mass log-odds initialize the gate (Equation 6), and zero terminal residuals recover the native distribution renormalized within the pool.

Candidate memory entries are precomputed, but only the chosen entry becomes visible. In Figure 3, repairing C to D changes the next prefix to $[\mathrm{A}, \mathrm{B}, \mathrm{D}]$. The target then verifies the resulting single path using its ordinary greedy rule. Global reads run once, followed by H dependent decisions; the 35.06M-parameter selector does not repeat a full-vocabulary projection. Its benefit must cover both static read cost and sequential selection cost, which we separate in the evaluation.

5 Joint Training for Support and Repair

A selector cannot recover a token that its pool omits, and a strong pool alone does not determine which choice will extend the accepted prefix. DRelay therefore trains the backbone and selector jointly, while freezing the target, token embedding, and vocabulary readout. During training, prefix memory contains earlier target-greedy tokens under a strict $j < i$ mask. Their identities are fixed; gradients pass through draft features and memory projections. Teacher proposal tokens never enter Global Read.

Learn support and selection together. For target label y_i, define coverage $c_i = \mathbb{I}[y_i \in \mathcal{V}_i]$ and its pool index k_i when covered. A covered base-correct token teaches KEEP; a covered alternative teaches both REPAIR and its identity:

$$\ell_i^{\text{sel}} = \begin{cases} \text{BCE}(g_i, 0), & y_i = v_{i0}, \\ \text{BCE}(g_i, 1) - \log r_{ik_i}, & c_i = 1,\ k_i > 0, \\ 0, & c_i = 0. \end{cases} \tag{3}$$

BCE takes the gate logit as its first argument. A pool miss supplies no valid selector action. Dense vocabulary supervision still teaches the backbone to include the missing token. Combining both responsibilities gives

$$\mathcal{L}_t = \frac{\sum_{b,i} m_{b,i}[-\log q_{b,i}(y_{b,i}) + \alpha_t w_{b,i} \ell_{b,i}^{\text{sel}}]}{\sum_{b,i} m_{b,i}}, \tag{4}$$

where b indexes anchor blocks and m excludes anchors, padding, and noncontiguous response positions. The common valid-token denominator keeps support and action learning on the same scale. Misses are neither KEEP labels nor a separate output class.

Value a decision by the prefix it enables. The one-edit diagnostic shows why equal token credit can overlook a useful repair: a decision matters only if decoding reaches it, and its value grows with the continuation it enables. Let $p_i = p_i^{\text{act}}(k_i)$ when covered and zero otherwise, and set $\bar{p}_i = c_i(1 + \text{stopgrad}(p_i))/2$. We assign raw credit

$$v_i = c_i \underbrace{\prod_{j<i} \bar{p}_j}_{\text{reach}} \underbrace{\left(1 + \sum_{t=i+1}^{H} \prod_{j=i+1}^{t} \bar{p}_j\right)}_{\text{continuation value}}. \tag{5}$$

These factors have the form of marginal prefix value, excluding the current position's own probability. Within each valid block, we normalize credit to mean one, retain a uniform floor, and bound its largest weight by four. The resulting w_i gradually replaces uniform weighting. Detachment and bounded normalization prevent uncertain long products from dominating learning or eliminating downstream supervision. This is a weighted likelihood surrogate, not an unbiased gradient of greedy accepted length.

Learn at boundaries the current drafter reaches. Uniformly sampled anchors provide broad coverage but need not resemble verification boundaries. We mix them with boundaries traced by a no-gradient teacher-prefix scout. Before its first failure, the scout and incremental native selection see the same earlier tokens; the scout can therefore advance by $A + 1$, where A is the accepted proposal count. Sampled anchors receive fresh differentiable forwards. Dense supervision continues under teacher prefixes after a rejection, rather than treating those suffixes as visited native states. The reference schedule increases the native-anchor fraction from 30% to 50% and the selector coefficient from approximately 0.05 to 1; bounded credit activates progressively. The implementation details specify the exact schedule and information boundaries.

6 Experiments

6.1 Experimental setup

Models and benchmarks. We evaluate Qwen3-4B and Qwen3-8B at temperatures 0 and 1 on eight datasets: GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2023), and AIME25 (OpenCompass, 2025) for mathematics; HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), and LiveCodeBench (Jain et al., 2025) for code; and MT-Bench (Zheng et al., 2023) and Alpaca (Taori et al., 2023) for dialogue and instruction following. All eight datasets are evaluated under the same decoding and hardware protocol. We compare DRelay with DFlash, Domino, and DSpark, using target-only autoregressive decoding as the reference; each method uses its native drafting and scheduling procedure.

Table 1: **Single-H800 system comparison on Qwen3-4B and Qwen3-8B.** Each temperature reports speedup S over AR with the *same target and temperature*, and committed tokens per round τ. LCB denotes LiveCodeBench. The reported average is the arithmetic mean over the eight listed datasets.

Method	MATH						CODE						CHAT				OVERALL	
	GSM8K		MATH-500		AIME25		HumanEval		MBPP		LCB		MT-Bench		Alpaca		Avg.	
	S	τ	S	τ	S	τ	S	τ	S	τ	S	τ	S	τ	S	τ	S	τ
Temperature = 0 (greedy)																		
Qwen3-4B																		
DFlash	3.05×	6.40	3.99×	7.89	3.60×	7.32	3.24×	6.61	3.47×	6.01	3.76×	7.40	1.96×	4.11	1.83×	3.48	3.11×	6.15
Domino	3.98×	9.21	4.21×	9.04	3.33×	7.25	3.31×	6.92	3.62×	6.68	3.41×	6.89	2.23×	4.94	2.04×	4.30	3.27×	6.90
DSpark	3.94×	9.24	4.25×	9.17	3.48×	7.47	3.37×	6.94	3.64×	6.65	3.49×	7.07	2.24×	5.01	2.01×	4.27	3.30×	6.98
DRelay	**4.18×**	**9.97**	**4.49×**	**9.62**	**3.85×**	**8.37**	**3.63×**	**7.62**	**3.84×**	**7.09**	**4.07×**	**8.23**	**2.35×**	**5.27**	**2.15×**	**4.49**	**3.57×**	**7.58**
Qwen3-8B																		
DFlash	3.47×	6.29	4.60×	8.03	4.26×	7.29	3.70×	6.46	3.31×	5.88	3.66×	7.20	1.91×	4.09	1.72×	3.61	3.33×	6.11
Domino	4.72×	9.48	4.73×	9.39	4.15×	7.57	3.87×	7.18	3.52×	6.80	3.40×	7.13	2.16×	5.13	2.05×	4.34	3.58×	7.13
DSpark	4.64×	9.39	4.69×	9.31	4.18×	7.65	3.83×	7.24	3.41×	6.69	3.46×	7.23	2.25×	5.26	1.99×	4.29	3.56×	7.13
DRelay	**4.95×**	**10.03**	**5.01×**	**9.96**	**4.72×**	**8.61**	**4.20×**	**7.84**	**3.71×**	**7.18**	**4.01×**	**8.47**	**2.36×**	**5.39**	**2.14×**	**4.51**	**3.89×**	**7.75**
Temperature = 1 (sampling)																		
Qwen3-4B																		
DFlash	2.84×	5.93	3.09×	6.71	2.36×	4.71	2.98×	6.01	2.66×	5.56	2.94×	7.14	1.58×	3.92	1.44×	3.37	2.49×	5.42
Domino	3.36×	8.17	3.34×	7.36	2.17×	4.74	2.96×	6.27	2.75×	6.22	2.63×	6.73	1.72×	4.66	1.64×	4.14	2.57×	6.04
DSpark	3.43×	8.30	3.38×	7.39	2.28×	4.82	2.93×	6.18	2.77×	6.26	2.76×	6.89	1.69×	4.61	1.59×	4.07	2.60×	6.07
DRelay	**3.63×**	**8.86**	**3.54×**	**7.85**	**2.58×**	**5.56**	**3.14×**	**6.71**	**2.90×**	**6.57**	**3.17×**	**8.16**	**1.81×**	**4.78**	**1.68×**	**4.25**	**2.81×**	**6.59**
Qwen3-8B																		
DFlash	3.17×	5.86	3.44×	6.57	2.71×	4.80	3.14×	5.50	2.91×	5.25	3.52×	7.17	1.70×	3.67	1.61×	3.48	2.78×	5.29
Domino	3.88×	7.90	3.24×	7.02	2.47×	4.67	3.10×	5.73	3.02×	5.87	3.08×	6.87	1.86×	4.53	1.83×	4.00	2.81×	5.82
DSpark	3.81×	7.78	3.17×	6.94	2.51×	4.75	3.16×	5.85	2.99×	5.78	3.14×	6.90	1.91×	4.62	1.72×	3.87	2.80×	5.81
DRelay	**4.16×**	**8.51**	**3.49×**	**7.57**	**2.88×**	**5.43**	**3.41×**	**6.34**	**3.23×**	**6.15**	**3.64×**	**8.13**	**1.95×**	**4.74**	**1.94×**	**4.22**	**3.09×**	**6.39**

Training and metrics. We train on the complete 1.42M PerfectBlend dataset (Xu et al., 2024) through SpecForge (Li et al., 2026) using eight H800 GPUs and run inference on one H800. DRelay uses $H = 15$ proposals and $K = 16$ candidates. We report end-to-end speedup S, including prefill and all decoding overhead, and committed tokens per verification round τ, including the target bonus or replacement token. Unless stated otherwise, all comparisons use the same target model, decoding budget, hardware, and evaluation protocol.

6.2 System results

All DRelay and baseline results in this subsection are obtained with the SGLang serving framework (Zheng et al., 2024).

Single-request decoding. Table 1 reports speedup and committed tokens per round over all eight datasets. DRelay has the highest average in every panel, from $2.81\times$ at Qwen3-4B/temperature 1 to $3.89\times$ at Qwen3-8B/temperature 0; the corresponding τ values rise with it. Each panel uses its matching AR reference and an eight-dataset arithmetic mean.

Concurrent serving. Table 2 reports throughput on AIME25 and HumanEval for concurrency 2–32. DRelay is fastest at every setting; at concurrency 32 its speedups over AR are $2.34\times$/$2.27\times$ on Qwen3-4B and $2.63\times$/$2.89\times$ on Qwen3-8B (AIME25/HumanEval). The lead persists as speedup over AR decreases with load. Baseline rows are tokens/s; speculative rows add same-concurrency speedups as subscripts.

Efficiency. Figure 4 decomposes per-round latency into verification, drafting, language-model head computation, and method-specific overhead. DRelay adds 0.55 ms of selector computation to the 6.38 ms shared by verification, drafting, and the language-model head, yielding a total of 6.93 ms. This is 8.6% above DFlash's 6.38 ms and slightly below Domino's 6.99 ms. Verification remains the largest component at 4.99 ms. The end-to-end results in Tables 1 and 2 show that the longer verified prefixes compensate for this additional per-round cost.

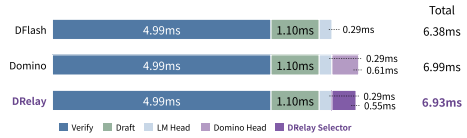


Figure 4: Latency breakdown of DFlash, Domino, and DRelay. Bars decompose latency into verification, drafting, LM-head computation, and method-specific head or selector overhead.

Tables 3 and 4 compare acceptance length τ, using the same definition as Table 1; they isolate architectural and training-objective contributions.

Table 3: Architecture ablation.

Variant	Accepted length
Full	**7.58**
No Global Read	7.16

Table 4: Training ablation.

Variant	Accepted length
Full	**7.58**
No support loss	7.47
No selector loss	7.22
Uniform repair weights	7.19

Table 2: Concurrent serving.

Throughput is in tokens/s; speculative rows include the speedup over AR at the same concurrency as subscripts. Results use greedy decoding on AIME25 and HumanEval.

	Concurrency				
Method	2	4	8	16	32
Qwen3-4B (greedy)					
AIME25					
AR	433	811	1,415	2,490	4,310
DFlash	$1{,}455_{3.36\times}$	$2{,}757_{3.40\times}$	$4{,}780_{3.38\times}$	$6{,}638_{2.67\times}$	$9{,}290_{2.16\times}$
Domino	$1{,}423_{3.29\times}$	$2{,}477_{3.06\times}$	$4{,}299_{3.04\times}$	$6{,}563_{2.64\times}$	$9{,}006_{2.09\times}$
DSpark	$1{,}447_{3.34\times}$	$2{,}524_{3.11\times}$	$4{,}418_{3.12\times}$	$6{,}596_{2.65\times}$	$9{,}104_{2.11\times}$
DRelay	$\mathbf{1{,}614}_{3.73\times}$	$\mathbf{2{,}965}_{3.66\times}$	$\mathbf{4{,}914}_{3.47\times}$	$\mathbf{7{,}181}_{2.88\times}$	$\mathbf{10{,}077}_{2.34\times}$
HumanEval					
AR	442	853	1,571	2,771	3,880
DFlash	$1{,}352_{3.06\times}$	$2{,}425_{2.84\times}$	$3{,}939_{2.51\times}$	$6{,}089_{2.20\times}$	$8{,}157_{2.10\times}$
Domino	$1{,}419_{3.21\times}$	$2{,}673_{3.13\times}$	$4{,}190_{2.66\times}$	$6{,}322_{2.28\times}$	$8{,}483_{2.18\times}$
DSpark	$1{,}443_{3.26\times}$	$2{,}712_{3.18\times}$	$4{,}217_{2.68\times}$	$6{,}390_{2.30\times}$	$8{,}511_{2.19\times}$
DRelay	$\mathbf{1{,}484}_{3.36\times}$	$\mathbf{2{,}790}_{3.27\times}$	$\mathbf{4{,}362}_{2.78\times}$	$\mathbf{6{,}586}_{2.38\times}$	$\mathbf{8{,}805}_{2.27\times}$
Qwen3-8B (greedy)					
AIME25					
AR	282	532	943	1,636	3,120
DFlash	$1{,}162_{4.11\times}$	$2{,}020_{3.80\times}$	$3{,}591_{3.81\times}$	$5{,}082_{3.11\times}$	$7{,}630_{2.45\times}$
Domino	$1{,}092_{3.87\times}$	$1{,}957_{3.68\times}$	$3{,}344_{3.55\times}$	$4{,}885_{2.99\times}$	$7{,}351_{2.36\times}$
DSpark	$1{,}114_{3.95\times}$	$2{,}027_{3.81\times}$	$3{,}502_{3.71\times}$	$4{,}972_{3.04\times}$	$7{,}428_{2.38\times}$
DRelay	$\mathbf{1{,}226}_{4.35\times}$	$\mathbf{2{,}331}_{4.38\times}$	$\mathbf{3{,}807}_{4.04\times}$	$\mathbf{5{,}365}_{3.28\times}$	$\mathbf{8{,}203}_{2.63\times}$
HumanEval					
AR	292	566	1,074	1,955	2,582
DFlash	$1{,}047_{3.59\times}$	$1{,}914_{3.38\times}$	$3{,}300_{3.07\times}$	$5{,}088_{2.60\times}$	$6{,}797_{2.63\times}$
Domino	$1{,}084_{3.72\times}$	$2{,}008_{3.55\times}$	$3{,}429_{3.19\times}$	$5{,}295_{2.71\times}$	$7{,}120_{2.76\times}$
DSpark	$1{,}108_{3.79\times}$	$2{,}087_{3.69\times}$	$3{,}501_{3.26\times}$	$5{,}386_{2.75\times}$	$7{,}231_{2.80\times}$
DRelay	$\mathbf{1{,}153}_{3.95\times}$	$\mathbf{2{,}094}_{3.70\times}$	$\mathbf{3{,}566}_{3.32\times}$	$\mathbf{5{,}497}_{2.81\times}$	$\mathbf{7{,}464}_{2.89\times}$

6.3 Ablation studies

Both tables report mean accepted length τ over the same eight datasets used in Table 1, using the same target/temperature setting as Full and the same hardware and decoding protocol. Each ablation changes only the component named in its row; all other components, positions, data, and training settings remain identical.

Global candidate context. Table 3 compares the full model with No Global Read, which removes global slot mixing and candidate reads. The accepted length falls from 7.58 to 7.16, a reduction of 0.42 tokens (5.5%). This drop shows that cross-position candidate evidence contributes beyond the remaining local and prefix information.

Training objective. Table 4 compares the full objective with three single-component controls. Removing the support loss reduces accepted length to 7.47 (-0.11), removing selector supervision reduces it to 7.22 (-0.36), and replacing prefix-value weights with uniform weights reduces it to 7.19 (-0.39). The larger degradation from removing selector supervision or prefix-value weighting indicates that learning when to repair and assigning more credit to prefix-enabling repairs are both important. The support-loss control retains selector supervision, the selector-loss control retains the draft objective but disables repair, and the uniform-weight control retains both losses; all remaining settings are unchanged.

7 Conclusion

DRelay treats a parallel draft as evidence for its own token choices. Native-draft diagnostics connect three opportunities: early mistakes often retain the correct candidate, reliable later predictions help distinguish that candidate, and a precise repair can recover a consecutive continuation. The architecture follows this logic by reading candidate-level global context once, then making selective repairs against the actual selected prefix. Joint support and repair learning aligns these decisions with prefix reach and continuation value. The system results quantify the end-to-end trade-off, while the latency breakdown identifies the cost of the selector relative to verification and drafting.

AI USE STATEMENT

AI was used only for literature retrieval and discovery (e.g., identifying related work) and for assistance with LaTeX formatting and typesetting. All research ideas, methodological and experimental design, analyses, figures, scientific claims, and the manuscript text were independently developed and written by the authors.

REFERENCES

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

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.

Jian Chen, Yesheng Liang, and Zhijian Liu. DFlash: Block Diffusion for Flash Speculative Decoding. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/64301.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Xin Cheng, Xingkai Yu, Chenze Shao, Jiashi Li, Yunfan Xiong, Yi Qian, Jiaqi Zhu, Shirong Ma, Xiaokang Zhang, Jiasheng Ye, Qinyu Chen, Chengqi Deng, Jiping Yu, Damai Dai, Zhengyan Zhang, Yixuan Wei, Yixuan Tan, Wenkai Yang, Runxin Xu, Yu Wu, Zhean Xu, Xuanyu Wang, Muyang Chen, Rui Tian, Xiao Bi, Zhewen Hao, Shaoyuan Chen, Huanqi Cao, Wentao Zhang, Anyi Xu, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. DSpark: Confidence-Scheduled Speculative Decoding with Semi-Autoregressive Generation. *arXiv preprint arXiv:2607.05147*, 2026. URL https://arxiv.org/abs/2607.05147.

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

Geonmo Gu, Byeongho Heo, HeeJae Jun, Yoohoon Kang, Sangmin Lee, Sangdoo Yun, and Dongyoon Han. Verification-Aware Training for Speculative Decoding. *arXiv preprint arXiv:2608.30135*, 2026. URL https://arxiv.org/abs/2608.30135.

Jianuo Huang, Yaojie Zhang, Qituan Zhang, Hao Lin, Hanlin Xu, and Linfeng Zhang. Domino: Decoupling Causal Modeling from Autoregressive Drafting in Speculative Decoding. *arXiv preprint arXiv:2605.29707*, 2026. URL https://arxiv.org/abs/2605.29707.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/94074dd5a072d28ff75a76dabed43767-Abstract-Conference.html.

Haodi Lei, Yafu Li, Haoran Zhang, Shunkai Zhang, Qianjia Cheng, Xiaoye Qu, Ganqu Cui, Bowen Zhou, Ning Ding, Yun Luo, and Yu Cheng. Draft-OPD: On-Policy Distillation for Speculative Draft Models. *arXiv preprint arXiv:2605.29343*, 2026. URL https://arxiv.org/abs/2605.29343.

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, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Shenggui Li, Chao Wang, Yikai Zhu, Yubo Wang, Fan Yin, Shuai Shi, Yefei Chen, Xiaomin Dong, Qiaoling Chen, Jin Pan, Ji Li, Laixin Xie, Yineng Zhang, Lei Yu, Yonggang Wen, Ivor Tsang, and Tianwei Zhang. SpecForge: A Flexible and Efficient Open-Source Training Framework for Speculative Decoding. *arXiv preprint arXiv:2603.18567*, 2026. URL https://arxiv.org/abs/2603.18567.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28935–28948. PMLR, 2024. URL https://proceedings.mlr.press/v235/li24bt.html.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-3: Scaling up inference acceleration of large language models via training-time test. In *Advances in Neural Information Processing Systems*, volume 38, pp. 136737–136756. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/c7b5a35ea98b62512a869c19ea7b03cb-Paper-Conference.pdf.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. URL https://arxiv.org/abs/2305.20050.

Luxi Lin, Zhanpeng Zeng, Shuang Peng, Songwei Liu, and Rongrong Ji. ReTrace: Rejected-Trajectory Conditioning for Speculative Decoding. *arXiv preprint arXiv:2608.29748*, 2026. URL https://arxiv.org/abs/2608.29748.

OpenCompass. AIME 2025. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/opencompass/AIME2025.

Yu-Yang Qian, Hao-Cong Wu, Chen Chen, Jiacheng Sun, Zhenhua Dong, Peng Zhao, and Zhi-Hua Zhou. AdaFlash: Adaptive Speculative Decoding via On-Policy Distilled Diffusion Drafters. *arXiv preprint arXiv:2607.19223*, 2026. URL https://arxiv.org/abs/2607.19223.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following LLaMA model. GitHub repository, 2023. URL https://github.com/tatsu-lab/stanford_alpaca.

Zhuoyu Wang, Junnan Huang, and Xinyu Chen. TAPS: Target-Aware Prefix Tree Selection for Diffusion-Drafted Speculative Decoding. *arXiv preprint arXiv:2606.00487*, 2026a. URL https://arxiv.org/abs/2606.00487.

Zili Wang, Zhaopeng Qiu, Yuekai Zhang, Shuang Yu, and Junjie Lai. Online Draft Co-Training for Speculative Decoding in Large-Scale, Long-Context RL Post-Training. *arXiv preprint arXiv:2609.07108*, 2026b. URL https://arxiv.org/abs/2609.07108.

Tianyu Wu, Yu Yao, Zhenting Qi, Han Zheng, Zhuohan Wang, Haoran Ma, Lawrence Liao, Himabindu Lakkaraju, Ju Li, and Yilun Du. D-PACE: Dynamic Position-Aware Cross-Entropy for Parallel Speculative Drafting. *arXiv preprint arXiv:2605.18810*, 2026. URL https://arxiv.org/abs/2605.18810.

Tengyu Xu, Eryk Helenowski, Karthik Abinav Sankararaman, Di Jin, Kaiyan Peng, Eric Han, Shaoliang Nie, Chen Zhu, Hejia Zhang, Wenxuan Zhou, Zhouhao Zeng, Yun He, Karishma Mandyam, Arya Talabzadeh, Madian Khabsa, Gabriel Cohen, Yuandong Tian, Hao Ma, Sinong Wang, and Han Fang. The perfect blend: Redefining rlhf with mixture of judges. *arXiv preprint arXiv:2409.20370*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Jiebin Zhang, Zhenghan Yu, Song Liu, Eugene J. Yu, Zheng Li, Dawei Zhu, Jiangshan Duo, Weimin Xiong, Yifan Song, Guanghua Yu, Jianchen Zhu, and Sujian Li. DFlare: Scaling Up Draft Capacity for Block Diffusion Speculative Decoding. *arXiv preprint arXiv:2606.02091*, 2026a. URL https://arxiv.org/abs/2606.02091.

Yaojie Zhang, Linfeng Zhang, Bin Cui, and Xupeng Miao. DFlow: Enabling Verifier Information Flow in Block Diffusion Speculative Decoding. *arXiv preprint arXiv:2609.06498*, 2026b. URL https://arxiv.org/abs/2609.06498.

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. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 62557–62583, 2024.

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*, volume 2024, pp. 32011–32050, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8766fbc68e1ed1cdef712ce273e0a363-Paper-Conference.pdf.

A Reference Diagnostics

This appendix defines the offline diagnostics used to motivate DRelay in Section 3. They are computed from frozen decoding traces and are separate from the trained-system evaluation in Section 6. In particular, the diagnostics measure whether useful evidence is present in a parallel draft; they do not use oracle labels as inputs to DRelay at inference time.

Reference traces and terminology. We collect native parallel-drafting traces from DFlash and Domino (Chen et al., 2026; Huang et al., 2026) with Qwen3-4B as the target model (Yang et al., 2025). At each position, the *base* token is the backbone argmax and the *served* token is the token selected by the corresponding native drafting procedure. The target's greedy continuation under the verified prefix supplies the reference token used to identify a mismatch. The original top-16 candidate pool is retained for the candidate-availability analyses. All traces use the same tokenizer and verification rule as the corresponding main-text diagnostic.

Candidate coverage. For a position i, let y_i be the target reference token and let $\mathcal{V}_i$ be the original candidate pool. We record $c_i = \mathbb{1}[y_i \in \mathcal{V}_i]$. Coverage is evaluated at the first base error and at the first error in the native served path, with the latter restricted to positions whose served pool is defined. These strata localize where a repair is possible; they do not imply that a deployed selector can observe the target token or the error location.

Continuation evidence. For a candidate $v \in \mathcal{V}_i$, we score a later predicted continuation under the verified prefix extended by v. The score uses the target model's conditional likelihood of the predicted tokens after position i and omits the current-token target probability. Replacing the continuation by a correctness-matched continuation from another prompt provides the control comparison in Figure 2. The control changes the future content while preserving the task and reliability strata, so a positive real-minus-control contrast indicates context-aligned evidence rather than candidate prior alone.

One-edit repair value. Let A be the number of proposals accepted before the first served error and let R be the number of immediately following proposals that agree with the target. If the target token is covered, changing only that error increases the accepted prefix by $1 + R$; later matches separated by another error do not count. We report this quantity as an oracle opportunity analysis over complete blocks. It is not an estimate of DRelay's realized gain, because DRelay must infer both whether to edit and which candidate to select.

Interpretation and scope. The diagnostics establish the existence of candidate availability and continuation evidence that a compact selector may exploit. They do not provide supervision from unverified target states, do not evaluate a learned repair policy, and do not replace the end-to-end speed and acceptance results reported in Section 6.

B Implementation and Training Details

This appendix records the implementation contract shared by the reported experiments. The equations in Sections 4 and 5 give the full computation; the points below make explicit the information boundaries and supervision used in the reported configuration.

Information boundary. At each round the parallel drafter produces hidden representations and a candidate distribution for each of $H = 15$ positions. DRelay retains the native top-K pool with $K = 16$. Global Read consumes only these draft-side representations and candidate distributions. It never reads a teacher proposal token or a future token that has not been verified by the target model. The Causal Selector receives the resulting candidate-specific global features, the strictly verified history, and a prefix memory whose entry at position j is visible only to decisions at positions $i > j$. Once a candidate is selected, it is the only token written into the prefix used by later decisions in the same block.

Trainable and frozen components. The target model, token embedding, and vocabulary readout remain frozen. The parallel drafting backbone, Global Read, and Causal Selector are trained jointly.

The selector has separate KEEP/REPAIR and ranking branches. Its gate starts from the native pool-mass log-odds, with a learned residual correction:

$$g_i = \log \sum_{k>0} q_i(v_{ik}) - \log q_i(v_{i0}) + \Delta g_i. \tag{6}$$

Here v_{i0} is the native candidate and q_i is the draft distribution over the retained pool. Residual corrections are learned from the draft-side features and verified history. This preserves the native choice when contextual evidence is insufficient and allows a repair to be carried forward causally.

Supervision. For a target token y_i, the dense support term trains the draft distribution to place mass on y_i, including when the token is absent from the retained pool. The selector term labels a covered base-correct token as KEEP and a covered alternative as REPAIR with its target-pool index. A pool miss has no valid selector action and therefore contributes only to support supervision. The selector term is multiplied by the prefix-value weights defined in Equation 5; these weights favor decisions that are reachable from the current prefix and that preserve a correct continuation. Padding, anchors, and non-contiguous response positions are masked from both terms. No future target token is supplied to Global Read or to the selector as an input.

Training and serving protocol. We train on the complete PerfectBlend corpus through SpecForge, using the training and hardware protocol described in Section 6. All comparisons use the same target model, candidate budget, decoding budget, and evaluation hardware. During serving, Global Read is computed once per parallel draft and is followed by the H causal selector decisions; the target then verifies the resulting single path and supplies the next anchor. Thus the reported latency includes both the shared draft/verification work and the method-specific selection overhead shown in Figure 4.

C Diagnostic Scope and Interpretation

The motivation analysis asks whether later parallel predictions contain information that can distinguish candidates at an earlier position. This appendix clarifies the diagnostic boundary used for that question.

What is scored. For a candidate v at position i, the offline assessor conditions on the verified prefix followed by v and scores only the later predicted draft tokens. The current target-token probability is excluded, so the score measures continuation compatibility rather than simply reproducing the native candidate prior. Real continuations are compared with correctness-matched continuations from other prompts. The target model is used as an offline assessor for this measurement; it is not queried for unverified future states by DRelay's Global Read.

What the diagnostic does not claim. First-error localization and correctness filters are used only to form interpretable offline strata. They are not available to the deployed selector, and they do not define an inference-time repair rule. Likewise, an oracle one-edit gain measures the value of a possible correction under the original candidate pool; it is not an end-to-end speedup or an accepted-length prediction for the trained system.

Connection to DRelay. The diagnostic supports the design choice of reading global candidate context while retaining a causal prefix selector. Section 4 implements that boundary using draft-side hidden states, candidate distributions, and verified history, and Section 6 measures the resulting trained system on all eight benchmarks. The appendix therefore serves as a definition of the evidence used in the motivation, rather than as a second set of model or benchmark results.