DSPARK-OPD: ACCEPTANCE-ALIGNED ON-POLICY DISTILLATION FOR SEMI-AUTOREGRESSIVE SPECULATIVE DRAFTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Standard rejection-sampling speculative decoding preserves the target distribution for any proposal model, but its speed depends on the length of the draft prefix that survives verification. Semi-autoregressive drafters face two training mismatches: offline training observes target-generated prefixes rather than draft-induced states, and tokenwise objectives do not reflect that an early rejection invalidates the remaining suffix. We present DSpark-OPD, an on-policy post-training method for the Markov-corrected DSpark architecture. During rollout, the frozen target verifies samples from the deployed Markov-corrected proposal. During replay, target and drafter distributions are evaluated on the same sampled predecessor chain, and optimization is applied to the corrected proposal used by the verifier. We introduce Survival Loss, an overlap-based pathwise objective that maximizes a differentiable lower bound on expected accepted length while assigning earlier transitions credit for later reachable positions. Across Qwen3-8B and Qwen3-14B, adding Survival Loss to the matched DSpark-OPD-Base objective improves every evaluated dataset–temperature comparison. Under greedy decoding, public-average engine acceptance length rises by 4.7% at 8B and 5.4% at 14B relative to DSpark-OPD-Base; relative to the released DSpark checkpoints, the complete pipeline gains are 12.8% and 6.0%. At concurrency 8, public-benchmark throughput improves by 4.9% and 6.1% over DSpark-OPD-Base, reaching 2.07× and 2.40× target-only throughput. Controlled Qwen3-8B ablations show that Survival Loss remains competitive with local and fixed-decay variants at block length 7 while providing state-adaptive prefix credit without a hand-designed decay schedule.

1 INTRODUCTION

Autoregressive decoding remains a central bottleneck in LLM serving because each generated token requires a target-model step, resulting in sequential latency and underutilized parallel hardware. Speculative decoding addresses this bottleneck with a lightweight proposal model and a parallel target verification pass (Leviathan et al., 2023; Chen et al., 2023). The verifier accepts the longest valid prefix and corrects the first mismatch, preserving the target distribution exactly. Its practical speedup depends on the proposal cost and on the length of the draft prefix that survives verification.

Parallel and semi-autoregressive drafters shift this trade-off by reducing proposal latency while preserving draft quality. DFlash generates an entire block in a single diffusion-style forward pass (Chen et al., 2026). DSpark further augments this parallel backbone with a lightweight Markov correction head and confidence-scheduled verification (Cheng et al., 2026). With proposal latency largely reduced, a complementary opportunity lies in aligning post-training with the states that the deployed drafter actually exposes to verification. At deployment, the drafter's own samples determine the Markov transitions, confidence inputs, and verifier-exposed states.

Draft-OPD establishes target-assisted rollout and acceptance-aware replay as an effective route to draft-induced supervision (Lei et al., 2026). It applies forward KL to accepted states and reverse KL to rejected states. Markov DSpark, however, introduces an additional alignment requirement. The verifier evaluates the corrected proposal conditioned on sampled predecessor tokens, not teacher-forced prefixes or intermediate backbone logits. DSpark-OPD therefore replays the sampled pre-

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.

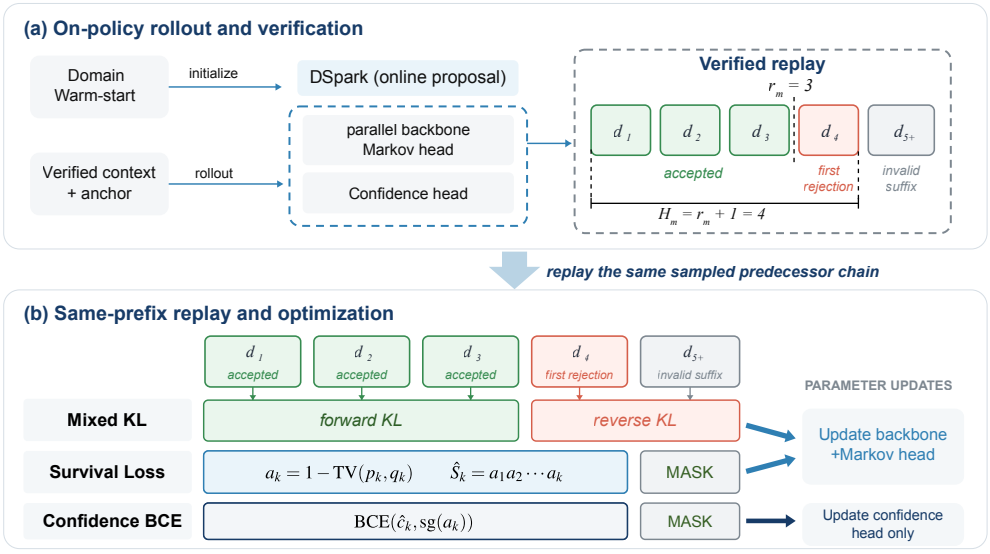

Figure 1: Training loop of DSpark-OPD. The frozen target verifies a Markov-corrected proposal and records its acceptance boundary. Replay follows the sampled predecessor chain. Mixed KL and Survival Loss update the final proposal, while confidence supervision updates the confidence head used by the scheduler.

decessor chain and optimizes the final corrected proposal used by the verifier, as summarized in Figure 1.

This state alignment determines where the drafter is trained. A second challenge is how credit is assigned across the verified block. Mixed KL remains additive over positions, whereas speculative verification is a stopping-time process. Once one token is rejected, the verifier discards the proposed suffix. Improving an early bottleneck can therefore preserve several later positions. Fixed position weights can emphasize early tokens, but they do not adapt credit to current target–draft agreement.

DSpark-OPD addresses both challenges. It performs target-assisted rollouts with the deployed Markov-corrected proposal, records the sampled predecessor chain, and replays the final corrected proposal on identical draft-induced prefixes. We then augment acceptance-aware mixed KL with Survival Loss. At each reached state, target–draft overlap equals the conditional one-step acceptance probability under the standard verifier. Summing these reached overlaps gives an exact on-policy representation of expected accepted length. Survival Loss replaces each local term with its prefix product, yielding a differentiable lower bound with explicit credit from later reachable positions to earlier transitions.

Our contributions are as follows.

1. We formulate deployed-policy replay for Markov DSpark, evaluating the frozen target and the Markov-corrected proposal on the same sampled predecessor chain.

2. We derive an exact reached-overlap representation of expected accepted length and introduce Survival Loss, a differentiable lower-bound objective that provides prefix-structured credit without a fixed positional schedule.

3. We provide comprehensive evaluations on Qwen3-8B and Qwen3-14B. Relative to the matched base online objective, adding Survival Loss improves every evaluated dataset–temperature comparison and increases end-to-end serving throughput at both scales. Controlled objective ablations show that the gain is robust across local-overlap, fixed-decay, and survival formulations, while Survival Loss retains adaptive prefix credit without an additional decay hyperparameter.

2 Related Work

Speculative decoding. Blockwise parallel decoding established the idea of predicting multiple future tokens before verification (Stern et al., 2018). Modern speculative decoding introduced rejection-sampling verification that exactly preserves the target distribution (Leviathan et al., 2023; Chen et al., 2023). Subsequent work explores token trees, multiple heads, early exits, and feature-level drafting (Miao et al., 2024; Cai et al., 2024; Elhoushi et al., 2024; Li et al., 2024a;b; 2026). These methods primarily modify candidate generation or scheduling. DSpark-OPD is orthogonal to these directions. It post-trains an existing semi-autoregressive drafter under its deployed proposal distribution and verifier-exposed states.

Parallel and semi-autoregressive drafters. DFlash replaces sequential drafting with a target-conditioned block diffusion model and shows that low draft latency can be as important as acceptance length (Chen et al., 2026). DFlare increases block-drafter capacity through richer target-feature fusion (Zhang et al., 2026). DSpark augments a DFlash-style parallel backbone with an inexpensive token-conditioned correction head and confidence-scheduled verification (Cheng et al., 2026). Recent systems extend this design space: Domino combines a parallel draft backbone with lightweight prefix-conditioned causal refinement, while AngelSpec introduces DFly with predecessor-conditioned refinement and load-aware verification (Huang et al., 2026; Liu et al., 2026). Our method trains the corrected proposal and confidence head on the state distribution induced by the deployed drafter.

Distillation and on-policy training. DistillSpec studies knowledge-distillation objectives for speculative draft models (Zhou et al., 2024). Generalized on-policy distillation trains on student-generated behavior (Agarwal et al., 2024). Draft-OPD applies this principle to speculative drafters (Lei et al., 2026). It combines target-assisted rollout with forward KL on accepted states and reverse KL on rejected states. AdaFlash applies reverse-KL on-policy distillation to diffusion drafters and jointly learns an adaptive candidate-length head (Qian et al., 2026). We retain this local correction mechanism, instantiate it for the deployed Markov-corrected DSpark proposal, and introduce Survival Loss, a pathwise objective constructed from target–draft overlap and motivated by prefix survival.

Confidence-based scheduling. EAGLE-2 and DSpark use confidence to adapt candidate topology or verified length (Li et al., 2024b; Cheng et al., 2026). To align confidence scheduling with verifier-exposed states, we update the DSpark confidence head using rollout-state statistics. Confidence guides scheduling, while Survival Loss changes the proposal distribution.

3 Background and Problem Setup

3.1 Lossless Speculative Sampling

Let p_θ denote the frozen target and q_ϕ the drafter. Given a verified context c, the drafter proposes a block $d_{1:K}$, where

$$d_k \sim q_\phi(\cdot \mid c, d_{<k}), \qquad k = 1, \dots, K. \tag{1}$$

At each position, a sampled token d_k is accepted with probability $\min(1, p_k(d_k)/q_k(d_k))$. At the first rejection, the verifier samples a correction from the normalized residual distribution $[p_k - q_k]_+$. The number of accepted draft tokens is $r \in \{0, \dots, K\}$. The engine metric used in our experiments includes the target bonus token, so we report

$$\tau_+ = \mathbb{E}[r + 1]. \tag{2}$$

3.2 Markov DSpark

The parallel DSpark backbone produces base logits ℓ_k^{base} for all block positions in one forward pass. Its Markov head adds a transition-dependent correction based on the previously sampled draft token.

$$q_k^{\text{corr}}(\cdot \mid d_{<k}) = \text{softmax}\big(\ell_k^{\text{base}} + b_k(d_{k-1})\big)\,. \tag{3}$$

Proposal tokens are sampled from q^{corr}, not the intermediate base distribution, and the verifier uses the same distribution in its acceptance ratios. The intermediate backbone distribution is therefore not the deployed proposal. DSpark also predicts confidence logits $\hat{z}_k$ and scores $\hat{c}_k = \sigma(\hat{z}_k)$. A scheduler uses their cumulative product to choose a verification length.

3.3 Acceptance-Aware On-Policy Distillation

Draft-OPD runs target-assisted decoding with the current drafter (Lei et al., 2026). It stores the context c_m, proposal $d_{m,1:K}$, and accepted length r_m. Target and draft distributions are then replayed on the recorded prefixes. Accepted positions receive forward KL. The first rejected position and sampled post-rejection suffix states receive reverse-KL regularization. This provides local state-aligned supervision. Our Survival Loss additionally organizes target–draft overlap along verification-reachable prefixes.

4 Method

An offline warm-start initializes a deployable DSpark drafter. Online training then replays the deployed Markov policy and optimizes dense acceptance overlap on reachable prefixes.

4.1 On-Policy Markov Adaptation

We first warm-start the complete drafter with the original DSpark objective. The Markov and confidence heads remain active. Online training therefore begins from a usable semi-autoregressive policy.

For each online round m, speculative decoding begins at verified context c_m and samples $d_{m,1:K} \sim q_\phi(\cdot \mid c_m)$. Confidence-based truncation is disabled during online rollout collection, so all K draft positions are verified. We store the block, accepted length r_m, and predecessor chain

$$\pi_m = (y_{a_m}, d_{m,1}, \ldots, d_{m,K-1}), \tag{4}$$

where y_{a_m} is the last verified token at the anchor. The target remains frozen.

For each stored anchor, replay evaluates

$$p_{m,k} = p_\theta(\cdot \mid c_m, d_{m,<k}), \tag{5}$$
$$q_{m,k} = q_\phi^{\text{corr}}(\cdot \mid c_m, d_{m,<k}). \tag{6}$$

Both distributions condition on the same draft-sampled prefix. Using a teacher-forced predecessor would change the Markov bias in Eq. 3, and base logits would define a proposal that the verifier never samples. We therefore replay the exact predecessor chain and optimize the corrected proposal q^{corr}.

Let $\mathcal{I}_{\text{acc}} = \{(m,k) : k \le r_m\}$ and $\mathcal{I}_{\text{rej}} = \{(m,k) : k > r_m\}$. Following Draft-OPD, we optimize

$$\mathcal{L}_{\text{acc}} = \frac{1}{|\mathcal{I}_{\text{acc}}|} \sum_{(m,k)\in\mathcal{I}_{\text{acc}}} D_{\text{KL}}(p_{m,k} \| q_{m,k}), \tag{7}$$

$$\mathcal{L}_{\text{rej}} = \frac{1}{Z} \sum_{(m,k)\in\mathcal{I}_{\text{rej}}} u_k D_{\text{KL}}(q_{m,k} \| p_{m,k}), \quad Z = \sum_{(m,k)\in\mathcal{I}_{\text{rej}}} u_k, \tag{8}$$

$$\mathcal{L}_{\text{OPD}}^{\text{corr}} = \mathcal{L}_{\text{acc}} + \beta_{\text{rej}} \mathcal{L}_{\text{rej}}, \tag{9}$$

The rejected stream uses $u_k = \gamma^{k-1}$ with $\gamma = 0.9$ and $\beta_{\text{rej}} = 2$. The accepted response stream has unit forward-KL weight and no additional reverse-KL term.

4.2 ACCEPTANCE-ALIGNED SURVIVAL OBJECTIVE

At each reachable replay state, define the target–draft overlap. For position k in rollout m, this overlap is

$$a_{m,k} = \sum_v q_{m,k}(v) \min\left(1, \frac{p_{m,k}(v)}{q_{m,k}(v)}\right) \tag{10}$$

$$= \sum_v \min(p_{m,k}(v), q_{m,k}(v)) = 1 - \mathrm{TV}(p_{m,k}, q_{m,k}). \tag{11}$$

At a fixed replay state, $a_{m,k}$ is exactly the expected one-step acceptance probability under standard speculative verification. Because it uses the complete target and draft distributions rather than one sampled accept/reject outcome, it provides a continuous local training signal at every reached position.

Speculative verification is prefix-based, so local overlap must be combined with reachability. Let $B_{m,k} \in \{0,1\}$ denote acceptance at position k, and let $R_{m,k} = \mathbf{1}\{r_m \geq k-1\} = \mathbf{1}\{k \leq H_m\}$, where $H_m = \min(r_m + 1, K)$. Conditioned on the reached replay state $s_{m,k} = (c_m, d_{m,<k})$, $\mathbb{E}[B_{m,k} \mid s_{m,k}, R_{m,k} = 1] = a_{m,k}$. Since $R_{m,k}$ is determined before the acceptance decision at position k, the tower property and tail-sum identity give

$$\mathbb{E}[r_m] = \sum_{k=1}^{K} \Pr(r_m \geq k) = \sum_{k=1}^{K} \mathbb{E}[R_{m,k} B_{m,k}]$$

$$= \mathbb{E}\left[\sum_{k=1}^{K} R_{m,k} a_{m,k}\right]. \tag{12}$$

Thus reached local overlap gives an exact on-policy value representation of expected accepted length.

During replay, sampled states and reachability indicators are fixed, so the local objective provides only positionwise gradients. We introduce the pathwise survival score

$$\widehat{S}_{m,k} = \prod_{j=1}^{k} a_{m,j}, \tag{13}$$

$$\mathcal{B}_{\mathrm{surv}} := \mathbb{E}\left[\sum_{k=1}^{K} R_{m,k} \widehat{S}_{m,k}\right] \leq \mathbb{E}\left[\sum_{k=1}^{K} R_{m,k} a_{m,k}\right] = \mathbb{E}[r_m]. \tag{14}$$

The inequality follows from $a_{m,j} \in [0,1]$, which implies $\widehat{S}_{m,k} \leq a_{m,k}$. Hence the survival score is an on-policy lower bound on expected accepted length, becoming tighter as earlier overlaps approach one.

The corresponding minibatch loss is

$$\mathcal{L}_{\mathrm{surv}} = 1 - \frac{1}{MK} \sum_{m=1}^{M} \sum_{k=1}^{H_m} \widehat{S}_{m,k}. \tag{15}$$

Accordingly, $\mathbb{E}[\mathcal{L}_{\mathrm{surv}}] \geq 1 - \mathbb{E}[r]/K$: minimizing the loss maximizes a differentiable lower bound on normalized expected accepted length.

The product also provides explicit prefix credit. For $J_m = \sum_{k=1}^{H_m} \widehat{S}_{m,k}$,

$$\frac{\partial J_m}{\partial a_{m,j}} = \sum_{k=j}^{H_m} \prod_{\substack{\ell \leq k \\ \ell \neq j}} a_{m,\ell}. \tag{16}$$

Thus an earlier transition receives credit from every retained suffix term that it gates, with weights determined by current target–draft agreement rather than a fixed positional schedule.

For the controlled objective ablations, we use the common form

$$\mathcal{L}_{\text{aux}}(g) = 1 - \frac{1}{MK} \sum_{m=1}^{M} \sum_{k=1}^{H_m} g_{m,k}, \tag{17}$$

where different choices of $g_{m,k}$ change only how the same dense overlap signal is organized within a block. Survival Loss corresponds to $g_{m,k} = \widehat{S}_{m,k}$. The complete ablation definitions are given in Appendix C.

4.3 Joint Online Objective and Lossless Verification

Although confidence-based truncation is disabled during online rollout collection, the DSpark confidence head is trained on the same reachable states for inference-time scheduling. We use the stop-gradient overlap as its supervision target,

$$\mathcal{L}_{\text{conf}} = \frac{1}{\sum_m H_m} \sum_m \sum_{k=1}^{H_m} \text{BCE}(\hat{z}_{m,k}, \text{sg}(a_{m,k}))\,. \tag{18}$$

Gradients from $\mathcal{L}_{\text{conf}}$ update only the confidence head and do not propagate to the backbone or Markov proposal.

The complete online-training objective is

$$\mathcal{L}_{\text{DSpark-OPD}} = \mathcal{L}_{\text{OPD}}^{\text{corr}} + \lambda_s \mathcal{L}_{\text{surv}} + \lambda_c \mathcal{L}_{\text{conf}}. \tag{19}$$

The first term aligns the deployed Markov proposal with the target on draft-induced states. The second applies prefix-structured credit to local target–draft overlap. The third trains the confidence scores used for inference-time scheduling.

5 Experiments

We first measure the complete on-policy pipeline at two model scales. We then isolate the objective, examine position-wise survival, and measure end-to-end serving efficiency.

5.1 Experimental Setup

Models and tasks. We conduct experiments on Qwen3-8B and Qwen3-14B with thinking mode disabled (Yang et al., 2025). For each scale, we compare the released block-7 DSpark checkpoint, a Warm-start checkpoint, DSpark-OPD-Base, and the final DSpark-OPD model. We evaluate seven public benchmarks spanning mathematical reasoning (GSM8K, MATH-500, and AIME 2024/2025) (Cobbe et al., 2021; Hendrycks et al., 2021; Mathematical Association of America, 2024; 2025), code generation (MBPP) (Austin et al., 2021), dialogue (MT-Bench) (Zheng et al., 2023), and software engineering (SWE-bench Lite, abbreviated as SWE-Lite in tables) (Jimenez et al., 2024). We additionally report results on a held-out private Business test set.

Training data. For the SFT stage, we fine-tune the released DSpark checkpoint on a private Business training set using the original DSpark objective, producing the Warm-start checkpoint. For the OPD stage, we use a disjoint Business training partition for on-policy rollout and replay.

Metrics. We report engine acceptance length τ_+, which includes one target bonus token. Position-wise prefix survival measures how often verification reaches each draft position. Serving metrics are output throughput and time per output token (TPOT). Public results are unweighted averages over the seven benchmark subsets. The Business test set is reported separately.

Loss-structure ablations. The controlled 8B suite compares four matched online-training objectives. L0 sets the auxiliary overlap loss to zero. L1 uses the local overlap $g_{m,k} = a_{m,k}$. L2 uses a normalized fixed decay $g_{m,k} = c_\rho \rho^{k-1} a_{m,k}$ with $\rho = 0.8$. L3 uses the dynamic survival score $g_{m,k} = \prod_{j \le k} a_{m,j}$. The constant $c_\rho = K / \sum_{k=1}^{K} \rho^{k-1}$ makes the maximum full-block score equal to K. We also sweep the L3 coefficient $\lambda_s \in \{0, 0.5, 1, 4, 16\}$.

5.2 STAGEWISE ACCEPTANCE

Table 1: Engine acceptance length τ_+ at concurrency 8 across three decoding temperatures. OPD-Base denotes the base online objective, and OPD+Surv. denotes the complete DSpark-OPD pipeline. Public Avg. excludes the Business test set. OPD+Surv. is highlighted with dark-blue bold text and a light blue-gray background.

	Qwen3-8B			Qwen3-14B		
Dataset	**DSpark**	**OPD-Base**	**OPD+Surv.**	**DSpark**	**OPD-Base**	**OPD+Surv.**
$T = 0$						
Business Test	2.459	3.993	**4.383**	2.676	4.270	**4.510**
GSM8K	5.658	6.133	**6.370**	5.977	5.933	**6.230**
MATH-500	5.477	5.974	**6.158**	5.691	5.757	**6.006**
AIME 2024	4.650	5.113	**5.322**	4.791	4.954	**5.228**
AIME 2025	4.647	5.073	**5.339**	4.775	5.018	**5.277**
MBPP	4.545	4.859	**5.030**	4.795	4.653	**4.921**
MT-Bench	3.771	3.874	**4.165**	3.833	3.811	**4.041**
SWE-Lite	3.138	3.337	**3.579**	3.180	3.112	**3.335**
Public Avg.	4.555	4.909	**5.138**	4.720	4.748	**5.005**
$T = 0.6$						
Business Test	2.423	3.942	**4.326**	2.668	4.233	**4.399**
GSM8K	5.519	5.980	**6.145**	5.750	5.790	**6.028**
MATH-500	5.282	5.719	**5.872**	5.460	5.573	**5.725**
AIME 2024	4.282	4.796	**4.865**	4.475	4.448	**4.734**
AIME 2025	4.336	4.639	**4.848**	4.494	4.552	**4.850**
MBPP	4.375	4.653	**4.814**	4.574	4.422	**4.660**
MT-Bench	3.631	3.817	**4.047**	3.732	3.717	**3.888**
SWE-Lite	2.971	3.153	**3.357**	2.982	2.900	**3.096**
Public Avg.	4.343	4.680	**4.850**	4.495	4.486	**4.712**
$T = 1$						
Business Test	2.375	3.767	**4.143**	2.589	4.143	**4.270**
GSM8K	5.273	5.696	**5.861**	5.553	5.515	**5.799**
MATH-500	4.927	5.271	**5.388**	5.067	5.186	**5.330**
AIME 2024	3.906	4.158	**4.280**	3.901	3.986	**4.179**
AIME 2025	3.806	4.093	**4.349**	3.930	3.972	**4.081**
MBPP	4.070	4.367	**4.536**	4.298	4.158	**4.346**
MT-Bench	3.460	3.644	**3.819**	3.520	3.543	**3.699**
SWE-Lite	2.696	2.818	**2.942**	2.667	2.628	**2.754**
Public Avg.	4.020	4.293	**4.454**	4.134	4.141	**4.313**

At temperature 0, DSpark-OPD raises public-average τ_+ from 4.555 to 5.138 (+12.8%) at 8B and from 4.720 to 5.005 (+6.0%) at 14B (Table 1). Relative to OPD-Base, Survival Loss raises the public averages from 4.909 to 5.138 at 8B and from 4.748 to 5.005 at 14B, corresponding to gains of 4.7% and 5.4%. The improvement is positive for every evaluated dataset–temperature pair at both scales.

5.3 CONTROLLED OBJECTIVE ABLATIONS

Table 2(a) compares three credit rules built from the same local target–draft overlap quantity. At coefficient 1, L1–L3 improve public-average acceptance length over L0 by 4.5–6.0% across temperatures and improve Business acceptance by 9.0–13.4%. The three variants form a tight performance group, showing that the improvement is robust to the exact within-block weighting rule. Survival Loss remains an attractive formulation because it derives prefix credit from current target–draft agreement and requires no hand-designed positional-decay hyperparameter. Table 2(b) shows a broad useful coefficient range for Survival Loss. Performance improves as λ_s increases through 4 and changes more modestly between the evaluated settings 4 and 16. The complete 8B and 14B pipelines use $\lambda_s = 1$ without scale-specific tuning.

Table 2: Controlled Qwen3-8B ablations at concurrency 8. Entries are acceptance length τ_+. Panel (a) compares objective structures at coefficient 1; panel (b) sweeps λ_s. Public Avg. excludes Business Test. Shading marks the default L3 setting; bold marks each column's best result.

		Public Avg.			Business Test		
Variant	**Position rule**	T=0	T=0.6	T=1	T=0	T=0.6	T=1
(a) Objective structure, coefficient 1							
L0: base	none	4.867	4.616	4.239	3.920	3.836	3.682
L1: local overlap	uniform	5.158	**4.856**	4.450	4.393	4.289	4.139
L2: decayed overlap	fixed ρ=0.8	**5.160**	4.845	**4.470**	**4.412**	**4.315**	**4.177**
L3: Survival	adaptive	5.137	4.836	4.431	4.292	4.218	4.013
(b) Survival coefficient							
λ_s=0	none	4.867	4.616	4.239	3.920	3.836	3.682
λ_s=0.5	adaptive	5.057	4.810	4.388	4.194	4.095	3.909
λ_s=1	adaptive (default)	5.137	4.836	4.431	4.292	4.218	4.013
λ_s=4	adaptive	5.209	**4.915**	4.466	4.460	4.326	**4.200**
λ_s=16	adaptive	**5.240**	4.905	**4.519**	**4.484**	**4.364**	4.178

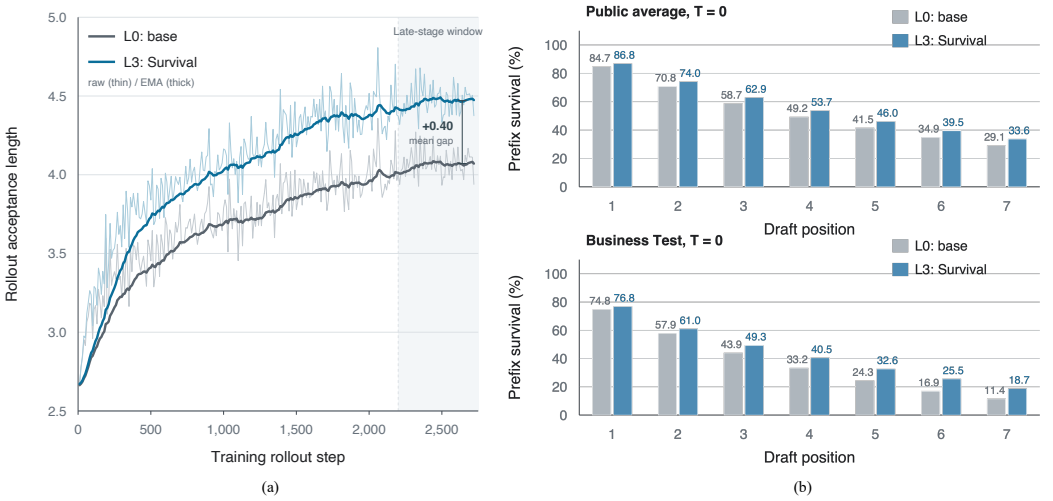

Figure 2: Training dynamics and position-wise effects in the controlled Qwen3-8B comparison. (a) raw (thin) and EMA-smoothed (thick) rollout acceptance length logged during online training for L0 and L3. (b) prefix survival at each draft position under greedy decoding and concurrency 8, averaged over the seven public subsets (top) and measured on the Business test set (bottom).

5.4 Position-Wise Effects

Figure 2 summarizes the training dynamics and the within-block prefix-survival profile. In the shaded late-stage window, the smoothed L3 curve remains consistently above L0, with an annotated mean acceptance-length gap of 0.40 tokens. The position-wise results show higher prefix survival for L3 at every draft position. On the public benchmarks, survival is 86.8% versus 84.7% at position 1 and 33.6% versus 29.1% at position 7. On the Business test set, the corresponding values are 76.8% versus 74.8% at position 1 and 18.7% versus 11.4% at position 7. The more pronounced separation at later positions is consistent with the intended prefix-credit mechanism, as improved early transitions allow longer draft prefixes to remain reachable during verification.

5.5 Serving Efficiency

Table 3 reports end-to-end throughput across concurrency levels from 1 to 32. DSpark-OPD achieves the highest throughput in every evaluated setting. On the public benchmarks, it improves over the released DSpark checkpoint by 8.7–12.7% at 8B and 1.7–5.0% at 14B. At concurrency 8, DSpark-

Table 3: Output throughput (tokens/s) at temperature 0. OPD-Base is the base online objective; OPD+Surv. is the complete DSpark-OPD pipeline. Speedup is relative to target-only decoding, and gain is relative to DSpark; both are means of per-dataset ratios. Public Avg. covers seven public subsets and excludes the Business test set. OPD+Surv. and its derived gains are shown in bold dark blue with a light blue-gray background.

Target	Conc.	Target-only	DSpark	OPD-Base	OPD+Surv.	Speedup	Gain vs. DSpark
Public Avg. (7)							
Qwen3-8B	1	151	319	343	**360**	**2.38×**	**+12.7%**
	8	1060	1986	2100	**2203**	**2.07×**	**+10.8%**
	16	1963	3273	3447	**3562**	**1.81×**	**+8.8%**
	32	3461	4972	5222	**5407**	**1.55×**	**+8.7%**
Qwen3-14B	1	90	236	234	**247**	**2.76×**	**+4.6%**
	8	650	1486	1473	**1563**	**2.40×**	**+5.0%**
	16	1216	2482	2464	**2584**	**2.12×**	**+4.0%**
	32	2167	3868	3722	**3943**	**1.81×**	**+1.7%**
Business test							
Qwen3-8B	1	145	131	248	**276**	**1.91×**	**+111.6%**
	8	909	727	1456	**1607**	**1.77×**	**+121.2%**
	16	1421	1108	2256	**2430**	**1.71×**	**+119.3%**
	32	1811	1238	2293	**2671**	**1.48×**	**+115.8%**
Qwen3-14B	1	86	95	181	**196**	**2.27×**	**+105.6%**
	8	525	500	908	**973**	**1.85×**	**+94.4%**
	16	805	708	1260	**1369**	**1.70×**	**+93.4%**
	32	1048	832	1440	**1512**	**1.44×**	**+81.8%**

OPD reaches 2,203 and 1,563 tokens/s, corresponding to 2.07× and 2.40× speedups over target-only decoding. These results are 10.8% and 5.0% higher than DSpark, and 4.9% and 6.1% higher than OPD-Base. The gains are larger on the held-out Business test set: at concurrency 8, DSpark-OPD reaches 1,607 tokens/s at 8B and 973 tokens/s at 14B, improving over DSpark by 121.2% and 94.4%, respectively. The improvement remains positive at higher concurrency, showing that the acceptance gains continue to translate into serving benefits as the request load increases.

6 Discussion

The results highlight the complementary roles of the two training components. Deployed-policy replay adapts Markov DSpark to draft-induced states and yields its largest gains on the held-out in-domain Business set. Adding Survival Loss further improves both public and Business results at both model scales, indicating that prefix-aware credit provides additional benefits beyond online state-distribution adaptation.

The current study establishes the method on block-7 Markov DSpark across two model scales, multiple task families, temperatures, and serving concurrencies. Natural next steps are to test longer draft blocks, additional drafter architectures, and repeated system measurements. Longer blocks are particularly interesting because the distinction between adaptive prefix credit and fixed positional schedules may become more pronounced as the survival horizon grows.

7 Conclusion

DSpark-OPD combines replay of the deployed Markov-corrected proposal with Survival Loss, a differentiable lower-bound objective for expected accepted length. Its product structure assigns adaptive credit along verification-reachable prefixes without a hand-designed positional schedule. Across Qwen3-8B and Qwen3-14B, the method consistently improves acceptance length and serving throughput over matched online and released DSpark baselines. Controlled ablations further show that dense target–draft overlap is effective across alternative within-block weighting rules. Standard verification and residual correction remain unchanged, preserving exact target-model sampling.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, including grammar correction, sentence refinement, and improvements to clarity and organization. The tools were not used to generate or alter experimental results. We reviewed and revised all AI-assisted text and independently verified the technical content, citations, and claims. We take responsibility for the final content of this work.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

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. *arXiv preprint arXiv:2401.10774*, 2024.

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.

Jian Chen, Yesheng Liang, and Zhijian Liu. Dflash: Block diffusion for flash speculative decoding. *arXiv preprint arXiv:2602.06036*, 2026.

Xin Cheng, Xingkai Yu, Chenze Shao, Jiashi Li, Yunfan Xiong, Yi Qian, Jiaqi Zhu, Shirong Ma, Xiaokang Zhang, Jiasheng Ye, et al. Dspark: Confidence-scheduled speculative decoding with semi-autoregressive generation. *arXiv preprint arXiv:2607.05147*, 2026.

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

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al. Layerskip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12622–12642, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, volume 2024, pp. 54107–54157, 2024.

Haodi Lei, Yafu Li, Haoran Zhang, Shunkai Zhang, Qianjia Cheng, Xiaoye Qu, Ganqu Cui, Bowen Zhou, Ning Ding, Yun Luo, et al. Draft-opd: On-policy distillation for speculative draft models. *arXiv preprint arXiv:2605.29343*, 2026.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International conference on machine learning*, pp. 19274–19286. PMLR, 2023.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024a.

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, 2024b.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *Advances in Neural Information Processing Systems*, 38:136737–136756, 2026.

Hong Liu, Rui Cen, Junhan Shi, Guangshuo Qin, Jiebin Zhang, Tianyu Liu, Runzhi Fan, Guoliang Zhao, Ruobing Xie, Kai Zhang, et al. Angelspec: Towards real-world high performance inference with speculative decoding. *arXiv preprint arXiv:2607.25852*, 2026.

Mathematical Association of America. American invitational mathematics examination (AIME) 2024, 2024.

Mathematical Association of America. American invitational mathematics examination (AIME) 2025, 2025.

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

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.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. *Advances in Neural Information Processing Systems*, 31, 2018.

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

Jiebin Zhang, Zhenghan Yu, Song Liu, Eugene J Yu, Zheng Li, Dawei Zhu, Jiangshan Duo, Weimin Xiong, Yifan Song, Guanghua Yu, et al. Dflare: Scaling up draft capacity for block diffusion speculative decoding. *arXiv preprint arXiv:2606.02091*, 2026.

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

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.

A Online Training Procedure

Input: Frozen target p_θ, warm-start DSpark drafter q_ϕ, prompt set $\mathcal{X}$, and block length K.
1. Rollout. Run speculative decoding with standard verification and q_ϕ. Store c_m, $d_{m,1:K}$, r_m, and the predecessor chain π_m.
2. Replay. Evaluate p_θ and q_ϕ^{corr} on the same sampled prefixes. Apply forward KL to accepted positions and reverse KL to the first rejected position and sampled post-rejection suffix states.
3. Survival. For $k \le H_m = \min(r_m + 1, K)$, compute $a_{m,k} = 1 - \mathrm{TV}(p_{m,k}, q_{m,k})$ and $\widehat{S}_{m,k} = \prod_{j \le k} a_{m,j}$. Mask positions with $k > H_m$.
4. Update. Minimize Eq. 19, fit confidence logits to $\mathrm{sg}(a_{m,k})$ using head-only BCE gradients, and synchronize the updated drafter with the rollout engine.

B Complete System Results

This section reports the complete per-dataset acceptance and serving results underlying the aggregate results in the main text. Public averages exclude the held-out Business test set.

B.1 Complete Acceptance Results

Table 4: Complete Qwen3-8B engine acceptance length τ_+ at concurrency 8. Public Avg. excludes the Business test set.

T	Dataset	DSpark	Warm-start	DSpark-OPD-Base	DSpark-OPD
0	Business Test	2.459	2.763	3.993	**4.383**
	GSM8K	5.658	5.653	6.133	**6.370**
	MATH-500	5.477	5.412	5.974	**6.158**
	AIME 2024	4.650	4.559	5.113	**5.322**
	AIME 2025	4.647	4.558	5.073	**5.339**
	MBPP	4.545	4.516	4.859	**5.030**
	MT-Bench	3.771	3.746	3.874	**4.165**
	SWE-Lite	3.138	3.146	3.337	**3.579**
	Public Avg.	4.555	4.513	4.909	**5.138**
0.6	Business Test	2.423	2.748	3.942	**4.326**
	GSM8K	5.519	5.479	5.980	**6.145**
	MATH-500	5.282	5.198	5.719	**5.872**
	AIME 2024	4.282	4.273	4.796	**4.865**
	AIME 2025	4.336	4.303	4.639	**4.848**
	MBPP	4.375	4.391	4.653	**4.814**
	MT-Bench	3.631	3.674	3.817	**4.047**
	SWE-Lite	2.971	2.963	3.153	**3.357**
	Public Avg.	4.343	4.326	4.680	**4.850**
1	Business Test	2.375	2.653	3.767	**4.143**
	GSM8K	5.273	5.299	5.696	**5.861**
	MATH-500	4.927	4.878	5.271	**5.388**
	AIME 2024	3.906	3.829	4.158	**4.280**
	AIME 2025	3.806	3.771	4.093	**4.349**
	MBPP	4.070	4.065	4.367	**4.536**
	MT-Bench	3.460	3.539	3.644	**3.819**
	SWE-Lite	2.696	2.680	2.818	**2.942**
	Public Avg.	4.020	4.009	4.293	**4.454**

Table 5: Complete Qwen3-14B engine acceptance length τ_+ at concurrency 8. Public Avg. excludes the Business test set.

T	Dataset	DSpark	Warm-start	DSpark-OPD-Base	DSpark-OPD
0	Business Test	2.676	2.896	4.270	**4.510**
	GSM8K	5.977	5.973	5.933	**6.230**
	MATH-500	5.691	5.654	5.757	**6.006**
	AIME 2024	4.791	4.895	4.954	**5.228**
	AIME 2025	4.775	4.787	5.018	**5.277**
	MBPP	4.795	4.745	4.653	**4.921**
	MT-Bench	3.833	3.809	3.811	**4.041**
	SWE-Lite	3.180	3.184	3.112	**3.335**
	Public Avg.	4.720	4.721	4.748	**5.005**
0.6	Business Test	2.668	2.861	4.233	**4.399**
	GSM8K	5.750	5.778	5.790	**6.028**
	MATH-500	5.460	5.429	5.573	**5.725**
	AIME 2024	4.475	4.336	4.448	**4.734**
	AIME 2025	4.494	4.408	4.552	**4.850**
	MBPP	4.574	4.536	4.422	**4.660**
	MT-Bench	3.732	3.743	3.717	**3.888**
	SWE-Lite	2.982	2.962	2.900	**3.096**
	Public Avg.	4.495	4.456	4.486	**4.712**
1	Business Test	2.589	2.790	4.143	**4.270**
	GSM8K	5.553	5.512	5.515	**5.799**
	MATH-500	5.067	5.045	5.186	**5.330**
	AIME 2024	3.901	3.941	3.986	**4.179**
	AIME 2025	3.930	3.838	3.972	**4.081**
	MBPP	4.298	4.283	4.158	**4.346**
	MT-Bench	3.520	3.558	3.543	**3.699**
	SWE-Lite	2.667	2.643	2.628	**2.754**
	Public Avg.	4.134	4.117	4.141	**4.313**

B.2 Complete Serving Results

Table 6: Complete Qwen3-8B output throughput (tokens/s) at temperature 0 and concurrency 8. Public Avg. excludes the Business test set.

Dataset	Target-only	DSpark	Warm-start	DSpark-OPD-Base	DSpark-OPD
Business Test	909	727	776	1456	**1607**
GSM8K	1091	2381	2361	2527	**2610**
MATH-500	1107	2394	2366	2551	**2641**
AIME 2024	1024	2113	2023	2233	**2391**
AIME 2025	1036	2077	2043	2182	**2362**
MBPP	1085	2007	1980	2146	**2202**
MT-Bench	993	1409	1398	1446	**1505**
SWE-Lite	1084	1522	1516	1612	**1709**
Public Avg.	1060	1986	1955	2100	**2203**

Table 7: Complete Qwen3-8B TPOT (ms/token) at temperature 0 and concurrency 8. Public Avg. excludes the Business test set.

Dataset	Target-only	DSpark	Warm-start	DSpark-OPD-Base	DSpark-OPD
Business Test	8.16	8.40	7.41	5.03	**4.69**
GSM8K	7.05	3.15	3.18	2.97	**2.85**
MATH-500	6.99	3.12	3.14	2.93	**2.84**
AIME 2024	7.05	3.45	3.50	3.20	**3.05**
AIME 2025	7.08	3.47	3.51	3.22	**3.06**
MBPP	7.03	3.74	3.78	3.49	**3.40**
MT-Bench	6.98	4.76	4.76	4.61	**4.38**
SWE-Lite	7.21	5.12	5.15	4.85	**4.57**
Public Avg.	7.06	3.83	3.86	3.61	**3.45**

Table 8: Complete Qwen3-14B output throughput (tokens/s) at temperature 0 and concurrency 8. Public Avg. excludes the Business test set.

Dataset	Target-only	DSpark	Warm-start	DSpark-OPD-Base	DSpark-OPD
Business Test	525	500	524	908	**973**
GSM8K	665	1829	1831	1827	**1920**
MATH-500	678	1793	1781	1809	**1889**
AIME 2024	634	1539	1541	1536	**1675**
AIME 2025	631	1532	1543	1589	**1657**
MBPP	662	1542	1505	1477	**1567**
MT-Bench	615	1070	1057	1009	**1074**
SWE-Lite	666	1096	1090	1066	**1157**
Public Avg.	650	1486	1478	1473	**1563**

Table 9: Complete Qwen3-14B TPOT (ms/token) at temperature 0 and concurrency 8. Public Avg. excludes the Business test set.

Dataset	Target-only	DSpark	Warm-start	DSpark-OPD-Base	DSpark-OPD
Business Test	13.95	11.79	10.84	8.22	**7.54**
GSM8K	11.60	4.09	4.03	4.04	**3.91**
MATH-500	11.51	4.16	4.19	4.12	**3.95**
AIME 2024	11.49	4.67	4.58	4.56	**4.33**
AIME 2025	11.52	4.64	4.64	4.52	**4.29**
MBPP	11.51	4.86	4.96	5.06	**4.78**
MT-Bench	11.50	6.35	6.38	6.59	**6.25**
SWE-Lite	11.69	7.12	7.10	7.28	**6.78**
Public Avg.	11.55	5.13	5.13	5.17	**4.90**

C Complete Controlled Ablation Results

All controlled variants start from the same Qwen3-8B checkpoint and use matched rollout, optimization, and evaluation settings. L0 uses no auxiliary overlap objective. L1 applies uniform per-position overlap, L2 applies fixed position decay with $\rho = 0.8$, and L3 uses the prefix-survival score in Eq. 13. The coefficient study evaluates $\lambda_s \in \{0, 0.5, 1, 4, 16\}$ while keeping all other settings fixed.

The following tables provide the complete per-dataset results for the objective-structure comparison and the Survival-coefficient sweep.

Table 10: Complete objective-structure results at concurrency 8. Public Avg. excludes the Business test set.

T	Dataset	L0: base	L1: local overlap	L2: decayed overlap	L3: Survival
0	Business Test	3.920	4.393	4.412	4.292
	GSM8K	6.115	6.373	6.390	6.338
	MATH-500	5.909	6.152	6.157	6.147
	AIME 2024	5.057	5.399	5.471	5.407
	AIME 2025	5.047	5.413	5.362	5.379
	MBPP	4.777	5.028	5.020	5.029
	MT-Bench	3.883	4.161	4.143	4.099
	SWE-Lite	3.285	3.576	3.579	3.562
	Public Avg.	4.867	5.158	5.160	5.137
0.6	Business Test	3.836	4.289	4.315	4.218
	GSM8K	5.844	6.082	6.133	6.139
	MATH-500	5.645	5.881	5.858	5.809
	AIME 2024	4.774	4.986	4.868	4.997
	AIME 2025	4.583	4.904	4.895	4.798
	MBPP	4.593	4.806	4.816	4.868
	MT-Bench	3.749	4.015	4.003	3.946
	SWE-Lite	3.126	3.317	3.343	3.298
	Public Avg.	4.616	4.856	4.845	4.836
1	Business Test	3.682	4.139	4.177	4.013
	GSM8K	5.673	5.846	5.844	5.839
	MATH-500	5.187	5.419	5.463	5.422
	AIME 2024	4.142	4.292	4.299	4.274
	AIME 2025	3.989	4.215	4.427	4.257
	MBPP	4.300	4.556	4.532	4.509
	MT-Bench	3.607	3.863	3.802	3.768
	SWE-Lite	2.777	2.957	2.922	2.951
	Public Avg.	4.239	4.450	4.470	4.431

Table 11: Complete Survival-coefficient sweep at concurrency 8. Public Avg. excludes the Business test set.

T	Dataset	λ_s=0	0.5	1	4	16
0	Business Test	3.920	4.194	4.292	4.460	4.484
	GSM8K	6.115	6.314	6.338	6.435	6.454
	MATH-500	5.909	6.087	6.147	6.211	6.226
	AIME 2024	5.057	5.295	5.407	5.448	5.544
	AIME 2025	5.047	5.242	5.379	5.395	5.411
	MBPP	4.777	4.938	5.029	5.095	5.121
	MT-Bench	3.883	4.024	4.099	4.200	4.250
	SWE-Lite	3.285	3.500	3.562	3.683	3.673
	Public Avg.	4.867	5.057	5.137	5.209	5.240
0.6	Business Test	3.836	4.095	4.218	4.326	4.364
	GSM8K	5.844	6.057	6.139	6.191	6.202
	MATH-500	5.645	5.842	5.809	5.909	5.932
	AIME 2024	4.774	4.940	4.997	4.962	4.908
	AIME 2025	4.583	4.868	4.798	4.987	4.919
	MBPP	4.593	4.749	4.868	4.925	4.874
	MT-Bench	3.749	3.960	3.946	4.054	4.115
	SWE-Lite	3.126	3.256	3.298	3.376	3.386
	Public Avg.	4.616	4.810	4.836	4.915	4.905
1	Business Test	3.682	3.909	4.013	4.200	4.178
	GSM8K	5.673	5.797	5.839	5.848	5.907
	MATH-500	5.187	5.430	5.422	5.504	5.473
	AIME 2024	4.142	4.221	4.274	4.264	4.447
	AIME 2025	3.989	4.177	4.257	4.293	4.308
	MBPP	4.300	4.460	4.509	4.534	4.567
	MT-Bench	3.607	3.716	3.768	3.829	3.935
	SWE-Lite	2.777	2.917	2.951	2.993	2.996
	Public Avg.	4.239	4.388	4.431	4.466	4.519