DiffSpec: Accelerating Long Sequence Generation with Differential Speculative Decoding

Zhongcheng Du, Yu Huang, Qihang Qiu, Yi Zhang, Deting Chen, Chencheng Ye, Xiaofei Liao, Hai Jin

Huazhong University of Science and Technology (HUST)

Overview

1. Introduction
2. Methodology
3. Experiments
4. Conclusion

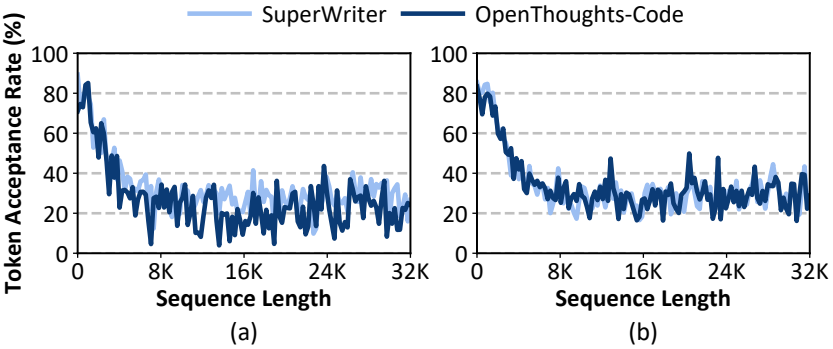
The Long Sequence Bottleneck

The Long Sequence Bottleneck

LLM inference for long sequences (up to 128K tokens) is dominated by memory bandwidth and attention costs.

- **KV Cache Growth:** Footprint scales linearly with sequence length, creating sustained memory bandwidth pressure.
- **Attention Cost:** Each token attends to all predecessors, causing linear scaling of compute overhead.
- **Decoding Inefficiency:** Standard autoregressive decoding becomes disproportionately expensive in later stages.
- **Speculation Limits:** Uniform speculative decoding fails as draft accuracy degrades with increasing context length.

Limitations of Existing Methods



- EAGLE-3 and SpecInfer: TAR falls from $> 80\%$ to $< 25\%$ over 0–32K tokens.
- Uniform speculation underuses later decoding stages, where successful speculation pays most.

The Core Insight

Rising Cost

- Decoding cost scales **linearly** with sequence length.
- Attention and KV cache footprint grow continuously.
- Later stages incur **30**$\times$ **higher** per-step latency.

Falling Acceptance

- Token Acceptance Rate (TAR) drops sharply.
- Draft quality degrades due to semantic drift.
- Uniform speculation wastes budget in low-impact regions.

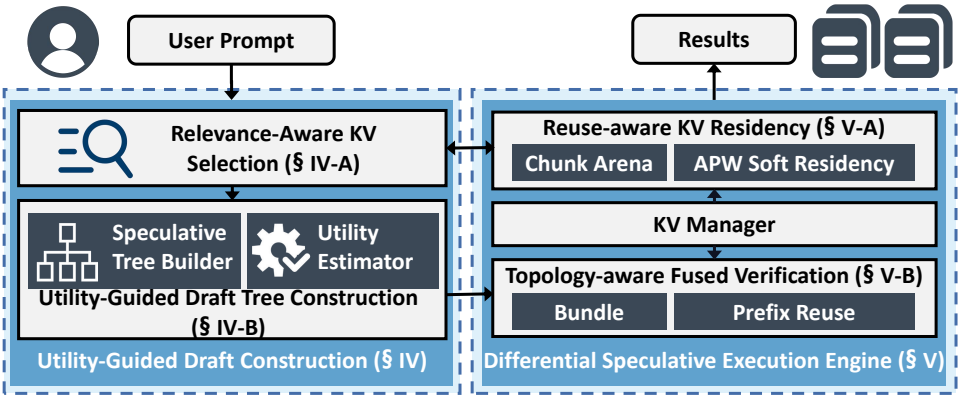
The Mismatch

Speculation fails most where it matters most.

DiffSpec Insight

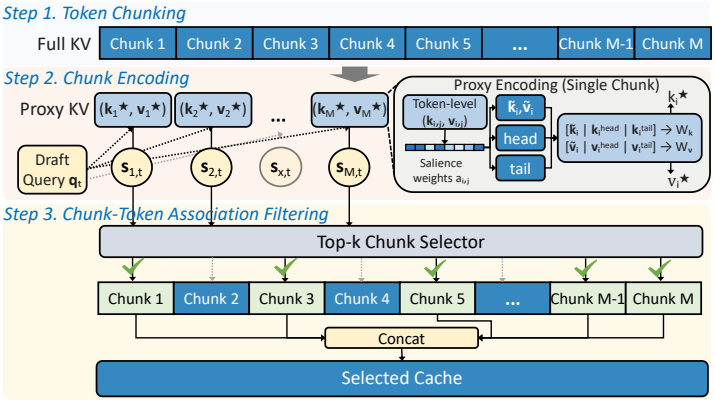
Align speculative effort with **position-specific utility**.

DiffSpec Overview



- **Draft construction:** relevance-aware KV selection and verifier-guided tree branching.
- **Execution engine:** reuse-aware KV residency and topology-aware fused verification.

Relevance-Aware KV Selection



- **Chunk and encode:** build proxy KV pairs with salience weights and boundary preservation.
- **Filter:** rank query–chunk associations and retain the top-*k* relevant chunks.

Utility-Guided Tree Construction

Rejection Hazard

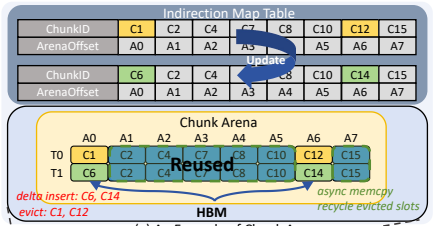
Compute hazard $h_d = \frac{R_d+\alpha}{N_d+2\alpha}$ using verifier feedback. R_d: first rejections at depth d; N_d: verifications reaching depth d.

- **Adaptive Branching**: Select top-$L_{\max}$ depths with highest hazard for branching.
- **Chain-like Growth**: Keep non-hazard depths as single chains to save budget.
- **Budget Allocation**: Concentrate width at failure points to extend verified prefix.
- **Hardware Friendly**: Regular sibling groups enable efficient GPU bundle verification.

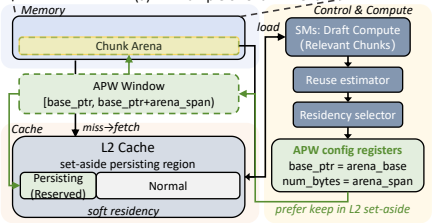
Addressing System Irregularity

Differential speculative decoding introduces sparse KV accesses and heterogeneous workloads.

- **Chunk Arena**: Packs selected KV blocks into contiguous memory.
- Converts logical sparsity into physical contiguity for better coalescing.
- **APW-Guided Soft Residency**: Favors L2 persistence for hot chunks; residency is a soft policy.
- Reduces memory traffic for sparse draft attention via temporal locality.

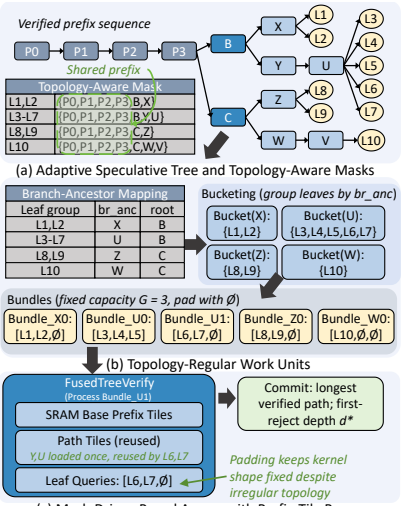


(a) An Example of Chunk Arena



(b) APW-Guided Soft Residency

Topology-Aware Verification



Topology-Aware Verification

DiffSpec handles irregular tree structures efficiently through:

- **Ancestor-Based Bucketing**: Groups leaf queries by nearest branching ancestor to maximize prefix coherence.
- **Fixed-Capacity Bundling**: Packs buckets into dense bundles, ensuring uniform workload distribution.
- **Paged KV Access**: Uses logical-to-physical indirection for flexible, coalesced memory access.
- **SRAM Prefix Reuse**: Stages shared prefix tiles once in on-chip memory for bundle-wide reuse.

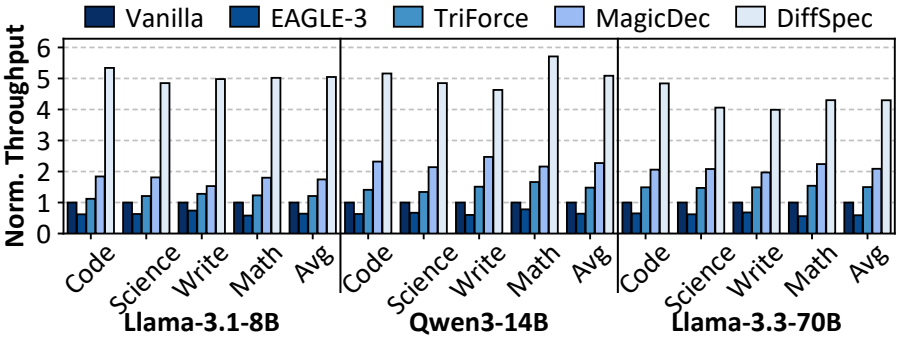
Models & Datasets

- **Models:** Llama-3.1-8B, Qwen3-14B, Llama-3.3-70B
- **Datasets:** OpenThoughts (Code, Math, Science), SuperWriter (Write)

Baselines & Hardware

- **Baselines:** Vanilla, EAGLE-3, TriForce, MagicDec
- **Hardware:** 4× NVIDIA H100 GPUs
- **Output length:** 8B/14B up to 128K; 70B capped at 32K tokens

Overall Performance



Key Results

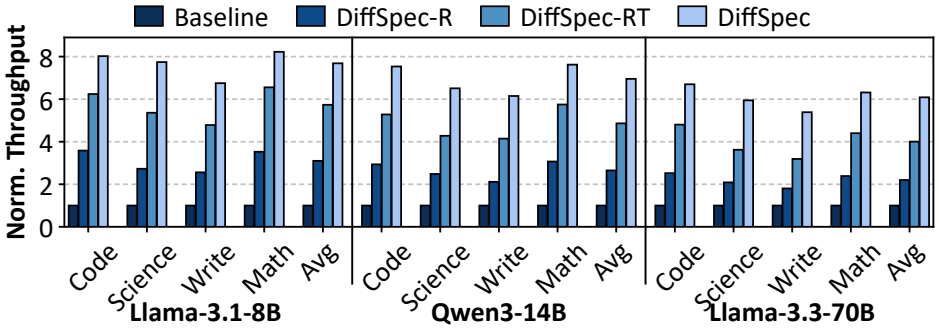
Average end-to-end speedups reported across the evaluated workloads and models.

Method	vs. EAGLE-3	vs. TriForce	vs. MagicDec
DiffSpec	**7.72x**	**3.45x**	**2.36x**

Scalability Insight

On Llama-3.1-8B, speedup over Vanilla rises from 2.48× to 6.22× over 0–32K generated tokens.

Scalability & Breakdown



Longer Generation

- Llama-3.1-8B, 0–32K generated tokens; speedup over Vanilla.
- DiffSpec: 2.48× → 6.22×.
- TriForce: 2.83× → 1.36×; MagicDec: 2.75× → 2.08×.

Incremental Ablations

- **KV selection (R):** 2.20–3.10× over EAGLE-3.
- **Add tree construction (RT):** 1.82–1.85× over R.
- **Add execution engine:** 1.34–1.52× over RT.

Conclusion & Impact

Core Insight

DiffSpec aligns speculative effort with position-specific utility to solve long-sequence mismatch.

- **Characterized Inefficiency**: Uniform speculation wastes budget in low-impact regions, causing failures in costly late stages.
- **Utility-Guided Drafting**: Relevance-aware KV selection and adaptive tree construction improve candidate quality where payoff is highest.
- **Differential Execution Engine**: Reuse-aware KV residency and topology-aware verification handle irregular access patterns efficiently.
- **Significant Speedups**: Achieves 7.72×, 3.45×, and 2.36× speedups over EAGLE-3, TriForce, and MagicDec, respectively.

Thank You!