Efficient Diffusion Language Models: A Comprehensive Survey

Haokun Lin[*†1,4], **Xinle Jia**[*3], **Shaozhen Liu**[*1], **Shujun Xia**[*4], **Weitao Huang**[*6], **Haobo Xu**[7], **Junyang Li**[1], **Yicheng Xiao**[7,8], **Xingrun Xing**[1], **Ziyu Guo**[2], **Renrui Zhang**[2], **Qi Li**[1], **Yichen Wu**[‡4,5], **Renzhen Wang**[6], **Xiaojuan Qi**[8], **Caifeng Shan**[3], **Hongsheng Li**[2], **Zhenan Sun**[‡1]

[1] NLPR & MAIS, Institute of Automation, CAS [2]The Chinese University of Hong Kong
[3]Nanjing University [4]City University of Hong Kong [5]Harvard University
[6]Xi'an Jiaotong University [7]Tsinghua University [8]The University of Hong Kong

[*]Equal Contribution [†]Project Leader [‡]Corresponding Author

GitHub Repository: https://github.com/FelixMessi/Awesome-Efficient-dLLMs

Abstract

Diffusion language models (dLLMs) have recently emerged as a promising alternative to autoregressive (AR) language models, offering competitive performance with a fundamentally different generation paradigm. Instead of producing tokens strictly left-to-right, dLLMs iteratively refine partially masked sequences, enabling bidirectional context modeling and supporting parallel token updates. This design creates new opportunities for accelerating generation and improving controllability. However, practical deployment of dLLMs remains challenging due to the high cost of training at scale and the lack of mature inference optimizations such as cache-friendly decoding and robust parallel sampling. In this work, we provide a comprehensive overview of recent progress on *efficient* dLLMs. To reflect how efficiency bottlenecks arise across the full lifecycle of dLLMs, spanning training, decoding, and deployment, we organize existing approaches into five categories: *Training*, *Inference*, *Context*, *Framework*, and *Multimodality*. For each category, we summarize representative methods and highlight their motivations, key techniques, and empirical trade-offs. Finally, we discuss open challenges and outline promising directions for future research toward high-throughput, robust, and practically deployable diffusion language models.

1 Introduction

Autoregressive (AR) large language models (LLMs) have shown strong performance across tasks, but their left-to-right decoding generates tokens sequentially, limiting parallelism and making inference inefficient [Yang *et al.*, 2025a]. Diffusion language models (dLLMs) offer an alternative paradigm by iteratively refining partially masked sequences [Gong *et al.*, 2022; Song *et al.*, 2025b]. This enables bidirectional context modeling and allows multiple tokens to be updated in parallel, making dLLMs a promising direction for efficient generation and improving controllability [Yang *et al.*, 2025c; You *et al.*, 2025; Zhu *et al.*, 2025a; Liu *et al.*, 2025a].

However, deploying dLLMs in practice still faces several efficiency bottlenecks. First, training dLLMs from scratch is highly data- and compute-intensive, and early open-source models (e.g., LLaDA [Nie *et al.*, 2025]) remain less competitive than strong AR counterparts. Second, during inference, dLLMs have not yet shown consistent speed advantages over AR models with optimized serving engines like vLLM. This is largely due to the incompatibility of standard KV caching with bidirectional attention and the generation quality degradation of parallel decoding [Wu *et al.*, 2025b]. Third, supporting long-context and variable-length generation is nontrivial, as many dLLMs rely on fixed denoising schedules and pre-specified output lengths.

In this survey, we provide the first comprehensive review of works that aim to improve the efficiency of dLLMs and enable their practical deployment. As illustrated in Figure 1, we organize the literature from multiple perspectives. For *training efficiency*, we summarize approaches that convert pretrained AR models into dLLMs [Ye *et al.*, 2025; Wu *et al.*, 2025a] as well as architectural designs [Zhu *et al.*, 2025b] that enable efficient dLLMs. For *inference efficiency*, we review parallel decoding strategies [Wu *et al.*, 2025b; Chen *et al.*, 2025b], dynamic KV-cache management [Ma *et al.*, 2025a; Jiang *et al.*, 2025], and compression techniques [Lin *et al.*, 2025b; Agrawal *et al.*, 2025] that improve the speed–quality trade-off. We further discuss *context scalability*, including methods for variable-length and long-context generation [Liu *et al.*, 2025b; Arriola *et al.*, 2025a]. In addition, we highlight emerging *frameworks* that support dLLM training, evaluation, and production serving [Zhou *et al.*, 2025c; Ma *et al.*, 2025b], and summarize recent progress on efficient *multimodal diffusion language models* [Yu *et al.*, 2025b].

Although existing surveys [Li *et al.*, 2025g; Yu *et al.*, 2025a] have reviewed diffusion language models broadly, our survey focuses specifically on efficiency-related research. Our main contributions are threefold: (1) we propose a unified taxonomy that organizes efficient dLLM techniques into five categories, including *Training*, *Inference*, *Context*, *Framework*, and *Multimodality*; (2) we systematically review representative works in each category and analyze how they address key bottlenecks; and (3) we outline open challenges and promising future directions for efficient dLLMs.

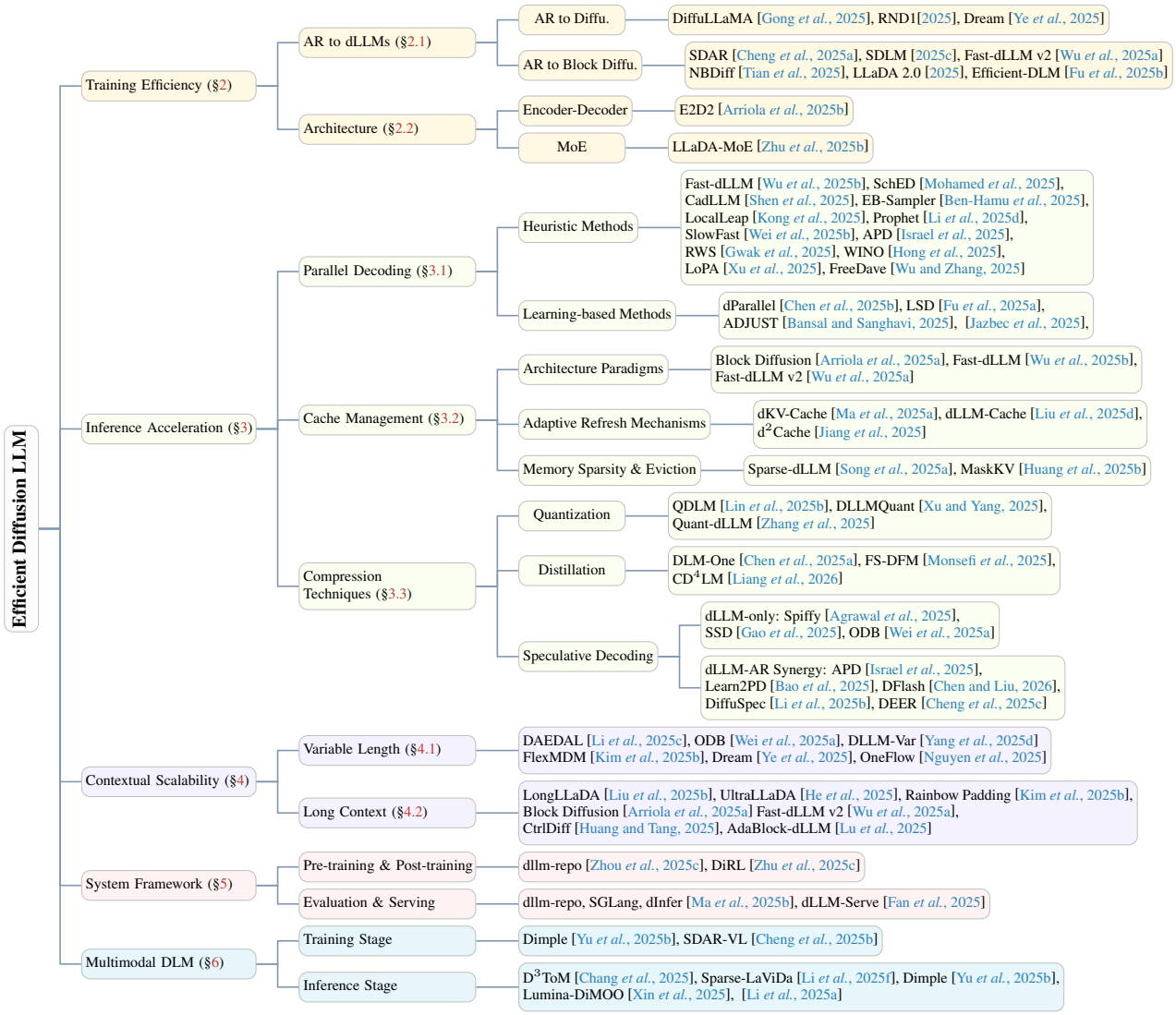

Figure 1: Our taxonomy tree of Efficient Diffusion Language Models (dLLMs).

2 Training Efficiency

2.1 AR to dLLMs

AR to Diffusion. Training diffusion language models from scratch at scale is computationally expensive. A practical alternative is to start from a pretrained AR model and focus on the core challenge of transition: how to convert an AR model into a diffusion-ready backbone without losing its linguistic competence. DiffuLLaMA [Gong *et al.*, 2025] explicitly targets this AR-to-diffusion transfer, introducing dedicated transition techniques, such as attention-mask annealing, shift operations, and a time-embedding-free architecture. Dream-7B [Ye *et al.*, 2025] pushes this direction further by turning the transition into a full training pipeline, notably incorporating context-adaptive token-level noise scheduling to improve downstream performance. In contrast, RND1 [Chandrasegaran *et al.*, 2025] argues that the transition can be simplified: it directly switches from AR initialization to dLLM training, avoiding explicit intermediate adaptation steps (e.g., attention annealing).

AR to Block Diffusion. Recent work on block diffusion language models [Arriola *et al.*, 2025a], which generates text block-by-block while denoising the tokens within each block rather than denoising the entire sequence iteratively, shows that this approach preserves a structured dependency pattern more naturally aligned with AR pretraining. Subsequent studies indicate that adapting AR models to block-wise diffusion can be substantially more efficient than converting them to fully diffusion-based language models. SDAR [Cheng *et al.*, 2025a] is the first to propose a highly efficient AR-to-block diffusion conversion pipeline. It leverages dense token-level supervision inherited from AR pretraining, enabling the diffusion objective to be aligned with only a short adaptation stage. Building on this efficiency-oriented view, SDLM [Liu *et al.*, 2025c] further introduces a Next Sequence Prediction (NSP) objective that interleaves noise blocks with tar-

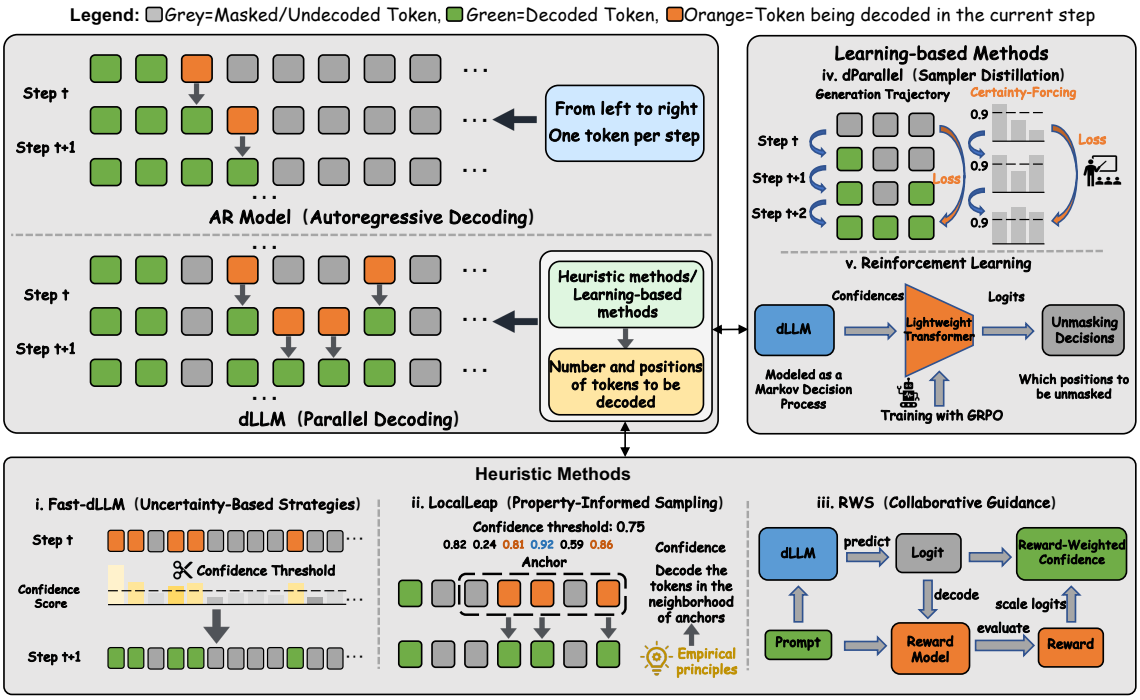

Figure 2: **Illustration of Parallel Decoding for dLLMs.** Unlike AR models that decode one token per step in a left-to-right manner, dLLMs can update multiple tokens simultaneously. In practice, heuristic or learning-based strategies are used to decide *which* token positions to decode and *how many* tokens to update at each step. Subfigures illustrate representative and state-of-the-art decoding methods.

get blocks, allowing parallel training while retaining block-level structure. Several subsequent works further refine the adaptation mechanism. NBDiff [Tian *et al.*, 2025] designs a context-causal attention mask tailored for block diffusion adaptation and augments training with an auxiliary AR loss to improve convergence and knowledge retention. Fast-dLLM v2 [Wu *et al.*, 2025a] identifies AR-friendly block-wise attention patterns and shows that effective block diffusion models can be obtained through lightweight post-training. Similarly, Efficient-DLM [Fu *et al.*, 2025b] demonstrates that continuous pretraining with block-wise attention initialized from AR weights yields strong efficiency gains at low cost. More recently, LLaDA 2.0 [Bie *et al.*, 2025] proposes a Warmup-Stable-Decay continual pretraining schedule to further stabilize the AR-to-block diffusion transition. Overall, these methods highlight that the core advantage of block diffusion lies in structurally aligning diffusion objectives with AR pretraining, making efficient adaptation possible without heavy architectural or training-stage interventions.

2.2 Architecture

Architecture choices in dLLMs are primarily motivated by efficiency considerations, as diffusion training and inference inherently involve repeated processing of partially noised sequences. Two design patterns are prevalent: decoupling clean and noisy representations to avoid redundant computation, and increasing model capacity through sparsity without proportional compute growth. E2D2 [Arriola *et al.*, 2025b] adopts an **encoder–decoder** architecture for block diffusion, where the encoder processes clean inputs and the decoder focuses on denoising, effectively reusing representations and reducing training cost by roughly half compared to decoder-only models of equal size. Orthogonally, **Mixture-of-Experts** (MoE) introduces sparse expert routing to scale model capacity efficiently. For instance, LLaDA-MoE [Zhu *et al.*, 2025b] is the first to integrate MoE into diffusion language models, showing that sparse MoE-based dLLMs can outperform dense diffusion baselines and remain competitive with autoregressive models while activating substantially fewer parameters at inference time.

3 Inference Acceleration

3.1 Parallel Decoding

Currently, the open-source dLLMs have yet to fully achieve their high-throughput potential in real-world applications, primarily due to the performance degradation caused by decoding multiple tokens in parallel. To address this, several approaches have been proposed by improving the parallel decoding and sampling strategy, which can be categorized into two types: heuristic methods and learning-based methods.

Heuristic Methods leverage handcrafted metrics and empirical observations to optimize the decoding process, offering plug-and-play efficiency gains without extensive retraining. These approaches can be categorized into three dimensions:

- **Uncertainty-Based Strategies**: These methods leverage token-level uncertainty signals (e.g., confidence scores or entropy) to determine *which* tokens to decode and *how many* tokens to update at each step. Fast-dLLM [Wu *et al.*, 2025b] selects tokens whose confidence exceeds a pre-

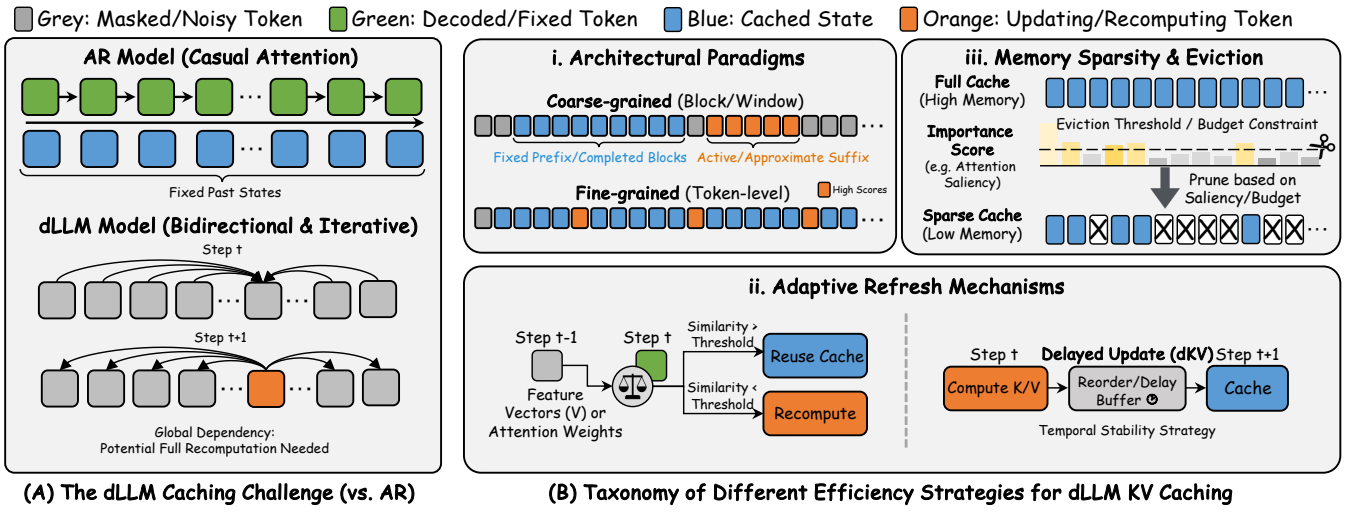

Figure 3: **Illustration of Cache Management for dLLMs.** (A) The key challenge in dLLM caching. Unlike AR models, dLLMs rely on bidirectional attention and iteratively refine the entire sequence, so KV states are not fixed and cannot be reused. (B) A taxonomy of efficiency strategies for dLLM KV caching: (i) *granularity control*, from coarse-grained blocks to fine-grained tokens; (ii) *adaptive refresh*, using similarity thresholds or delayed updates; and (iii) *memory sparsity and eviction*, guided by saliency signals and budget constraints.

set threshold, enabling safe and effective parallel decoding. SchED [Mohamed *et al.*, 2025] proposes a training-free early-exit mechanism that halts decoding based on smooth, progress-dependent confidence thresholds, delivering substantial speedups. CadLLM [Shen *et al.*, 2025] dynamically adjusts block size, step size, and unmasking thresholds based on confidence signals to improve throughput with competitive accuracy. EB-Sampler [Ben-Hamu *et al.*, 2025] uses an entropy-bound scheduler to adaptively select the number and order of token updates per step, controlling both model and dependency errors.

- **Property-Informed Sampling**: Rather than relying on explicit uncertainty scores, these methods exploit empirical decoding patterns such as locality, progressive consistency, and early stabilization. LocalLeap [Kong *et al.*, 2025] leverages local determinism and spatial consistency to perform parallel decoding within neighborhoods around high-confidence anchor tokens. Prophet [Li *et al.*, 2025d] utilizes an early answer convergence phenomenon and terminates decoding once token transitions stabilize. Slow-Fast Sampling [Wei *et al.*, 2025b] summarizes three principles for the decoding of dLLMs: certainty, convergence, and positional principle. Guided by these three principles, it designs a cyclical "exploration-acceleration" two-phase sampling strategy, significantly enhancing the inference efficiency of dLLMs.
- **Collaborative Guidance & Speculative Verification**: To further boost reliability, some methods introduce external guidance or internal verification loops. APD [Israel *et al.*, 2025] and RWS [Gwak *et al.*, 2025] utilize auxiliary autoregressive or reward models to refine the joint distribution and prioritize token decoding. On the architectural side, WINO [Hong *et al.*, 2025], LoPA [Xu *et al.*, 2025], and FreeDave [Wu and Zhang, 2025] adopt a speculative paradigm: they first draft multiple candidate tokens and then use bidirectional context or internal self-verification to prune errors. This allows for revocable decoding, enabling the model to adaptively correct earlier stages as more contextual information becomes available.

Discussion. Heuristic methods offer strong practicality due to their simplicity and plug-and-play nature, while their effectiveness largely depends on the quality of uncertainty signals and the robustness of verification under aggressive settings.

Learning-based Methods generally encompass sampler distillation, trained auxiliary modules, and reinforcement learning, enabling significant inference acceleration in both semi-autoregressive and fully parallel scenarios:

- **Sampler Distillation**: dParallel [Chen *et al.*, 2025b] designs a novel training strategy named certainty-forcing distillation, which distills the model to adhere to its original sampling trajectories and enables it to achieve high certainty over the masked tokens more rapidly and concurrently. This strategy significantly reduces the number of decoding steps while preserving the output quality of the model. LSD [Fu *et al.*, 2025a] employs a distillation approach that enables a student sampler with fewer steps to learn and align with the intermediate score trajectories of a teacher sampler that utilizes more steps. Additionally, LSD+ further learns a non-uniform time schedule to adaptively allocate sampling steps.
- **Trained Auxiliary Modules**: ADJUST [Bansal and Sanghavi, 2025] trains a lightweight single-layer sampler based on an existing dLLM. Through several forward passes of this lightweight sampler after a single full-model forward pass, approximately sampling multiple tokens from the joint distribution can be achieved.
- **Reinforcement Learning**: [Jazbec *et al.*, 2025] formulates dLLM sampling as a Markov Decision Process and use GRPO to train a lightweight policy network that adaptively selects token positions to unmask at each timestep.

Discussion. Overall, learning-based methods provide a systematic path to accelerate diffusion decoding by optimizing sampling policies directly, though they typically introduce extra training overhead and require careful design to maintain competitiveness across tasks and prompts.

3.2 Cache Management

The deployment of dLLMs is fundamentally hindered by the incompatibility of standard Key-Value (KV) caching with bidirectional attention mechanisms. Unlike AR LLMs, where past states remain fixed and cached KV pairs can be reused across decoding steps [Hao *et al.*, 2025b; Liu *et al.*, 2024], dLLMs iteratively refine the entire sequence and thus require repeated updates to token representations. As illustrated in Figure 3, current works have introduced caching mechanisms to dLLMs by architectural adaptations, algorithmic refinements, and memory optimizations.

Architecture Paradigms. Architecture approaches to caching in dLLMs mainly differ in the granularity at which generation dependencies are fixed and reused, as shown in Figure 3(i). Coarse-grained approaches, such as Block Diffusion [Arriola *et al.*, 2025a], adopt a window-based scheme that caches completed prefixes as fixed "Blue" blocks and restricts diffusion to an active "Orange" suffix. Fast-dLLM [Wu *et al.*, 2025b] and Fast-dLLM v2 [Wu *et al.*, 2025a] refine this idea with a DualCache design that separates exact prefix states from approximate masked suffix representations. In contrast, fine-grained strategies operate at the token level and selectively recompute only a small set of unstable tokens, which are identified using criteria such as confidence scores or attention weights, while directly reusing cached states for the remaining stable tokens.

Adaptive Refresh Mechanisms. For standard diffusion architectures, the main challenge in caching is KV drift, where token representations evolve across denoising steps, as shown in Figure 3(ii). Adaptive Refresh Mechanisms address this issue by dynamically deciding whether cached states should be reused or recomputed based on stability signals such as feature similarity or attention. dLLM-Cache [Liu *et al.*, 2025d] proposes a V-Verify strategy that compares Value vectors against a similarity threshold, reusing the cache when changes are small and recomputing otherwise. dKV-Cache [Ma *et al.*, 2025a] adopts a temporal stability strategy with delayed updates, postponing cache commits until token states stabilize. d^2Cache [Jiang *et al.*, 2025] further refines this idea by prioritizing recomputation for high-attention groups while freezing low-attention states.

Memory Sparsity & Eviction. For long-context generation, cache management in dLLMs must operate under strict memory budgets, see Figure 3(iii). Memory sparsity and eviction methods [Nguyen-Tri *et al.*, 2025] address this by retaining only the most informative token states. Sparse-dLLM [Song *et al.*, 2025a] assigns attention-based importance scores to tokens and dynamically evicts low-importance entries, forming a sparse cache that preserves critical context. MaskKV [Huang *et al.*, 2025b] extends this idea with the *Mask as Prophet* principle, aggregating importance signals from mask tokens to guide more reliable eviction decisions. By pruning based on global importance, these approaches cut memory footprint with minimal impact on denoising.

Discussion. These innovations mitigate the quadratic complexity of bidirectional attention, enabling dLLMs to approach the inference latency of AR models without sacrificing generation quality. Future research is expected to pivot from pure acceleration to broader applications, such as optimizing cache dynamics for multi-modal synthesis [Xin *et al.*, 2025] and utilizing the KV cache as a compressed protocol for multi-agent communication, thereby solidifying dLLMs as a scalable foundation for unified generative systems.

3.3 Compression Techniques

Quantization. To mitigate the substantial memory cost and computational overhead, Post-Training Quantization [Yang *et al.*, 2024; Yang *et al.*, 2025b; Lin *et al.*, 2024b] has been extended from AR LLMs to dLLMs. QDLM [Lin *et al.*, 2025b] establishes the first comprehensive benchmark, identifying distinct activation outliers in dLLMs and evaluating state-of-the-art PTQ baselines. Subsequent works address dLLM-specific features: DLLMQuant [Xu and Yang, 2025] introduces a fine-grained quantization scheme that accounts for temporal embedding, mask ratios, and attention distributions. Meanwhile, Quant-dLLM [Zhang *et al.*, 2025] identifies the critical role of timestep-dependent masking and proposes a data-aware, any-order quantizer, enabling extreme 2-bit weight-only quantization. Despite these algorithmic advances, two primary gaps remain: (1) a lack of specialized low-bit kernel support to translate theoretical compression into wall-clock speedup, (2) the unaddressed synergy between quantization and dLLM-specific caching mechanisms and (3) the need for Quantization-aware Training [Huang *et al.*, 2025a; Huang *et al.*, 2026] to better preserve quality.

Distillation. To bridge the sampling speed gap between dLLMs and AR LLMs, several works utilize distillation techniques to compress the iterative denoising process into minimal steps. DLM-One [Chen *et al.*, 2025a] focuses on optimizing continuous dLLMs and distilling a teacher's score function into a one-step generator in continuous embedding space, achieving a 500$\times$ speedup without iterative refinement. For discrete dLLM generation, FS-DFM [Monsefi *et al.*, 2025] introduces the few-step discrete flow-matching, achieving competitive long-text perplexity with only eight sampling steps, effectively balancing generation quality and throughput. Furthermore, CD^{4}LM [Liang *et al.*, 2026] uses discrete-space consistency distillation to train the model to denoise correctly even when intermediate steps are skipped. Collectively, these approaches suggest that distillation-based methods provide a promising pathway for accelerating dLLM inference. However, preserving generation quality, especially long-form coherence, reasoning ability, and robustness, under aggressive step reduction remains a central challenge. Moreover, jointly exploring distillation with compression techniques such as pruning [Zhang *et al.*, 2024; Lin *et al.*, 2024a; Xing *et al.*, 2025; Hao *et al.*, 2025a] is still underexplored.

Speculative Decoding (SD). Speculative decoding is a family of inference acceleration techniques that improves

throughput by *decoupling proposal and validation*: the system first proposes multiple candidate token updates in parallel and then verifies them with a stronger decision rule to avoid quality degradation [Chen *et al.*, 2023; Li *et al.*, 2024]. While classical SD for autoregressive LLMs often relies on a separate lightweight draft model, recent dLLM work generalizes this paradigm into two directions: *dLLM-only speculation* and *dLLM-AR synergy*. Below, we organize existing methods under these two categories and summarize how they improve the throughput and quality.

dLLM-only Speculation optimizes the internal denoising process or uses dLLMs for both drafting and verification.

- **Self-Speculation**: This category focuses on unleashing the dLLM's inherent potential to accelerate itself without external draft models. Spiffy [Agrawal *et al.*, 2025] introduces an "Auto-Speculation" mechanism, where the model predicts future intermediate states within its own denoising process. These predictions are structured into "Directed Draft Graphs" calibrated offline to maximize the acceptance rate during parallel verification. Similarly, SSD [Gao *et al.*, 2025] leverages the model to self-draft multiple tokens and verifies them using a "Hierarchical Verification Tree", enabling the acceptance of valid sub-sequences in a single forward pass.
- **Iteration Optimization**: Addressing the memory-bound nature of decoding stage, ODB-dLLM [Wei *et al.*, 2025a] introduces a "Jump-Share" speculative decoding mechanism that selectively skips denoising steps (Jump) and shares KV caches across layers (Share). This strategy optimizes arithmetic intensity, significantly reducing memory bandwidth pressure during iterative denoising phase.

dLLM-AR Synergy Speculation bridges dLLM and AR models to leverage their complementary strengths.

- **Adaptive Speculation**: Motivated by the fact that token difficulty varies across contexts, these methods adapt the speculation budget by dynamically deciding how many tokens to draft in parallel and which drafts require further refinement. APD [Israel *et al.*, 2025] leverages a small auxiliary AR model to estimate the joint plausibility of dLLM drafts and adjusts the parallel lookahead accordingly, improving the throughput–quality trade-off. Learn2PD [Bao *et al.*, 2025] takes a learning-based route by training a lightweight filter model to predict draft correctness; low-confidence tokens are selectively re-masked and re-denoised, enabling selective verification and early acceptance for "easy" tokens.
- **Hybrid Verifier Architectures**: These approaches marry the parallel generation capabilities of Diffusion (as drafters) with the precise reasoning of Autoregression (as verifiers). DEER [Cheng *et al.*, 2025c] addresses the "uncertainty accumulation" typical of AR drafters by employing a discrete dLLM to generate high-quality block drafts, which are subsequently verified by the target AR model. DiffuSpec [Li *et al.*, 2025b] proposes a training-free solution using "Causal-Consistency Path Search" (CPS) to extract AR-aligned sequences directly from the dLLM's internal token lattice. Furthermore, DFlash [Chen and Liu, 2026] enhances drafting by conditioning a lightweight diffusion model on the "hidden context features" of the large target AR model, bridging the representation gap between the compact drafter and the large verifier.

Discussion. Looking forward, effective SD algorithms for dLLMs will likely require adaptive speculation strategies that are tightly coupled with diffusion-specific decoding dynamics, as well as synergistic integration with model compression techniques. Such co-design is essential to fully exploit hardware parallelism while preserving generation quality, particularly under increasingly aggressive acceleration regimes.

4 Contextual Scalability

4.1 Variable Length

Early diffusion language models rely on a fixed, pre-specified output length, which limits practical usability and wastes computation by denoising padding tokens or over-allocated sequence slots [Han *et al.*, 2023]. Enabling native variable-length generation is therefore essential for efficient dLLMs, as it allows computation to adapt to the actual response length while preserving diffusion's parallel decoding advantages.

Inference-Time Adaptive Length Control. This line of work determines output length on the fly during inference using confidence-based stopping signals, rather than fixing it in advance. DAEDAL [Li *et al.*, 2025c] starts from a short sequence, adjusts length based on End-Of-Sequence (EOS) token confidence, and then inserts masks at low-confidence positions for iterative refinement. dLLM-Var [Yang *et al.*, 2025d] enforces explicit EOS modeling by always masking the EOS token during training and conditionally appending new blocks at inference when no EOS is produced, enabling open-ended generation. Complementarily, ODB-dLLM [Wei *et al.*, 2025a] introduces adaptive length prediction to terminate decoding at the first high-confidence EOS.

Dynamic-Canvas and Any-Order Generation. This paradigm frames generation as iterative refinement on a dynamic canvas, where length emerges implicitly from insertion and editing rather than explicit length control. FlexMDM [Kim *et al.*, 2025b] decouples length from token prediction by first estimating insertion counts between existing positions and then filling the inserted masks via posterior denoising, enabling flexible-length generation under non-causal diffusion. Dream [Ye *et al.*, 2025] combines autoregressive initialization with diffusion-based infilling to support the generation and completion of text segments with arbitrary lengths. Extending this idea further, OneFlow [Nguyen *et al.*, 2025] generalizes dynamic-canvas refinement to multimodal through an edit-flow mechanism, enabling parallel, interleaved synthesis of variable-length text and image content without fixed-length allocation.

Discussion. These approaches make variable-length generation practical for efficient dLLMs in real-world settings. Future work should develop more robust length-adaptive decoding to maintain stable termination and coherence.

4.2 Long Context

Despite significant progress in AR LLM for long-context processing, dLLMs have historically struggled with context scal-

ability due to fundamental architectural mismatches [Dat *et al.*, 2024]. Recent advances have enabled context windows to expand from thousands to over 128K tokens while preserving generation quality, primarily through following techniques:

- **Position Encoding & Extrapolation:** LongLLaDA [Liu *et al.*, 2025b] presents the first systematic analysis of long-context extrapolation in dLLMs, showing that dLLMs degrade more gracefully beyond their training context and retain strong sensitivity to recent context segments. Based on this, it derives a training-free RoPE scaling rule by computing the critical dimension and scaling factor, enabling effective context extension without parameter modification. UltraLLaDA [He *et al.*, 2025] further extends the context window to 128K tokens through diffusion-aware NTK position encoding and a multi-document concatenation masking strategy during post-training.
- **Blockwise Diffusion:** To enable long-context generation without prohibitive computation, blockwise diffusion decomposes extended sequences into segments and performs parallel denoising within blocks while coordinating dependencies across blocks. Block Diffusion [Arriola *et al.*, 2025a] combines sequential block generation with intra-block parallel refinement to maintain coherence at scale, while Fast-dLLM v2 [Wu *et al.*, 2025a] further improves long-context efficiency through bidirectional block modeling and hierarchical caching. CtrlDiff [Huang and Tang, 2025] and AdaBlock-dLLM [Lu *et al.*, 2025] adapt block granularity during generation based on semantic or attention cues, allocating computation where long-range dependencies are most critical.
- **Memory optimization:** These techniques mitigate the quadratic attention bottleneck in long-sequence generation by reducing memory overhead. Rainbow Padding [Kim *et al.*, 2025a] tackles early termination in instruction-tuned dLLMs through cyclic multi-token padding patterns that prevent premature EOS prediction.

Discussion. Together, these advances establish a promising foundation for long-context dLLMs. However, fully realizing their practical benefits will require deeper co-design across acceleration techniques to maintain global coherence and superior performance under extreme context lengths.

5 System Framework

As diffusion language models move toward large-scale deployment, the focus has shifted from algorithmic feasibility to system-level efficiency and scalability. While vLLM [Kwon *et al.*, 2023] and SGLang [Zheng *et al.*, 2024] are widely utilized as industry standards for autoregressive LLM serving, the specialized ecosystem for dLLMs is still in its nascent stages and requires broader community support to match this maturity. Current progress indicates a trend toward integrating dLLM support into existing high-performance engines and developing dedicated frameworks:

- **Unified Training and Evaluation**: dllm-repo [Zhou *et al.*, 2025c] provides a comprehensive, research-oriented library that standardizes the full diffusion LLM lifecycle, including training, fine-tuning, and evaluation. It supports LoRA, DeepSpeed, and FSDP, while providing reference implementations for training algorithms such as masked diffusion, block diffusion, and edit flows. [Peng *et al.*, 2025] benchmark the throughput of dLLMs against AR models and provide insights into practical acceleration strategies for dLLMs.
- **Post-training Optimization**: DiRL [Zhu *et al.*, 2025c] addresses the critical training-inference mismatch in dLLMs by establishing an efficient post-training framework that supports scalable reinforcement learning via online policy updates and training-inference co-design.
- **Mainstream Serving Integration**: SGLang has recently introduced initial support for the LLaDA 2.0 series models. SGLang leverages its existing Chunked-Prefill pipeline to implement computational support for Block Diffusion LLM, enabling high-performance deployment of 16B and 100B parameter diffusion models.
- **Specialized Inference Engines**: dInfer [Ma *et al.*, 2025b] develops a modular inference pipeline that decouples model logic from diffusion iteration management, achieving a 10× speedup over baseline frameworks through the integration of CUDA Graphs and loop unrolling.
- **Production Serving**: To tackle the highly variable memory footprint of dLLM inference in production GPUs, dLLM-Serve [Fan *et al.*, 2025] employs Logit-Aware Activation Budgeting and a Phase-Multiplexed Scheduler to manage the massive memory overhead of logit tensors, effectively enhancing throughput in resource-constrained scenarios.

Discussion. Collectively, these frameworks form a growing ecosystem that addresses the full lifecycle of dLLM deployment, from optimized single-request latency to high-concurrency production serving and efficient alignment. However, as dLLMs move toward more complex non-autoregressive architectures and multimodal settings, there is a pressing need for specialized open-source frameworks that provide production-level robustness, portable performance optimizations across different hardware backends, and seamless integration with existing serving infrastructures.

6 Multimodal DLM Efficiency

Training Stage. To alleviate the inherent token-by-token decoding latency of autoregressive multimodal models [Zhou *et al.*, 2025a; Zhou *et al.*, 2025b; Lin *et al.*, 2025a], recent work has explored Multimodal Diffusion Language Model (MDLM) as a parallel and more controllable generation paradigm. Dimple [Yu *et al.*, 2025b] develops a MDLM by employing a hybrid "autoregressive-then-diffusion" training paradigm to stabilize convergence. Pushing block-wise efficiency, SDAR-VL [Cheng *et al.*, 2025b] introduces an asynchronous noise scheduling strategy, demonstrating that block-wise discrete diffusion can outperform strong autoregressive baselines in vision-language understanding tasks.

Inference Stage. To address the computational intensity of multimodal denoising, recent works propose acceleration from algorithmic perspectives. D^3ToM [Chang *et al.*, 2025] dynamically merges redundant visual tokens via "decider-guided" importance maps, while Sparse-LaViDa [Li *et al.*,

2025f] truncates unnecessary masked tokens by utilizing specialized register tokens for compact representation. Complementing these, [Li *et al.*, 2025a] identifies a progressive information recovery mechanism in discrete diffusion, providing a theoretical basis for aggressive token pruning without sacrificing fidelity. Furthermore, Dimple optimizes the inference pipeline through "confident decoding," which dynamically adjusts the number of tokens generated per step to reduce iterations by two-thirds, and re-incorporates the prefilling mechanism from AR models to jointly minimize generation complexity. Lumina-DiMOO [Xin *et al.*, 2025] applies dLLM efficiency to multi-modal generation using a "Max-Logit" cache, freezing representations of tokens with high prediction confidence to accelerate image-text synthesis.

Discussion. Together, these advancements transition multimodal dLLMs from theoretical exploration to high-performance, sparse-inference systems. However, the rapid development of MDLM still necessitates multi-level optimizations, spanning from data-centric noise scheduling to hardware-aware kernel designs, to fully unlock their potential in real-time, large-scale multimodal applications. This remains a fertile ground for future research into unified, cross-modal diffusion architectures.

7 Future Research Directions

Training Efficiency. Future work should improve the data efficiency of AR-to-diffusion transfer and move beyond naive initialization. Designing curricula and data structures that fully exploit bidirectional conditioning, especially data with strong long-range dependencies, may help dLLMs surpass AR models in global reasoning. Moreover, training-inference co-design remains crucial: incorporating efficient decoding structures (e.g., learned schedules, block-wise refinement, cache-friendly attention) directly into training could significantly reduce the need for serving stage optimization.

Inference Optimization. A key challenge still lies in developing a unified paradigm for the throughput–quality tradeoff in parallel decoding, together with diffusion-aware cache management. Unlike AR serving, which benefits from mature primitives such as PageAttention and FlashAttention, dLLMs still lack standardized abstractions for parallel sampling and KV cache. Another important direction is hardware-aligned compression and kernel optimization. Attaining wall-clock speedups comparable to AR engines requires a full-stack orchestration of algorithmic parallelization, system-level memory management, and customized Triton/CUDA kernels

Context and Framework. Bridging the context scalability gap between dLLMs and modern AR models (which can reach 1M-token windows) requires mitigating the quadratic cost of bidirectional attention while preserving global coherence under parallel denoising. Blockwise diffusion is a promising direction, but it still requires more effective memory management and stable decoding strategies to remain efficient as sequence lengths continue to grow. In addition, the dLLM ecosystem still lacks mature open-source frameworks covering pretraining, post-training, and production serving. Building robust, reusable tooling will be essential for making dLLMs practically deployable in real-world applications.

Multimodal DLM. Discrete diffusion-based models provide a unified generation paradigm that naturally extends from text to multimodal understanding and generation [Luo *et al.*, 2025; Li *et al.*, 2025e; Shi *et al.*, 2025]. A critical frontier involves enhancing spatial-semantic alignment in unified MDLMs to support high-fidelity perception tasks, such as document parsing and precise visual grounding. Reinforcement learning is also promising, as discrete tokens enable token-level rewards to better optimize quality and controllability [Yang *et al.*, 2025c; Xin *et al.*, 2025].

8 Conclusion

This survey provides a comprehensive overview of existing research efforts toward improving the efficiency of dLLMs. To capture efficiency bottlenecks across the dLLM pipeline, from pretraining and decoding to long-context scalability and multimodal extension, we organize the literature into five categories, summarize representative methods, and discuss their practical implications. Finally, we highlight open challenges and future directions, with the hope of inspiring further progress toward efficient and widely deployable dLLMs.

References

[Agrawal *et al.*, 2025] Sudhanshu Agrawal, Risheek Garrepalli, et al. Spiffy: Multiplying diffusion llm acceleration via lossless speculative decoding. *arXiv preprint arXiv:2509.18085*, 2025.

[Arriola *et al.*, 2025a] Marianne Arriola, Aaron Gokaslan, et al. Block diffusion: Interpolating between autoregressive and diffusion language models. In *ICLR*, 2025.

[Arriola *et al.*, 2025b] Marianne Arriola, Yair Schiff, et al. Encoder-decoder diffusion language models for efficient training and inference. *ArXiv, abs/2510.22852*, 2025.

[Bansal and Sanghavi, 2025] Parikshit Bansal and Sujay Sanghavi. Enabling approximate joint sampling in diffusion lms. *arXiv preprint arXiv:2509.22738*, 2025.

[Bao *et al.*, 2025] Wenrui Bao, Zhiben Chen, et al. Learning to parallel: Accelerating diffusion large language models via learnable parallel decoding. *arXiv preprint arXiv:2509.25188*, 2025.

[Ben-Hamu *et al.*, 2025] Heli Ben-Hamu, Itai Gat, et al. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *arXiv preprint arXiv:2505.24857*, 2025.

[Bie *et al.*, 2025] Tiwei Bie, Maosong Cao, et al. Llada2. 0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

[Chandrasegaran *et al.*, 2025] Keshigeyan Chandrasegaran, Armin W. Thomas, et al. Rnd1: Simple, scalable ar-to-diffusion conversion. Oct 2025.

[Chang *et al.*, 2025] Shuochen Chang, Xiaofeng Zhang, et al. D3tom: Decider-guided dynamic token merging for accelerating diffusion mllms. *arXiv preprint arXiv:2511.12280*, 2025.

[Chen and Liu, 2026] Jian Chen and Zhijian Liu. Dflash: Block diffusion for flash speculative decoding, 2026.

[Chen *et al.*, 2023] Charlie Chen, Sebastian Borgeaud, et al. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

[Chen *et al.*, 2025a] Tianqi Chen, Shujian Zhang, et al. Dlm-one: Diffusion language models for one-step sequence generation. *arXiv preprint arXiv:2506.00290*, 2025.

[Chen *et al.*, 2025b] Zigeng Chen, Gongfan Fang, et al. dparallel: Learnable parallel decoding for dllms. *arXiv preprint arXiv:2509.26488*, 2025.

[Cheng *et al.*, 2025a] Shuang Cheng, Yihan Bian, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*, 2025.

[Cheng *et al.*, 2025b] Shuang Cheng, Yuhua Jiang, et al. Sdar-vl: Stable and efficient block-wise diffusion for vision-language understanding. *arXiv preprint arXiv:2512.14068*, 2025.

[Cheng *et al.*, 2025c] Zicong Cheng, Guo-Wei Yang, et al. Deer: Draft with diffusion, verify with autoregressive models. *arXiv preprint arXiv:2512.15176*, 2025.

[Dat *et al.*, 2024] Do Huu Dat, Duc Anh Do, et al. Discrete diffusion language model for efficient text summarization. In *North American Chapter of the Association for Computational Linguistics*, 2024.

[Fan *et al.*, 2025] Jiakun Fan, Yanglin Zhang, et al. Taming the memory footprint crisis: System design for production diffusion llm serving. *arXiv preprint arXiv:2512.17077*, 2025.

[Fu *et al.*, 2025a] Feiyang Fu, Tongxian Guo, et al. Learnable sampler distillation for discrete diffusion models. *arXiv preprint arXiv:2509.19962*, 2025.

[Fu *et al.*, 2025b] Yonggan Fu, Lexington Whalen, et al. Efficient-dlm: From autoregressive to diffusion language models, and beyond in speed, 2025.

[Gao *et al.*, 2025] Yifeng Gao, Ziang Ji, et al. Self speculative decoding for diffusion large language models. *arXiv preprint arXiv:2510.04147*, 2025.

[Gong *et al.*, 2022] Shansan Gong, Mukai Li, et al. Diffuseq: Sequence to sequence text generation with diffusion models. *ArXiv*, abs/2210.08933, 2022.

[Gong *et al.*, 2025] Shansan Gong, Shivam Agarwal, et al. Scaling diffusion language models via adaptation from autoregressive models. In *ICLR*, 2025.

[Gwak *et al.*, 2025] Daehoon Gwak, Minseo Jung, et al. Reward-weighted sampling: Enhancing non-autoregressive characteristics in masked diffusion llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

[Han *et al.*, 2023] Xiaochuang Han, Sachin Kumar, et al. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11575–11596, 2023.

[Hao *et al.*, 2025a] Jitai Hao, Qiang Huang, et al. A token is worth over 1,000 tokens: Efficient knowledge distillation through low-rank clone. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[Hao *et al.*, 2025b] Jitai Hao, Yuke Zhu, et al. Omnikv: Dynamic context selection for efficient long-context llms. In *The Thirteenth International Conference on Learning Representations*, 2025.

[He *et al.*, 2025] Guangxin He, Shen Nie, et al. Ultrallada: Scaling the context length to 128k for diffusion large language models. *ArXiv*, abs/2510.10481, 2025.

[Hong *et al.*, 2025] Feng Hong, Geng Yu, et al. Wide-in, narrow-out: Revokable decoding for efficient and effective dllms. *arXiv preprint arXiv:2507.18578*, 2025.

[Huang and Tang, 2025] Chihan Huang and Hao Tang. Ctrldiff: Boosting large diffusion language models with dynamic block prediction and controllable generation. *ArXiv*, abs/2505.14455, 2025.

[Huang *et al.*, 2025a] Hong Huang, Decheng Wu, et al. Tequila: Trapping-free ternary quantization for large language models. *arXiv preprint arXiv:2509.23809*, 2025.

[Huang *et al.*, 2025b] Jianuo Huang, Yaojie Zhang, et al. Mask tokens as prophet: Fine-grained cache eviction for efficient dllm inference. *arXiv preprint arXiv:2510.09309*, 2025.

[Huang *et al.*, 2026] Hong Huang, Decheng Wu, et al. Sherry: Hardware-efficient 1.25-bit ternary quantization via fine-grained sparsification. *arXiv preprint arXiv:2601.07892*, 2026.

[Israel *et al.*, 2025] Daniel Israel, Guy Van den Broeck, et al. Accelerating diffusion llms via adaptive parallel decoding. *arXiv preprint arXiv:2506.00413*, 2025.

[Jazbec *et al.*, 2025] Metod Jazbec, Theo X Olausson, et al. Learning unmasking policies for diffusion language models. *arXiv preprint arXiv:2512.09106*, 2025.

[Jiang *et al.*, 2025] Yuchu Jiang, Yue Cai, et al. d2 cache: Accelerating diffusion-based llms via dual adaptive caching. *arXiv preprint arXiv:2509.23094*, 2025.

[Kim *et al.*, 2025a] Bumjun Kim, Dongjae Jeon, et al. Rainbow padding: Mitigating early termination in instruction-tuned diffusion llms. *ArXiv*, abs/2510.03680, 2025.

[Kim *et al.*, 2025b] Jaeyeon Kim, Cheuk Kit Lee, et al. Any-order flexible length masked diffusion. *ArXiv*, abs/2509.01025, 2025.

[Kong *et al.*, 2025] Fanheng Kong, Jingyuan Zhang, et al. Accelerating diffusion llm inference via local determinism propagation. *arXiv preprint arXiv:2510.07081*, 2025.

[Kwon *et al.*, 2023] Woosuk Kwon, Zhuohan Li, et al. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[Li *et al.*, 2024] Yuhui Li, Fangyun Wei, et al. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024.

[Li *et al.*, 2025a] Duo Li, Zuhao Yang, et al. A comprehensive study on visual token redundancy for discrete diffusion-based multimodal large language models. *arXiv preprint arXiv:2511.15098*, 2025.

[Li *et al.*, 2025b] Guanghao Li, Zhihui Fu, et al. Diffuspec: Unlocking diffusion language models for speculative decoding. *arXiv preprint arXiv:2510.02358*, 2025.

[Li *et al.*, 2025c] Jinsong Li, Xiao wen Dong, et al. Beyond fixed: Training-free variable-length denoising for diffusion large language models. *ArXiv*, abs/2508.00819, 2025.

[Li *et al.*, 2025d] Pengxiang Li, Yefan Zhou, et al. Diffusion language models know the answer before decoding. *arXiv preprint arXiv:2508.19982*, 2025.

[Li *et al.*, 2025e] Shufan Li, Jiuxiang Gu, et al. Lavida-o: Elastic large masked diffusion models for unified multimodal understanding and generation. *arXiv preprint arXiv:2509.19244*, 2025.

[Li *et al.*, 2025f] Shufan Li, Jiuxiang Gu, et al. Sparselavida: Sparse multimodal discrete diffusion language models. *arXiv preprint arXiv:2512.14008*, 2025.

[Li *et al.*, 2025g] Tianyi Li, Mingda Chen, et al. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025.

[Liang *et al.*, 2026] Yihao Liang, Ze Wang, et al. Cd4lm: Consistency distillation and adaptive decoding for diffusion language models. *arXiv*, 2026.

[Lin *et al.*, 2024a] Haokun Lin, Haoli Bai, et al. Mope-clip: Structured pruning for efficient vision-language models with module-wise pruning error metric. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 27370–27380, 2024.

[Lin *et al.*, 2024b] Haokun Lin, Haobo Xu, et al. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37:87766–87800, 2024.

[Lin *et al.*, 2025a] Haokun Lin, Teng Wang, et al. Toklip: Marry visual tokens to clip for multimodal comprehension and generation. *arXiv preprint arXiv:2505.05422*, 2025.

[Lin *et al.*, 2025b] Haokun Lin, Haobo Xu, et al. Quantization meets dllms: A systematic study of post-training quantization for diffusion llms. *arXiv preprint arXiv:2508.14896*, 2025.

[Liu *et al.*, 2024] Ruikang Liu, Haoli Bai, et al. Intactkv: Improving large language model quantization by keeping pivot tokens intact. *arXiv preprint arXiv:2403.01241*, 2024.

[Liu *et al.*, 2025a] Aiwei Liu, Minghua He, et al. Wedlm: Reconciling diffusion language models with standard causal attention for fast inference. *arXiv preprint arXiv:2512.22737*, 2025.

[Liu *et al.*, 2025b] Xiaoran Liu, Zhigeng Liu, et al. Longllada: Unlocking long context capabilities in diffusion llms. *ArXiv*, abs/2506.14429, 2025.

[Liu *et al.*, 2025c] Yangzhou Liu, Yue Cao, et al. Sequential diffusion language models, 2025.

[Liu *et al.*, 2025d] Zhiyuan Liu, Yicun Yang, et al. dllm-cache: Accelerating diffusion large language models with adaptive caching. *arXiv preprint arXiv:2506.06295*, 2025.

[Lu *et al.*, 2025] Guanxi Lu, Hao Mark Chen, et al. Adablock-dllm: Semantic-aware diffusion llm inference via adaptive block size. *ArXiv*, abs/2509.26432, 2025.

[Luo *et al.*, 2025] Run Luo, Xiaobo Xia, et al. Next-omni: Towards any-to-any omnimodal foundation models with discrete flow matching. *arXiv preprint arXiv:2510.13721*, 2025.

[Ma *et al.*, 2025a] Xinyin Ma, Runpeng Yu, et al. dkv-cache: The cache for diffusion language models. *arXiv preprint arXiv:2505.15781*, 2025.

[Ma *et al.*, 2025b] Yuxin Ma, Lun Du, et al. dinfer: An efficient inference framework for diffusion language models. *arXiv preprint arXiv:2510.08666*, 2025.

[Mohamed *et al.*, 2025] Amr Mohamed, Yang Zhang, et al. Fast-decoding diffusion language models via progress-aware confidence schedules. *arXiv preprint arXiv:2512.02892*, 2025.

[Monsefi *et al.*, 2025] Amin Karimi Monsefi, Nikhil Bhendawade, et al. Fs-dfm: Fast and accurate long text generation with few-step diffusion language models. *arXiv preprint arXiv:2509.20624*, 2025.

[Nguyen *et al.*, 2025] John Nguyen, Marton Havasi, et al. Oneflow: Concurrent mixed-modal and interleaved generation with edit flows. *ArXiv*, abs/2510.03506, 2025.

[Nguyen-Tri *et al.*, 2025] Quan Nguyen-Tri, Mukul Ranjan, et al. Attention is all you need for kv cache in diffusion llms. *arXiv preprint arXiv:2510.14973*, 2025.

[Nie *et al.*, 2025] Shen Nie, Fengqi Zhu, et al. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

[Peng *et al.*, 2025] Han Peng, Peiyu Liu, et al. How efficient are diffusion language models? a critical examination of efficiency evaluation practices. *arXiv preprint arXiv:2510.18480*, 2025.

[Shen *et al.*, 2025] Jucheng Shen, Gaurav Sarkar, et al. Improving the throughput of diffusion-based large language models via a training-free confidence-aware calibration. *arXiv preprint arXiv:2512.07173*, 2025.

[Shi *et al.*, 2025] Qingyu Shi, Jinbin Bai, et al. Muddit: Liberating generation beyond text-to-image with a unified discrete diffusion model. *arXiv preprint arXiv:2505.23606*, 2025.

[Song *et al.*, 2025a] Yuerong Song, Xiaoran Liu, et al. Sparse-dllm: Accelerating diffusion llms with dynamic cache eviction. *arXiv preprint arXiv:2508.02558*, 2025.

[Song *et al.*, 2025b] Yuxuan Song, Zheng Zhang, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

[Tian *et al.*, 2025] Yuchuan Tian, Yuchen Liang, et al. From next-token to next-block: A principled adaptation path for diffusion llms. *arXiv preprint arXiv:2512.06776*, 2025.

[Wei *et al.*, 2025a] Linye Wei, Wenjue Chen, et al. Orchestrating dual-boundaries: An arithmetic intensity inspired acceleration framework for diffusion language models. *arXiv preprint arXiv:2511.21759*, 2025.

[Wei *et al.*, 2025b] Qingyan Wei, Yaojie Zhang, et al. Accelerating diffusion large language models with slowfast: The three golden principles. *arXiv preprint arXiv:2506.10848*, 2025.

[Wu and Zhang, 2025] Shutong Wu and Jiawei Zhang. Free draft-and-verification: Toward lossless parallel decoding for diffusion large language models. *arXiv preprint arXiv:2510.00294*, 2025.

[Wu *et al.*, 2025a] Chengyue Wu, Hao Zhang, et al. Fast-dllm v2: Efficient block-diffusion llm. *arXiv preprint arXiv:2509.26328*, 2025.

[Wu *et al.*, 2025b] Chengyue Wu, Hao Zhang, et al. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025.

[Xin *et al.*, 2025] Yi Xin, Qi Qin, et al. Lumina-dimoo: An omni diffusion large language model for multi-modal generation and understanding. *arXiv preprint arXiv:2510.06308*, 2025.

[Xing *et al.*, 2025] Xingrun Xing, Zheng Liu, et al. Efficientllm: Scalable pruning-aware pretraining for architecture-agnostic edge language models. *arXiv preprint arXiv:2502.06663*, 2025.

[Xu and Yang, 2025] Chen Xu and Dawei Yang. Dllmquant: Quantizing diffusion-based large language models. *arXiv preprint arXiv:2508.14090*, 2025.

[Xu *et al.*, 2025] Chenkai Xu, Yijie Jin, et al. Lopa: Scaling dllm inference via lookahead parallel decoding. *arXiv preprint arXiv:2512.16229*, 2025.

[Yang *et al.*, 2024] Lianwei Yang, Haisong Gong, et al. Dopq-vit: Towards distribution-friendly and outlier-aware post-training quantization for vision transformers. *arXiv preprint arXiv:2408.03291*, 2024.

[Yang *et al.*, 2025a] An Yang, Anfeng Li, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[Yang *et al.*, 2025b] Lianwei Yang, Haokun Lin, et al. Lrq-dit: Log-rotation post-training quantization of diffusion transformers for image and video generation. *arXiv preprint arXiv:2508.03485*, 2025.

[Yang *et al.*, 2025c] Ling Yang, Ye Tian, et al. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025.

[Yang *et al.*, 2025d] Yicun Yang, Cong Wang, et al. Diffusion llm with native variable generation lengths: Let [eos] lead the way. *ArXiv*, abs/2510.24605, 2025.

[Ye *et al.*, 2025] Jiacheng Ye, Zhihui Xie, et al. Dream 7b: Diffusion large language models. *ArXiv*, abs/2508.15487, 2025.

[You *et al.*, 2025] Zebin You, Shen Nie, et al. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025.

[Yu *et al.*, 2025a] Runpeng Yu, Qi Li, et al. Discrete diffusion in large language and multimodal models: A survey, 2025.

[Yu *et al.*, 2025b] Runpeng Yu, Xinyin Ma, et al. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025.

[Zhang *et al.*, 2024] Yingtao Zhang, Haoli Bai, et al. Plug-and-play: An efficient post-training pruning method for large language models. 2024.

[Zhang *et al.*, 2025] Tianao Zhang, Zhiteng Li, et al. Quant-dllm: Post-training extreme low-bit quantization for diffusion large language models. *arXiv preprint arXiv:2510.03274*, 2025.

[Zheng *et al.*, 2024] Lianmin Zheng, Liangsheng Yin, et al. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37:62557–62583, 2024.

[Zhou *et al.*, 2025a] Yinan Zhou, Yuxin Chen, et al. Dogr: Towards versatile visual document grounding and referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3596–3606, 2025.

[Zhou *et al.*, 2025b] Yinan Zhou, Yaxiong Wang, et al. Scale up composed image retrieval learning via modification text generation. *arXiv preprint arXiv:2504.05316*, 2025.

[Zhou *et al.*, 2025c] Zhanhui Zhou, Lingjie Chen, et al. dllm: Simple diffusion language modeling, 2025.

[Zhu *et al.*, 2025a] Fengqi Zhu, Rongzhen Wang, et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025.

[Zhu *et al.*, 2025b] Fengqi Zhu, Zebin You, et al. Llada-moe: A sparse moe diffusion language model. *arXiv preprint arXiv:2509.24389*, 2025.

[Zhu *et al.*, 2025c] Ying Zhu, Jiaxin Wan, et al. Dirl: An efficient post-training framework for diffusion language models. *arXiv preprint arXiv:2512.22234*, 2025.