Diffusion Language Models for Mobile Edge Agentic AI: Foundations, Applications, and Challenges

Chenqi Li[1], Minghui Min[1*], Dusit Niyato[2], Wei Ni[3]

[1*]School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, 221116, Jiangsu, China .
[2]College of Computing and Data Science, Nanyang Technological University, Singapore, 639798, Singapore .
[3]School of Engineering, Edith Cowan University, Perth, Western Australia, 6027, Australia .

*Corresponding author(s). E-mail(s): minmh@cumt.edu.cn;
Contributing authors: chenqi.li@cumt.edu.cn; dniyato@ntu.edu.sg;
wei.ni@ieee.org;

Abstract

Diffusion language models (DLMs) offer a non-autoregressive alternative for mobile edge agentic artificial intelligence (AI) by refining tokens through iterative denoising rather than left-to-right decoding. Compared with autoregressive Transformer-based large language models (LLMs), DLMs can update multiple uncertain tokens in parallel and exploit bidirectional context throughout the generation process, enabling more flexible quality-latency trade-offs beyond fixed sequential decoding. These properties are particularly attractive for edge agents, where partial refinement, early exit, and constraint-guided correction can reduce response delay and communication overhead while improving robustness under noisy, incomplete, or dynamic contexts. This survey reviews DLM foundations and analyzes their suitability for edge settings under latency, memory, energy, bandwidth, privacy, and reliability constraints. We cover resource-efficient architectures, training and inference acceleration, compression, edge/cloud deployment, communication-aware serving, Internet of Things (IoT)/wireless applications, and evaluation of DLM-based agents. We further discuss open issues in long-context state management, split inference, trustworthy execution, multimodal grounding, and reproducible benchmarking. The goal is to connect DLM modeling properties, including bidirectionality, parallel refinement,

controllability, and quality-latency elasticity, with system-level requirements of future mobile edge intelligence.

Keywords: Diffusion Language Models, Mobile Edge Agentic AI, Edge Intelligence, Resource-Efficient Inference, Distributed Inference, Communication-Aware Serving

1 Introduction

In recent years, artificial intelligence (AI) has been increasingly incorporated into the Internet of Things (IoT). The proliferation of smart devices, edge sensors, and cyber–physical infrastructures has generated large volumes of heterogeneous data that require intelligent processing and decision-making capabilities. At the same time, large language models (LLMs) have shown strong capabilities in natural language understanding (NLU), reasoning, and multimodal interaction Xiao et al. (2025); Saheed and Chukwuere (2026). Integrating such capabilities into IoT environments may help future IoT systems become more autonomous, adaptive, and context-aware Karunanayake et al. (2025). However, the use of LLMs in real-time IoT applications remains constrained by resource limitations Xiao et al. (2025); Saheed and Chukwuere (2026). In addition, privacy and security requirements may prevent institutions from accessing commercial state-of-the-art LLM services, requiring local deployment within private networks Xiao et al. (2025). Thus, deploying LLMs directly on resource-constrained mobile edge IoT devices remains challenging, making LLM efficiency optimization a core research topic for mobile edge deployment Xiao et al. (2025); Saheed and Chukwuere (2026). This highlights the growing need for adaptive and resilient systems that can operate efficiently at the mobile edge Karunanayake et al. (2025).

However, realizing this vision at the mobile edge remains nontrivial because modern generative models impose substantial computational, memory, and communication requirements. For example, GPT-4 has been estimated to contain approximately 1.76 trillion parameters Wang et al. (2025a), a scale that far exceeds the hardware capacities of typical edge nodes Zheng et al. (2025). Most current LLMs rely on autoregressive architectures that generate tokens sequentially from left to right. Although this paradigm excels in natural language processing (NLP), it introduces inherent inefficiencies that limit its applicability in resource-constrained IoT environments. The sequential nature of autoregressive decoding leads to high inference latency and poor parallelization efficiency, presenting a major bottleneck for latency-sensitive edge applications Shi et al. (2026). Furthermore, the intensive memory access patterns of these models place significant pressure on both on-device resources and wireless communication infrastructures. Specifically, processing long inputs, such as the 128,000-token context window supported by advanced GPT models, requires the maintenance of large Key-Value (KV) caches Sun et al. (2026). This operation can consume gigabytes (GB) of random-access memory (RAM) during inference, making deployment on memory-limited IoT devices difficult Zheng et al. (2025); Sun et al.

Table 1 Representative LLM paradigms.

Model	Paradigm	Scale	Backbone	Inference Pattern / Edge Bottleneck
Autoregressive Transformer-based LLMs				
GPT-3 Brown et al. (2020)	Autoregressive LM	175B	Transformer decoder with sparse attention	Sequential next-token generation; KV cache grows with context.
GPT-4 OpenAI et al. (2024)	Autoregressive multimodal LM	1.76T	Transformer-style model with closed technical details	Frontier-scale capability; unsuitable for direct edge deployment.
Llama 3.1 Meta AI (2024)	Autoregressive LM	8B / 70B / 405B	Optimized Transformer with GQA and 128K context	Open-weight autoregressive baseline; long-context inference incurs substantial KV-cache pressure.
Mistral 7B Jiang et al. (2023)	Efficient autoregressive LM	7B	Transformer with GQA, SWA, and rolling KV cache	More deployment-friendly than larger autoregressive models, but generation remains token-by-token.
Mixtral 8×7B Jiang et al. (2024)	Sparse MoE autoregressive LM	47B total / 13B active	Transformer MoE with top-2 expert routing	Reduces active compute per token, but requires large weight storage and routing overhead.
Diffusion-based Language Models				
Diffusion-LM Li et al. (2022)	Continuous diffusion LM	Task-scale	Transformer-based denoiser over continuous word vectors	Non-left-to-right refinement; useful for control and infilling, but not yet a general large-scale LLM.
LLaDA 8B Nie et al. (2025)	Masked diffusion LM	8B	Transformer mask predictor with full attention	Predicts multiple masked tokens in parallel; avoids causal KV decoding but requires iterative denoising.
Diffusion Gemma 26B-A4B Google (2026a,b)	Masked discrete diffusion MoE LM	26B total / about 4B active	Gemma-family MoE backbone with block-diffusion decoding	Open-weight DLM checkpoint; parallel refinement can improve local GPU throughput, but iterative denoising and model size remain edge bottlenecks.
Dream 7B Ye et al. (2025)	Discrete diffusion LLM	7B	Transformer-based DLM with autoregressive initialization	Parallel iterative denoising with arbitrary-order generation and tunable quality–speed trade-offs.

Note: **B**: Billion (parameters); **GQA**: Grouped-Query Attention; **GPU**: Graphics Processing Unit; **SWA**: Sliding Window Attention; **MoE**: Mixture-of-Experts.

(2026). As IoT systems typically operate under stringent constraints regarding computational capability, energy availability, and network bandwidth, exploring alternative generative paradigms that align with edge computing architectures is important Zheng et al. (2025).

Table 1 provides a qualitative comparison of autoregressive LLMs, diffusion models (DMs), and diffusion language models (DLMs) from the perspective of modeling mechanisms. To complement this paradigm-level discussion, Table 2 summarizes representative quantitative evidence reported in recent technical papers. These results suggest that diffusion-based LLMs have begun to approach autoregressive LLMs on general reasoning and code benchmarks, while showing particular advantages in planning-oriented tasks and emerging faster-than-autoregressive inference settings.

Table 2 Reported quantitative evidence comparing autoregressive LLMs and diffusion-based LLMs.

Study	Compared Models	Evaluation Focus	Reported Results	Implication
Nie et al. (2025)	**LLaDA 8B Base** (DLM) vs. **Llama3 8B Base** (autoregressive)	General, math, and code benchmarks	**MMLU**: 65.9 vs. 65.4 **GSM8K**: 70.3 vs. 48.7 **MATH**: 31.4 vs. 16.0 **HumanEval**: 35.4 vs. 34.8 **HumanEval-FIM**: 73.8 vs. 73.3	DLMs can reach performance comparable to similarly sized autoregressive LLMs on several standard benchmarks.
Ye et al. (2025)	**Dream 7B** (DLM) vs. **Qwen2.5 7B** / **Llama3 8B** (autoregressive)	Math reasoning and code generation	**GSM8K**: 77.2 vs. 78.9 / 55.3 **MATH**: 39.6 vs. 41.1 / 18.0 **HumanEval**: 57.9 vs. 56.7 / 35.4	DLMs can remain competitive with strong autoregressive baselines on reasoning and code tasks.
Ye et al. (2025)	**Dream 7B** (DLM) vs. **Qwen2.5 7B** / **Llama3 8B** (autoregressive)	Planning-oriented tasks	**Countdown**: 16.0 vs. 6.2 / 3.7 **Sudoku**: 81.0 vs. 21.0 / 0.0 **Trip planning**: 17.8 vs. 3.6 / 8.7	Bidirectional refinement may benefit constraint satisfaction and long-horizon planning.
Wang et al. (2025c)	**D2F-Dream-Base-7B** / **D2F-LLaDA-Instruct-8B** vs. **autoregressive baselines**	Inference throughput	**GSM8K**: 119.9 vs. 48.0 / 52.7 tokens/s **HumanEval**: 81.6 vs. 50.9 / 55.4 tokens/s	Recent DLM acceleration methods can enter a faster-than-autoregressive inference regime.

Note: The numbers are reported from the original papers under their respective evaluation settings. Therefore, the table is intended as a compact literature-level summary rather than a controlled unified benchmark. **MMLU**: Massive Multitask Language Understanding; **GSM8K**: Grade School Math 8K; **FIM**: fill-in-the-middle.

DLMs have emerged as a non-autoregressive alternative to conventional autoregressive frameworks for edge intelligence Li et al. (2022); Sahoo et al. (2024). As shown in Fig. 1, research on DLMs has grown rapidly in recent years, especially since 2024, driven by mobile agentic AI and the need to migrate generative capabilities to resource-constrained edge environments. The trend serves as an indicative literature-level signal since the three sources differ in indexing scope and update policy.

Unlike token-by-token autoregressive generation, DLMs synthesize text through iterative denoising from a known prior, such as Gaussian noise in embedding space or an absorbing [MASK] state. This process updates multiple tokens in parallel and incorporates bidirectional context during generation. For edge intelligence, these properties enable better hardware utilization, flexible compute-quality trade-offs, inference-time controllability, and distributed inference across cloud-edge-device hierarchies Yang et al. (2025b). They also align with decentralized IoT systems, where diffusion-based solution generators can support complex optimization in Multi-access Edge Computing (MEC) networks Liang et al. (2025a).

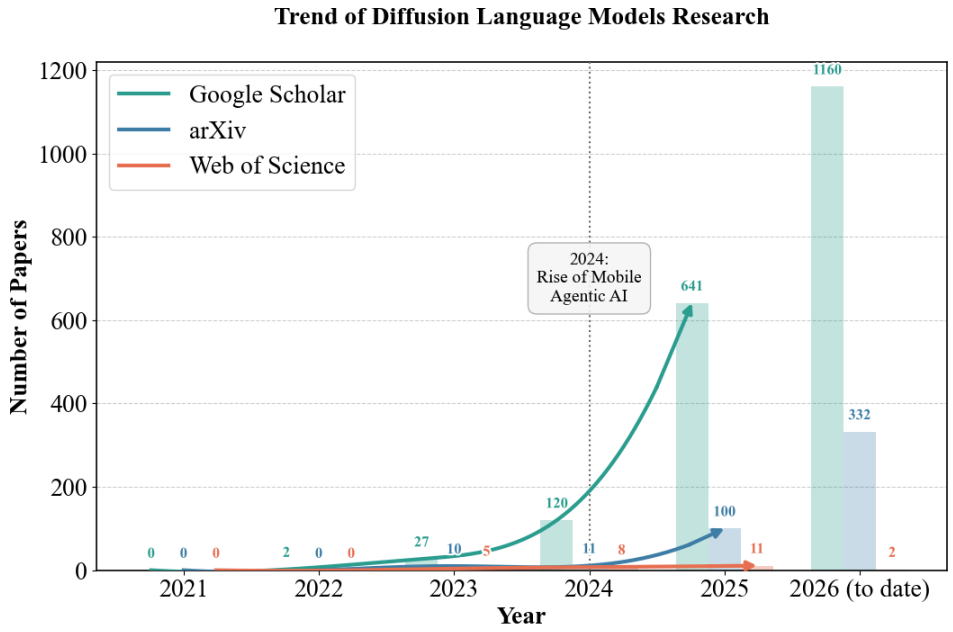


Fig. 1 Research trends of DLMs across Google Scholar, arXiv, and Web of Science

Distributed DLMs are suitable for mobile edge computing because their iterative denoising trajectory can be partitioned across device-edge-cloud tiers, reducing the local memory and computation burden of standalone mobile or IoT devices Sung et al. (2025). Their fixed-length latent states are also more predictable than token-by-token autoregressive synchronization with expanding KV caches, which can reduce bandwidth overhead and transmission latency in device-edge collaboration Du et al. (2024). Moreover, DLMs support anytime decoding, allowing systems to adjust denoising depth and offloading ratios according to network conditions, latency targets, and battery status Miles et al. (2026). By keeping initial noising and final decoding on devices while offloading only obfuscated intermediate states, distributed DLMs may further support privacy-preserving collaborative generation Allmendinger et al. (2026).

Existing diffusion-related surveys can be broadly grouped into three categories, as summarized in Table 3. A first line of work reviews general DM foundations, model variants, and broad applications. For instance, Croitoru et al. (2023) focus on denoising DMs in computer vision, covering denoising diffusion probabilistic models (DDPMs), noise conditional score networks (NCSNs), stochastic differential equations (SDEs), and vision tasks such as image generation, super-resolution, inpainting, and editing. Yang et al. (2023a) provide a broader methodological survey of DMs, emphasizing efficient sampling, likelihood estimation, structured data, and applications across vision, NLP, temporal modeling, and scientific domains. Ahsan et al. (2025) further summarize DM foundations and cross-domain applications, including

image synthesis, audio generation, healthcare, time-series forecasting, anomaly detection, and molecular dynamics. These surveys provide foundations for general diffusion modeling. Yet, they do not study DLMs as edge-deployable backbones for mobile agents.

A more recent line of work specifically investigates DLMs and discrete diffusion large models. In particular, Li et al. (2025f) review continuous, discrete, and hybrid DLMs, together with training, inference optimization, caching, multimodal extensions, and applications. Yu et al. (2025a) focus on discrete diffusion large language and multimodal models, covering mathematical formulations, representative diffusion large language models (dLLMs) and diffusion multimodal large language models (dMLLMs), training, inference, quantization, privacy, safety, and applications. Although these works establish important DLM taxonomies, they do not systematically connect DLM structures with mobile edge deployment, communication-computation cooperation, IoT/wireless applications, or agent-oriented evaluation.

Another related line of work studies DMs in networks, wireless systems, and semantic communications. For example, Luong et al. (2025) review applications of DMs in future networks, including channel modeling, signal reconstruction, integrated sensing and communication (ISAC), edge resource management, semantic communications, and security. Qin et al. (2026) focus on DMs for generative semantic communications, emphasizing conditional diffusion, efficient diffusion, generalized diffusion, and semantic decoding as an inverse problem. Fan et al. (2026) survey generative diffusion models (GDMs) for wireless networks from sensing, transmission, application, and security perspectives.

Distinct from these works, this current survey centers on DLMs rather than continuous DMs/GDMs, and investigates how DLM properties—bidirectional context, parallel refinement, controllability, and quality-latency elasticity—can support mobile edge agentic AI under latency, memory, energy, bandwidth, privacy, and reliability constraints. Although DLMs have attracted growing attention, most studies still focus on model design, generation quality, and inference acceleration, while their role in mobile edge agentic AI remains underexplored. Mobile edge and IoT systems require low latency, small memory and computing footprint, energy efficiency, bandwidth awareness, privacy, reliability, and closed-loop interaction, calling for a system-level view beyond model-centric progress. The main contributions of this survey are summarized as follows:

- **Edge-oriented view of DLMs.** We frame DLMs as an edge-oriented generative paradigm for mobile agents by connecting their iterative denoising, bidirectional context modeling, parallel token refinement, controllable generation, and quality-latency elasticity with the requirements of mobile edge intelligence.
- **Unified review of foundations and efficiency techniques.** We review the modeling foundations of DLMs and summarize resource-efficient techniques for lightweight architectures, efficient training, accelerated inference, and model compression.

Table 3 Comparison between this survey and representative diffusion-related surveys.

Survey	Main Focus	Uncovered Aspects
Croitoru et al. (2023)	Denoising DMs in computer vision, including DDPMs, NCSNs, SDEs, relations to other generative models, and vision tasks such as generation, super-resolution, inpainting, editing, and translation.	Lacks DLM-specific analysis and does not consider edge deployment, IoT/wireless constraints, or agentic execution.
Yang et al. (2023a)	General DM methods and applications, including efficient sampling, likelihood estimation, structured data, connections to other generative models, and applications in vision, NLP, temporal data, multimodal learning, and science.	Does not map diffusion modeling properties to mobile edge systems, resource-constrained deployment, or agent-oriented intelligence.
Ahsan et al. (2025)	Cross-domain DM foundations and applications, including image synthesis, image transformation, text-to-image generation, audio synthesis, healthcare, time-series forecasting, anomaly detection, and molecular dynamics.	Provides broad application coverage but lacks DLM-centered inference, compression, edge serving, and agent evaluation perspectives.
Li et al. (2025f)	DLM ecosystem, including continuous and discrete DLMs, hybrid autoregressive-DMs, pre-training, post-training, inference optimization, caching, multimodal DLMs, and applications.	Does not address edge/cloud deployment, wireless bandwidth limits, IoT scenarios, or structure-aware evaluation for DLM-based agents.
Yu et al. (2025a)	Discrete diffusion in large language and multimodal models, including mathematical formulations, representative dLLMs/dMLLMs, training, inference, quantization, privacy, safety, and applications.	Lacks a system-level treatment of mobile edge deployment, communication-computation co-design, and closed-loop agent execution.
Luong et al. (2025)	DMs for future networks and communications, including channel modeling, signal reconstruction, ISAC, edge resource management, semantic communications, security, and other wireless applications.	Focuses on DMs as wireless generation, reconstruction, and optimization tools, without studying token-level DLMs for edge-native agents.
Qin et al. (2026)	DMs for generative semantic communications, including score-based foundations, conditional diffusion, efficient diffusion, generalized diffusion, semantic decoding as inverse problems, and human-/machine-/agent-centric scenarios.	Centers on semantic reconstruction and transmission, leaving DLM-based agent architecture, split inference, and agent evaluation underexplored.
Fan et al. (2026)	GDMs for wireless networks, organized around sensing, transmission, application, and security planes, with emphasis on channel modeling, semantic communications, network optimization, vertical applications, and security.	Emphasizes continuous GDMs for wireless networks rather than DLM-native modeling, edge deployment, and agentic decision loops.
This survey	**DLMs for mobile edge agentic AI, covering DLMs foundations, resource-efficient DLMs, edge/cloud deployment, communication-efficient diffusion intelligence, IoT and wireless applications, and evaluation frameworks.**	**Bridges the above gaps by connecting DLM modeling properties with latency, memory, energy, bandwidth, privacy, reliability, and agent-oriented system requirements.**

- **System-level deployment and communication perspective.** We organize edge DLM deployment from device-only execution to edge-assisted, cloud-edge collaborative, distributed, and adaptive serving paradigms, and discuss communication-computation co-optimization under dynamic wireless conditions.

- **Applications, evaluation, and open challenges.** We summarize the potential of DLMs in IoT and wireless systems, review structure-aware evaluation across capability, interaction, deployment, safety, and reproducibility, and identify open challenges toward DLM-native mobile edge intelligence.

The overall structure of this survey is illustrated in Fig. 2. The figure provides a roadmap from DLM foundations and resource-efficient techniques to edge deployment, communication-efficient diffusion intelligence, IoT and wireless applications, evaluation frameworks, and open challenges. For ease of reference, Table 4 summarizes the abbreviations and acronyms frequently used throughout this survey.

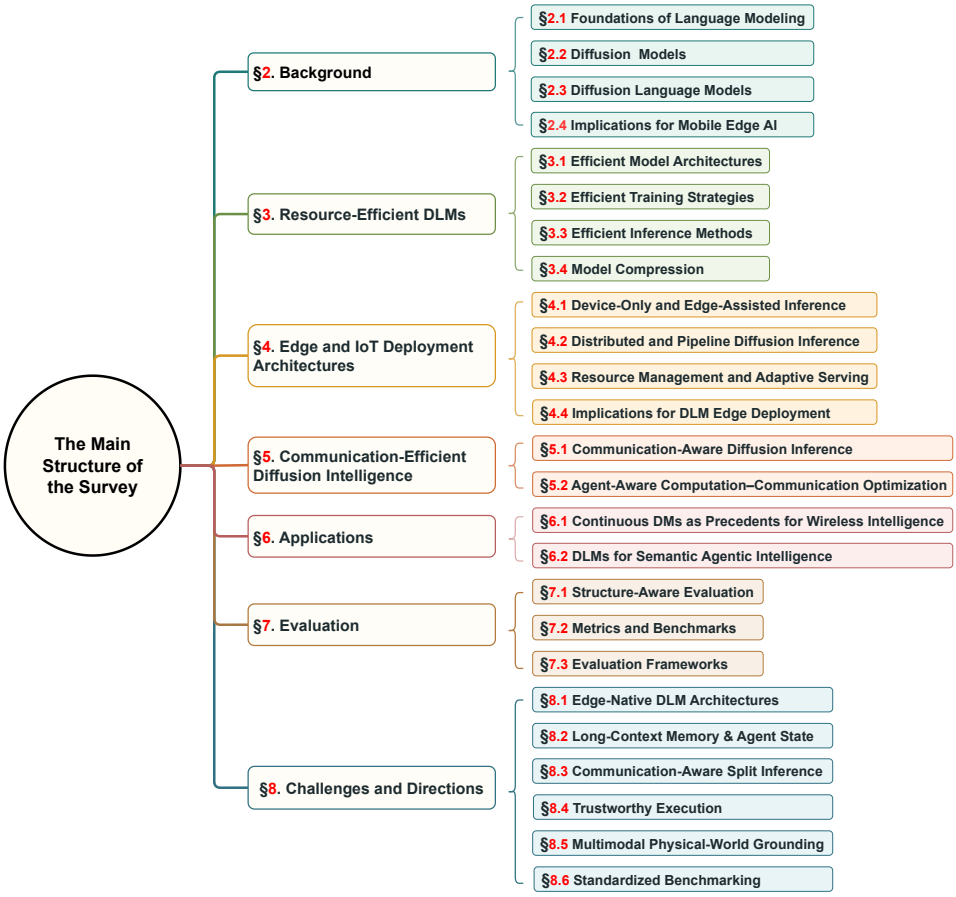


Fig. 2 Survey framework for DLMs for mobile edge agentic AI, organized around foundations, edge-oriented applications, evaluation, and research challenges

Table 4 Frequently used abbreviations.

Abbreviation	Full Term
AI	Artificial Intelligence
DM	Diffusion Model
DLM	Diffusion Language Model
LLM	Large Language Model
IoT	Internet of Things
MEC	Multi-access Edge Computing
NLP	Natural Language Processing
MLM	Masked Language Model
KV	Key-Value
MoE	Mixture-of-Experts
SDE	Stochastic Differential Equation
D3PM	Discrete Denoising Diffusion Probabilistic Model
CFG	Classifier-Free Guidance
MLLM	Multimodal Large Language Model
VLA	Vision-Language-Action
UI	User Interface
API	Application Programming Interface
PEFT	Parameter-Efficient Fine-Tuning
LoRA	Low-Rank Adaptation
DP	Differential Privacy
NFE	Number of Function Evaluations
DRL	Deep Reinforcement Learning

2 Background: Diffusion Language Models

2.1 Language Modeling Paradigms

2.1.1 Autoregressive Models

Autoregressive models form the dominant paradigm of modern LLMs. They generate sequences through next-token prediction, where each token depends only on previous tokens Gruver et al. (2023). This factorizes the joint sequence probability as:

$$p(x_1, x_2, \ldots, x_T) = p(x_1) \prod_{t=2}^{T} p(x_t \mid x_1, x_2, \ldots, x_{t-1}), \tag{1}$$

where x_t denotes the token at position t, T denotes the sequence length, $p(x_1)$ denotes the marginal probability of the first token, and $p(x_t \mid x_1, x_2, \ldots, x_{t-1})$ denotes the conditional probability of x_t given all preceding tokens. Accordingly, this left-to-right factorization supports data compression, pattern extrapolation, and in-context learning Gruver et al. (2023), and has shaped several representative model families relevant to mobile edge agents.

Closed-source autoregressive Transformers, represented by GPT-3 and GPT-4, established the scaling paradigm by showing that larger parameter capacity and training data can enable emergent capabilities Touvron et al. (2023). Their strong zero-shot

and few-shot generalization supports complex cognitive tasks without downstream fine-tuning Gruver et al. (2023), making them cloud-based upper-bound references for mobile edge agent intelligence. Open-source models, represented by Llama 1 and Llama 2, use optimized autoregressive Transformer designs with mechanisms such as grouped-query attention (GQA) Touvron et al. (2023). Llama 2 further combines large-scale self-supervised pre-training with Reinforcement Learning from Human Feedback (RLHF) to align with human preferences Touvron et al. (2023). Their openness and efficiency make them key foundations for on-device and lightweight edge research.

Beyond natural language, autoregressive modeling has been extended to time-series and visual generation. Time-LLM maps time-series data into textual prototypes through patch reprogramming, enabling frozen LLaMA or GPT-2 models to perform forecasting and spatiotemporal reasoning Jin et al. (2024). This improves the ability of edge agents to process physical sensor data. In visual generation, traditional Vector-Quantized Generative Adversarial Network (VQGAN)-style raster-scan autoregressive generation is inefficient Tian et al. (2024). Visual autoregressive modeling addresses this limitation by reformulating generation as next-scale prediction, using coarse-to-fine multi-scale parallel autoregressive modeling to surpass Diffusion Transformers (DiTs) in visual fidelity and scalability Tian et al. (2024).

2.1.2 Masked Language Models for Edge Intelligent Perception

Masked Language Models (MLMs) have introduced bidirectional representation learning by masking part of an input sequence and training the model to reconstruct the missing tokens from surrounding contexts Devlin et al. (2019). Unlike autoregressive models, which impose unidirectional causality, MLMs jointly exploit left and right contexts to learn deep semantic representations Devlin et al. (2019). For mobile edge agents, this contextual reconstruction ability provides compact semantic perception for robust decision-making under limited computation, memory, and bandwidth Deng et al. (2020).

BERT has established the canonical MLM architecture through cloze-style masked prediction, showing that large-scale masked pre-training on unlabeled corpora yields transferable semantic embeddings for many downstream tasks Devlin et al. (2019). Its variants, such as RoBERTa, verify the effectiveness of optimized masked pre-training for representation learning Devlin et al. (2019); Liu et al. (2019). To address the low pre-training efficiency of standard MLMs, DeBERTaV3 introduces a sample-efficient Replaced Token Detection (RTD) objective and Gradient-Disentangled Embedding Sharing (GDES), which reduces conflicts between discriminative and generative objectives and improves NLU performance He et al. (2023a).

MLM embeddings also support large-scale unstructured data analysis. BERTopic uses document embeddings from BERT or RoBERTa, combines density-based clustering with class-based Term Frequency-Inverse Document Frequency (c-TF-IDF), and enables coherent latent-topic discovery in large text corpora Grootendorst (2022). Beyond text, masked prediction has been extended to time-series analysis and visual generation. In mobile edge computing, this contextual completion mechanism can impute missing sensor readings and support environmental situational awareness,

making MLMs an important precursor to DLMs that also rely on bidirectional reconstruction and iterative refinement.

2.2 Diffusion Models

DMs are generative latent variable models inspired by non-equilibrium thermodynamics. They gradually corrupt a complex data distribution into a simple prior through a parameterized Markov chain, and then synthesize data by learning the reverse denoising trajectory Lipman et al. (2023). According to the topology of the data state space, DMs are commonly divided into continuous and discrete paradigms.

2.2.1 Continuous Diffusion Models

For continuous state spaces, such as images and audio, DMs define a forward process that incrementally injects Gaussian noise and a reverse process that reconstructs data through denoising. The forward corruption process can be formulated under the continuous-time framework of SDEs:

$$dx = f(x,t)dt + g(t)dw, \tag{2}$$

where $f(\cdot,t)$ is the drift coefficient, $g(t)$ is the diffusion coefficient, and w is the standard Wiener process Song et al. (2021). The generative phase follows the reverse-time evolution of this SDE. By Anderson's theorem, the reverse SDE is given by

$$dx = \left[f(x,t) - g^2(t)\nabla_x \log p_t(x)\right] dt + g(t)d\bar{w}. \tag{3}$$

Solving this process requires neural networks to estimate the score function $\nabla_x \log p_t(x)$ of the perturbed data distribution, commonly through objectives such as Denoising Score Matching Karras et al. (2022). To reduce the cost of score estimation in high-dimensional spaces, Latent Diffusion Models (LDMs) perform diffusion in the low-dimensional latent space of a pre-trained autoencoder, supporting lighter edge deployment Rombach et al. (2022). Flow Matching (FM) further accelerates training and sampling by regressing the vector field of a fixed conditional probability path for Continuous Normalizing Flows (CNFs) Lipman et al. (2023).

2.2.2 Discrete Diffusion Models

Because text is inherently discrete, continuous DMs are unsuitable for direct natural language modeling. Discrete Denoising Diffusion Probabilistic Models (D3PMs) extend diffusion to categorical spaces by replacing Gaussian noising with Markov transition matrices Q_t, which define probabilistic token-state transitions Austin et al. (2021). A major advantage of this paradigm is the ability to design structured transition matrices, including absorbing states, such as the [MASK] token. When the forward process iteratively masks text, the reverse process predicts masked tokens in a hierarchical and layer-wise manner. This creates a mathematical connection between DMs and conventional MLMs, forming the theoretical basis of modern DLMs Austin et al. (2021).

2.3 Diffusion Language Models

DLMs represent a non-autoregressive paradigm for language generation. Instead of generating tokens strictly from left to right, they synthesize text by iteratively reconstructing corrupted sequences. For mobile edge agents, this mechanism offers parallel token refinement, bidirectional context modeling, controllable generation, and flexible quality-latency adaptation under resource constraints.

2.3.1 Generation Pipeline of Diffusion Language Models

DLM architectures rely on the duality between forward noising and reverse generation. Due to the discrete nature of language, existing methods can commonly be grouped into three pipelines: Continuous Embedding Diffusion, Discrete State-Space Diffusion, and Masked Diffusion with Maximum Likelihood.

Continuous Embedding Diffusion. Methods, such as Diffusion-LM, map discrete tokens into continuous embedding space, inject Gaussian noise in the forward process, denoise latent vectors through a learned reverse network, and finally project continuous variables back to tokens through rounding. This design enables fine-grained controllable generation with gradient-guided optimization, but may suffer from rounding errors, embedding sensitivity, and slow sampling Li et al. (2022).

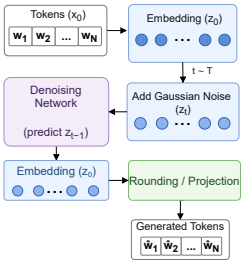
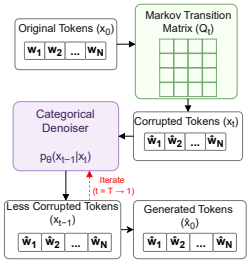
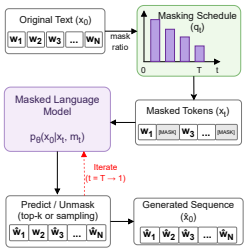
Discrete State-Space Diffusion. D3PMs perform diffusion directly in discrete token space using structured Markov transition matrices. Tokens evolve according to categorical distributions, and absorbing states can stabilize selected token states. The reverse process recovers sequences through iterative categorical sampling. Unlike masked prediction, these absorbing placeholders are not designed as explicit masked-token targets. This pipeline avoids continuous-space quantization errors and provides a principled framework for discrete sequence modeling Austin et al. (2021).

Masked Diffusion and Maximum Likelihood Estimation. Masked Diffusion Language Models (MDLMs) use a masking-based forward process, in which tokens of a clean sequence x_0 are corrupted according to a time-dependent masking probability. The reverse model predicts the original tokens at masked positions from the corrupted sequence x_t. Under the linear masking process adopted by LLaDA, where $t \sim \mathcal{U}(0, 1]$ and each token is independently replaced with the `[MASK]` token with probability t, a principled masked-token training objective can be written as Nie et al. (2025):

$$\mathcal{L}_{\text{mask}}(\theta) = -\mathbb{E}_{x_0,t,x_t}\left[\frac{1}{t}\sum_{i\in\mathcal{M}_t}\log p_\theta(x_0^i \mid x_t)\right], \tag{4}$$

where $x_0 \sim p_{\text{data}}$ is a clean sequence sampled from the data distribution, $x_t \sim q_{t|0}(\cdot \mid x_0)$ is sampled from the forward masking distribution, and $\mathcal{M}_t$ denotes the set of masked positions at time step t. This objective computes the cross-entropy loss only over masked tokens and upper-bounds the negative log-likelihood of the induced generative model, thereby providing a structured approximate maximum-likelihood training framework Nie et al. (2025). From a complementary variational perspective, Sahoo et al. (2024) derive a continuous-time Rao–Blackwellized negative Evidence Lower

Table 5 Comparison of generation pipelines in Diffusion Language Models.

Continuous Embedding Diffusion	**Discrete State-Space Diffusion**	**Masked Diffusion & Maximum Likelihood**
		
Structure: Tokens → embeddings → Gaussian noising → denoising → rounding. **Description:** Maps tokens into continuous embeddings, performs Gaussian diffusion in latent space, and projects denoised embeddings back to tokens Li et al. (2022). **Pros:** Gradient-based control; fine-grained guidance. **Cons:** Rounding errors; embedding sensitivity; slow sampling.	**Structure:** Tokens → Markov corruption → categorical denoising → output tokens. **Description:** Performs diffusion directly in token space using structured Markov transition matrices and reverse categorical sampling Austin et al. (2021). **Pros:** Avoids rounding; models discrete states directly. **Cons:** Transition design complexity; costly for large vocabularies.	**Structure:** Tokens → masking → masked-token prediction → iterative unmasking. **Description:** Uses a masking-based forward process and Rao-Blackwellized MLM-style MLE for semi-autoregressive generation Sahoo et al. (2024); Nie et al. (2025). **Pros:** Efficient sampling; variable-length generation; scalable. **Cons:** Masking-policy dependence; multi-step refinement; costly pretraining.

Bound (ELBO) under a substitution-based reverse-process parameterization, showing that masked diffusion training can be expressed as a schedule-weighted average of masked language modeling losses.

Beyond these training formulations, DiffusionGemma further demonstrates that masked discrete diffusion is moving from research prototypes toward open-weight checkpoints in the Gemma family. The public DiffusionGemma-26B-A4B-it model combines masked diffusion decoding with a mixture-of-experts (MoE) backbone, providing a recent reference point for studying scalable DLM inference in practical deployment settings Google (2026a,b).

Table 5 compares these three pipelines in terms of structure, forward and reverse processes, advantages, and limitations, providing a compact reference for later discussions on DLM design and deployment.

2.3.2 Comparison with Autoregressive LLMs

Although autoregressive LLMs remain dominant, DLMs address several structural limitations of left-to-right generation. To clarify this structural shift, Fig. 3 contrasts autoregressive language modeling, general diffusion modeling, and diffusion language modeling from the perspective of their generation trajectories. Panel (a) represents

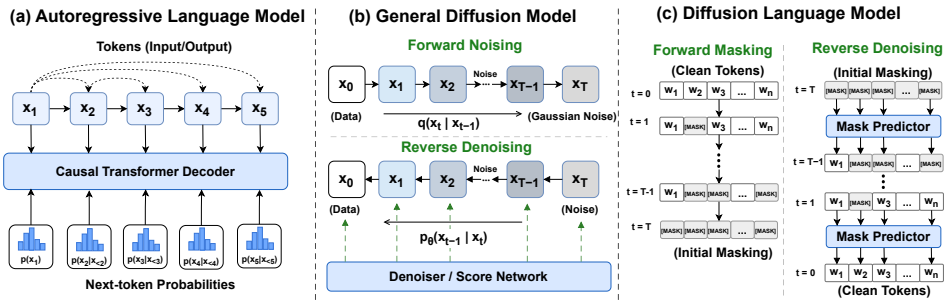


Fig. 3 Generation mechanisms of autoregressive language models, general DMs, and DLMs

left-to-right causal decoding, panel (b) shows continuous forward noising and reverse denoising, and panel (c) maps diffusion-style generation to token masking and iterative token denoising.

Generation Direction and Contextual Modeling. Autoregressive models follow temporal causality and unidirectional decoding, which can cause error accumulation and limit global structural planning. In contrast, DLMs use bidirectional refinement. LLaDA, for example, improves global contextual reasoning and mitigates the Reversal Curse, namely the difficulty of inferring B is A after learning A is B Nie et al. (2025).

Compute-Quality Trade-off. Autoregressive generation is constrained by parameter scale and fixed decoding trajectories. The Score Entropy Discrete Diffusion models (SEDD) shows that DMs can trade computation for quality by adjusting reverse sampling steps. At a comparable scale, it outperforms autoregressive counterparts, such as GPT-2, and maintains high-fidelity generation with far fewer network evaluations Lou et al. (2024); Feng et al. (2025a).

Controllability and Infilling. Because of unidirectionality, autoregressive models struggle with infilling and zero-shot constraint satisfaction. DLMs can incorporate conditioning guidance during inference. Diffusion-LM uses gradient-based guidance, while SEDD uses classifier-free guidance (CFG), enabling controllable generation without task-specific retraining Li et al. (2022); Lou et al. (2024).

2.3.3 Multimodal Large Diffusion Language Models

Recent studies have extended masked and discrete diffusion from text-only generation to multimodal understanding, reasoning, and generation. As summarized in recent surveys, a growing body of dMLLMs has explored diverse architectures for integrating visual and textual information within diffusion-based generation frameworks Li et al. (2025f); Yu et al. (2025a). Representative examples include Dimple, which combines a vision encoder with a discrete DLM backbone and adopts an autoregressive-then-diffusion training strategy together with its Confident Decoding mechanism for parallel generation Yu et al. (2025b); LaViDa, which equips DLMs with visual encoders and introduces complementary masking and Prefix-DLM caching for multimodal understanding Li et al. (2025e); and LLaDA-V, which extends a large language DM to multimodal understanding through visual instruction tuning

and bidirectional attention within a purely diffusion-based multimodal large language model (MLLM) framework You et al. (2026). Beyond these understanding-oriented dMLLMs, MMaDA exemplifies a more unified multimodal diffusion direction by modeling text and image tokens within a shared discrete diffusion framework, enabling textual reasoning, multimodal understanding, and text-to-image generation under a unified denoising formulation Yang et al. (2025a).

Rather than exhaustively enumerating all multimodal DLMs, this survey highlights these representative models to illustrate major architectural directions relevant to mobile edge agents. Such developments suggest a flexible token-level interface for processing instructions, visual observations, user interface (UI) states, sensor streams, and action-related representations under tight latency, memory, and energy budgets. However, practical edge deployment still depends on efficient visual tokenization, denoising acceleration, cache optimization, model quantization, and adaptive device-edge offloading.

2.4 Implications for Mobile Edge System Design

Integrating DLMs into the cognitive architecture of mobile edge agents provides system-level implications along three complementary dimensions: compute–quality elasticity, memory-bandwidth efficiency, and deterministic and safe on-device execution.

2.4.1 Elastic Compute Adaptation and Deployment

Edge devices are characterized by substantial heterogeneity and temporal dynamics in computing resources (e.g., GPU/NPU availability) and power budgets. Exploiting the compute-quality trade-off validated by SEDD Lou et al. (2024), edge agents can elastically adjust diffusion sampling steps through early stopping based on real-time energy thresholds. Under favorable resource conditions, the agent can execute multi-step denoising for complex reasoning; under battery constraints, it can truncate the sampling trajectory to deliver core semantics with ultra-low latency, offering flexibility not available in the same form in standard autoregressive models.

2.4.2 Alleviating Memory Bandwidth Bottlenecks via Parallel Decoding

Mobile edge devices (e.g., smartphones and IoT nodes) are generally bottlenecked by memory bandwidth rather than raw floating-point operations (FLOPs). The pervasive KV-caching and serial token generation of autoregressive models impose substantial memory access overhead. In contrast, DLMs (e.g., MDLM and LLaDA) can predict multiple tokens in parallel during the reverse diffusion phase Sahoo et al. (2024); Nie et al. (2025). This parallelization improves the utilization of edge AI accelerators and reduces end-to-end perception-to-action latency for intelligent agents.

Table 6 Mapping DLM properties to edge-native agent requirements.

DLM Property	Edge-Agent Implication	Deployment Concern
Bidirectional context	Supports command repair, global planning, and context completion by using both left and right contexts Nie et al. (2025).	Long-context memory and full-sequence refinement cost.
Parallel refinement	Updates multiple uncertain tokens together for low-latency structured generation and action-token synthesis Wu et al. (2025).	Memory access, cache behavior, and synchronization overhead.
Adaptive denoising	Adjusts refinement steps for early exit, anytime response, and quality-latency trade-offs Zhu et al. (2026).	Semantic drift or insufficient correction after truncation.
Structured controllability	Guides generation toward valid formats, tool calls, API commands, and safe executable plans Li et al. (2022).	Need for constraint checking before final action execution.
Noising and split states	Enables privacy-aware offloading by transmitting obfuscated intermediate states, instead of raw prompts Allmendinger et al. (2026).	Possible leakage from latent states, token distributions, or reasoning traces.
Quality-latency elasticity	Adapts generation to battery level, wireless condition, task urgency, and interaction risk Peng et al. (2025).	Requires joint reporting of latency, energy, memory, communication, and hardware settings.

2.4.3 Deterministic and Safe On-Device Execution

Edge agents frequently translate cloud-issued directives into physical actions via local operating systems or actuators (e.g., smart home control and application programming interface (API) invocation). This requires reliable format control and strict safety constraints. By using gradient-based or gradient-free controllable generation in DLMs Li et al. (2022); Lou et al. (2024), generation can be constrained to follow predefined syntactic structures and security boundaries during action generation, thereby improving the robustness and safety of on-device AI systems.

The above discussion shows that DLMs are relevant to mobile edge agents, not only because they avoid strictly left-to-right decoding, but also because their denoising dynamics can be translated into concrete system-level functions. Table 6 summarizes this mechanism-to-requirement mapping from edge-agent perspectives.

2.5 Lessons Learned

Autoregressive LLMs remain preferable for open-ended and long-form generation, but their sequential decoding and KV-cache dependence create latency and memory bottlenecks at the edge Li et al. (2025a); Peng et al. (2025). DLMs better support infilling, structured generation, global planning, and constraint-aware action synthesis; yet, their practical gains depend on stable sampling, cache compatibility, and hardware-aware parallel decoding Nie et al. (2025); Wu et al. (2025). Thus, edge systems should treat autoregressive models and DLMs as complementary backbones selected by task structure, latency budget, and ecosystem maturity.

3 Resource-Efficient Diffusion Language Models

3.1 Efficient Model Architectures

Deploying agentic AI in MEC is limited by edge devices' constrained computation, memory bandwidth, and power budgets. Traditional DMs incur high latency and redundancy because of iterative multi-step denoising. Recent work targets sub-second mobile inference through architectural redesign and lightweight optimization Zhao et al. (2024); Li et al. (2023b). This section reviews efficient architectures in three directions: lightweight diffusion Transformers, block and hybrid autoregressive-DMs, and sparse MoE diffusion models.

3.1.1 Lightweight Diffusion Transformers

Conventional DMs rely on computation-heavy convolutional U-Nets. DiT, built entirely on attention, has become a simpler foundation for on-device diffusion. Token merging and pruning further reduce the attention memory wall, improving throughput on resource-limited hardware while preserving fidelity Bolya and Hoffman (2023). Table 7 summarizes representative efficient diffusion and DLM architectures in terms of model scale, application scenarios, and mobile/IoT-related deployment characteristics.

U-ViT Architecture. For feature mapping in visual and multimodal generation, U-ViT proposes a streamlined universal Transformer. It treats timesteps, conditional signals, and noised sequence patches as uniform tokens. Long skip connections between shallow and deep layers replace the convolutional downsampling and upsampling modules of U-Nets, reducing complexity while maintaining feature representation Bao et al. (2023).

On-Device Extreme Compression (SnapGen++). For computation-limited mobile devices, SnapGen++ demonstrates strong lightweight generation. Architecture-level optimization and efficient diffusion training reduce DiT parameters to 0.4 billion (0.4B), achieving high-fidelity generation with 1.8-second latency Hu et al. (2026). Diffusion-specific post-training quantization (PTQ) further compresses weights to low-bit representations, fitting low-memory edge gateways He et al. (2023b).

3.1.2 Block Diffusion and Hybrid Autoregressive Diffusion Models

Lightweight diffusion Transformers reduce the cost of individual backbones, but they do not fully resolve the structural mismatch between parallel diffusion and flexible-length language generation. Autoregressive models capture long-range dependencies and support flexible-length decoding but suffer from sequential generation latency, whereas DMs enable parallel decoding but face challenges in variable-length generation and exact likelihood modeling Li et al. (2022). Block and hybrid autoregressive-DMs therefore address this mismatch by reorganizing the decoding process to combine diffusion-style parallelism with autoregressive-style length flexibility and cache reuse.

Block Diffusion Architecture. Block Diffusion addresses fixed-length limits by interpolating discrete denoising diffusion with autoregressive mechanisms. It supports

Table 7 Representative efficient diffusion and DLM architectures for edge deployment.

Category	Model	Model Size	Application Scenario	Edge/Mobile Implication
Lightweight and Compressed Diffusion Architectures				
Lightweight mobile DM	MobileDiffusion Zhao et al. (2024)	<400M parameters; one-step generation	Text-to-image generation, controllable generation, style/object LoRA, and inpainting	On-device latency: 0.2 s; enables instant local generation.
Lightweight mobile DM	SnapFusion Li et al. (2023b)	Text encoder: 123M; UNet: 848M; decoder: 13M	Mobile text-to-image generation at 512×512 resolution	Mobile latency: 1.96 s; reduces diffusion to 8 denoising steps.
Lightweight DiT backbone	U-ViT Bao et al. (2023)	U-ViT-M: 131M; U-ViT-L: 287M; U-ViT-H: 501M + 84M autoencoder	Unconditional, class-conditional, and text-to-image generation	Simplified tokenized backbone; easier hardware-friendly deployment.
Edge-oriented DiT compression	SnapGen++ Hu et al. (2026)	On-device model: 0.4B; full model: 1.6B	High-fidelity text-to-image generation at approximately 1024^2 resolution	Mobile latency: 1.8 s; elastic sub-models fit heterogeneous devices.
Post-training quantized DM	PTQD He et al. (2023b)	LDM-4: 1603.35 MB FP32; 430.06 MB W8A8; 234.51 MB mixed / W4A4	Compressed image diffusion inference	Low-bit quantization; lowers memory footprint and compute cost.
DLMs and Hybrids				
Block / hybrid DLM	Block Diffusion Arriola et al. (2025)	Backbone-dependent; block size $L' \in \{4, 8, 16, 128\}$	Flexible-length text generation and language modeling	Block-wise decoding; supports KV cache and variable-length text.
Training-free DLM acceleration	Fast-dLLM Wu et al. (2025)	No extra parameters; applied to LLaDA and Dream	Diffusion LLM inference for reasoning, code, and text generation	KV cache + parallel decoding; accelerates practical DLM inference.
MoE DLM	DiffusionGemma Google (2026a,b)	26B / about 4B active	Text generation	MoE lowers active compute; memory-bound.
Hybrid autoregressive-DM language model	Evo Wu et al. (2026)	Evo-8B	Text generation, reasoning, code generation, and language understanding	Adaptive autoregressive-diffusion balance; avoids unnecessary refinement steps.
Sparse MoE and Distributed Architectures				
Sparse MoE DM	Switch-DiT Park et al. (2025)	DiT-B: 131M; Switch-DiT-B: 137M–151M; common setting: 144M	Unconditional and class-conditional image generation	Timestep-aware sparse routing; activates fewer experts per step.
Billion-scale sparse DiT	DiT-MoE Fei et al. (2024)	199M/71M active to 16.5B/3.1B active parameters	Large-scale class-conditional image generation	Large total capacity; only sparse experts are activated during inference.
Distributed diffusion serving	DistriFusion Li et al. (2024)	Distributed execution across multiple GPUs/devices; no retraining	High-resolution image generation with SDXL-style DMs	Patch-parallel inference; distributes high-resolution generation across devices.

flexible-length generation and combines KV caching with adaptive parallel token sampling during inference Wu et al. (2025). Block-wise parallel decoding preserves diffusion acceleration, removes redundant long-sequence computation, and lowers single-pass memory footprint Arriola et al. (2025). Fully masked DMs, such as LLaDA, validate scalable parallel prediction on large datasets Nie et al. (2025).

Evolution-Balanced Hybrid Latent Trajectory (Evo). Evo mathematically unifies autoregressive and diffusion paradigms in a Duality Latent Trajectory Model. It views text generation as a continuous evolutionary flow, assigning each token $t_i \in [0, 1]$ to represent semantic maturity. Low-maturity tokens receive autoregressive-style high-confidence local refinement, while high-maturity tokens trigger diffusion-style global parallel planning. This balance adaptively trades off inference uncertainty and computational efficiency Wu et al. (2026).

3.1.3 Sparse and MoE Diffusion Models

While block and hybrid designs improve decoding flexibility, they still rely on relatively persistent model capacity during each inference pass. Sparse and MoE diffusion models provide another architectural direction by increasing the total model capacity while activating only a subset of parameters for each timestep, token group, or expert route. Because diffusion timesteps differ in task complexity, static dense networks may waste computation, whereas sparse activation decouples the total parameter capacity from per-step inference FLOPs Park et al. (2025); Fei et al. (2024). Token-partitioned distributed inference can further support collaborative computing across edge devices Li et al. (2024).

Sparse Routing for Collaborative Denoising (Switch-DiT). Switch-DiT treats denoising timesteps as independent tasks and inserts Sparse Mixture-of-Experts (SMOE) into each Transformer block. Its Diffusion Prior Loss routes similar denoising tasks through shared experts while isolating conflicting parameters. This timestep-aware routing improves complex multimodal generation without increasing single-pass computational overhead Park et al. (2025).

Billion-Scale Sparse Diffusion (DiT-MoE). DiT-MoE enables large-model emergence under bounded inference overhead through Shared Expert Routing and Expert-level Balance Loss. Empirical results show temporal expert preferences: early denoising queries low-frequency global-feature experts, whereas late denoising uses high-frequency detail experts. For mobile edge agents, such sparse activation is potentially beneficial because only a subset of experts is activated for each input, reducing inference computation and offering a configurable trade-off between model capacity and memory requirements Fei et al. (2024).

3.2 Efficient Training Strategies

Mobile edge agent systems face strict computation, memory, and communication constraints, making resource-efficient training and fine-tuning essential for scalable deployment. Current research addresses resource-efficient DLM training at three complementary levels: (a) autoregressive-to-diffusion adaptation reduces the cost of model initialization by reusing pretrained knowledge, (b) parameter-efficient fine-tuning

(PEFT) limits the number of parameters updated during adaptation, and (c) federated diffusion coordinates training across distributed and privacy-sensitive data sources. Together, these strategies target initialization cost, local update efficiency, and cross-device training coordination, respectively.

3.2.1 Autoregressive-to-Diffusion Adaptation and Joint Training

Training large DLMs from scratch is costly. Since pre-trained autoregressive models already encode rich sequential priors, adapting them to diffusion offers a resource-efficient path Gong et al. (2025).

A key strategy unifies discrete token prediction and continuous denoising. Transfusion jointly trains text and image generation in one Transformer backbone by combining next-token prediction with diffusion losses, improving scalability without heterogeneous networks Zhou et al. (2025). Block Diffusion addresses fixed-length limits through interpolation-based adaptation between discrete denoising and autoregressive generation, improving optimization with KV-caching and parallel sampling Arriola et al. (2025). Evo dynamically switches between high-confidence autoregressive refinement and global diffusion planning through evolutionary variational inference Wu et al. (2026). These methods reuse foundation weights and avoid the energy cost of de novo training.

3.2.2 PEFT Mechanisms

Reusing pretrained autoregressive weights reduces the need for de novo DLM training, but full-parameter adaptation can still be prohibitively expensive on resource-constrained devices. PEFT targets the next bottleneck by restricting parameter updates to lightweight trainable components. Low-Rank Adaptation (LoRA) freezes pre-trained weights and inserts trainable low-rank matrices into the Transformer bypass, reducing trainable parameters by orders of magnitude while matching or surpassing full fine-tuning Hu et al. (2022). DiffFit targets diffusion spatio-temporal dynamics by fine-tuning only bias terms and lightweight spatial scaling factors, instead of dense layer approximations, reducing storage and gradient costs for edge domain transfer Xie et al. (2023).

3.2.3 Federated Diffusion Training Strategies

PEFT and federated diffusion address resource-efficient training at different levels. While PEFT reduces the number of parameters updated during local or domain-specific adaptation, federated training coordinates learning across distributed and privacy-sensitive data sources. The two directions are distinct but complementary: the former targets local adaptation cost, whereas the latter addresses cross-device coordination, data locality, and communication overhead Sun et al. (2024); Li et al. (2025b). In mobile edge networks, however, federated diffusion still faces major challenges arising from non-independent and identically distributed (non-IID) data, communication overhead, and heterogeneous edge resource constraints Mendieta et al. (2025).

To handle data heterogeneity, recent frameworks locally fine-tune DMs to synthesize high-quality data, aligning global distributions without exposing client data

and reducing typical federated learning (FL) performance gaps Hoefler et al. (2024). FedDiff introduces one-shot federated diffusion to reduce eavesdropping risks and multi-round synchronization latency. With Differential Privacy (DP) and Fourier Magnitude Filtering (FMF), it extracts task features in heterogeneous settings, accelerates convergence, and provides rigorous privacy bounds Mendieta et al. (2025); Dockhorn et al. (2023). On-demand quantization further adapts weight compression and wireless transmission under battery and bandwidth limits, enabling energy-efficient federated aggregation without degrading model precision Lai et al. (2024).

3.3 Efficient Inference Methods

To deploy DLMs on mobile edge agents, inference must meet strict latency and computation limits. Diffusion generation offers acceleration opportunities distinct from autoregressive models. This section organizes DLM inference acceleration along three complementary dimensions. Parallel decoding reduces serial dependence across token positions by updating multiple tokens jointly; few-step diffusion shortens the denoising trajectory by reducing refinement rounds or function evaluations; and speculative draft–verify strategies seek to reduce expensive target-model computation through lightweight proposal and verification.

3.3.1 Parallel Decoding

Unlike autoregressive language models that generate tokens left to right, DLMs can update multiple tokens simultaneously during reverse denoising Wu et al. (2023); Gong et al. (2023). MaskGIT establishes this paradigm with a bidirectional Transformer and dynamic mask scheduling, generating all tokens in a constant number of steps Chang et al. (2022). Extending this non-autoregressive design, the autoregressive-diffusion framework uses multi-level diffusion with position-dependent denoising steps Wu et al. (2023). Left tokens undergo fewer steps, emerge earlier, and guide right-token generation, preserving parallel decoding while implicitly modeling language dependencies Wu et al. (2023). SSD-LM further adopts semi-autoregressive block-wise generation, enabling parallel context updates within each block and bridging full parallel generation with serial decoding Han et al. (2023).

3.3.2 Few-Step Diffusion

Parallel decoding increases the amount of sequence progress made within each model evaluation by updating multiple token positions jointly. However, generation may still require many iterative refinement rounds, and standard DMs can require hundreds or thousands of Markov-chain steps for one high-quality sample Song et al. (2022); Salimans and Ho (2022). Few-step diffusion targets a different bottleneck by shortening the denoising trajectory itself. Since each denoising step usually requires one forward pass of the denoising network, inference cost is commonly characterized by the number of function evaluations (NFEs) Lu et al. (2022). These two directions are complementary: parallel decoding reduces serial dependence across token positions, whereas few-step diffusion reduces the number of denoiser evaluations required for generation.

Denoising Diffusion Implicit Models (DDIMs) generalizes the forward process to a non-Markovian form while keeping the standard training objective, enabling deterministic generation with 10x to 50x wall-clock acceleration Song et al. (2022). Progressive distillation iteratively distills a deterministic sampler into models needing half as many sampling steps, reducing samplers from up to 8192 steps to as few as 4 steps with limited perceptual degradation Salimans and Ho (2022).

Beyond DDIM and progressive distillation, solver- and consistency-based approaches provide additional routes to few-step generation. DPM-Solver reduces sampling cost through high-order ordinary differential equation (ODE) solvers, enabling high-quality generation with a small number of NFEs, while Consistency Models learn mappings that support one-step or few-step generation Lu et al. (2022); Song et al. (2023).

For discrete text generation, data-distribution ratio estimation removes dependence on autoregressive distribution annealing, enabling flexible compute-quality trade-offs and high-fidelity generation with fewer sampling steps Lou et al. (2024). FM provides simulation-free training with CNFs and can use Optimal Transport (OT) displacement interpolation to form straight conditional probability paths, improving generation and sampling efficiency Lipman et al. (2023). Latent Consistency Models (LCMs) directly predict solutions of augmented probability flow ODEs in latent spaces, achieving high-fidelity generation in 2 to 4 steps or even one step Luo et al. (2023).

3.3.3 Emerging Speculative Decoding Directions

Few-step diffusion compresses the denoising trajectory globally, but may require retraining, distillation, or carefully designed solvers. A complementary emerging inference-time direction is speculative decoding, where a lightweight draft model proposes multiple candidate tokens or intermediate states and a stronger target model performs parallel verification Leviathan et al. (2023). In autoregressive settings, this draft-and-verify strategy can preserve the target model's output distribution while reducing expensive target-model computation. While originally designed for autoregressive Transformers, the principle may be extended to diffusion by approximating simpler subtasks or selected intermediate states to accelerate parts of the denoising process. Thus, few-step diffusion reduces the length of the trajectory, whereas speculative approaches seek to reduce the cost of traversing or verifying that trajectory.

For distributed edge systems, lightweight drafting and initial trajectory prediction can run on IoT devices, while costly parallel verification is offloaded to edge servers, reducing end-to-end latency.

3.4 Model Compression

Despite strong generation quality, DLMs remain difficult to deploy on mobile edge agents because of their large parameter scales, memory-bandwidth demand, and costly iterative denoising. Model compression reduces storage, computation, and runtime memory costs with limited quality loss, making it essential for edge deployment. Since

model compression has been extensively studied in diffusion and language models, this subsection only outlines its relevance to edge DLMs, while detailed techniques can be found in dedicated surveys on efficient DMs, diffusion quantization, and LLM compression Shen et al. (2025); Ma et al. (2025); Zeng et al. (2025a); Zhu et al. (2024).

Quantization compresses weights and activations into low-bit formats, such as INT8 and INT4, but DMs require timestep-aware treatment because activation distributions vary along the denoising trajectory Li et al. (2023a); He et al. (2023b).

Distillation transfers sampling trajectories or guidance behavior from large teacher models to lightweight students, reducing denoising steps or merging CFG passes for faster conditional generation Salimans and Ho (2022); Meng et al. (2023).

Pruning and sparsity remove redundant parameters, substructures, or denoising steps; methods exploiting spatial-frequency or temporal-gradient redundancy indicate useful directions for compressing Transformer-based DLMs under edge resource constraints Yang et al. (2023b); Fang et al. (2023).

3.5 Lessons Learned

Resource-efficient DLMs must balance denoising depth, memory footprint, and generation quality, rather than optimize parameter count alone Peng et al. (2025); Shen et al. (2025). Lightweight blocks, sparse routing, compression, and adaptive decoding can reduce deployment cost, but may introduce semantic drift or hardware complexity Arriola et al. (2025); Park et al. (2025); Wu et al. (2025). Future edge DLMs should prioritize memory-bandwidth reduction, cache reuse, adaptive steps, and quality reports under fixed latency, energy, and device settings.

4 DLM-Native Edge and IoT Agent Deployment

Although DLMs are increasingly relevant to mobile edge agentic AI, direct studies on DLM deployment in edge and IoT systems remain scarce. This section takes a DLM-centered perspective: it uses system-level schemes developed mainly for standard DMs as deployment references and discusses how they should be adapted for future DLM agent execution. The key insight is that device-only execution and lightweight on-device diffusion provide useful principles for reducing local latency and memory cost Zhao et al. (2024); Zheng (2025). Meanwhile, edge-assisted inference, cloud–edge collaboration, distributed denoising, and adaptive resource management suggest architectural directions for partitioning iterative generation across heterogeneous devices and edge servers Li et al. (2024); Xu et al. (2026). On the other hand, DLMs introduce additional challenges, such as discrete token generation, long-context memory, KV-cache reuse, multimodal inputs, and multi-round agent interactions, which require DLM-specific deployment designs rather than direct reuse of visual diffusion serving pipelines Zheng et al. (2025). Accordingly, wireless and MEC conditions are treated here as deployment constraints rather than the main conceptual focus; the main focus remains how DLM mechanisms affect inference, memory, privacy, and interaction for edge agents.

4.1 Device-Only and Edge-Assisted DLM Agent Inference

Deploying DLM-based agents on mobile and IoT devices is limited, not only by computation, memory, energy, and wireless bandwidth, but also by the iterative token-refinement process itself. Standard diffusion systems mainly follow three patterns: local execution, device–edge workload splitting, and cloud–edge–device collaboration. Although not yet mature for DLMs, these patterns provide practical directions for edge DLM serving.

Device-only inference removes network dependence and enables local generation on smartphones, wearables, and standalone IoT nodes. Existing diffusion systems rely on architecture pruning, hardware-aware redesign, operator approximation, dynamic loading, and step reduction. To achieve device-only execution, researchers have explored optimizations across different system levels. At the architectural level, frameworks, such as SnapFusion Li et al. (2023b) and EdgeDiT Kodavanti et al. (2026), focus on simplifying the backbone (e.g., UNet or DiT) through rigorous pruning and distillation techniques to meet mobile latency constraints. At the operator level, solutions, such as LUT-Diff Wang et al. (2025b), bypass costly matrix multiplications altogether, relying instead on lookup tables and mobile co-scheduling to maintain efficiency. MobileDiffusion further enables single-step generation with a hybrid Diffusion-GAN design, and on-device video generation uses dynamic loading for large backbones under mobile memory limits Zhao et al. (2024); Kim et al. (2025). For DLMs, these schemes suggest lightweight token backbones, cache-aware loading, low-bit execution, fewer denoising steps, and early-exit strategies for local agent reasoning and generation.

For DLM agents, edge-assisted inference should be viewed not only as offloading heavy computation to edge servers, but also as deciding which denoising stages, token blocks, or reasoning states should remain on devices. Standard diffusion systems split workloads by denoising stages, intermediate activations, or feature extractors. Early denoising or shared semantic extraction can run at the edge, with personalized refinement on devices Yang et al. (2025b). Dynamic splitting further adapts partition points to modality complexity, network state, and device capacity Liu et al. (2026b). For DLMs, similar designs may support token-block, multimodal-feature, or reasoning-trajectory splitting, letting edge servers handle heavy denoising or fusion while devices preserve private prompts, local states, permission-sensitive context, and final actions.

Cloud-edge-device collaboration forms a three-tier system: the cloud provides high-capacity generation, the edge supports low-latency caching or preview inference, and devices handle interaction and personalization. For DLM agents, this hierarchy should be organized around agent-level objectives, such as time-to-first-action, context reuse, and safe final action decoding. DiffusionX uses lightweight edge preview generation before invoking cloud models for final rendering Wei et al. (2025). Other frameworks route tasks across local, edge, and cloud nodes according to latency, energy, and quality of service (QoS) constraints, while edge caching reduces redundant generation for repeated prompts Hu et al. (2025); Yao et al. (2025). For DLM agents, this architecture supports iterative interaction: preliminary reasoning, cached context, or draft responses can run at the edge, while complex reasoning or multimodal synthesis is escalated to the cloud.

4.2 Distributed and Pipeline DLM Agent Inference

The deployment modes discussed above determine which computing tiers participate in DLM agent inference. Building on this placement perspective, distributed and pipeline inference specify how denoising steps, token blocks, model components, or reasoning states can be scheduled and executed across these tiers. Distributed diffusion inference partitions generation across heterogeneous nodes, including devices, edge servers, and device clusters, to address the mismatch between large models, iterative denoising, and limited edge resources.

Although most existing systems target standard DMs, their principles can guide DLM deployment for long sequences, multimodal inputs, and multi-agent reasoning. For DLM agents, the split boundary should be considered at token blocks, denoising rounds, reasoning states, and action-related outputs, rather than only at model layers.

Collaborative denoising splits diffusion steps between devices and edge servers. Typically, early compute-intensive denoising runs at the edge, and compact latent representations are sent to devices for task-specific refinement Du et al. (2024). This asymmetric allocation reduces device energy use, avoids full-data offloading, and supports privacy because devices share intermediate activations rather than raw prompts, conditions, or final outputs, as in CollaFuse Allmendinger et al. (2026). Adaptive controllers tune local and offloaded denoising steps based on bandwidth, device compute capacity, and latency-quality preferences Kong et al. (2026). For DLMs, this suggests splitting denoising iterations, token blocks, reasoning traces, or multimodal latent states across device and edge nodes.

Pipeline diffusion inference raises throughput by overlapping computation and communication. Recent systems use patch-level or sequence-level parallelism, rather than simple layer-wise partitioning, to distribute high-resolution generation across nodes Fang et al. (2025). Since adjacent diffusion timesteps are often similar, systems can reuse stale activations while asynchronously transmitting updated features, hiding communication latency within the pipeline Li et al. (2024). For DLMs, similar designs may exploit redundancy across denoising steps, token refinement rounds, or intermediate reasoning states. This is especially useful when stable tokens or reusable agent states can be cached across interaction turns.

Distributed systems may also adapt parallelism across inference stages. Under CFG, conditional and unconditional branches can show time-varying discrepancies, motivating dynamic switching among data parallelism, pipeline parallelism, and serial execution Jung et al. (2026). Operator-level optimizations, such as pipelined attention, double buffering, quantized communication, and graph compilation, reduce synchronization and kernel-launch overhead in high-speed edge-cloud environments Li (2026). These techniques matter for DLMs because long contexts and KV caches make communication and memory movement comparable to computation.

For reasoning-oriented DLMs, distributed inference can support test-time scaling. Edge servers can generate candidate reasoning trajectories in parallel, evaluate or stitch them with reward models, and return compact reasoning states to devices for final response generation Miles et al. (2026). This separates exploration from synthesis and may enable low-latency agentic reasoning under edge constraints.

4.3 Agent-State-Aware Resource Management and Adaptive Serving

Distributed and pipeline inference provide the structural basis for splitting DLM execution, but static partitions are insufficient in mobile edge environments where wireless channels, user demands, battery budgets, device capabilities, and agent states vary over time. Resource management and adaptive serving therefore move from static partitioning to runtime decisions over offloading, scheduling, caching, and refinement allocation. For diffusion-based agents, these decisions must balance latency, energy, privacy, and generation quality across device, edge, and cloud resources.

Diffusion offloading differs from binary offloading because generation can be split by denoising steps, model blocks, or intermediate representations. Mobile Edge Generation (MEG) keeps some denoising stages on devices and offloads heavier stages to edge servers or nearby devices Feng et al. (2025b); Xu et al. (2026). These decisions adapt to wireless signal-to-noise ratio (SNR), latency, and energy constraints, while diffusion-based reinforcement learning supports offloading under complex mobile trajectories Xu et al. (2026); Xu (2024). For DLMs, similar mechanisms may decide whether token refinement, multimodal fusion, long-context reasoning, or final response synthesis runs locally or remotely.

Compute scheduling extends offloading to multi-user mobile scenarios. In vehicle-to-everything (V2X) and dense edge networks, diffusion-based reinforcement learning can schedule serverless containers across roadside units and mobile terminals Huang et al. (2026); Yang et al. (2025c). Two-timescale designs combine long-term model-state caching across base stations with short-term computation allocation for dynamic requests Liu et al. (2026c); Yang and Chen (2025). For DLM agents, interactive continuity may require migrating model states, cached contexts, stable token states, or partial reasoning trajectories across edge nodes.

Adaptive serving adds runtime elasticity. Visual diffusion frameworks, such as FlexGen, adjust network width, inference steps, or execution paths according to real-time network and device conditions Li et al. (2025c). Energy-aware edge inference combines lightweight compression and adaptive quantization to avoid service disruption on low-power IoT devices Xie and Fang (2025). Future DLM serving should be cache-aware, privacy-aware, and context-aware because agentic DLMs must process persistent user context, multimodal observations, and multi-turn decisions under tight resource budgets. Thus, adaptive serving should be evaluated not only by communication cost, but also by interaction success, state reuse, and safe action completion.

4.4 Implications for Edge-Native DLM Agent Deployment

Existing edge diffusion systems provide useful but indirect guidance for DLM deployment Zheng (2025). The main adaptation is to reinterpret these systems through DLM-specific generation dynamics rather than treating DLMs as ordinary visual diffusion backbones. Their core mechanisms, including device-only compression, edge-assisted split inference, cloud–edge routing, collaborative denoising, pipeline parallelism, caching, and adaptive scheduling, can be transferred to DLMs at the

architectural level. Nevertheless, DLMs differ from visual DMs on several important aspects. DLMs operate over discrete or hybrid token spaces and interact heavily with KV caches Li et al. (2025f). Additionally, they often rely on masked-token refinement Yu et al. (2025a) and must support long-context memory for multi-round reasoning and action planning Li et al. (2025a). To this end, future edge DLM systems cannot simply inherit legacy image-based diffusion deployment schemes. Instead, they must be redesigned to handle discrete token-level dynamics Yu et al. (2025a). This adaptation must also account for the memory overhead of caching reasoning trajectories Li et al. (2025f), while ensuring secure device-edge-cloud state management for privacy-sensitive prompts Zheng (2025). This gap indicates a potential research direction: building deployment frameworks specifically designed for DLM-based mobile edge agents.

4.5 Lessons Learned

Edge DLM deployment should choose among device-only, edge-assisted, and cloud-edge collaboration according to privacy, latency, energy, and bandwidth constraints Zheng et al. (2025); Zheng (2025). Existing diffusion offloading offers useful primitives, but visual-generation schemes cannot be directly reused for discrete tokens, long-context states, reasoning traces, and tool calls Du et al. (2024); Li et al. (2025f). A practical design keeps private prompts and final actions on-device, offloads heavy intermediate refinement when necessary, and adapts partitioning to network conditions, task risk, token uncertainty, and interaction latency.

5 Adaptive DLM Agent Serving under Communication Constraints

This section treats communication efficiency as one deployment condition for DLM agents, rather than as the core objective by itself. The focus is on how DLM serving can adapt denoising depth, split execution, privacy exposure, and interaction latency under edge constraints.

5.1 Communication-Aware DLM Agent Inference

For edge DLM serving, limited device resources and stochastic wireless channel conditions primarily affect how iterative denoising, split inference, and interaction latency should be managed. Traditional centralized inference often incurs high communication latency and bandwidth consumption. To address these challenges, communication-aware inference leverages model partitioning and collaborative generation to optimize performance within wireless environments while preserving agent-level goals, such as time-to-first-action, privacy exposure, and action reliability.

5.1.1 DLM-Oriented Model Partitioning

Partitioning strategies reconfigure DMs to align with heterogeneous mobile hardware and fluctuating wireless states. To minimize communication costs and satisfy dynamic

latency constraints, inference frameworks optimize end-edge collaboration through fine-grained subgraph partitioning and adaptive, bandwidth-aware model width scaling Huo et al. (2026); Li et al. (2025c). Furthermore, spatial and temporal techniques, such as patch-based diffusion and split learning, facilitate elastic model distribution across mobile edge nodes Li et al. (2026). For Transformer-based architectures (e.g., U-ViT), multi-exit mechanisms allow mobile devices to adaptively truncate the denoising process during network congestion, mitigating wireless channel pressure by selecting optimal exit points Tian et al. (2025a). For DLMs, partitioning should further consider token blocks, denoising rounds, reasoning states, and tool/action boundaries rather than only the model layers.

5.1.2 Collaborative DLM Agent Generation

DLM-oriented model partitioning defines the execution boundaries, at which computation and intermediate states can be placed across devices, edge servers, and cloud resources. Building on these boundaries, collaborative generation coordinates when and what each tier executes and exchanges during the generation process. Thus, partitioning primarily addresses the placement of computation, whereas collaborative generation focuses on cross-tier orchestration of partial denoising results, intermediate states, and semantic outputs.

In practice, such collaborative generation can optimize task allocation across the mobile-edge-cloud hierarchy to reduce wireless data traffic. Decoupling the diffusion process into cloud-side semantic planning and edge-side visual refinement reduces round-trip communication Yan et al. (2024). In scenarios involving multi-round prompt evolution, DiffusionX employs lightweight on-device DMs for rapid previews while invoking cloud-based high-fidelity refinement only after the prompt is finalized, thereby reducing repeated cloud-side generation and avoiding unnecessary device–cloud interactions during iterative prompt refinement Wei et al. (2025). For dynamic environments, generative AI-assisted reinforcement learning (GAIRL) and transfer learning enable real-time optimization of offloading strategies and local denoising ratios Tian et al. (2025a). Additionally, multi-stage relay diffusion facilitates efficient task migration across heterogeneous mobile nodes under stringent communication and energy constraints Li et al. (2026). For DLM agents, these methods should be adapted to support draft-and-revise interaction and to avoid exposing permission-critical outputs before verification.

5.2 Agent-Aware Computation–Communication Co-Optimization

Deploying DLMs within mobile agentic ecosystems requires a paradigm shift from static resource allocation to DLM-aware adaptive serving under edge constraints. Unlike traditional autoregressive models, DLMs operate under a multi-step iterative paradigm in which decoding complexity scales with the number of denoising rounds T and sequence length N. Peng et al. (2025) show that this iterative generation is fundamentally memory-bound rather than purely compute-bound; repeated

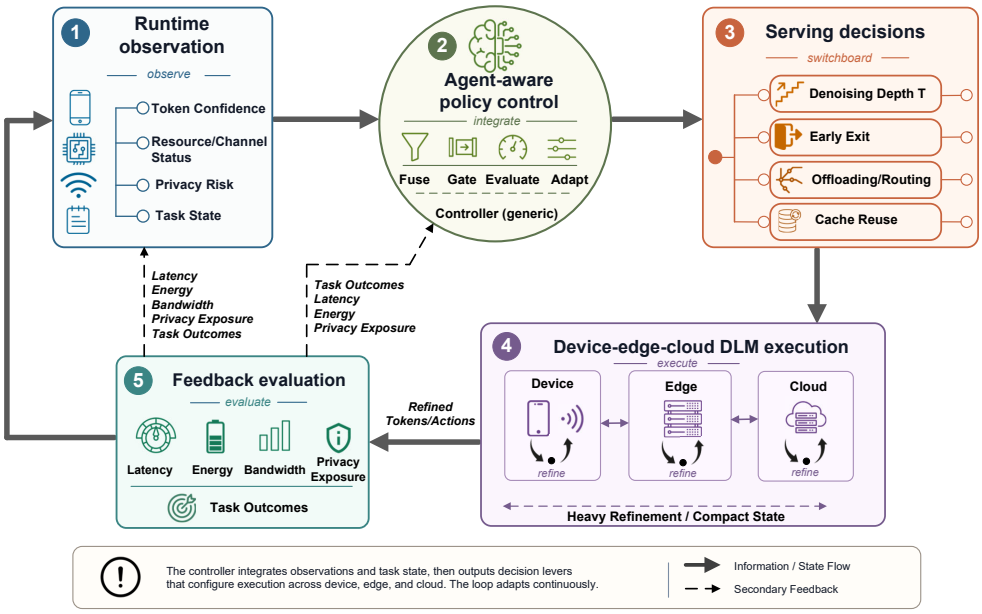


Fig. 4 Agent-aware adaptive DLM serving loop under communication constraints. Modules (1)–(5) denote runtime observation, policy control, serving decisions, tiered execution, and feedback evaluation

memory accesses across T steps increase bandwidth overhead, and standard parallel decoding strategies exhibit diminishing returns when scaled. Consequently, deployment on resource-constrained edge devices creates a combined latency-memory constraint. Strict latency deadlines and the stochastic convergence of semantic tokens require adaptive denoising schedules that accelerate or truncate generation steps when appropriate. At the same time, the trade-off between mobile battery lifetime and memory-intensive high-fidelity decoding requires energy-aware inference mechanisms, such as elastic task offloading and distributed architectures, to reduce the local hardware burden.

To clarify this dynamic serving process, Fig. 4 illustrates an agent-aware adaptive DLM serving loop under communication constraints. Instead of treating computation offloading, denoising scheduling, and cache reuse as independent decisions, the loop organizes runtime observation, policy control, serving decisions, device-edge-cloud execution, and feedback evaluation into a closed process. This closed-loop view explains how DLM agent serving can continuously adapt denoising depth, early exit, offloading/routing, and cache reuse, according to token confidence, resource/channel status, privacy risk, and task outcomes, thereby providing the basis for adaptive denoising schedules and energy-aware DLM agent inference, as will be discussed in the following subsections.

5.2.1 Adaptive Denoising Schedules

In edge environments, balancing interaction latency with generation quality is critical. For agents, adaptive denoising also determines how much refinement should be allocated to uncertain tokens, high-risk actions, or complex reasoning steps. Efficient DLM inference manipulates denoising depth based on the bits-to-rounds principle, which lower-bounds decoding rounds by the target sequence's total information content Fu et al. (2025). To overcome the low per-round information gain of traditional decoding, Fu et al. (2025) propose a training-free Explore-Then-Exploit (ETE) mechanism. By actively exploring high-information tokens to trigger confidence cascades and exploiting the induced high-confidence tokens, ETE reduces decoding rounds without compromising fidelity.

To navigate this information-theoretic boundary, modern frameworks employ semantic-aware truncation strategies that trade some refinement accuracy for lower latency. Since the latent variables in DLMs converge toward discrete word embeddings early in the reverse process, the system can terminate denoising once the entropy of the token distribution falls below a task-specific threshold. For example, the end-of-text prediction (EoTP) and learnable filtering models enable the system to halt computation on padding tokens or redundant iterations without compromising the final semantic output Bao et al. (2025). Similarly, early-skipping mechanisms leverage intermediate tensor variations to bypass computationally expensive layers for low-importance tokens, reallocating saved cycles to meet strict interaction deadlines Zhu et al. (2026). By integrating communication-aware guidance, the inference process can be guided toward outputs that are robust to packet loss or bandwidth fluctuations, helping the agent remain responsive even under degraded network conditions He et al. (2025). This adaptive scheduling transforms inference from fixed-length execution into a flexible process conditioned on token confidence, task urgency, and network state Liu et al. (2024a).

5.2.2 Energy-Aware DLM Agent Inference

Adaptive denoising schedules control how much refinement is performed for a given task, but they do not determine where this refinement should be executed. In mobile edge settings, the same denoising budget may lead to different energy and latency costs, depending on whether it is processed locally, offloaded to an edge server, or distributed across nearby devices. This issue is particularly important because multi-step DLM inference can increase mobile energy consumption through repeated memory access and computation. Energy-aware DLM inference complements adaptive scheduling by jointly considering denoising depth, computation placement, and device power budgets. Addressing this trade-off requires architectural elasticity across the device-edge continuum.

A key strategy for addressing the power-compute conflict is split diffusion, where the edge device only executes the lightweight forward noising phase, while the energy-intensive reverse denoising is offloaded to a proximal edge server Ai et al. (2026).

This offloading decision is not a simple binary choice but a multi-objective optimization involving remaining battery life, transmission power, and privacy requirements. Under strict energy budgets, systems can use deep reinforcement learning (DRL) to solve mixed-integer nonlinear programming problems, dynamically determining the optimal split points, which, in the context of DLMs, dictate how much local noising, token refinement, or final decoding should remain on the device to balance communication overhead, data obfuscation, and system lifetime Zhou et al. (2022). However, as deployments scale to heterogeneous IoT environments, this single-server paradigm becomes a bottleneck. To address this issue, multimodal parallel offloading can be implemented through content-aware semantic partitioning, which minimizes transmission overhead by isolating critical region proposals Zeng et al. (2025b). To navigate the larger action space of multi-node routing, integrating a generative diffusion module into the DRL policy enables dynamic allocation under fluctuating edge resources. For DLM agents, the routing policy should also account for token uncertainty, action risk, and privacy exposure.

Building on this multi-node foundation, the paradigm is shifting from simple task offloading toward multi-device collaborative decoding. While intra-device optimizations, such as the ETE mechanism, mitigate local compute limits, deploying the large parameter space of advanced DLMs requires architectural elasticity at the model level. A relevant approach is to apply the MoE mechanism directly to DMs. Previous studies have broadly explored the integration of MoE mechanisms with mobile edge computing Qin et al. (2025); Jin et al. (2026); Gao et al. (2026), proposing foundational solutions, such as hierarchical expert selection for device heterogeneity Jin et al. (2026), joint optimization of routing and radio resources Qin et al. (2025), and privacy-preserving federated inference Gao et al. (2026). Extending this paradigm to DLMs allows different specialized "expert" modules to be deployed across distinct proximal devices. By decoupling the dense generative backbone, the system can dynamically route intermediate token representations, or specific iterative denoising steps, to sparsely activated experts, enabling parallel processing without overloading single terminals. Fig. 5 provides a general technical architecture for sparse DLM execution through multi-device collaborative decoding. In one reverse denoising round, the current token state (x_t, t) is routed by a gating module to heterogeneous edge experts, and the weighted expert outputs are aggregated to estimate x_{t-1}.

Beyond efficiency, elastic partitioning of DLMs creates a privacy-energy-performance trade-off. By performing the initial noising stages locally, the device can provide a noising-based obfuscation layer that reduces the exposure of the original discrete text before transmission Ai et al. (2026). This can help ensure that the agent uses server-side compute without the high energy overhead of heavy encryption or the risk of exposing sensitive prompts. Thus, energy-aware inference in the DLM context requires a resource-allocation strategy in which the split point is continuously repositioned to balance device lifetime, privacy exposure, interaction latency, and generation quality Liu et al. (2024a).

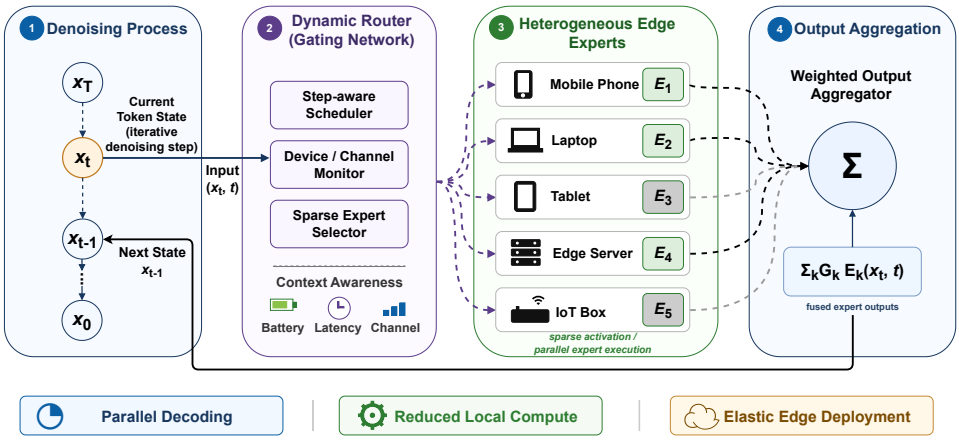


Fig. 5 Distributed DLM-MoE architecture for sparse edge-agent collaborative decoding. Numbers (1)–(4) denote token-state input, dynamic routing, parallel edge experts, and output aggregation

5.3 Lessons Learned

Adaptive DLM serving must jointly manage offloading, denoising depth, interaction latency, energy use, and privacy exposure Liu et al. (2024a); Xu et al. (2026). Fixed split points are fragile because mobile systems face channel fluctuation, mobility, contention, packet loss, and heterogeneous hardware Li et al. (2026); Zheng (2025). Future systems should use communication-aware partitioning, adaptive denoising, and energy-aware collaboration to optimize time-to-first-action, selectively refine uncertain or risky tokens, and retain permission-critical steps locally.

6 Applications and Emerging Use Cases in IoT and Wireless Systems

Since direct DLM applications in IoT and wireless systems remain limited, this section first reviews continuous DMs as established precedents for diffusion-based wireless intelligence, and then discusses how DLMs may extend this progress toward discrete semantic and agentic tasks.

6.1 Continuous DMs as Precedents for Wireless Intelligence

Continuous DMs have already shown the value of diffusion-style iterative refinement in wireless and IoT systems, especially for continuous signal modeling, resource optimization, trajectory generation, and network control. Recent surveys have systematically reviewed GDMs for wireless networks across sensing, transmission, application, and security dimensions, providing a broader context for the representative continuous-DM precedents summarized below Fan et al. (2026). Against this broader landscape, representative studies have adapted DMs across diverse wireless and IoT domains. For network security, DMs have been used to synthesize scarce attack samples and

improve intrusion detection systems Sánchez-Carballo et al. (2026). For physical and aerial operations, diffusion-based methods enable tasks such as forecasting electricity loads, restoring aerial imagery, and generating optimal unmanned aerial vehicle (UAV) trajectories Pan et al. (2025). At the infrastructure level, these models have been shown to be effective for complex radio resource management, particularly in Open Radio Access Network (O-RAN) and network slicing scenarios Yan et al. (2026). These applications operate on numerical features, physical dynamics, or resource variables rather than language tokens. They should be viewed as system-level precedents for diffusion intelligence at the edge.

6.2 DLMs for Semantic Agentic Intelligence

Building on these precedents, DLMs extend diffusion-style refinement from continuous signals and resource variables to discrete semantic and executable representations. As mobile edge environments evolve from data-driven optimization toward intent-driven semantic control, this token-level modeling capability becomes particularly relevant for instruction repair, action-token synthesis, protocol configuration, executable plan generation, and agent-level decision making.

Existing DLMs studies have demonstrated broad applicability in conventional NLP tasks, code generation, and computational biology, covering sequence generation, information extraction, summarization, dialogue, retrieval, translation, molecular modeling, and protein design Li et al. (2025f); Reyes-Ramírez et al. (2024). In addition, discrete language modeling and discrete diffusion offer a unified token-denoising interface that can extend diffusion-based generation beyond language-centric tasks. By representing multimodal observations, trajectories, actions, passwords, design topologies, or semantic codewords as discrete token sequences, DLM-style iterative refinement can support semantic reasoning, command repair, action-token synthesis, executable plan generation, and structured decision making for mobile edge agents under latency and resource constraints. Fig. 6 summarizes this capability–interface–application relationship. It first highlights four key DLM capabilities, namely bidirectional context, parallel refinement, iterative denoising, and structured controllability. These capabilities are then connected to a unified token-denoising interface over discrete token representations, through which multimodal observations, trajectories, actions, passwords, design topologies, and semantic codewords can be refined for different semantic agentic tasks. The resulting application map covers foundation tasks, robotics/vision-language-action (VLA), vehicular systems, UAV and aerial systems, and edge semantic applications.

In robotics, recent VLA studies provide direct evidence that DLMs can serve as semantic-to-action engines. LLaDA-VLA builds on pretrained diffusion-based vision-language models (VLMs) and adapts masked diffusion to robotic manipulation through localized special action-token classification and hierarchical action-structured decoding, enabling coherent action-sequence generation in simulation and real-robot tasks Wen et al. (2025b). Discrete Diffusion VLA formulates action decoding as discrete diffusion over tokenized action chunks inside a unified Transformer, where high-confidence action tokens are fixed first and uncertain ones are re-masked for

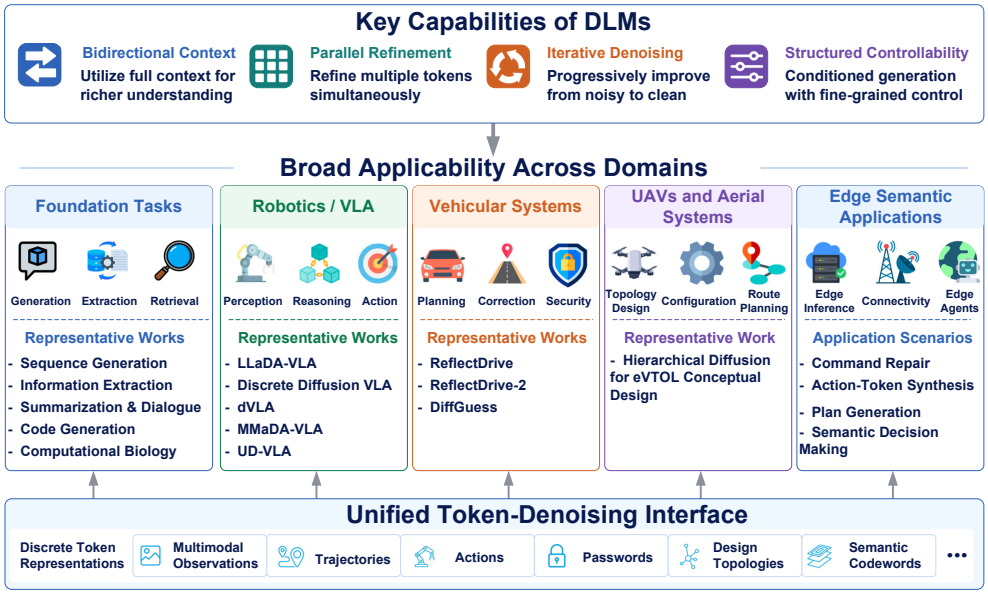


Fig. 6 Applications of DLMs for semantic agentic intelligence through a unified token-denoising interface

later refinement Liang et al. (2025c). Similarly, dVLA adopts a discrete diffusion language backbone with separate tokenizers for images, text, and actions, and introduces multimodal chain-of-thought generation to jointly optimize visual subgoals, textual reasoning, and robot actions Wen et al. (2025a). More recent unified VLA models, such as MMaDA-VLA and UD-VLA, extend this direction by mapping language, images, and continuous robot controls into a shared discrete token space and synchronously denoising future visual observations and action chunks Liu et al. (2026a); Chen et al. (2026). These studies suggest that DLMs can transform robotic control from sequential action prediction into parallel, revisable, and semantically grounded action refinement.

In vehicular networks and autonomous driving, DLMs have begun to support trajectory-level semantic planning and safety-aware correction. ReflectDrive discretizes the two-dimensional driving space into an action codebook, fine-tunes pretrained DLMs for trajectory planning, and introduces a reflection mechanism that identifies unsafe trajectory tokens and regenerates them through gradient-free inpainting Li et al. (2025d). ReflectDrive-2 extends this direction with a decision–draft–reflect pipeline, where masked discrete diffusion generates editable trajectory drafts and AutoEdit rewrites selected tokens within the same discrete action space; it further couples drafting and editing through reinforcement learning and improves runtime efficiency with shared-prefix KV reuse and fused on-device unmasking Wang et al. (2026). Beyond driving control, DiffGuess applies a DiffuSeq-based DLM to V2X and intelligent transportation systems (ITS) password strength assessment by modeling password modification as an iterative diffusion process, showing that DLM-style

refinement can also enhance authentication security in ITS Xiong et al. (2025). These works demonstrate that DLMs can support vehicular agents, not only in planning and correction but also in semantic security and access-control reasoning.

For UAVs and aerial edge systems, existing DLM-related evidence is still preliminary but technically relevant. In eVTOL aircraft conceptual design, a hierarchical diffusion framework combines a Riemannian DLM with masked diffusion to sample discrete aircraft topologies and corresponding continuous parameters under simulation-based inference Ghiglino et al. (2026). Although this work focuses on aircraft design rather than real-time UAV networking, it illustrates how DLM-like discrete diffusion can model topology-level design choices, variable-dimensional structures, and constraint-conditioned generation. This suggests a possible direction for future UAV edge agents, where mission plans, swarm formations, component configurations, and route constraints can also be represented as discrete structured tokens and refined through iterative denoising.

6.3 Lessons Learned

Continuous DMs have already demonstrated the value of diffusion-style refinement for numerical signals, physical dynamics, trajectory generation, resource allocation, and network optimization in IoT and wireless systems Pan et al. (2025); Liang et al. (2025b). DLMs extend this paradigm toward discrete semantic and executable structures, making them potentially useful for command repair, structured API and tool generation, semantic coordination, protocol configuration, and intent translation Li et al. (2025f); Huh et al. (2026); Habib et al. (2026). Since current DLM applications in IoT and wireless systems remain early and often indirect, they should be viewed as semantic and action-level complements to continuous DMs, and practical edge DLM agents should integrate safety constraints, execution verification, closed-loop feedback, and real-device validation.

7 Evaluation Metrics, Benchmarks, and Frameworks

Existing evaluation protocols for LLMs traditionally emphasize static foundation capabilities. For instance, Massive Multitask Language Understanding (MMLU) and Holistic Evaluation of Language Models (HELM) baseline general reasoning and knowledge Hendrycks et al. (2021); Liang et al. (2023), while frameworks like AgentBench measure instruction following and coding proficiency Liu et al. (2024b). However, mobile edge agentic AI requires a broader perspective that jointly considers generation mechanisms, interaction latency, resource constraints, and autonomous task execution. Consequently, recent benchmarks have introduced more interactive setups: Mobile-Bench evaluates multi-application workflows Deng et al. (2024), AndroidWorld tests executable mobile UI states Rawles et al. (2025), and OSWorld incorporates process-aware rewards for open-ended environments Xie et al. (2024). Furthermore, MLPerf Mobile highlights the importance of hardware-aware and reproducible deployment metrics Janapa Reddi et al. (2022). This section organizes evaluation around structure-aware comparison among autoregressive models, general DMs, and DLMs.

7.1 Structure-Aware Evaluation with Emphasis on DLMs

Structure-aware evaluation links generation mechanisms to deployability. Autoregressive models require sequential decoding and KV-cache management, so their edge performance depends on prefill latency, decoding latency, cache growth, and time-to-first-action. General DMs generate through iterative denoising, which makes sampling steps, function evaluations, energy per sample, and downstream utility important for mobile deployment Ho et al. (2020); Rombach et al. (2022). DLMs combine language generation with iterative refinement, bidirectional token prediction, and mask scheduling, so they should not be evaluated as ordinary LLM backbones.

For DLMs, high language accuracy alone is insufficient. Models, such as LLaDA and DiffuLLaMA, show strong language and reasoning potential Nie et al. (2025); Gong et al. (2025), but practical efficiency can be weakened by repeated denoising, remasking instability, limited cache compatibility, and inconsistent latency protocols Peng et al. (2025). To address this issue, a structure-aware evaluation protocol for DLMs should jointly report model-level quality and generation-process behavior, including denoising-step efficiency, mask-schedule sensitivity, parallel decoding gain, early-exit quality, cache compatibility, and quality-latency-energy trade-offs.

A recent inference-only analysis of the public DiffusionGemma checkpoint illustrates why such process-level reporting is necessary: its token commitment is neither strictly parallel nor strictly sequential, but exhibits a task- and granularity-dependent left-to-right bias, large commit batches, remasking events, and entropy-based commitment behavior Asaria et al. (2026). For edge evaluation, this suggests that DLM reports should include commit-order statistics, same-call commit ratios, entropy or confidence signals, denoising-step budgets, and task-regime differences, rather than relying only on aggregate accuracy or throughput.

7.2 Metrics and Benchmarks

Given these structure-specific differences, evaluation metrics should not be selected as isolated accuracy scores. Instead, they should answer practical deployment questions: whether the model can reason, whether the generation process is stable, whether interaction succeeds, and whether the system remains efficient under edge constraints. Accordingly, rather than enumerating metrics separately from benchmarks, we group them by the practical questions they answer. Capability benchmarks measure whether the backbone can reason, follow instructions, code, or remain factual Chen et al. (2021); agent and mobile benchmarks assess UI grounding, tool use, multi-step task completion, and error recovery; and edge benchmarks evaluate whether these capabilities remain usable under latency, memory, energy, thermal, and communication constraints.

For DLM-based mobile agents, benchmark reports should also specify generation-specific and system-specific settings, including denoising steps, mask schedule, output length, prompt length, batch size, cache strategy, quantization level, runtime, accelerator, and power measurement method. Without these details, claims about DLM speed, energy efficiency, or deployability are difficult to compare in a fair fashion, especially when similar aggregate scores may hide sample-level instability caused by

Table 8 A structure-aware evaluation framework for mobile edge agentic AI.

Evaluation Dimension	Main Focus	Representative Elements
Backbone Capability	Evaluate the intrinsic capability of the foundation model Hendrycks et al. (2021); Liang et al. (2023); Lin et al. (2022)	Knowledge, reasoning, coding, factuality, instruction following
Generation Dynamics	Evaluate how outputs are generated under different model structures Ho et al. (2020); Nie et al. (2025); Peng et al. (2025)	**Autoregressive:** decoding latency, KV-cache cost **Diffusion:** denoising cost, sampling efficiency **DLMs:** step count, mask schedule, cache compatibility, early exit
Mobile Interaction	Evaluate closed-loop task execution in mobile environments Xie et al. (2024); Deng et al. (2024); Rawles et al. (2025)	UI grounding, tool/API use, planning, action execution, feedback, recovery
Edge Deployment	Evaluate deployability on mobile and edge platforms Janapa Reddi et al. (2022)	Latency, energy, memory, communication cost, thermal behavior, offloading
Safety and Reliability	Evaluate trustworthiness and robustness of mobile edge agents Lin et al. (2022); Rawles et al. (2025); Fang et al. (2026)	Unsafe actions, permission violations, privacy leakage, adversarial UI robustness, non-determinism, reproducibility

diffusion steps, sampling strategies, batch size, hardware, or numerical precision Fang et al. (2026).

7.3 Evaluation Framework

The metrics discussed above cover heterogeneous aspects of model capability, generation behavior, interaction, deployment, and safety. To make these aspects easier to apply in practice, we organize them into a layered evaluation framework. Table 8 summarizes the above discussion into five layers covering backbone capability, generation dynamics, mobile interaction, edge deployment, and safety and reliability. The table is intended as a compact checklist, rather than a fixed benchmark, while keeping these dimensions comparable across autoregressive models, general DMs, and DLMs.

Recent DLM acceleration studies further justify this layered framework. Consistency Diffusion Language Models (CDLM) improves sampling with consistency modeling, block-wise causal attention, multi-token finalization, and cache compatibility Kim et al. (2026), while R2-dLLM reduces redundant spatio-temporal decoding by finalizing confident and stable tokens Du et al. (2026). These studies show that comprehensive DLM evaluation must jointly report accuracy, denoising trajectory, caching behavior, latency, and hardware efficiency. Overall, Table 8 provides a concise basis for comparing autoregressive models, general DMs, and DLMs under the practical requirements of mobile edge agentic AI.

7.4 Lessons Learned

DLM-based mobile edge agents should be evaluated beyond model accuracy, with generation dynamics, interaction reliability, system efficiency, and safety measured jointly Peng et al. (2025); Fang et al. (2026). High benchmark scores do not imply deployability unless latency, memory, energy, communication, cache behavior, and tool execution are reported on real hardware Rawles et al. (2025); Janapa Reddi et al. (2022). Future benchmarks should standardize fixed-latency and fixed-energy settings, time-to-first-action, invalid-action rates, wireless dynamics, and reproducible hardware measurements.

8 Open Challenges and Future Directions

DLMs offer useful properties for mobile edge agentic AI, including non-autoregressive generation, bidirectional context modeling, parallel token refinement, and adaptive denoising Li et al. (2025f); Yu et al. (2025a). However, most current studies remain model-centric and cloud-oriented, while edge systems require tight control of latency, memory, energy, bandwidth, privacy, and reliability Zheng et al. (2025); Peng et al. (2025). This section summarizes the main challenges and directions toward edge-native DLM systems.

8.1 Edge-Native DLM Architectures

A core challenge is that iterative denoising repeatedly accesses memory and may offset the parallelism gained from simultaneous token updates Peng et al. (2025); Wu et al. (2025). This issue is more severe on mobile and IoT devices, where memory bandwidth, cache behavior, and accelerator support often dominate latency and energy. Future DLM architectures should jointly optimize denoising depth, token update ratio, mask schedule, cache compatibility, and hardware-friendly operators. Lightweight backbones, block-wise diffusion, sparse routing, early exit, and anytime generation are potential directions for adapting generation quality to real-time resource budgets Arriola et al. (2025); Zhu et al. (2026).

8.2 Long-Context Memory and Agent State Management

Mobile edge agents must process multi-turn instructions, UI histories, sensor observations, tool calls, and previous actions. While autoregressive LLMs mainly suffer from KV-cache growth, DLMs face full-sequence refinement, repeated remasking, and multi-step token updates, which increase memory-bandwidth pressure on constrained devices Li et al. (2025a); Peng et al. (2025). Future systems should avoid refining all tokens at every step. Instead, they should identify stable tokens, selectively remask uncertain regions, compress historical context, and cache reusable agent states. Cache-aware and confidence-aware decoding also provides a useful starting point for efficient long-context DLM agents Wu et al. (2025); Du et al. (2026).

8.3 Communication-Aware Split and Distributed DLM Inference

Existing edge diffusion deployment primarily targets image or video generation, where latent features or denoising stages are split across devices and servers Du et al. (2024); Zheng (2025). DLMs are harder to distribute because they involve discrete tokens, reasoning traces, tool-call structures, and multimodal semantic states. Future distributed DLM systems should assign denoising steps, token blocks, or expert modules across devices, edge servers, and cloud nodes according to bandwidth, channel quality, latency, and privacy. A practical direction is to keep private prompt encoding and final action decoding on-device, while offloading heavy intermediate refinement or sparse experts to nearby edge nodes Qin et al. (2025); Jin et al. (2026); Liu et al. (2024a).

8.4 Privacy, Security, and Trustworthy Edge DLM Agents

DLMs may support privacy-preserving split inference because forward noising and intermediate denoising states can obscure raw prompts before offloading Allmendinger et al. (2026); Dockhorn et al. (2023). However, token distributions, latent states, reasoning traces, and tool-call intentions may still leak sensitive information. The risk is amplified when edge agents control real applications, devices, or physical environments. Future research should integrate privacy-preserving noising, secure split inference, trusted execution, and formal leakage analysis into distributed DLM pipelines. Iterative refinement also enables step-wise safety checking, constraint enforcement, and rollback before final action execution, which should be evaluated in realistic mobile-agent environments Deng et al. (2024); Rawles et al. (2025).

8.5 Multimodal and Physical-World DLM Agents

Future mobile edge agents must handle camera inputs, speech, sensor streams, wireless signals, maps, UI states, and executable actions. Recent multimodal DLMs show that text reasoning, multimodal understanding, and text-to-image generation can be unified within diffusion architectures Yang et al. (2025a); Tian et al. (2025b). Yet, real-time deployment on mobile and IoT platforms remains underexplored. A potential direction is to represent instructions, observations, sensor readings, and actions as structured tokens refined through shared denoising. Such systems may support semantic communication, mobile assistants, vehicular coordination, UAV swarms, smart homes, and intent-driven network management, but require strict latency control, robustness, safety guarantees, closed-loop evaluation, and real-device validation Huh et al. (2026); Habib et al. (2026).

8.6 Standardized Evaluation and Reproducibility

Current benchmarks evaluate language ability, mobile interaction, or hardware efficiency separately, but DLM-based edge agents require joint evaluation of generation dynamics and system constraints Rawles et al. (2025); Janapa Reddi et al. (2022); Peng et al. (2025). Future benchmarks should report denoising steps, mask schedules,

remasking stability, parallel decoding gain, cache behavior, energy use, communication overhead, and closed-loop task success. They should also include time-to-first-action, invalid action rate, privacy leakage risk, and safety failures. Reproducibility requires fixed-latency and fixed-energy protocols, together with hardware platform, accelerator type, quantization level, prompt length, output length, batch size, and power measurement details Fang et al. (2026); Janapa Reddi et al. (2022).

9 Conclusion

DLMs offer new control dimensions for mobile edge agents, including parallel token refinement, adaptive denoising, constrained generation, and privacy-aware split execution. This survey has reviewed their modeling foundations, resource-efficient techniques, edge deployment architectures, communication-aware serving, IoT/wireless applications, and evaluation criteria. Our analysis suggests that DLMs should complement rather than replace autoregressive LLMs: they are suitable for structured, constraint-aware, and latency-adaptive tasks, but still face challenges in long-context state management, hardware-efficient inference, distributed collaboration, safety, multimodal grounding, and reproducibility. This survey aims to connect diffusion language modeling, edge computing, wireless networking, and trustworthy agentic AI.

Declarations

- **Funding:** This work was supported by the National Natural Science Foundation of China under Grants 62571529 and U25A20388.
- **Competing interests:** The authors have no competing interests to declare that are relevant to the content of this article.
- **Author contributions:** Conceptualization, C.L., M.M., D.N. and W.N.; methodology, C.L., M.M. and D.N.; investigation, C.L.; data curation, C.L.; visualization, C.L.; writing—original draft preparation, C.L.; writing—review and editing, C.L., M.M., D.N. and W.N.; supervision, M.M., D.N. and W.N.; project administration, M.M.; funding acquisition, M.M. All authors have read and approved the final manuscript.
- **Data availability:** No datasets were generated or analysed during the current study.
- **Ethics approval and consent to participate:** Not applicable.
- **Consent for publication:** Not applicable.
- **Materials availability:** Not applicable.
- **Code availability:** Not applicable.

References

Ahsan MM, Raman S, Liu Y, et al (2025) A comprehensive survey on diffusion models and their applications. Applied Soft Computing 181:113470. https://doi.org/10.1016/j.asoc.2025.113470

Ai Y, Chen Q, Wen D, et al (2026) Cross-modal collaborative diffusion models for distributed AI-generated content. IEEE Transactions on Cognitive Communications and Networking 12:5552–5565. https://doi.org/10.1109/TCCN.2026.3656389

Allmendinger S, Zipperling D, Struppek L, et al (2026) CollaFuse: Collaborative diffusion models. URL https://arxiv.org/abs/2406.14429, arXiv:2406.14429

Arriola M, Gokaslan A, Chiu J, et al (2025) Block diffusion: Interpolating between autoregressive and diffusion language models. In: International Conference on Learning Representations, pp 50726–50753

Asaria A, Salomone T, Gandhi D (2026) Neither parallel nor sequential: How DiffusionGemma actually commits tokens. URL https://arxiv.org/abs/2606.14620, arXiv:2606.14620

Austin J, Johnson DD, Ho J, et al (2021) Structured denoising diffusion models in discrete state-spaces. In: Advances in Neural Information Processing Systems, pp 17981–17993

Bao F, Nie S, Xue K, et al (2023) All are worth words: A ViT backbone for diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 22669–22679

Bao W, Chen Z, Xu D, et al (2025) Learning to parallel: Accelerating diffusion large language models via learnable parallel decoding. URL https://arxiv.org/abs/2509.25188, arXiv:2509.25188

Bolya D, Hoffman J (2023) Token merging for fast Stable Diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp 4599–4603

Brown T, Mann B, Ryder N, et al (2020) Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol 33. Curran Associates, Inc., pp 1877–1901

Chang H, Zhang H, Jiang L, et al (2022) MaskGIT: Masked generative image transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11315–11325

Chen J, Song W, Ding P, et al (2026) Unified diffusion VLA: Vision-language-action model via joint discrete denosing diffusion process. In: The Fourteenth International Conference on Learning Representations

Chen M, Tworek J, Jun H, et al (2021) Evaluating large language models trained on code. URL https://arxiv.org/abs/2107.03374, arXiv:2107.03374

Croitoru FA, Hondru V, Ionescu RT, et al (2023) Diffusion models in vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 45(9):10850–10869. https://doi.org/10.1109/TPAMI.2023.3261988

Deng S, Zhao H, Fang W, et al (2020) Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet of Things Journal 7(8):7457–7469. https://doi.org/10.1109/JIOT.2020.2984887

Deng S, Xu W, Sun H, et al (2024) Mobile-Bench: An evaluation benchmark for LLM-based mobile agents. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 8813–8831, https://doi.org/10.18653/v1/2024.acl-long.478

Devlin J, Chang MW, Lee K, et al (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp 4171–4186, https://doi.org/10.18653/v1/N19-1423

Dockhorn T, Cao T, Vahdat A, et al (2023) Differentially private diffusion models. Transactions on Machine Learning Research

Du H, Zhang R, Niyato D, et al (2024) Exploring collaborative distributed diffusion-based AI-generated content (AIGC) in wireless networks. IEEE Network 38(3):178–186. https://doi.org/10.1109/MNET.006.2300223

Du Z, Xia K, Zhong X, et al (2026) R2-dLLM: Accelerating diffusion large language models via spatio-temporal redundancy reduction. URL https://arxiv.org/abs/2604.18995, arXiv:2604.18995

Fan D, Meng R, Xu X, et al (2026) Generative diffusion models for wireless networks: Fundamental, architecture, and state-of-the-art. IEEE Communications Surveys & Tutorials 28:5632–5677. https://doi.org/10.1109/COMST.2026.3671110

Fang G, Ma X, Wang X (2023) Structural pruning for diffusion models. In: Thirty-seventh Conference on Neural Information Processing Systems

Fang J, Pan J, Li A, et al (2025) PipeFusion: Patch-level pipeline parallelism for diffusion transformers inference. In: Advances in Neural Information Processing Systems, pp 98434–98455

Fang Z, Jiang Z, Chen H, et al (2026) Dataset-level metrics attenuate non-determinism: A fine-grained non-determinism evaluation in diffusion language models. URL https://arxiv.org/abs/2604.13413, arXiv:2604.13413

Fei Z, Fan M, Yu C, et al (2024) Scaling diffusion transformers to 16 billion parameters. URL https://arxiv.org/abs/2407.11633, arXiv:2407.11633

Feng G, Geng Y, Guan J, et al (2025a) Theoretical benefit and limitation of diffusion language model. In: Advances in Neural Information Processing Systems, pp 24415–24459

Feng W, Zhang R, Zhu Y, et al (2025b) Exploring collaborative diffusion model inferring for AIGC-enabled edge services. IEEE Transactions on Cognitive Communications and Networking 11(2):946–960. https://doi.org/10.1109/TCCN.2024.3519320

Fu H, Huang B, Adams V, et al (2025) From bits to rounds: Parallel decoding with exploration for diffusion language models. URL https://arxiv.org/abs/2511.21103, arXiv:2511.21103

Gao S, Zhang S, Jing S, et al (2026) Towards efficient federated learning of networked mixture-of-experts for mobile edge computing. URL https://arxiv.org/abs/2511.01743, arXiv:2511.01743

Ghiglino A, Elenius D, Roy A, et al (2026) Do diffusion models dream of electric planes? Discrete and continuous simulation-based inference for aircraft design. URL https://arxiv.org/abs/2603.13284, arXiv:2603.13284

Gong S, Li M, Feng J, et al (2023) DiffuSeq: Sequence to sequence text generation with diffusion models. URL https://arxiv.org/abs/2210.08933, arXiv:2210.08933

Gong S, Agarwal S, Zhang Y, et al (2025) Scaling diffusion language models via adaptation from autoregressive models. In: International Conference on Learning Representations, pp 5046–5073

Google (2026a) DiffusionGemma 26B-A4B-IT Model Card. https://huggingface.co/google/diffusiongemma-26B-A4B-it, accessed: 2026-06-26

Google (2026b) DiffusionGemma: 4x faster text generation. https://blog.google/innovation-and-ai/technology/developers-tools/diffusion-gemma-faster-text-generation/, accessed: 2026-06-26

Grootendorst M (2022) BERTopic: Neural topic modeling with a class-based TF-IDF procedure. URL https://arxiv.org/abs/2203.05794, arXiv:2203.05794

Gruver N, Finzi M, Qiu S, et al (2023) Large language models are zero-shot time series forecasters. In: Advances in Neural Information Processing Systems, pp 19622–19635

Habib MA, Elsayed M, Ozcan Y, et al (2026) Generative AI for intent-driven network management in 6G RAN: A case study on the Mamba model. IEEE Wireless Communications pp 1–8. https://doi.org/10.1109/MWC.2026.3667666

Han X, Kumar S, Tsvetkov Y, et al (2023) 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). Association for Computational Linguistics, pp 11575–11596, https://doi.org/10.18653/v1/2023.acl-long.647

He J, Chen K, Wang B, et al (2025) Communication-aware diffusion models for multi-agent trajectory forecasting in connected and autonomous vehicles. In: Proceedings of the Tenth ACM/IEEE Symposium on Edge Computing, https://doi.org/10.1145/3769102.3774636

He P, Gao J, Chen W (2023a) DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In: The Eleventh International Conference on Learning Representations

He Y, Liu L, Liu J, et al (2023b) PTQD: Accurate post-training quantization for diffusion models. Advances in Neural Information Processing Systems 36:13237–13249

Hendrycks D, Burns C, Basart S, et al (2021) Measuring massive multitask language understanding. In: International Conference on Learning Representations

Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems, pp 6840–6851

Hoefler MA, Mazouka T, Mueller K, et al (2024) Boosting federated learning with diffusion models for non-IID and imbalanced data. In: 2024 IEEE International Conference on Big Data (BigData), IEEE, pp 7790–7799

Hu D, Gupta A, Gabidolla M, et al (2026) SnapGen++: Unleashing diffusion transformers for efficient high-fidelity image generation on edge devices. URL https://arxiv.org/abs/2601.08303, arXiv:2601.08303

Hu EJ, Shen Y, Wallis P, et al (2022) LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations, URL https://openreview.net/forum?id=nZeVKeeFYf9

Hu Y, Ye D, Kang J, et al (2025) A cloud–edge collaborative architecture for multimodal LLM-based advanced driver assistance systems in IoT networks. IEEE Internet of Things Journal 12(10):13208–13221. https://doi.org/10.1109/JIOT.2024.3509628

Huang X, Zou R, Zhong W, et al (2026) Diffusion-based deep reinforcement learning for service scheduling in serverless vehicular edge computing. IEEE Internet of Things Journal 13(5):8424–8435. https://doi.org/10.1109/JIOT.2025.3641386

Huh Y, Kang J, Choi W (2026) Markov-enforced discrete diffusion model for digital semantic symbol error correction. URL https://arxiv.org/abs/2603.22983, arXiv:2603.22983

Huo D, Zhou Y, Hao Y, et al (2026) Multi-modal model partition strategy for end-edge collaborative inference. Journal of Parallel and Distributed Computing 208:105189. https://doi.org/10.1016/j.jpdc.2025.105189

Janapa Reddi V, Kanter D, Mattson P, et al (2022) MLPerf mobile inference benchmark: An industry-standard open-source machine learning benchmark for on-device AI. In: Proceedings of Machine Learning and Systems, pp 352–369

Jiang AQ, Sablayrolles A, Mensch A, et al (2023) Mistral 7B. URL https://arxiv.org/abs/2310.06825, arXiv:2310.06825

Jiang AQ, Sablayrolles A, Roux A, et al (2024) Mixtral of experts. URL https://arxiv.org/abs/2401.04088, arXiv:2401.04088

Jin L, Zhang Y, Li Y, et al (2026) MoE^2: Optimizing collaborative inference for edge large language models. IEEE Transactions on Networking 34:4637–4651. https://doi.org/10.1109/TON.2026.3677371

Jin M, Wang S, Ma L, et al (2024) Time-LLM: Time series forecasting by reprogramming large language models. In: Kim B, Yue Y, Chaudhuri S, et al (eds) International Conference on Learning Representations, pp 23857–23880

Jung E, Kim B, Kim H, et al (2026) Accelerating diffusion via hybrid data-pipeline parallelism based on conditional guidance scheduling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 9374–9383

Karras T, Aittala M, Aila T, et al (2022) Elucidating the design space of diffusion-based generative models. In: Advances in Neural Information Processing Systems, pp 26565–26577

Karunanayake B, Khalil I, Yi X, et al (2025) Toward LLM-driven adaptive policy orchestration for host-based intrusion detection systems in IoT environments. IEEE Network 39(5):66–73. https://doi.org/10.1109/MNET.2025.3579532

Kim B, Lee K, Jeong I, et al (2025) On-device Sora: Enabling training-free diffusion-based text-to-video generation for mobile devices. URL https://arxiv.org/abs/2502.04363, arXiv:2502.04363

Kim M, Xu C, Hooper C, et al (2026) CDLM: Consistency diffusion language models for faster sampling. URL https://arxiv.org/abs/2511.19269, arXiv:2511.19269

Kodavanti S, Arveti M, Vajrala S, et al (2026) EdgeDiT: Hardware-aware diffusion transformers for efficient on-device image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp 3801–3809

Kong Y, Yang P, Qin X, et al (2026) Distributed and controllable mobile text-to-image generation with user preference guarantee. IEEE Transactions on Mobile Computing 25(3):3712–3727. https://doi.org/10.1109/TMC.2025.3620352

Lai B, He J, Kang J, et al (2024) On-demand quantization for green federated generative diffusion in mobile edge networks. In: ICC 2024 - IEEE International Conference on Communications, pp 2883–2888, https://doi.org/10.1109/ICC51166.2024.10622695

Leviathan Y, Kalman M, Matias Y (2023) Fast inference from transformers via speculative decoding. In: Proceedings of the 40th International Conference on Machine Learning, vol 202. PMLR, pp 19274–19286

Li G (2026) FastUSP: A multi-level collaborative acceleration framework for distributed diffusion model inference. URL https://arxiv.org/abs/2602.10940, arXiv:2602.10940

Li H, Li Y, Tian A, et al (2025a) A survey on large language model acceleration based on KV cache management. URL https://arxiv.org/abs/2412.19442, arXiv:2412.19442

Li M, Cai T, Cao J, et al (2024) DistriFusion: Distributed parallel inference for high-resolution diffusion models. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 7183–7193, https://doi.org/10.1109/CVPR52733.2024.00686

Li N, Yang W, Siew M, et al (2025b) Edge-assisted collaborative fine-tuning for multi-user personalized artificial intelligence generated content (aigc). URL https://arxiv.org/abs/2508.04745, arXiv:2508.04745

Li P, Dong H, Qian L, et al (2025c) FlexGen: Efficient on-demand generative AI service with flexible diffusion model in mobile edge networks. IEEE Transactions on Cognitive Communications and Networking 11(2):961–973. https://doi.org/10.1109/TCCN.2024.3522084

Li P, Zheng Y, Wang Y, et al (2025d) Discrete diffusion for reflective vision-language-action models in autonomous driving. URL https://arxiv.org/abs/2509.20109, arXiv:2509.20109

Li P, Qian L, Niyato D, et al (2026) Toward resource-efficient collaboration of large AI models in mobile edge networks. IEEE Network 40(3):51–59. https://doi.org/10.1109/MNET.2025.3650049

Li S, Kallidromitis K, Bansal H, et al (2025e) LaViDa: A large diffusion language model for multimodal understanding. In: Advances in Neural Information Processing Systems, pp 105101–105134, URL https://proceedings.neurips.cc/paper_files/paper/2025/file/975affbe7b5f3b55fba62247b6877b1c-Paper-Conference.pdf

Li T, Chen M, Guo B, et al (2025f) A survey on diffusion language models. URL https://arxiv.org/abs/2508.10875, arXiv:2508.10875

Li X, Thickstun J, Gulrajani I, et al (2022) Diffusion-LM improves controllable text generation. In: Advances in Neural Information Processing Systems, pp 4328–4343

Li X, Liu Y, Lian L, et al (2023a) Q-Diffusion: Quantizing diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 17535–17545

Li Y, Wang H, Jin Q, et al (2023b) SnapFusion: Text-to-image diffusion model on mobile devices within two seconds. In: Advances in Neural Information Processing Systems, pp 20662–20678

Liang P, Bommasani R, Lee T, et al (2023) Holistic evaluation of language models. Transactions on Machine Learning Research

Liang R, Yang B, Chen P, et al (2025a) GDSG: Graph diffusion-based solution generator for optimization problems in MEC networks. IEEE Transactions on Mobile Computing 24(10):10264–10277. https://doi.org/10.1109/TMC.2025.3568248

Liang R, Yang B, Chen P, et al (2025b) Diffusion models as network optimizers: Explorations and analysis. IEEE Internet of Things Journal 12(10):13183–13193. https://doi.org/10.1109/JIOT.2025.3528955

Liang Z, Li Y, Yang T, et al (2025c) Discrete diffusion VLA: Bringing discrete diffusion to action decoding in vision-language-action policies. URL https://arxiv.org/abs/2508.20072, arXiv:2508.20072

Lin S, Hilton J, Evans O (2022) TruthfulQA: Measuring how models mimic human falsehoods. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 3214–3252, https://doi.org/10.18653/v1/2022.acl-long.229

Lipman Y, Chen R, Ben-Hamu H, et al (2023) Flow matching for generative modeling. In: International Conference on Learning Representations

Liu H, Wu J, Zhuang X, et al (2024a) Joint communication and computation scheduling for MEC-enabled AIGC services based on generative diffusion model. In: 2024 22nd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pp 345–352

Liu X, Yu H, Zhang H, et al (2024b) AgentBench: Evaluating LLMs as agents. In: Kim B, Yue Y, Chaudhuri S, et al (eds) International Conference on Learning Representations, pp 52989–53046

Liu Y, Ott M, Goyal N, et al (2019) RoBERTa: A robustly optimized BERT pretraining approach. URL https://arxiv.org/abs/1907.11692, arXiv:1907.11692

Liu Y, Ding P, Jiang T, et al (2026a) MMaDA-VLA: Large diffusion vision-language-action model with unified multi-modal instruction and generation. URL https://arxiv.org/abs/2603.25406, arXiv:2603.25406

Liu Y, Xiao L, Wang C, et al (2026b) Reinforcement learning-based edge-assisted inference with multimodal data. IEEE Transactions on Mobile Computing 25(4):5016–5031. https://doi.org/10.1109/TMC.2025.3627276

Liu Z, Du H, Hou X, et al (2026c) Two-timescale model caching and resource allocation for edge-enabled AI-generated content services. IEEE Transactions on Mobile Computing 25(4):4822–4838. https://doi.org/10.1109/TMC.2025.3627263

Lou A, Meng C, Ermon S (2024) Discrete diffusion modeling by estimating the ratios of the data distribution. In: International Conference on Machine Learning. JMLR.org, ICML'24

Lu C, Zhou Y, Bao F, et al (2022) DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In: Advances in Neural Information Processing Systems, pp 5775–5787

Luo S, Tan Y, Huang L, et al (2023) Latent consistency models: Synthesizing high-resolution images with few-step inference. URL https://arxiv.org/abs/2310.04378, arXiv:2310.04378

Luong NC, Hai ND, Le DV, et al (2025) Diffusion models for future networks and communications: A comprehensive survey. URL https://arxiv.org/abs/2508.01586, arXiv:2508.01586

Ma Z, Zhang Y, Jia G, et al (2025) Efficient diffusion models: A comprehensive survey from principles to practices. IEEE Transactions on Pattern Analysis and Machine Intelligence 47(9):7506–7525. https://doi.org/10.1109/TPAMI.2025.3569700

Mendieta M, Sun G, Chen C (2025) Navigating heterogeneity and privacy in one-shot federated learning with diffusion models. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp 2601–2610, https://doi.org/10.1109/WACV61041.2025.00258

Meng C, Rombach R, Gao R, et al (2023) On distillation of guided diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14297–14306

Meta AI (2024) Llama 3.1 Model Card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md, accessed: 2026-05-18

Miles R, Toker A, Oncescu AM, et al (2026) Test-time scaling with diffusion language models via reward-guided stitching. URL https://arxiv.org/abs/2602.22871, arXiv:2602.22871

Nie S, Zhu F, You Z, et al (2025) Large language diffusion models. In: Advances in Neural Information Processing Systems, pp 50608–50646

OpenAI, Achiam J, Adler S, et al (2024) GPT-4 technical report. URL https://arxiv.org/abs/2303.08774, arXiv:2303.08774

Pan H, Lin B, Liu Y, et al (2025) Diffusion-model-enhanced multiobjective optimization for improving forest monitoring efficiency in UAV-enabled Internet of Things. IEEE Internet of Things Journal 12(19):40980–40996. https://doi.org/10.1109/JIOT.2025.3590507

Park B, Go H, Kim JY, et al (2025) Switch diffusion transformer: Synergizing denoising tasks with sparse mixture-of-experts. In: Computer Vision – ECCV 2024, Cham, pp 461–477, https://doi.org/10.1007/978-3-031-43546-9_27

Peng H, Liu P, Dong Z, et al (2025) How efficient are diffusion language models? A critical examination of efficiency evaluation practices. URL https://arxiv.org/abs/2510.18480, arXiv:2510.18480

Qin HL, Dai J, Lu G, et al (2026) Generative AI meets 6G and beyond: Diffusion models for semantic communications. IEEE Communications Surveys & Tutorials 28:6241–6281. https://doi.org/10.1109/COMST.2026.3690542

Qin S, Wu H, Du H, et al (2025) Optimal expert selection for distributed mixture-of-experts at the wireless edge. URL https://arxiv.org/abs/2503.13421, arXiv:2503.13421

Rawles C, Clinckemaillie S, Chang Y, et al (2025) AndroidWorld: A dynamic benchmarking environment for autonomous agents. In: International Conference on Learning Representations, pp 406–441

Reyes-Ramírez AD, Aragón ME, Sánchez-Vega F, et al (2024) Improving aggressiveness detection using a data augmentation technique based on a diffusion language model. In: Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024), pp 171–177, https://doi.org/10.18653/v1/2024.woah-1.13

Rombach R, Blattmann A, Lorenz D, et al (2022) High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10684–10695

Saheed YK, Chukwuere JE (2026) Autonomous LLM agent: A memory-augmented, edge-optimized SHAP explanations with zero-day attack resilience in IoT/industrial IoT networks. IEEE Internet of Things Journal 13(7):14213–14228. https://doi.org/10.1109/JIOT.2025.3648649

Sahoo SS, Arriola M, Schiff Y, et al (2024) Simple and effective masked diffusion language models. Advances in Neural Information Processing Systems 37:130136–130184

Salimans T, Ho J (2022) Progressive distillation for fast sampling of diffusion models. URL https://arxiv.org/abs/2202.00512, arXiv:2202.00512

Shen H, Zhang J, Xiong B, et al (2025) Efficient diffusion models: A survey. Transactions on Machine Learning Research

Shi X, Wan J, Dong L, et al (2026) SimpleTool: Parallel decoding for real-time LLM function calling. URL https://arxiv.org/abs/2603.00030, arXiv:2603.00030

Song J, Meng C, Ermon S (2022) Denoising diffusion implicit models. URL https://arxiv.org/abs/2010.02502, arXiv:2010.02502

Song Y, Sohl-Dickstein J, Kingma DP, et al (2021) Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations

Song Y, Dhariwal P, Chen M, et al (2023) Consistency models. In: Proceedings of the 40th International Conference on Machine Learning

Sun G, Khalid U, Mendieta M, et al (2024) Exploring parameter-efficient fine-tuning to enable foundation models in federated learning. In: 2024 IEEE International Conference on Big Data (BigData), pp 8015–8024, https://doi.org/10.1109/BigData62323.2024.10825712

Sun H, Liu S, Li L, et al (2026) HillInfer: Efficient long-context LLM inference on the edge with hierarchical KV eviction using SmartSSD. URL https://arxiv.org/abs/2602.18750, arXiv:2602.18750

Sung M, Palakonda V, Im S, et al (2025) Memory- and latency-constrained inference of large language models via adaptive split computing. URL https://arxiv.org/abs/2511.04002, arXiv:2511.04002

Sánchez-Carballo E, Melgarejo-Meseguer FM, Rojo-Álvarez JL (2026) Latent diffusion for Internet of Things attack data generation in intrusion detection. URL https://arxiv.org/abs/2601.16976, arXiv:2601.16976

Tian K, Jiang Y, Yuan Z, et al (2024) Visual autoregressive modeling: Scalable image generation via next-scale prediction. In: Advances in Neural Information Processing

Systems, pp 84839–84865, https://doi.org/10.52202/079017-2694

Tian M, Liu Z, Hou C, et al (2025a) Accelerating AI-generated content collaborative inference via transfer reinforcement learning in dynamic edge networks. IEEE Transactions on Cloud Computing 13(3):1011–1025. https://doi.org/10.1109/TCC.2025.3586878

Tian Y, Yang L, Yang J, et al (2025b) MMaDA-Parallel: Multimodal large diffusion language models for thinking-aware editing and generation. URL https://arxiv.org/abs/2511.09611, arXiv:2511.09611

Touvron H, Martin L, Stone K, et al (2023) Llama 2: Open foundation and fine-tuned chat models. URL https://arxiv.org/abs/2307.09288, arXiv:2307.09288

Wang H, Wang Y, Cui B, et al (2026) ReflectDrive-2: Reinforcement-learning-aligned self-editing for discrete diffusion driving. URL https://arxiv.org/abs/2605.04647, arXiv:2605.04647

Wang L, Chen S, Jiang L, et al (2025a) Parameter-efficient fine-tuning in large language models: A survey of methodologies. Artificial Intelligence Review 58:227. https://doi.org/10.1007/s10462-025-11236-4

Wang Q, Jiang S, Yang Y, et al (2025b) Efficient and adaptive diffusion model inference through lookup table on mobile devices. IEEE Transactions on Mobile Computing 24(9):8729–8746. https://doi.org/10.1109/TMC.2025.3558203

Wang X, Xu C, Jin Y, et al (2025c) Diffusion LLMs can do faster-than-AR inference via discrete diffusion forcing. URL https://arxiv.org/abs/2508.09192, arXiv:2508.09192

Wei Y, Tang S, Zhao L, et al (2025) DiffusionX: Efficient edge-cloud collaborative image generation with multi-round prompt evolution. URL https://arxiv.org/abs/2510.16326, arXiv:2510.16326

Wen J, Zhu M, Liu J, et al (2025a) dVLA: Diffusion vision-language-action model with multimodal chain-of-thought. URL https://arxiv.org/abs/2509.25681, arXiv:2509.25681

Wen Y, Li H, Gu K, et al (2025b) LLaDA-VLA: Vision language diffusion action models. URL https://arxiv.org/abs/2509.06932, arXiv:2509.06932

Wu C, Zhang H, Xue S, et al (2025) Fast-dLLM: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. URL https://arxiv.org/abs/2505.22618, arXiv:2505.22618

Wu J, Hu M, Zhu J, et al (2026) Evo: Autoregressive-diffusion large language models with evolving balance. URL https://arxiv.org/abs/2603.06617, arXiv:2603.06617

Wu T, Fan Z, Liu X, et al (2023) AR-Diffusion: Auto-regressive diffusion model for text generation. Advances in Neural Information Processing Systems 36:39957–39974

Xiao B, Kantarci B, Kang J, et al (2025) Efficient prompting for LLM-based generative Internet of Things. IEEE Internet of Things Journal 12(1):778–791. https://doi.org/10.1109/JIOT.2024.3470210

Xie E, Yao L, Shi H, et al (2023) DiffFit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp 4230–4239

Xie T, Zhang D, Chen J, et al (2024) OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. Advances in Neural Information Processing Systems 37:52040–52094

Xie Y, Fang Q (2025) An energy-aware generative AI edge inference framework for low-power IoT devices. Electronics 14(20). URL https://doi.org/10.3390/electronics14204086

Xiong H, Wu X, Yang X, et al (2025) DiffGuess: Password strength assessment with diffusion models in intelligent vehicle-to-everything (V2X) communications. IEEE Transactions on Intelligent Transportation Systems pp 1–11. https://doi.org/10.1109/TITS.2025.3623319

Xu C (2024) PhD forum abstract: Diffusion-based task scheduling for efficient AI-generated content in edge networks. In: 2024 23rd ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), pp 333–334, https://doi.org/10.1109/IPSN61024.2024.10697426

Xu X, Mu X, Liu Y, et al (2026) Large model at edge: An optimal mobile edge generation (MEG) design. IEEE Transactions on Wireless Communications 25:5355–5369. https://doi.org/10.1109/TWC.2025.3617845

Yan C, Liu S, Liu H, et al (2024) Hybrid SD: Edge-cloud collaborative inference for Stable Diffusion models. URL https://arxiv.org/abs/2408.06646, arXiv:2408.06646

Yan P, Zeng H, Hou YT (2026) xDiff: Online diffusion model for collaborative inter-cell interference management in 5G O-RAN. IEEE Transactions on Networking 34:1363–1376. https://doi.org/10.1109/TON.2025.3622520

Yang L, Zhang Z, Song Y, et al (2023a) Diffusion models: A comprehensive survey of methods and applications. ACM Comput Surv 56(4). https://doi.org/10.1145/3626235

Yang L, Tian Y, Li B, et al (2025a) MMaDA: Multimodal large diffusion language models. In: Advances in Neural Information Processing Systems, pp 138867–138907

Yang W, Xiong Z, Guo S, et al (2025b) Efficient multi-user offloading of personalized diffusion models: A DRL-convex hybrid solution. IEEE Transactions on Mobile Computing 24(9):9092–9109. https://doi.org/10.1109/TMC.2025.3560582

Yang X, Zhou D, Feng J, et al (2023b) Diffusion probabilistic model made slim. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 22552–22562, https://doi.org/10.1109/CVPR52729.2023.02160

Yang Y, Chen Z (2025) Diffusion modeling meets deep reinforcement learning for edge task scheduling. In: 2025 7th International Conference on Next Generation Data-driven Networks (NGDN), pp 116–121, https://doi.org/10.1109/NGDN66208.2025.11182116

Yang Y, Ma W, Sun W, et al (2025c) Diffusion-based multi-agent reinforcement learning for semantic vehicular edge computing. IEEE Transactions on Services Computing 18(6):3668–3681

Yao Z, Tang Z, Yang W, et al (2025) Enhancing LLM QoS through cloud-edge collaboration: A diffusion-based multi-agent reinforcement learning approach. IEEE Transactions on Services Computing 18(3):1412–1427. https://doi.org/10.1109/TSC.2025.3562362

Ye J, Xie Z, Zheng L, et al (2025) Dream 7B: Diffusion large language models. URL https://arxiv.org/abs/2508.15487, arXiv:2508.15487

You Z, Nie S, Zhang X, et al (2026) LLaDA-V: Large language diffusion models with visual instruction tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 10093–10105

Yu R, Li Q, Wang X (2025a) Discrete diffusion in large language and multimodal models: A survey. URL https://arxiv.org/abs/2506.13759, arXiv:2506.13759

Yu R, Ma X, Wang X (2025b) Dimple: Discrete diffusion multimodal large language model with parallel decoding. URL https://arxiv.org/abs/2505.16990, arXiv:2505.16990

Zeng Q, Hu C, Song M, et al (2025a) Diffusion model quantization: A review. URL https://arxiv.org/abs/2505.05215, arXiv:2505.05215

Zeng W, Zheng J, Gao L, et al (2025b) Generative AI-aided multimodal parallel offloading for AIGC metaverse service in IoT networks. IEEE Internet of Things Journal 12(10):13273–13285. https://doi.org/10.1109/JIOT.2025.3535623

Zhao Y, Xu Y, Xiao Z, et al (2024) MobileDiffusion: Instant text-to-image generation on mobile devices. In: European Conference on Computer Vision, Springer, pp 225–242, https://doi.org/10.1007/978-3-031-73033-7_13

Zheng D (2025) Diffusion models on the edge: Challenges, optimizations, and applications. URL https://arxiv.org/abs/2504.15298, arXiv:2504.15298

Zheng Y, Chen Y, Qian B, et al (2025) A review on edge large language models: Design, execution, and applications. ACM Comput Surv 57(8). https://doi.org/10.1145/3719664

Zhou C, Yu L, Babu A, et al (2025) Transfusion: Predict the next token and diffuse images with one multi-modal model. In: International Conference on Learning Representations, pp 6446–6469

Zhou H, Jiang K, Liu X, et al (2022) Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing. IEEE Internet of Things Journal 9(2):1517–1530. https://doi.org/10.1109/JIOT.2021.3091142

Zhu X, Li J, Liu Y, et al (2024) A survey on model compression for large language models. Transactions of the Association for Computational Linguistics 12:1556–1577. https://doi.org/10.1162/tacl_a_00704

Zhu Z, Ren F, Tan Z, et al (2026) ES-dLLM: Efficient inference for diffusion large language models by early-skipping. In: The Fourteenth International Conference on Learning Representations