Diffusion Models for Text Generation: Foundations, Scaling, Reasoning, Multimodality, Test-Time Computation, and Trustworthiness

Nima Niroumand *Department of Computer Engineering - Sharif University of Technology*

Amirhosein Attari *Department of Computer Engineering - Sharif University of Technology*

Pooria Kazem *Department of Computer Engineering - Sharif University of Technology*

Pouya Behzadifar *Department of Computer Engineering - Sharif University of Technology*

Hanieh Sartipi *Department of Computer Engineering - Sharif University of Technology*

Shaygan Adim *Department of Computer Engineering - Sharif University of Technology*

Abstract

Diffusion language models replace fixed left-to-right factorization with iterative denoising of a corrupted sequence. This interface supports infilling, flexible generation order, global revision, and parallel prediction, but it also introduces representation mismatch, repeated network cost, and configuration-dependent failures.

Across 30 primary works, this survey makes four explicit contributions. It provides a common notation for Gaussian latent, categorical, and absorbing-mask diffusion; organizes the field by design layers and bottleneck-driven evolution rather than by model names alone; introduces an evidence-attribution protocol that separates representation, architecture, data and scale, initialization, post-training, and inference systems; and connects reasoning, scientific and multimodal specialization, test-time computation, and trustworthiness through the shared object of the reverse process.

The evidence supports three conclusions. Parallel advantage depends on the evaluation metric; test-time methods trade different resources rather than forming one interchangeable family; and bidirectional editability is both a capability and an attack surface. Diffusion language models should therefore be evaluated as complete reverse-process stacks, including data, sampler, compute, validity, and deployment configuration.

1 Introduction

Autoregressive language models generate a sequence one token at a time. At each step, they predict the next token from the prefix already produced. This paradigm has led to highly capable language models, but its generation order is fixed: once an early token has been selected, later decisions can condition on it but cannot directly revise it. This becomes inconvenient when an output must satisfy properties involving distant or future positions, including bidirectional infilling, lexical constraints, program repair, structured generation, and reasoning processes that benefit from revising an early hypothesis.

Diffusion models offer a different generative mechanism. They begin from a noisy sample and iteratively transform it into a structured one. The original Denoising Diffusion Probabilistic Model (DDPM) was developed for continuous data such as images (Ho et al., 2020). Its forward process gradually destroys information, while its learned reverse process removes noise step by step. Each reverse step can inspect a global hypothesis rather than only a left prefix. In principle, this enables joint revision of several positions.

Applying diffusion to text is difficult because tokens are categorical objects. A Gaussian-perturbed token representation does not automatically correspond to a meaningful partially corrupted word. Early diffusion language models therefore faced a representation question: should text be diffused in a learned continuous embedding space, or directly in a discrete categorical state space? Diffusion-LM follows the continuous route and uses differentiable latent guidance for control (Li et al., 2022). D3PM introduces categorical transition matrices, including absorbing states such as `[MASK]`, for direct discrete diffusion (Austin et al., 2021). Subsequent work aligns absorbing diffusion with bidirectional masked-language-model pretraining (He et al., 2023; Sahoo et al., 2024).

Conditional generation introduces a second design problem. A useful text diffusion model must map a source sequence to a target sequence for tasks such as paraphrasing, question generation, simplification, dialogue, and translation. DiffuSeq keeps source embeddings clean while denoising the target (Gong et al., 2023). SeqDiffuSeq replaces concatenated conditioning with an encoder-decoder architecture, self-conditioning, and position-level schedules (Yuan et al., 2022). Meta-DiffuB treats the schedule itself as a source-conditioned policy (Chuang et al., 2024).

A second line of development studies the reverse process itself. Reparameterized Discrete Diffusion Models (RDMs) expose a latent routing decision in discrete reverse transitions and use confidence-aware decoding (Zheng et al., 2024). A stochastic-integral framework connects discrete diffusion to continuous-time jump processes, gives a path-KL interpretation of score entropy, and distinguishes numerical inference errors from score-estimation errors (Ren et al., 2025). Control can enter at inference through constrained projection or during post-training through reward optimization (Cardei et al., 2025; Zekri & Boulle, 2025).

Recent work broadens the scope from small conditional generators to general-purpose foundation models, systems-aware inference, and reward-optimized reasoning. Adaptation from pretrained autoregressive models converts GPT-2 and LLaMA2 checkpoints into diffusion models with continual pretraining (Gong et al., 2025). LLaDA scales masked diffusion from scratch to 8B parameters and tests whether in-context learning and instruction following require autoregressive factorization (Nie et al., 2025). Seed Diffusion targets the speed–quality frontier through trajectory selection, on-policy speed training, and block-parallel decoding (ByteDance Seed et al., 2025). Diffusion-of-Thought (DoT), d1, and DCoLT investigate different meanings of reasoning in a diffusion process: latent thought refinement, policy-gradient post-training of masked dLLMs, and reinforcement of the entire reverse trajectory (Ye et al., 2024; Zhao et al., 2025; Huang et al., 2025).

A complementary line extends the same reverse-process interface to new domains and modalities while revisiting deployment. Dream 7B combines AR initialization with context-adaptive token-level noise rescheduling (Ye et al., 2025). DiffuCoder tests masked diffusion in program synthesis and introduces complementary-mask policy optimization (Gong et al., 2026). TransDLM treats multi-property molecular optimization as text-conditioned continuous diffusion rather than predictor-guided search (Xiong et al., 2026). LLaDA-V and Dimple extend masked diffusion to visual instruction tuning through different training routes (You et al., 2026; Yu et al., 2026), whereas MMaDA places text, image understanding, and image generation under one discrete diffusion objective and one multimodal reinforcement-learning interface (Yang et al., 2025). These methods make specialization and modality first-class design layers rather than downstream afterthoughts.

The newest results make inference-time computation and trustworthiness equally central. A metric-dependent theory proves that the apparent speed advantage of masked diffusion can hold for token-level quality yet disappear for worst-case sequence correctness (Feng et al., 2025). ReMDM restores iterative correction by allowing visible tokens to be masked again and demonstrates compute scaling in language, images, and guided molecular generation (Wang et al., 2025). RFG converts the log-density ratio of enhanced and reference dLLMs into step-wise reward-free guidance (Chen et al., 2026). Prophet treats answer-region convergence as an adaptive stopping problem (Li et al., 2026), while Fast-dLLM reduces cost per step through approximate cache reuse and parallel commitment (Wu et al., 2026). Finally, TrustLDM shows

that bidirectional conditioning creates distinctive safety, privacy, and fairness failures under editable suffixes, prefixes, orders, and generation lengths (Mo et al., 2026). Thus the reverse process is not merely a sampler: it is the locus of correction, guidance, compute allocation, and attack.

The corpus comprises 30 primary works selected to trace the main mechanisms from foundations to deployment. This is a focused, mechanism-oriented review rather than a systematic meta-analysis. We compare objectives, architectures, ablations, evaluation protocols, and stated limitations. Cross-paper scores are treated as breadth evidence unless model scale, data, compute, and decoding conditions are sufficiently matched.

This survey therefore asks four connected questions:

1. **State and learning:** Should diffusion operate in continuous embeddings or discrete tokens, how should conditions enter, and which metric represents success?
2. **Reverse policy and control:** Which positions or transitions should change, when may a decision be revised, and where should constraints, rewards, or reasoning supervision act?
3. **Capability and interface:** How are large-model capabilities acquired, and which changes are needed for code, scientific sequences, visual understanding, and image generation?
4. **Deployment and trust:** When do remasking, guidance, stopping, parallel updates, and caching improve the end-to-end frontier, and how do those choices alter safety, privacy, and fairness?

Recent surveys provide broad taxonomies of diffusion language and multimodal models, including their training and inference techniques (Li et al., 2025; Yu et al., 2025). Building on that landscape rather than duplicating a model catalogue, the distinctive contributions of this survey are:

- **A mechanism-aligned mathematical interface.** We translate continuous, categorical, absorbing-mask, routing, remasking, guidance, stopping, and policy-optimization formulations into one notation, while preserving paper-specific symbols only when a theorem requires them.
- **A bottleneck-driven taxonomy and evolution map.** We organize progress by the limitation each wave addresses—representation mismatch, conditioning, scalability, reasoning, specialization, inference cost, or deployment risk—and identify the design layer at which each method intervenes.
- **An evidence-attribution protocol.** Instead of reading unmatched benchmark numbers as a universal leaderboard, we separate gains due to objective or architecture, data and scale, source initialization, supervised or reinforcement post-training, sampling policy, and systems or hardware.
- **A reverse-process synthesis.** We jointly analyze correction, control, adaptive computation, scientific validity, multimodal grounding, and trust, then derive open problems that concern compatibility and end-to-end evaluation of the complete deployed stack.

The remainder of the survey is organized as follows. Section 2 introduces the background and unified notation. Section 3 presents the integrated taxonomy. Sections 4–6 review representation, conditional generation, metric-dependent theory, reverse-process design, remasking, and control. Section 7 examines large-scale training and systems co-design, and Section 8 studies reasoning, reinforcement learning, and reward-free test-time guidance. Sections 9 and 10 cover code, molecular design, visual instruction tuning, and unified multimodal diffusion. Section 11 compares adaptive stopping, cache reuse, and parallel decoding, while Section 12 studies safety, privacy, and fairness under editable context. Section 13 integrates the evidence, and Sections 14 and 15 give future directions and conclusions.

2 Background

2.1 Autoregressive Language Modeling

Let $x_0 = (x_0^1, \ldots, x_0^L)$ denote a clean token sequence of length L. An autoregressive language model factorizes its probability as

$$p_{\mathrm{AR}}(x_0) = \prod_{i=1}^{L} p_{\mathrm{AR}}(x_0^i \mid x_0^{<i}), \tag{1}$$

where $x_0^{<i}$ is the preceding prefix. This factorization gives an exact and convenient likelihood decomposition, but fixes a left-to-right generation order. Diffusion replaces this positional factorization with a reverse-time refinement process over a complete, partially corrupted sequence.

2.2 Continuous Diffusion

DDPMs define a forward Markov chain from a clean continuous sample $\mathbf{z}_0$ to noisy variables $\mathbf{z}_{1:T}$:

$$q(\mathbf{z}_{1:T} \mid \mathbf{z}_0) = \prod_{t=1}^{T} q(\mathbf{z}_t \mid \mathbf{z}_{t-1}), \tag{2}$$

with a common Gaussian transition

$$q(\mathbf{z}_t \mid \mathbf{z}_{t-1}) = \mathcal{N}\Big(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t \mathbf{I}\Big). \tag{3}$$

Generation reverses this chain with learned transitions. In practice, the reverse model is often trained by a noise-prediction loss:

$$\mathcal{L}_{\mathrm{simple}} = \mathbb{E}_{t,\mathbf{z}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, c)\|_2^2\right]. \tag{4}$$

The optional condition c is omitted for unconditional modeling. This formulation is natural for images but does not directly solve discrete text generation.

2.3 Categorical and Absorbing-State Diffusion

For one token position with vocabulary size $|\mathcal{V}|$, let $\mathbf{x}_t^i \in \{0,1\}^{|\mathcal{V}|}$ be its one-hot representation and $\mathbf{Q}_t$ a row-stochastic transition matrix. The categorical transition is

$$q(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i) = \mathrm{Cat}(\mathbf{x}_t^i; \mathbf{x}_{t-1}^i \mathbf{Q}_t). \tag{5}$$

D3PM makes the transition structure a modeling choice, supporting uniform replacement, structured transitions, and absorbing masks (Austin et al., 2021). In absorbing-state diffusion, a token remains visible with cumulative survival probability $\bar{\alpha}_t = \prod_{u=1}^{t}(1-\beta_u)$ or becomes the mask state m with probability $1-\bar{\alpha}_t$. This process aligns naturally with masked language modeling because the reverse model predicts missing content from bidirectional context.

2.4 Unified Notation Across the Survey

Earlier text-diffusion work reuses x, z, t, α, and M for incompatible objects. To make methods directly comparable, this survey uses the convention in Table 1. Discrete token identities use x; bold $\mathbf{x}$ is reserved for one-hot vectors in categorical derivations. Continuous embeddings use bold $\mathbf{z}$, c is any non-diffused condition, and m is the mask state. Discrete time uses $t \in \{0, \ldots, T\}$, while continuous mask time uses $t \in [0,1]$. A larger t always means more corruption.

Under this notation, independent absorbing corruption has the common marginal

$$q(x_t \mid x_0) = \prod_{i=1}^{L}\left[\alpha(t)\,\mathbf{1}\{x_t^i = x_0^i\} + (1-\alpha(t))\,\mathbf{1}\{x_t^i = m\}\right], \tag{6}$$

and a broad class of masked objectives can be written as

$$\mathcal{L}_{\text{mask}}(\theta) = \mathbb{E}_{x_0,c,t,x_t}\left[\omega(t,x_t)\sum_{i\in\mathcal{M}_t} -\log p_\theta(x_0^i \mid x_t, c, t)\right]. \tag{7}$$

Methods differ in the weight ω, the condition c, architecture, and routing policy; they do not require a new alphabet for every paper.

Symbol	Meaning	Consistency rule
x_0, x_t	Clean and corrupted discrete sequences	$x_t^i \in \mathcal{V} \cup \{m\}$; bold $\mathbf{x}_t^i$ denotes its one-hot vector only in categorical derivations.
$\mathbf{z}_0, \mathbf{z}_t$	Clean and corrupted continuous latent sequences	Used only for embedding-space Gaussian diffusion, including Diffusion-LM, DiffuSeq, and TransDLM.
c	Fixed condition	May contain source text, prompt, image features, molecular description, or a tuple of these; it is not corrupted unless stated.
m	Absorbing mask token	Replaces the inconsistent symbols M, `[MASK]`, and one-hot mask vectors in prose and survey equations.
$\beta_t, \bar{\alpha}_t$	One-step noise and cumulative survival in discrete time	$\bar{\alpha}_t = \prod_{u=1}^{t}(1-\beta_u)$.
$\alpha(t)$	Survival function in continuous mask time	For the linear schedule, $\alpha(t) = 1-t$ and the mask probability is t.
$\mathcal{M}_t, \mathcal{V}_t$	Masked and visible position sets	$\mathcal{M}_t = \{i : x_t^i = m\}$ and $\mathcal{V}_t = \{1,\dots,L\} \setminus \mathcal{M}_t$.
$\gamma_i(t)$	Model confidence at position i	$\gamma_i(t) = \max_{v\in\mathcal{V}} p_\theta(x_0^i = v \mid x_t, c, t)$.
$\mathcal{U}_t$	Positions committed at one reverse update	Routing policies differ mainly in how $\mathcal{U}_t \subseteq \mathcal{M}_t$ is selected.
σ_t^i	Probability of remasking a visible token	Reserved for ReMDM-style reverse transitions; $\sigma_t^i = 0$ recovers irreversible masked diffusion.
p_θ, p_{ref}	Enhanced/policy and reference reverse models	Used consistently for policy ratios and RFG; neither symbol denotes the forward corruption process q.
$g_t^i, \bar{g}_t$	Top-two logit gap and its answer-region mean	Used for adaptive stopping, with $g_t^i = \ell_{t,i}^{(1)} - \ell_{t,i}^{(2)}$.
$R(x_0), A_i$	Completion reward and group-relative advantage	Rewards are task-level quantities; A_i is normalized within a sampled completion group unless stated otherwise.

Table 1: Unified notation used throughout the survey. Paper-specific symbols are translated into this interface unless preserving the original symbol is necessary for a theorem.

The notation is shared, but the generation mechanisms are not. Figure 1 separates four interfaces that are often conflated in comparisons: sequential AR prediction, global categorical denoising, continuous latent denoising followed by rounding, and diffusion inside causally ordered blocks. The figure also makes the source of parallelism explicit. It lies in how many positions one network call may update, not in a guarantee that every update is independent or inexpensive.

3 An Integrated Taxonomy of Diffusion Language Models

A representation-only taxonomy is insufficient for current diffusion language models. Representation and corruption define the stochastic state space, but they do not determine the complete system. Capability also depends on the evaluation metric, conditioning interface, revisability, routing, solver, reward, scale, modality, reusable computation, and threat model. Table 2 therefore classifies methods by their primary intervention while retaining dependencies on adjacent layers.

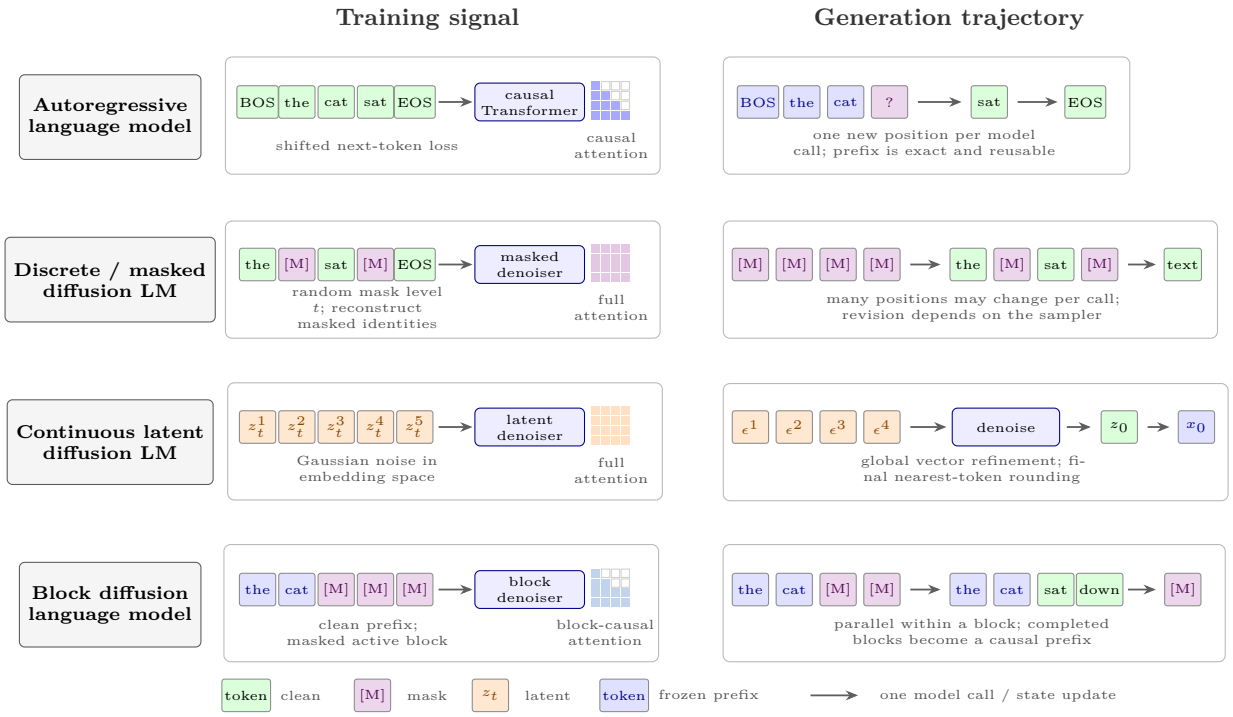


Figure 1: Training and inference interfaces in a common visual language. AR models predict a shifted next token under causal attention. Discrete DLMs reconstruct categorical masks under bidirectional attention; visible-token revisability depends on the reverse policy. Continuous DLMs denoise vectors and then round them to tokens. Block DLMs denoise several positions within a causal block and freeze completed blocks. The diagrams show computational dependency, not a fixed implementation or quality ranking.

3.1 Evolution and Attribution of Progress

The field did not advance through one monotonic architecture sequence. Each wave addressed a bottleneck exposed by the previous one. Continuous text diffusion made gradient control possible but introduced rounding and slow decoding. Discrete and masked objectives respected token identity, yet left routing, likelihood, and irreversible commitment unresolved. Scaling then improved capability, while making it harder to separate architecture from data, initialization, and compute. Recent work consequently shifts attention to post-training, domain validity, test-time allocation, and deployment risk. Figure 2 summarizes this bottleneck–response evolution using representative milestones rather than an exhaustive chronology.

No single benchmark or survey table provides a controlled total order over these papers. Reported gains mix changes in model family, parameter count, training corpus, source checkpoint, reward, decoding budget, and hardware. We therefore use within-paper ablations for causal attribution and cross-paper results mainly to establish breadth. Table 3 makes that comparison rule explicit.

This taxonomy changes the main question of the survey. The issue is no longer whether continuous or discrete diffusion is universally preferable. A practical DLM combines a representation, corruption process, quality metric, conditioning architecture, correction and routing policy, route to scale, domain or modality interface, post-training objective, inference implementation, and threat model. Figure 3 reorganizes the same literature as a hierarchical map. It is a taxonomy constructed for this review, not an empirical ranking.

4 Representation Design: Continuous, Discrete, and Masked Diffusion

The first bottleneck is representational. DDPMs assume a Euclidean state in which Gaussian perturbations remain meaningful, whereas a token is categorical and must eventually be decoded exactly. Diffusion-LM

Design layer	Central question	Representative methods
Representation and conditioning	Where should diffusion operate, and how should fixed evidence guide reconstruction?	Continuous latents enable differentiable control but require rounding; categorical masks preserve token identity. Seq2Seq methods then separate, encode, or schedule the condition.
Reverse process and theory	Which states should change, can visible tokens be revised, and which quality metric justifies parallelism?	Routing, remasking, and jump-process solvers separate prediction from commitment. Metric-dependent theory distinguishes token error from exact sequence error.
Control and reasoning	Where should constraints, rewards, or reasoning supervision modify a trajectory?	Projection acts at sampling; policy methods optimize completions or reverse actions; density-ratio guidance changes test-time transitions.
Scaling and systems	How can DLMs acquire broad capability and practical throughput?	AR transfer, from-scratch masked pretraining, and trajectory–systems co-design offer distinct routes whose data, initialization, and hardware remain confounded.
Domains and modalities	What changes when outputs must execute, satisfy chemical properties, or combine visual and textual tokens?	Code emphasizes executable correctness; molecular design requires semantic validity; multimodal systems add fixed evidence, response canvases, and heterogeneous rewards.
Deployment and trust	Which computations can be reused or stopped safely, and how does flexible context change risk?	Correction, guidance, adaptive stopping, caching, and parallel commitment alter the quality–latency surface; editable context also alters safety, privacy, and fairness.

Table 2: Compact design-layer taxonomy. Detailed method-level comparisons appear in later sections; here the rows identify the scientific question and the main source of improvement without turning the taxonomy into a model-name inventory.

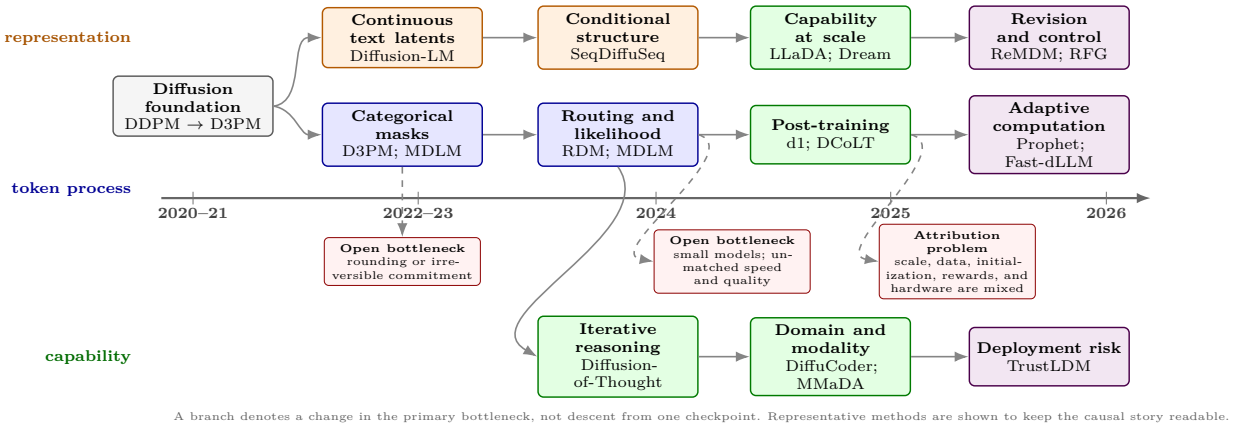


Figure 2: Branching evolution of diffusion language modeling. The map follows three interacting threads—representation, token-level reverse processes, and capability—from probabilistic foundations to scale, specialization, adaptive computation, and deployment risk. Red boxes mark the unresolved limitation that redirected the next wave. The methods are representative rather than exhaustive, and placement is not a performance ranking.

tests whether a learned embedding space can bridge that mismatch. D3PM, DiffusionBERT, and MDLM instead move corruption into token space. The transition between these works is therefore motivated by a concrete limitation—rounding and latent geometry—rather than by a claim that one Transformer architecture is uniformly better.

Primary driver	Representative evidence	What the evidence supports	What remains confounded
Representation and objective	D3PM, DiffusionBERT, and MDLM	Categorical or masked states remove latent rounding and can improve likelihood-oriented training.	Architecture, data recipe, and sampler are not held fixed across the full sequence.
Conditioning architecture and schedule	DiffuSeq, SeqDiffuSeq, and Meta-DiffuB	Stable source conditioning, encoder–decoder structure, self-conditioning, and contextual schedules improve Seq2Seq generation.	Candidate count, backbone, metric suite, and inference steps differ.
Scale, data, and initialization	DiffuLLaMA, LLaDA, Dream, and Seed Diffusion	Both AR transfer and from-scratch masked pretraining can acquire broad capabilities at billion-parameter scale.	Training tokens, source checkpoints, model sizes, and hardware stacks prevent a clean architecture ranking.
Post-training and verifier	SEPO, d1, DCoLT, DiffuCoder, MMaDA, and RFG	Rewards can act on completions, complementary masks, transitions, or whole reverse trajectories.	Reward quality, online sampling cost, policy approximation, and task-specific verifiers differ.
Domain or modality interface	TransDLM, LLaDA-V, Dimple, and MMaDA	Diffusion can support chemical constraints, visual conditioning, and image-token generation.	Domain validity and multimodal scores are not commensurate with text likelihood or exact-answer accuracy.
Sampler, systems, and evaluation	ReMDM, Prophet, FastdLLM, metric-dependent theory, and TrustLDM	Correction, stopping, caching, and configuration search change the quality–cost–risk surface without redefining the base denoiser.	Step budgets, kernels, batch sizes, answer extraction, and threat models must be matched.

Table 3: Evidence-attribution protocol used in this survey. Cross-paper numbers are interpreted through the dominant source of change rather than as a universal leaderboard.

4.1 Diffusion-LM: Gaussian Diffusion in Learned Embedding Spaces

Diffusion-LM resolves the categorical-token problem by mapping $x_0 = (x_0^1, \ldots, x_0^L)$ to a trainable continuous representation

$$\mathbf{z}_0 = \mathrm{Emb}(x_0) = [\mathrm{Emb}(x_0^1), \ldots, \mathrm{Emb}(x_0^L)] \tag{8}$$

and running Gaussian diffusion over these vectors (Li et al., 2022). The embedding table, denoiser, and rounding interface are trained jointly. In contrast with pure noise prediction, the denoiser predicts the clean latent sequence, which is useful because the final decoder must associate every vector with a vocabulary item.

The continuous state makes external guidance differentiable. Given an attribute c,

$$p(\mathbf{z}_{t-1} \mid \mathbf{z}_t, c) \propto p(\mathbf{z}_{t-1} \mid \mathbf{z}_t) p(c \mid \mathbf{z}_{t-1}). \tag{9}$$

Its score separates into fluency and control terms:

$$\nabla_{\mathbf{z}_{t-1}} \log p(\mathbf{z}_{t-1} \mid \mathbf{z}_t, c) = \nabla_{\mathbf{z}_{t-1}} \log p(\mathbf{z}_{t-1} \mid \mathbf{z}_t) \tag{10}$$

$$+ \nabla_{\mathbf{z}_{t-1}} \log p(c \mid \mathbf{z}_{t-1}). \tag{11}$$

This supports sentiment, syntax, length, infilling, and span-level control without retraining the generator for every attribute. The price is a representation mismatch: a denoised vector need not lie near a valid token embedding, so nearest-neighbor rounding can erase meaningful latent changes. Long reverse trajectories add a second cost. These weaknesses motivate direct categorical diffusion, although the continuous route remains useful when differentiable control is primary.

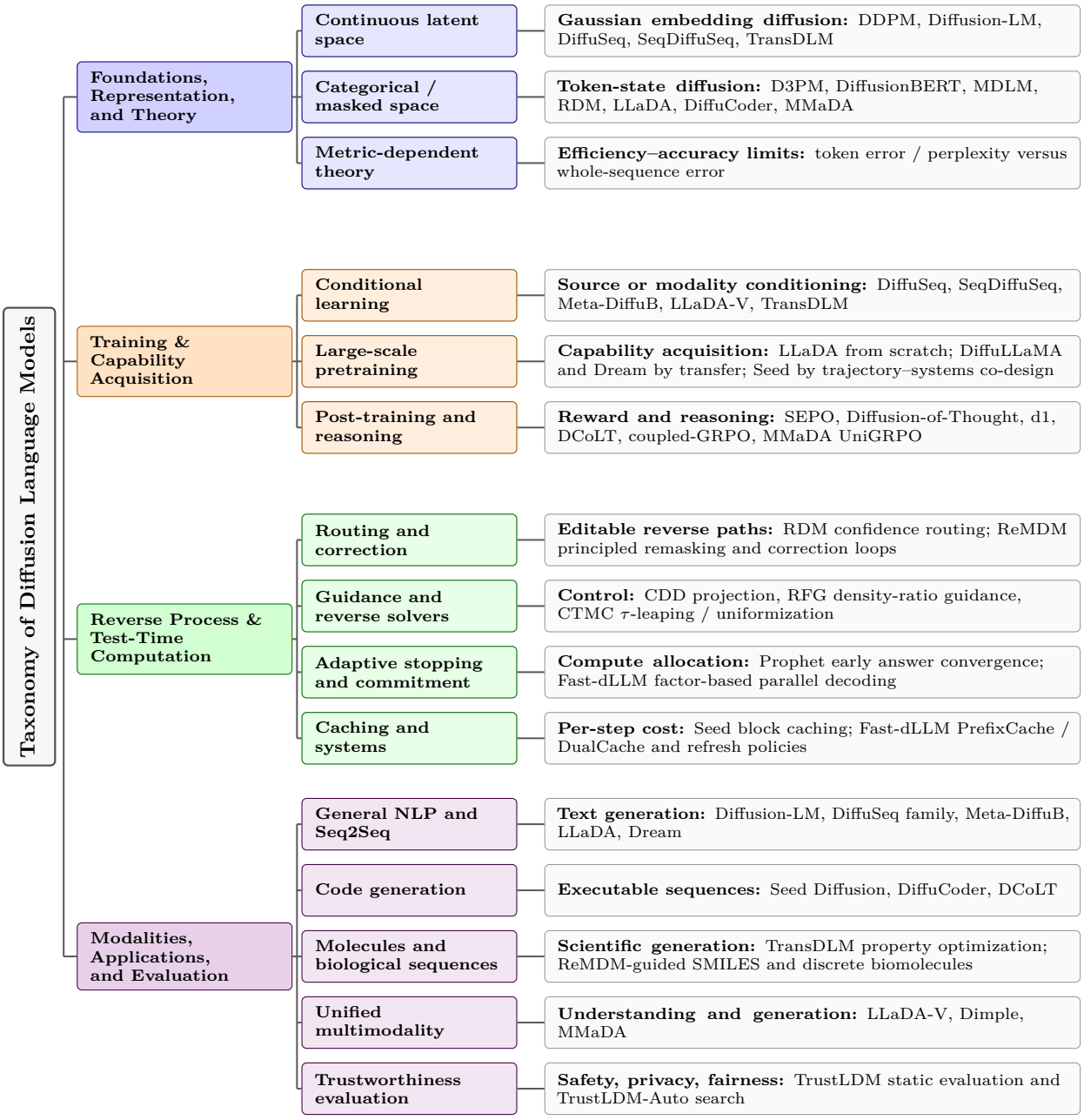


Figure 3: Hierarchical taxonomy constructed for this survey. The first level separates foundations and theory, capability acquisition, reverse-process and test-time computation, and modalities, applications, or evaluation. The second level identifies modeling choices, and the final column lists representative methods. Placement indicates a primary role; many systems span several branches.

4.2 D3PM: Direct Diffusion in Categorical State Space

D3PM keeps generation in token space and makes the transition matrix an explicit inductive bias (Austin et al., 2021). With $\bar{\mathbf{Q}}_t = \mathbf{Q}_1 \cdots \mathbf{Q}_t$ and one-hot token states $\mathbf{x}_t^i$, the forward marginal and analytical posterior are

$$q(\mathbf{x}_t^i \mid \mathbf{x}_0^i) = \mathrm{Cat}(\mathbf{x}_t^i; \mathbf{x}_0^i \bar{\mathbf{Q}}_t), \tag{12}$$

$$q(\mathbf{x}_{t-1}^i \mid \mathbf{x}_t^i, \mathbf{x}_0^i) \propto (\mathbf{x}_t^i \mathbf{Q}_t^\top) \odot (\mathbf{x}_0^i \bar{\mathbf{Q}}_{t-1}), \tag{13}$$

where $\odot$ is elementwise multiplication. The learned model predicts a clean-token distribution $\widetilde{p}_\theta(\mathbf{x}_0^i \mid \mathbf{x}_t^i)$ and substitutes it into this posterior. Training uses the discrete variational bound, often supplemented by auxiliary clean-token cross-entropy.

Uniform replacement, structured transitions, and absorbing states express different meanings of uncertainty. In language experiments, absorbing masks are markedly more effective than uniform replacement: `[MASK]` says that identity is unknown, whereas a randomly substituted token can falsely appear meaningful. Nearest-neighbor transitions defined by static embeddings do not reliably help. D3PM therefore resolves latent rounding and makes the corruption kernel part of the language model. It does not yet resolve long reverse chains, and its early text likelihood remains behind strong AR models.

4.3 DiffusionBERT: Matching Corruption to Pretraining

DiffusionBERT asks whether a pretrained bidirectional masked language model already contains the appropriate reverse-process skill (He et al., 2023). BERT has learned to reconstruct randomly masked tokens from two-sided context, while absorbing diffusion asks it to solve the same problem across many corruption levels. Initializing the denoiser from BERT therefore transfers linguistic knowledge and aligns pretraining with the discrete state space.

The method also studies a spindle-shaped schedule intended to vary masking according to token information content and compares several ways of providing t: a layer-wise time embedding, a prefix time token, and no explicit time input. In the time-agnostic case, the model infers the effective noise level from the mask pattern. This becomes relevant to later large DLMs that likewise omit dedicated time modules. The gains must still be interpreted jointly with BERT initialization, and iterative inference remains more expensive than one masked-LM pass.

4.4 MDLM: A Likelihood-Oriented Masked Objective

MDLM revisits continuous-time absorbing diffusion with a substitution parameterization and Rao–Blackwellization (Sahoo et al., 2024). Let m denote the mask token and $\alpha(t)$ the survival probability. Under the restriction that visible tokens copy themselves, the continuous-time negative evidence lower bound reduces to a weighted masked-language-modeling loss of the form

$$\mathcal{L}_{\mathrm{MDLM}} = \mathbb{E}_{t,x_0,x_t}\left[-\frac{\dot{\alpha}(t)}{1-\alpha(t)}\sum_{i:x_t^i=m}\log p_\theta(x_0^i \mid x_t, t)\right]. \tag{14}$$

The schedule-derived weight and integration over corruption levels distinguish this objective from fixed-ratio BERT masking and keep it connected to a sequence likelihood bound.

MDLM narrows the perplexity gap between masked diffusion and autoregressive baselines and supports variable-length, semi-autoregressive block generation. One block is completed by diffusion and then used as context for the next. The remaining issue is computational: improving the objective does not remove repeated full-context evaluations. Nevertheless, masked diffusion is no longer merely a discrete approximation to Gaussian diffusion; it is a likelihood-oriented language-modeling interface in which corruption removes identity and a bidirectional model reconstructs it. LLaDA, d1, and LLaDOU build directly on this interface.

5 Conditional Sequence-to-Sequence Diffusion

Improved unconditional likelihood does not by itself solve conditional generation. A source must remain semantically stable while the target changes, and repeatedly processing that source can dominate the cost of a long reverse chain. Let $c = x^{\mathrm{src}}$ and $x_0 = x^{\mathrm{tgt}}$. Conditional models differ in whether c is concatenated, encoded once, or used to determine the noise schedule. This progression motivates DiffuSeq, SeqDiffuSeq, and Meta-DiffuB in turn.

5.1 DiffuSeq: Partial Noising and Stable Conditioning

DiffuSeq introduces classifier-free conditional diffusion with a simple rule: concatenate source and target latents but keep the source clean (Gong et al., 2023):

$$\mathbf{h}_t = \mathbf{c} \oplus \mathbf{z}_t, \qquad \mathbf{c}_t = \mathbf{c}_0 = \mathrm{Emb}(c), \tag{15}$$

Only the target follows a Gaussian transition,

$$q(\mathbf{z}_t \mid \mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t \mathbf{I}). \tag{16}$$

The denoiser receives the clean source, noisy target, and timestep simultaneously. During sampling, an anchoring operation restores the source portion after every transition, preventing the condition from drifting.

Training combines clean-target reconstruction over timesteps, a final embedding reconstruction term, and latent regularization. Inference begins with $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$. DiffuSeq also uses minimum Bayes risk decoding: from candidate set $\mathcal{S}$, it selects

$$\widehat{x}_0 = \arg\min_{x \in \mathcal{S}} \frac{1}{|\mathcal{S}|} \sum_{x' \in \mathcal{S}} R(x, x'), \tag{17}$$

using negative BLEU as the reported risk. Partial noising gives a clear conditioning mechanism, but the concatenated source is recomputed at every denoising step, latent rounding remains, and the strongest quality often relies on multiple candidates.

5.2 SeqDiffuSeq: Architecture, Self-Conditioning, and Position Schedules

SeqDiffuSeq encodes the source once and repeatedly denoises only the target through an encoder-decoder architecture (Yuan et al., 2022):

$$p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t, c, t). \tag{18}$$

Source representations can be cached, and the decoder processes only target positions rather than a concatenated source-target sequence. In the reported QQP experiment with 2,000 reverse steps, this design takes 89 seconds per batch on one V100, compared with 317 seconds for DiffuSeq, a reported 3.56× speed-up.

Self-conditioning supplies the previous clean estimate to the next denoising call:

$$\widehat{\mathbf{z}}_0^t = f_\theta(\mathbf{z}_{t+1}, c, t+1), \tag{19}$$

$$\widehat{\mathbf{z}}_0 = f_\theta(\mathbf{z}_t, \mathrm{sg}[\widehat{\mathbf{z}}_0^t], c, t), \tag{20}$$

where sg stops gradients. The reverse process now refines its own previous hypothesis instead of restarting from the current noisy state alone.

The method also estimates the loss at position i and timestep t,

$$L_t^{(i)} = \mathbb{E}\left[\|\mathbf{z}_{0,i}^\theta - \mathbf{z}_{0,i}\|_2^2\right], \tag{21}$$

and learns a separate cumulative schedule $\bar{\alpha}_t^{(i)}$ so denoising difficulty changes more uniformly over time. Later positions often receive different schedules, producing a weak left-to-right recovery bias without imposing an AR factorization. Replacing the adaptive schedule with a fixed square-root schedule reduces BLEU by 2.29 points on average; removing self-conditioning costs 1.34 points; removing both costs 5.71. Stronger single-sample quality reduces candidate diversity, so MBR adds less than it does for DiffuSeq. The reported speed gain should be attributed to encoder–decoder source reuse and shorter target-side computation, not to the diffusion objective alone; 2,000 reverse steps are still required in that comparison.

5.3 Meta-DiffuB: Contextualized Scheduling as a Learned Curriculum

Meta-DiffuB moves from position-specific schedules to source-specific curricula (Chuang et al., 2024). It contains a scheduler B_ψ and a sequence-to-sequence diffusion exploiter D_θ. For source condition c, the

scheduler predicts binary instructions $\iota_t^c \in \{0, 1\}$ instead of thousands of unconstrained real-valued noise levels:

$$\boldsymbol{\beta}^c = B_\psi(c), \qquad \boldsymbol{\iota}^c = (\iota_1^c, \ldots, \iota_T^c). \tag{22}$$

An instruction advances to the next base noise level or repeats the current level. Repetition pauses additional corruption, allowing difficult examples to move more cautiously through the trajectory.

The source-conditioned forward transition is

$$q_{B_\psi}(\mathbf{z}_t \mid \mathbf{z}_{t-1}, c) = \mathcal{N}\Big(\mathbf{z}_t; \sqrt{1-\beta_t^c}\mathbf{z}_{t-1}, \beta_t^c \mathbf{I}\Big). \tag{23}$$

To train the scheduler, a temporary exploiter update produces $D_{\theta'}$. If the old and updated exploiters receive R_{D_θ} and $R_{D_{\theta'}}$, the meta-reward is

$$R_{B_\psi} = R_{D_{\theta'}} - R_{D_\theta}, \qquad \nabla_\psi J(\psi) = \sum_{t=1}^{T} \nabla_\psi \log B_\psi(\iota_t^c \mid c) R_{B_\psi}. \tag{24}$$

Exploration epochs keep the exploiter fixed so schedules are compared under the same model. Across QQP, Wiki-Auto, Quasar-T, and Commonsense Conversation, the paper reports improvements over diffusion and fine-tuned LM baselines and observes less aggressive noising for difficult examples. The contribution is a source-conditioned allocation of denoising difficulty. Its limitations are equally important: BLEU does not fully represent factuality, fluency, or human preference; meta-optimization is expensive; and the underlying DiffuSeq-style sampler still uses a long trajectory.

6 Reverse-Process Design, Theory, and Control

A better token objective still leaves the generation policy underspecified. The denoiser predicts what a clean token could be, but the sampler decides *which positions change*, *when a position becomes stable*, and whether an earlier decision can be reopened. Numerical approximation adds a separate source of error. This section therefore moves from representation to executable reverse-process design, where routing, correction, solver accuracy, and control must be analyzed explicitly.

6.1 Metric-Dependent Efficiency: Token Error Versus Sequence Error

Claims that masked diffusion is faster than autoregression are incomplete unless they specify what "accurate" means. Feng et al. (2025) assume that the target distribution q is generated by a hidden Markov model. They also require the denoiser to approximate each true conditional with uniformly small error,

$$D_{\mathrm{KL}}\big(q(x_0^i \mid x_t) \,\|\, p_\theta(x_0^i \mid x_t)\big) < \epsilon_{\mathrm{learn}} \quad \text{for all } t, x_t, i. \tag{25}$$

The analysis separates a token-level metric, written in perplexity form,

$$\mathrm{TER}(p) = 2^{\mathbb{E}_{x\sim q}\left[-|x|^{-1}\log p(x)\right]}, \tag{26}$$

from the probability of generating an invalid whole sequence,

$$\mathrm{SER}(p) = 1 - \sum_{x \in \mathcal{L}_q} p(x), \qquad \mathcal{L}_q = \{x : q(x) > 0\}. \tag{27}$$

TER rewards locally fluent probability assignment. SER is stricter: one inconsistent transition makes the complete sample erroneous, so it is closer to exact reasoning-chain or formal-language correctness.

For an n-gram target and accuracy tolerance ϵ, the positive result constructs a masking schedule requiring

$$N = O\big((n-1)\epsilon^{-n}\big) \tag{28}$$

reverse steps—independent of sequence length L—such that

$$\log \mathrm{TER}(p) \le \log \mathrm{TER}(q) + \epsilon_{\mathrm{learn}} + 4\epsilon \log |\mathcal{V}|. \tag{29}$$

Thus constant-depth parallel refinement is theoretically meaningful when the target is near-optimal token-level quality and the dependency order n is fixed.

The sequence-level result is fundamentally different. The paper constructs an HMM over a vocabulary of size 16 for which, even with $\epsilon_{\text{learn}} < 1/128$, a linear budget $N = CL$ can leave

$$\mathrm{SER}(p) > \tfrac{1}{2} \tag{30}$$

for a constant C. Therefore sublinear denoising cannot guarantee low worst-case sequence error, and linearly many full-sequence Transformer evaluations may cost more than cached AR decoding. The lower-bound analysis also covers the remasking strategy available at the time and cached samplers that skip network calls when no position changes; remasking can improve practical trajectories without invalidating this worst-case statement.

Controlled experiments on synthetic 2–4-gram languages and HMMs support the metric split. Around 512 MDM steps approach the AR token-level metric across lengths 512, 1024, and 2048, whereas sequence accuracy deteriorates with length and remains far below AR even at 2048 steps. The scope must be stated carefully: the theory assumes formal HMM targets, a uniformly accurate conditional model, and a worst-case construction. It does not prove that every natural-language reasoning task needs L steps, nor that adaptive stopping or guidance is useless. Instead, it provides a lens for later methods. ReMDM changes revisability, RFG changes transition quality, Prophet allocates steps per example, and Fast-dLLM changes their cost. None should be evaluated only by perplexity or nominal parallelism.

6.2 RDM: Making the Hidden Routing Decision Explicit

For a one-hot categorical token state $\mathbf{x}_t^i$, a common forward process mixes the previous state with a noise distribution q_{noise}:

$$q(\mathbf{x}_t^i \mid \mathbf{x}_{t-1}^i) = (1-\beta_t)\mathbf{x}_{t-1}^i + \beta_t q_{\text{noise}}, \qquad q(\mathbf{x}_t^i \mid \mathbf{x}_0^i) = \bar{\alpha}_t \mathbf{x}_0^i + (1-\bar{\alpha}_t) q_{\text{noise}}, \tag{31}$$

where $\bar{\alpha}_t = \prod_{u=1}^{t}(1-\beta_u)$. Uniform noise yields multinomial diffusion, whereas a point mass on m yields absorbing diffusion. A conventional reverse model predicts $f_\theta(x_t) \approx x_0$ and substitutes that prediction into the analytical posterior.

RDM observes that this posterior contains an implicit binary routing variable v_{t-1} (Zheng et al., 2024):

$$q(x_{t-1}, v_{t-1} \mid x_t, x_0) = q(v_{t-1} \mid x_t, x_0) q(x_{t-1} \mid v_{t-1}, x_t, x_0). \tag{32}$$

For a noisy position, the router decides whether the token should be restored or remain noisy; for a clean position, it decides whether the state should be preserved or reset. Marginalizing v_{t-1} recovers the original discrete reverse transition, so the router is a reparameterization rather than an extra heuristic model.

The resulting surrogate is a weighted cross-entropy over currently noisy positions. If $b_{t,n} = \mathbf{1}[x_{t,n} = x_{0,n}]$, a representative term is

$$\mathcal{L}_t^{\text{RDM}}(\theta) = \mathbb{E}\left[-\lambda_{t-1}^{(2)} \sum_{n=1}^{N} (1-b_{t,n}) x_{0,n}^\top \log f_\theta(x_t)_n\right]. \tag{33}$$

At inference $b_{t,n}$ is unknown, so RDM replaces it with model confidence $s_{t,n} = \max_j f_{\theta,j}(x_t)_n$ and commits the highest-confidence positions first. The empirical implication is important: a model can use a valid categorical diffusion objective and still decode poorly if its routing order is inefficient. RDM reports competitive sequence-to-sequence generation with roughly ten decoding iterations in settings where DiffuSeq commonly uses thousands. The evidence is nevertheless concentrated in relatively small, short conditional tasks, and confidence routing remains only as reliable as the denoiser's calibration.

6.3 ReMDM: Restoring Revision and Inference-Time Scaling

Standard absorbing diffusion has a structural asymmetry: once $x_t^i \neq m$, the analytical posterior keeps that token fixed. Consequently, a wrong early prediction is locked in even though later context could expose the

mistake. ReMDM introduces a reverse posterior with a remasking probability σ_t^i while preserving the usual masked-diffusion marginals (Wang et al., 2025). For one position and $s < t$, translated into our notation,

$$q_\sigma(x_s^i \mid x_t^i, x_0^i) = \begin{cases} (1-\sigma_t^i)\,\delta_{x_0^i} + \sigma_t^i\,\delta_m, & x_t^i \neq m, \\ \dfrac{\alpha(s) - (1-\sigma_t^i)\alpha(t)}{1-\alpha(t)}\,\delta_{x_0^i} + \dfrac{1-\alpha(s)-\sigma_t^i\alpha(t)}{1-\alpha(t)}\,\delta_m, & x_t^i = m. \end{cases} \tag{34}$$

Validity requires

$$0 \le \sigma_t^i \le \min\left\{1, \frac{1-\alpha(s)}{\alpha(t)}\right\}. \tag{35}$$

Setting $\sigma_t^i = 0$ recovers ordinary masked diffusion. Because a state may move from visible back to m, the joint construction is non-Markovian when conditioned only on the current noisy state; conditioning on x_0 yields the tractable posterior above, analogous in spirit to a discrete DDIM-style alternative path.

ReMDM's marginal $q_\sigma(x_t \mid x_0)$ is identical to the classical absorbing marginal. Its negative ELBO differs mainly by a timestep weight,

$$\mathcal{L}_\sigma = \mathbb{E}[-\log p_\theta(x_0 \mid x_{0^+})] + \mathbb{E}_{t,x_t}\left[T\frac{(1-\sigma_t)\alpha(t) - \alpha(s)}{1-\alpha(t)}\log p_\theta(x_0 \mid x_t)\right], \tag{36}$$

so $\sigma_t = 0$ at training permits direct reuse of pretrained MDLM or LLaDA weights, followed by $\sigma_t > 0$ only during sampling. This plug-and-play claim is empirical rather than an equality of training bounds: the standard MDLM bound is tighter, but the reused denoiser performs well under the alternative sampler.

The sampler exposes several correction policies. A capped or rescaled schedule limits remasking globally. A confidence schedule assigns larger σ_t^i to tokens that had low probability when last decoded, while a switch activates revision only late in generation. A loop instead holds $\alpha(t)$ constant over an interval and repeatedly repairs a nearly complete candidate. The authors further prove that masked forward–backward and discrete-flow-matching correctors are special cases of the ReMDM posterior; a constant-noise correction loop lies outside those cases.

On OpenWebText with sequence length 1024, ReMDM with nucleus sampling and a correction loop improves MAUVE from 0.042 for MDLM at 1024 steps to 0.403, and reaches 0.656 at 4096 steps; the paper emphasizes MAUVE because generative perplexity can be made deceptively low by collapsing diversity. On LLaDA-8B-Instruct it improves Countdown pass@1 from 45.2 ± 0.2 to 46.1 ± 0.2 and the reported TruthfulQA ΔROUGE values. These are useful but modest downstream gains compared with the large unconditional-generation change.

The biological application is especially informative. Molecules are represented as character-level SMILES strings from QM9, so validity and global ring structure depend on coupled discrete tokens. ReMDM is combined with discrete classifier-free or classifier-based guidance to maximize ring count and, in the appendix, drug-likeness. Across 32, 64, and 128 steps, its novelty–property curves move the Pareto frontier beyond AR, uniform diffusion, and irreversible MDLM baselines. This is not evidence of laboratory efficacy: the dataset is small, strings are limited to length 32, and properties are computed by RDKit. It is evidence that revisable discrete generation can better negotiate global chemical constraints than one-shot commitment. Together with TransDLM, it reveals two distinct scientific routes: latent continuous optimization for source-conditioned multi-property change, and discrete remasking for guided exploration of biological strings.

6.4 Continuous-Time Jump Processes and Solver Error

Ren et al. represent discrete diffusion as a continuous-time Markov chain on a finite state space $\mathcal{X}$ (Ren et al., 2025). If p_t is the state distribution and Q_t a rate matrix, the forward dynamics satisfy

$$\frac{dp_t}{dt} = Q_t p_t. \tag{37}$$

For $x \neq y$, the discrete score is a probability ratio $s_t(x,y) = p_t(y)/p_t(x)$ rather than a Euclidean gradient. The time-reversed jump rate combines this score with the off-diagonal forward rate:

$$Q_s^{\leftarrow}(y,x) = s_{T-s}(x,y)\widetilde{Q}_{T-s}(x,y). \tag{38}$$

Representing jumps through Poisson random measures gives a path-space change-of-measure theorem. Consequently, score entropy is not merely a tokenwise fitting loss: it controls a dynamic contribution to the KL divergence between the true and learned reverse path measures.

6.4.1 Poisson random measures

A discrete diffusion trajectory is a sequence of random jumps rather than a continuous Brownian path. Ren et al. express the forward process as the stochastic integral

$$X_t = X_0 + \int_0^t \int_{\mathcal{X}} (y - X_{\tau^-}) N^{[\lambda]}(d\tau, dy), \qquad \lambda_\tau(y) = \widetilde{Q}_\tau(y, X_{\tau^-}), \tag{39}$$

where $N^{[\lambda]}$ is a Poisson random measure with evolving intensity and X_{τ^-} is the state immediately before a jump. The reverse process has the same form with intensity

$$\mu_\tau(y) = s_{T-\tau}(\bar{X}_{\tau^-}, y)\widetilde{Q}_{T-\tau}(\bar{X}_{\tau^-}, y). \tag{40}$$

This is the jump-process analogue of an SDE representation for continuous diffusion: the rate matrix supplies candidate transitions and the score ratio changes their reverse-time intensities.

6.4.2 Change of measure and score entropy

The change-of-measure theorem for evolving Poisson intensities plays the role that Girsanov's theorem plays for Brownian SDEs. If $\widehat{s}_\theta$ approximates the true score ratio, the dynamic path-KL term contains

$$\mathbb{E}\left[\int_0^T \sum_{y \in \mathcal{X}} K\left(\frac{\widehat{s}_{\theta,T-\tau}(X_\tau, y)}{s_{T-\tau}(X_\tau, y)}\right) s_{T-\tau}(X_\tau, y)\widetilde{Q}_{T-\tau}(X_\tau, y), d\tau\right], \tag{41}$$

where $K(r) = r - 1 - \log r \geq 0$. Thus, score-entropy training minimizes a dynamic contribution to an upper bound on divergence between complete true and learned reverse paths. This result provides a probabilistic interpretation stronger than treating score entropy as a convenient token loss.

This framework separates three errors that are often conflated in empirical work:

$$D_{\mathrm{KL}}(p_\delta \| \widehat{q}_{T-\delta}) \lesssim \underbrace{e^{-\rho T} \log |\mathcal{X}|}_{\text{finite horizon}} + \underbrace{\varepsilon}_{\text{score error}} + \underbrace{D^2 \kappa T}_{\text{numerical discretization}}. \tag{42}$$

6.4.3 τ-leaping versus uniformization

τ-leaping freezes the learned intensity in an interval $[s_n, s_{n+1})$,

$$\widehat{\mu}_{\theta,s}(y) \approx \widehat{\mu}_{\theta,s_n}(y), \tag{43}$$

and samples Poisson jump counts under this frozen rate. It is simple and parallelizable, but contributes the third term of Eq. 42. Uniformization instead chooses a block-wise upper bound

$$\Lambda_b \geq \sup_{s \in (s_b, s_{b+1}]} \sum_{y \in \mathcal{X}} \widehat{\mu}_{\theta,s}(y), \tag{44}$$

samples candidate Poisson events at rate Λ_b, and accepts actual jumps according to relative intensities. It simulates the *learned* continuous-time reverse jump process without the τ-leaping discretization term. It does not remove finite-horizon or learned-score error. This distinction becomes central when scaling the model and shortening its trajectory: reducing nominal denoising steps is useful only if the solver and policy still approximate a good reverse process.

6.5 Two Control Locations: Projection and Policy Adaptation

CDD enforces hard sequence-level constraints by changing the sampling distribution while leaving the base model fixed (Cardei et al., 2025). The feasible set may represent lexical inclusion, count requirements, molecule validity, safety rules, or other structured conditions. Conceptually, each reverse step solves

$$p_t^\star = \arg\min_{p \in \mathcal{C}} \left[D_{\mathrm{KL}}(p \| p_{\theta,t}) - \tau \mathcal{H}(p) \right], \tag{45}$$

where $\mathcal{C}$ is a feasible set and the entropy term prevents unnecessary collapse. Unlike prompting, filtering, or finite-candidate reranking, the constraint enters while uncertainty remains distributed across token positions. This is suitable for strict request-specific conditions, but projection adds optimization to every reverse step and requires a mathematically tractable constraint representation.

SEPO moves control into post-training (Zekri & Boulle, 2025). For a reverse policy π_θ and possibly non-differentiable sequence reward $R(x)$, the objective and score-function gradient are

$$J(\theta) = \mathbb{E}_{x \sim \pi_\theta}[R(x)], \qquad \nabla_\theta J(\theta) = \mathbb{E}_{x \sim \pi_\theta}[R(x) \nabla_\theta \log \pi_\theta(x)]. \tag{46}$$

In practice, clipped ratios limit destructive updates, advantages reduce variance, and a reference-policy penalty can preserve base fluency. CDD is therefore an inference-time projection onto an externally specified feasible set, whereas SEPO changes the probability of rewarded sequences across future requests. The former is attractive when constraints vary by sample; the latter is appropriate when a stable preference should generalize. SEPO's text evidence depends on a model judge and its sequence-design evidence on a computational oracle, so it establishes optimizability rather than independent human or wet-lab validity. d1 and DCoLT inherit this policy perspective but confront a harder object: a large masked dLLM has neither the simple tokenwise factorization of an AR model nor a unique definition of the action being optimized.

6.6 Why Reverse-Process Design Becomes a Capability Question

These methods reveal four separable choices: token prediction, routing order, numerical solver, and control location. Later large-scale systems operationalize each choice differently. LLaDA uses low-confidence remasking as a routing rule; Seed Diffusion distills useful orders and optimizes trajectory length; d1 approximates completion likelihood for policy gradients; DCoLT directly parameterizes the unmask order. The progression is therefore not "better denoisers followed by applications." It is a shift from fitting local reverse conditionals to designing and optimizing a complete generative policy.

7 Scaling and Systems Co-Design

Strong small-model results do not establish that diffusion is a general language-modeling substrate. The next bottleneck is capability acquisition at scale. DiffuGPT/DiffuLLaMA inherit knowledge from AR checkpoints; LLaDA trains a large masked model from scratch; Dream adds token-adaptive weighting to transfer; and Seed Diffusion co-designs trajectory learning with deployment. Their scores cannot isolate architecture alone because data budget, initialization, model size, and inference stack all change.

7.1 Scaling by Adapting Autoregressive Checkpoints

7.1.1 Objective alignment

Gong et al. begin from an absorbing process with linear survival schedule $\alpha(t) = 1 - t$ (Gong et al., 2025). For a sequence $x_0^{1:L}$, the continuous-time discrete-diffusion loss becomes

$$\mathcal{L}^{\mathrm{DD}}(\theta) = \mathbb{E}\left[-\frac{1}{t} \sum_{i=1}^{L} \mathbf{1}[x_t^i = m] \log p_\theta(x_0^i \mid x_t, t) \right]. \tag{47}$$

The AR loss is also cross-entropy, but token $n-1$ predicts token n under a causal attention mask. The two discrepancies are therefore (i) which context is visible and (ii) whether logits predict the same position or the

next position. This observation makes conversion substantially more direct than mapping an AR checkpoint into a continuous Gaussian latent model.

7.1.2 Adaptation recipe

The method uses three coupled choices. First, *attention-mask annealing* gradually relaxes causal attention toward full bidirectional attention, avoiding an abrupt change in the representation geometry. Second, the *shift operation* is preserved: logits are shifted before evaluating the masked-denoising cross-entropy, so output features retain their pretrained next-token interface. During sampling, the sequence is shifted back with a start token after each reverse transition. Third, explicit time embeddings are omitted; the corrupted token pattern itself provides a noise-level signal and avoids introducing randomly initialized modules into the checkpoint.

The ablation on GSM8K-symbolic supports the recipe. With GPT2-M initialization, discrete diffusion reaches 49.7%, while removing the shift drops accuracy to 34.5% and removing mask annealing gives 47.2%. Continuous diffusion is substantially worse under the same adaptation setting. The study consequently identifies the shift as the dominant bridge and mask annealing as a useful but smaller stabilization mechanism.

7.1.3 Scale, capability, and limits

The paper converts GPT-2 and LLaMA2 checkpoints from 127M to 7B parameters using fewer than 200B continual-pretraining tokens. DiffuGPT improves on earlier diffusion baselines and displays self-correction, in-context learning, and strong infilling. On the CD4 global-planning task, DiffuGPT reaches 87.5%, compared with 51.1% for a much larger fine-tuned LLaMA baseline reported in the paper. The result is task-specific but demonstrates where bidirectional reconstruction directly matches the evaluation.

Adaptation does not fully recover all source-model capabilities. DiffuLLaMA trails the original LLaMA2 on several commonsense tasks, and generation quality depends on the number of diffusion steps. Nevertheless, long-sequence latency is promising: with FlashAttention-2 and $T = 256$, the reported single-batch time is 13.3 seconds versus 17.5 for LLaMA2 at length 1024, and 23.5 versus 35.7 seconds at length 2048. These numbers are implementation-specific, but they show that full-sequence recomputation can become competitive when the number of reverse steps is below sequence length.

7.2 LLaDA: From-Scratch Masked Diffusion at 8B

7.2.1 Probabilistic formulation and pretraining

LLaDA tests whether LLM-scale capability requires an autoregressive factorization (Nie et al., 2025). Each token is masked independently with probability $t \sim \mathcal{U}[0, 1]$. A bidirectional Transformer predicts all masked tokens, and the objective is

$$\mathcal{L}_{\mathrm{LLaDA}}(\theta) = -\mathbb{E}_{t,x_0,x_t}\left[\frac{1}{t}\sum_{i=1}^{L}\mathbf{1}[x_t^i = m]\log p_\theta(x_0^i \mid x_t, t)\right]. \tag{48}$$

The $1/t$ weighting is essential: the objective is a variational upper bound on negative log-likelihood rather than fixed-ratio BERT masking. LLaDA trains 1B and 8B models; the 8B model uses 2.3T tokens, sequence length 4096, and approximately 0.13 million H800 GPU-hours. It uses full multi-head attention rather than grouped-query attention because standard AR KV caching is incompatible with globally changing masked responses.

7.2.2 SFT and flexible reverse sampling

For instruction tuning, the prompt remains clean and only response tokens are masked. On 4.5M instruction pairs, the conditional loss has the same form as Eq. 48 with the prompt appended as visible context. Sampling begins from a fully masked response. At a step from t to $s < t$, the model predicts every masked position and remasks an expected fraction s/t of predictions. Random remasking follows the underlying diffusion transition, while low-confidence remasking is a practical decoding heuristic that revises uncertain tokens.

The same model supports pure diffusion, block diffusion, and an AR-like order without retraining. This flexibility is a property of the learned conditional family, not evidence that all modes are equally good. The main paper uses low-confidence pure diffusion because it is the strongest configuration; block sampling is relevant when generation length and caching matter.

7.2.3 Evidence and comparison discipline

LLaDA 8B Base reports MMLU 65.9, GSM8K 70.3, HumanEval 35.4, and HumanEval-FIM 73.8. Under the authors' evaluation, these values are competitive with LLaMA3 8B on several tasks and clearly above LLaMA2 7B on mathematics and code. However, the baselines do not share identical pretraining token counts: for example, the table lists 2.3T tokens for LLaDA and 15T for LLaMA3. The safest conclusion is that masked diffusion exhibits strong empirical scaling and supports in-context learning and instruction following; it is not that decoder factorization alone explains every benchmark difference.

7.3 Dream 7B: Transfer Plus Context-Adaptive Rescheduling

7.3.1 Preserving the AR prediction interface

Dream starts from Qwen2.5-7B rather than relearning language structure from random initialization (Ye et al., 2025). Like DiffuLLaMA, it replaces causal attention with full attention but preserves the AR shift: hidden state h_i continues to predict position $i+1$. This is an architectural compatibility constraint, not a left-to-right sampling requirement. At diffusion time the model may fill positions in arbitrary order while using an output head whose learned geometry remains aligned with the AR checkpoint.

Under the unified notation, the base objective is Eq. 7 with linear survival $\alpha(t) = 1 - t$ and $\omega(t, x_t) = 1/t$. Dream's main modification is to reject the assumption that all masked positions in one sequence experience the same effective difficulty. A token surrounded by visible context is easier than an equally masked token in an information-poor region, even if both share the same sampled t.

7.3.2 CART: token-specific information weighting

Context-Adaptive noise Rescheduling at Token-level (CART) assigns position-specific weights:

$$\mathcal{L}_{\mathrm{CART}}(\theta) = \mathbb{E}\left[\sum_{i \in \mathcal{M}_t} w_i(t, x_t)\left[-\log p_\theta(x_0^i \mid x_t, t)\right]\right], \tag{49}$$

with the implemented context score written in survey notation as

$$w_i(t, x_t) = \frac{1}{2}\sum_{j \in \mathcal{V}_t} \mathrm{Geo}(p, |i-j|-1)\,. \tag{50}$$

The geometric kernel makes nearby visible tokens contribute more strongly when p is large and spreads their contribution more uniformly when p is small. Equations 49–50 do not create a new forward process; they alter the learning signal so the same sentence-level mask pattern is interpreted at token resolution.

7.3.3 Training scale, evidence, and limits

Dream 7B matches the Qwen2.5-7B architecture and is continually pretrained on 580B open-source tokens spanning general text, mathematics, and code. Dream-Instruct then uses 1.8M instruction–response pairs with corruption restricted to the response. The reported base results include MMLU 69.5, GSM8K 77.2, MATH 39.6, HumanEval 57.9, and MBPP 56.2. Relative to LLaDA 8B under the paper's common protocol, the corresponding gains are 3.6, 6.3, 8.9, 25.0, and 17.2 points, while using approximately 0.6T rather than 2.3T training tokens. The comparison supports the combined transfer-and-rescheduling recipe, but does not isolate CART from checkpoint quality or data composition.

Planning tasks expose a more distinctive pattern. Dream reports 16.0 on Countdown, 81.0 on Sudoku, and 17.8 on trip planning, compared with 6.2, 21.0, and 3.6 for Qwen2.5-7B in the same table. By changing

the number of reverse steps, the model also traces a quality–speed curve rather than exposing one fixed decoding cost. It supports arbitrary-order completion and infilling without task-specific heads. These are useful capability demonstrations, but the strongest planning claims remain benchmark-specific and should not be generalized to unconstrained long-horizon planning without intervention-based evaluation.

7.4 Seed Diffusion: Co-Designing Corruption, Trajectory, and Hardware

7.4.1 Two-stage corruption curriculum

Seed Diffusion is code-focused and targets the speed-quality Pareto frontier (ByteDance Seed et al., 2025). During the first 80% of diffusion training it uses independent mask corruption:

$$q_{\text{mask}}(x_t \mid x_0) = \prod_i q_{\text{mask}}(x_t[i] \mid x_0[i]), \quad q(x_t^i = v \mid x_0^i) = \begin{cases} 1-\gamma_t, & v = x_0^i, \\ \gamma_t, & v = m. \end{cases} \tag{51}$$

For the final 20%, an edit-based process applies scheduled insertions, deletions, and substitutions. Writing its edit-ratio schedule as η_t to avoid overloading survival notation, the target edit count is $k_t = \lfloor L\eta_t \rfloor$, with a small maximum edit ratio. This augmentation removes the harmful assumption that every visible token is correct and explicitly trains revision of unmasked content.

7.4.2 Constraining and optimizing the trajectory space

Mask diffusion is equivalent to averaging over token generation permutations:

$$\mathcal{J}(\theta) = -\mathbb{E}_{\pi \sim \mathcal{U}(S_d)} \left[\sum_{r=1}^{d} \log p_\theta(x_{\pi(r)} \mid x_{\pi(<r)}) \right]. \tag{52}$$

This expressiveness also creates many redundant or linguistically poor orders. Seed generates a pool of trajectories, scores them with an ELBO-related criterion, and distills selected trajectories through constrained-order training. The next stage is on-policy speed learning. For a reverse trajectory τ and verifier V, the idealized objective is

$$\min_\theta \ \mathbb{E}_{\text{prompt}, \tau \sim p_\theta} \left[|\tau| - V(\tau[0]) \right], \tag{53}$$

implemented with a progressive surrogate because directly minimizing step count is unstable. The surrogate encourages larger meaningful Levenshtein changes between intermediate states while the verifier protects final correctness.

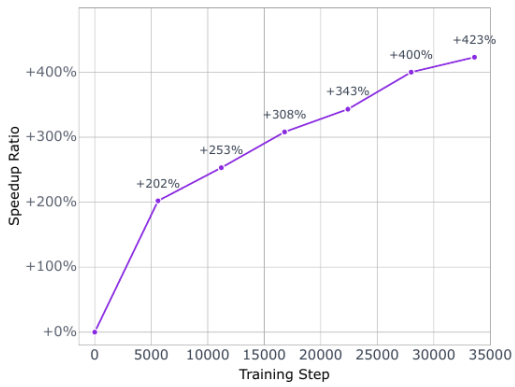
7.4.3 Block-parallel inference and evidence

Inference is semi-autoregressive across blocks and diffusion-parallel within a block. Completed blocks become a stable causal prefix and are cached; arbitrary block sizes remain available without block-specific retraining. Figure 4 shows the coupled trade-off: on-policy training increases the estimated speedup, while larger blocks make each forward pass more expensive. Thus, a useful block size balances tokens committed per step against attention cost and trajectory quality.

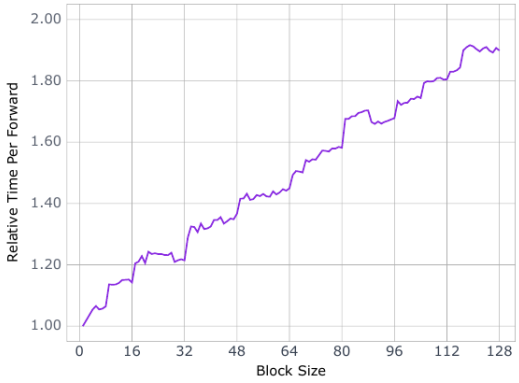
Seed Diffusion Preview reports 2,146 tokens/s on H20 GPUs while remaining competitive on a broad code suite. Selected results include 54.3 pass@1 on CanItEdit, 72.6 average on MBXP, and 42.2 overall on NaturalCodeBench. The left panel of Figure 4 reports an estimated speedup increase to 423% during on-policy training. The paper does not disclose a conventional parameter count for the preview model, and cross-system speed numbers for Mercury and Gemini use different hardware or protocols. Consequently, the robust contribution is the co-design recipe; the headline throughput is not a hardware-independent ranking.

8 Reasoning and Reinforcement Learning

Scaling improves broad benchmarks, but it does not show that intermediate revisions are valid reasoning or genuine self-correction. Reasoning papers therefore introduce supervision or reward at different units.



(a) The changes of speedup ratio estimated by sampling with a certain block size b during the on-policy training.



(b) The relative forward time($\frac{T(B=b)}{T(B=1)}$) changes over the different block size(b)

Figure 4: Seed Diffusion's reported on-policy speedup dynamics and the per-forward cost as block size increases. The figure illustrates why parallelism requires both trajectory learning and systems-aware block selection. Reproduced from Figure 2 of (ByteDance Seed et al., 2025).

Method	Scaling route	Training scale	Reverse/inference design	Representative evidence	Interpretation and caveat
DiffuGPT / DiffuLLaMA	Continual pre-training from GPT-2/LLaMA2	127M–7B; < 200B adaptation tokens	Shift-preserving DD; causal-to-full attention; arbitrary-order sampling	GSM8K-symbolic 61.8/63.1; CD4 87.5; long-sequence latency advantage in reported setup	Efficient knowledge transfer, but incomplete recovery of AR base capability
LLaDA	From-scratch masked diffusion	8B; 2.3T tokens; 4.5M SFT pairs	Full-context mask predictor; low-confidence remasking; pure/block/AR-like modes	MMLU 65.9, GSM8K 70.3, HumanEval 35.4, FIM 73.8	Strong scaling evidence; comparisons remain affected by data and alignment mismatch
Dream 7B	AR initialization plus continual diffusion training	7B; 580B tokens; 1.8M SFT pairs	Shift-preserving full attention; CART token-level rescheduling; arbitrary-order and infilling	MMLU 69.5, GSM8K 77.2, HumanEval 57.9; large reported planning gains	Efficient route beyond LLaDA in the paper's protocol; CART and checkpoint effects are not fully separated
Seed Diffusion	Code-only large-scale diffusion plus trajectory optimization	Size undisclosed; Seed-Coder data pipeline	Edit augmentation; constrained orders; on-policy shortening; cached block-parallel decoding	2,146 token/s on H20; MBXP 72.6; CanItEdit 54.3	Strong systems result, but throughput is hardware- and protocol-dependent

Table 4: Complementary scaling and systems strategies. The methods address different bottlenecks: initialization cost, from-scratch scalability, token-adaptive transfer, and end-to-end throughput.

DoT refines an explicit rationale across diffusion time. d1 treats the sampled completion as the action and approximates its likelihood. DCoLT treats each reverse transition, including unmask order, as an action. These definitions must remain separate because they imply different credit-assignment errors and different evidence for correction.

8.1 Diffusion-of-Thought: Reasoning as Iterative Latent Refinement

8.1.1 Single-pass and multi-pass formulations

For problem condition c, rationales $r_{1:n}$, and answer a, standard CoT uses $p(a \mid c, r_{1:n})$. DoT distributes the continuous rationale representation across diffusion time:

$$a \sim p_\theta^{\mathrm{DoT}}(a \mid c, \mathbf{z}_t), \tag{54}$$

where the prompt tokens are fixed and only the rationale/answer region is noised. Single-pass DoT refines all rationale positions jointly. Multi-pass DoT restores a causal inductive bias by generating one rationale at a time:

$$p(r_{1:n}, a \mid c) = \prod_{j=1}^{n} p_\theta^{\mathrm{DoT}}(r_j \mid c, r_{<j}, \mathbf{z}_t^{r_j})\; p_\theta^{\mathrm{DoT}}(a \mid c, r_{1:n}, \mathbf{z}_t^{a}). \tag{55}$$

The single-pass version has more parallelism; the multi-pass version provides cleaner conditional signals but reintroduces sequential dependence between thoughts.

8.1.2 Training for self-correction

DoT identifies two exposure gaps (Ye et al., 2024). First, ordinary diffusion training corrupts gold rationales, while inference denoises states generated by the model. *Scheduled sampling* occasionally creates the training state by denoising a noisier model state and then re-noising the model's prediction. The denoiser therefore learns to recover from its own global errors. Second, multi-pass training normally conditions on correct previous rationales. *Coupled sampling* also corrupts earlier rationales when training a later one, reducing thought-level exposure bias.

For continuous diffusion backbones, the objective is the standard variational bound with prior, diffusion, and rounding terms. DoT uses a conditional ODE solver to reduce Plaid's thousands of sampling steps. It also applies self-consistency across sampled reasoning trajectories:

$$\widehat{a} = \arg\max_{a} \sum_{m=1}^{M} \mathbf{1}[a^{(m)} = a]. \tag{56}$$

The diffusion-step count becomes an explicit test-time reasoning budget: easy multiplication can be solved with one or two model calls, while GSM8K benefits from more reverse steps.

8.1.3 Evidence and limitations

On 4-by-4 and 5-by-5 multiplication and Boolean logic, DoT reaches 100% accuracy and reports up to 27× the throughput of fine-tuned GPT-2 CoT without losing accuracy. On GSM8K-Aug, from-scratch DoT reaches only 4.6%, demonstrating that diffusion structure alone does not replace pretrained language understanding. With SEDD-medium, DoT reaches 53.5% and 59.4% with self-consistency, compared with 43.9% for GPT2-medium CoT in the paper's setting. Scheduled and coupled sampling provide smaller but consistent gains.

The experiments use relatively small task-specific models and augmented reasoning data. Intermediate denoising states demonstrate correction behavior, but they are not guaranteed to be faithful explanations of the final answer. The main result is a new controllable reasoning-time axis, not evidence that diffusion automatically reasons better at large scale.

8.2 d1: Completion-Level Policy Optimization for Masked dLLMs

8.2.1 Why AR GRPO does not transfer directly

AR GRPO uses token probabilities whose product exactly factorizes the sequence probability. A masked dLLM instead defines the completion distribution through many denoising calls, so exact policy ratios would require differentiating through a costly reverse trajectory. d1 uses a mean-field approximation (Zhao et al., 2025):

$$\log \pi_\theta(o \mid c) \approx \sum_{k=1}^{|o|} \log \phi_\theta(o_k \mid c'), \qquad \phi_\theta(o_k \mid c') = p_\theta(o_k \mid c' \oplus m^{|o|}), \tag{57}$$

where c' is obtained by independently masking prompt tokens. All completion positions remain masked, so one forward pass estimates every per-token log probability. This is much cheaper than the multi-sample conditional likelihood estimator used for evaluation in LLaDA.

Random prompt masking is not only a likelihood approximation. Each inner policy update sees a different view of the same prompt-completion pair. The stochastic views regularize repeated GRPO updates and allow more inner updates per expensive batch of online generations. Empirically, light masking probabilities 0.1–0.3 are stable, whereas aggressive masking 0.5–0.7 can destabilize late training.

8.2.2 diffu-GRPO objective and two-stage recipe

Let G completions receive rewards r_i and group-relative advantages $A_i = r_i - G^{-1}\sum_j r_j$. With the approximate token ratio

$$\rho_{i,k}(\theta) = \frac{\phi_\theta(o_{i,k} \mid c')}{\phi_{\theta_{\text{old}}}(o_{i,k} \mid c')}, \tag{58}$$

d1 optimizes the clipped surrogate

$$\mathcal{L}_{\text{diffu-GRPO}} = -\mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_k \min(\rho_{i,k}A_i, \text{clip}(\rho_{i,k}, 1-\epsilon, 1+\epsilon)A_i) - \beta D_{\text{KL}}(\phi_\theta \| \phi_{\text{ref}})\right]. \tag{59}$$

The complete d1 recipe first performs masked SFT on the s1K set of 1,000 curated reasoning traces, then applies task-specific diffu-GRPO. SFT teaches verification and backtracking patterns; online RL aligns the model with correctness and format rewards.

8.2.3 Results and what they establish

At the best reported generation length, d1-LLaDA improves LLaDA-8B-Instruct from 78.2 to 82.1 on GSM8K and from 36.2 to 40.2 on MATH500. Countdown rises from 20.7 to 42.2, and Sudoku from 11.7 to 22.1. Diffu-GRPO alone improves its initialization in all twelve task/length settings in Table 1 of the paper. The combined SFT+RL recipe wins eleven of twelve comparisons against pure diffu-GRPO. A unified model trained across four tasks remains competitive, and coding experiments show gains on HumanEval and MBPP for most lengths.

The evaluation reports the best checkpoint for task-specific RL models and uses an approximate policy probability. Generation length materially affects results; Sudoku even degrades with longer sequences. d1 therefore establishes that useful policy-gradient training is feasible for large masked dLLMs, while leaving exact likelihood ratios, efficient online generation, and robust cross-task reasoning open.

8.3 DCoLT: Reinforcing Every Reverse-Diffusion Action

8.3.1 Trajectory-level policy

DCoLT defines a reverse trajectory with a separate reverse-step index r to avoid reversing the survey's noise-time convention. Let $\xi_0 = x_T$ be the highly corrupted initial state and $\xi_S = x_0$ the final clean sequence:

$$\xi_r \sim \pi_{\theta,r}(\cdot \mid \xi_{r-1}), \qquad r = 1, \ldots, S, \tag{60}$$

and applies a final verifiable reward to the complete chain (Huang et al., 2025). A group of trajectories receives standardized group-relative advantages. Policy-gradient losses are computed at every denoising step, gradients are accumulated across the chain, and the parameters are updated only after the full trajectory. Intermediate states need not be grammatical or monotonically closer in human interpretation; they are latent actions optimized for the final outcome.

For continuous-time SEDD, a concrete score parameterizes the reverse rate, and τ-leaping gives a tractable product of token transition probabilities. This connects DCoLT directly to the jump-process interpretation of Section 6: the reverse numerical transition is also the RL action distribution.

8.3.2 Learning which tokens to unmask

For LLaDA, an action has two factors: selecting an ordered set $U_r = (u_1, \ldots, u_K)$ of masked positions and assigning token values. The Unmask Policy Module (UPM) produces scores $h^i_{\theta,r}$ and defines a Plackett–Luce

policy

$$\pi_{\theta,r}^{\text{unmask}}(U_r \mid \xi_{r-1}) = \prod_{k=1}^{K} \frac{\exp h_{\theta,r}^{u_k}}{\sum_{j \in \mathcal{M}(\xi_{r-1}) \setminus \{u_1,\ldots,u_{k-1}\}} \exp h_{\theta,r}^{j}}. \tag{61}$$

The complete transition factorizes as

$$\pi_{\theta,r}(\xi_r \mid \xi_{r-1}) = \pi_{\theta,r}^{\text{unmask}}(U_r \mid \xi_{r-1}) \prod_{i \in U_r} p_{\theta,r}(\xi_r^i \mid \xi_{r-1}). \tag{62}$$

Figure 5 shows that UPM is a small time-conditioned Transformer block above LLaDA hidden states. Its learned ranking turns the heuristic confidence router of earlier masked diffusion into an explicit trainable policy.

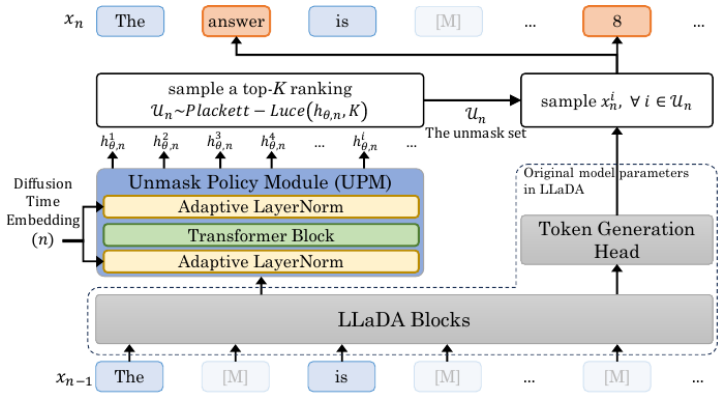


Figure 5: LLaDOU factorizes a masked-diffusion action into an ordered unmask decision and token generation. The UPM samples the top-K order with a Plackett–Luce policy. Reproduced from Figure 2 of (Huang et al., 2025).

8.3.3 Results, ablations, and interpretation

On SEDD-400M, DCoLT reaches 96.2% on Sudoku 4-by-4 and 57.0% on GSM8K-Aug, compared with 79.4% and 53.5% for SEDD+DoT. On LLaDA 8B, LLaDOU reports 88.1% on GSM8K, 44.6% on MATH, 59.1% on HumanEval, and 51.6% on MBPP. In the UPM ablation, training only the new unmask policy raises GSM8K from 47.27 to 69.24 in the paper's 64-step setting; training UPM and LLaDA jointly raises it to 81.06. This is unusually direct evidence that token order is not a secondary sampling detail.

The method uses verifiable final rewards and substantial online sampling, including 16 H800 GPUs for the LLaDA experiments. Comparisons with d1 must separate MATH from MATH500, full-parameter from LoRA training, prompt formats, data, sequence length, and checkpoint selection. Finally, "lateral thought" is a useful policy interpretation, not a measurement that intermediate states are faithful or human-like reasoning.

8.4 RFG: Reward-Free Guidance from Model Ratios

Reasoning-oriented post-training produces an enhanced policy p_θ, but applying the policy directly is not the only way to exploit it. RFG pairs it with a reference dLLM p_{ref} and interprets their trajectory log-density ratio as an implicit reward (Chen et al., 2026):

$$r_\theta(x_{0:T}) = \beta \log \frac{p_\theta(x_{0:T})}{p_{\text{ref}}(x_{0:T})}. \tag{63}$$

For reverse step t, the tail log-ratio

$$Q_\theta^t(x_{t-1}, x_{t:T}) = \sum_{j=t}^{T} \beta \log \frac{p_\theta(x_{j-1} \mid x_{j:T})}{p_{\text{ref}}(x_{j-1} \mid x_{j:T})} \tag{64}$$

equals a log exponential-utility expectation of the complete reward under the reference trajectory. Taking $Q_\theta^t - Q_\theta^{t+1}$ yields the step reward

$$r_\theta^t(x_{t-1} \mid x_t) = \beta \log \frac{p_\theta(x_{t-1} \mid x_t)}{p_{\text{ref}}(x_{t-1} \mid x_t)}. \tag{65}$$

Reward-guided sampling with strength γ then gives the normalized transition

$$\log p_{\text{RFG}}(x_{t-1} \mid x_t) = (1+w) \log p_\theta(x_{t-1} \mid x_t) - w \log p_{\text{ref}}(x_{t-1} \mid x_t) - \log Z_t, \tag{66}$$

$$w = \beta\gamma. \tag{67}$$

The signs matter: positive w extrapolates away from the reference and further along the enhanced model's direction; $w = 0$ recovers p_θ, and $w = -1$ recovers p_{ref}. This is analogous to classifier-free guidance, but the density ratio represents post-training improvement rather than a class condition.

Implementation requires two forward evaluations at each denoising step, combines their logits as $(1+w)\ell_\theta - w\ell_{\text{ref}}$, and passes the result to the existing unmasking and token-selection schedule. No process-reward model or additional optimization is required. The theoretical interpretation is exact when the enhanced policy is the KL-regularized optimum associated with an underlying reward. Applying the same formula to instruction-tuned checkpoints is a useful empirical extension, not a consequence of that optimum condition.

Across LLaDA and Dream families, RFG improves every reported post-trained policy and beats a compute-matched naive logit ensemble. Representative changes are Dream-Instruct MATH-500 43.6 → 46.4, DiffuCoder HumanEval 69.5 → 78.7, and DiffuCoder MBPP 72.8 → 74.3; the maximum gain is 9.2 points. Performance remains strong over a range of w, and mildly negative guidance can correct an over-specialized policy. The cost is approximately a second model evaluation per step plus memory for two checkpoints, so RFG scales *quality with compute*; it does not by itself reduce latency. This distinction makes it complementary to Prophet's step reduction and Fast-dLLM's per-step acceleration.

Method	Optimized object	Training signal	Diffusion-specific mechanism	Representative result	Main limitation
DoT	Explicit rationale and answer across diffusion time	Supervised rationales; scheduled/coupled sampling	Partial noising, joint or thought-wise denoising, ODE acceleration, self-consistency	SEDD-M: 53.5 GSM8K-Aug; 59.4 with self-consistency; up to 27× on simple tasks	Small/task-specific models; reasoning states are not guaranteed faithful
d1	Final completion under an approximate masked-dLLM policy	SFT traces plus correctness/format GRPO rewards	Mean-field sequence probability; one-step log probability; random prompt masking	82.1 GSM8K, 40.2 MATH500, 42.2 Countdown, 22.1 Sudoku	Approximate likelihood; best checkpoint/task-specific RL; expensive online sampling
DCoLT	Every reverse transition, unmask order, and token value	Final outcome reward over grouped trajectories	SEDD score policy; LLaDA UPM; Plackett–Luce ordered unmasking	LLaDOU: 88.1 GSM8K, 44.6 MATH, 59.1 HumanEval, 51.6 MBPP	Verifiable rewards and high compute; intermediate "thought" is hard to validate
RFG	Each test-time reverse transition	Implicit reward from enhanced/reference log-density ratio	$(1+w)\ell_\theta - w\ell_{\text{ref}}$ guidance with any existing route	Up to +9.2 HumanEval points; consistent gains over policy and ensemble	Two model evaluations per step; theory assumes a reward-related enhanced policy

Table 5: Distinct intervention units for reasoning in diffusion language models. DoT supervises a rationale trajectory, d1 optimizes an approximate completion policy, DCoLT optimizes the complete reverse chain, and RFG guides a fixed enhanced policy at test time.

9 Domain Specialization and Scientific Applications

General-language accuracy is an incomplete evaluation once outputs must execute or remain chemically valid. Code imposes long-range syntax and testable semantics, so commitment order and complete-program diversity matter. Molecular optimization adds scaffold retention and simultaneous property constraints. DiffuCoder keeps masked discrete diffusion but changes data, training, and RL; TransDLM keeps continuous latent diffusion but changes the scientific representation and conditioning channel. Their gains therefore support domain adaptation, not a universal ranking of discrete versus continuous diffusion.

9.1 DiffuCoder: Code Generation as Order-Adaptive Denoising

9.1.1 Four-stage specialization pipeline

DiffuCoder is a 7B masked diffusion model trained on 130B code tokens (Gong et al., 2026). Its pipeline contains adaptation pretraining, a mid-training stage on an annealed code mixture, supervised instruction tuning, and coupled-GRPO. The first three stages separate code knowledge from instruction following; the fourth targets executable correctness and the diffusion-specific probability estimator. This distinction matters because generic Dream and LLaDA models improve only marginally, or regress on some code benchmarks, after the same kind of SFT, whereas the code-specialized model provides a stronger policy for RL.

The base loss is the common masked objective in Eq. 7. The scientific contribution is not a new corruption kernel but an analysis of the resulting *generation order*. Let $\mathcal{U}_t$ be the positions unmasked at step t. DiffuCoder measures local and global "AR-ness": whether newly committed positions extend a local left-to-right frontier and whether the earliest remaining masks are selected. The experiments show that a masked dLLM is neither purely parallel nor obligatorily left-to-right. Code tends to produce stronger causal structure than some other modalities, yet the model may recover delimiters, keywords, or later constraints before earlier uncertain tokens.

9.1.2 Temperature changes both token identity and order

In an AR model, temperature mainly changes the categorical choice at the next fixed position. In a confidence-routed dLLM it also changes confidence ranks and therefore the set $\mathcal{U}_t$. Raising temperature can simultaneously diversify sampled token values and the order in which positions become stable. DiffuCoder reports substantially larger pass@k at suitable temperatures even when pass@1 falls, creating diverse complete programs for verifier-based RL. This is a diffusion-specific exploration mechanism: rollout diversity comes from both content and schedule.

9.1.3 Coupled-GRPO with complementary masks

GRPO requires likelihood ratios for sampled completions, but exact masked-diffusion likelihood estimation averages losses over noise levels and masks. A full-mask one-pass approximation, as used by d1, is efficient but evaluates every token under an unrealistic fully hidden response and can overemphasize high-entropy early positions. DiffuCoder instead samples a pair of complementary masks. For completion o_i, choose t and $\hat{t}$ with $t + \hat{t} = T$ and enforce

$$\mathbf{1}\{o^k_{i,t} = m\} + \mathbf{1}\{o^k_{i,\hat{t}} = m\} = 1 \quad \text{for every completion position } k. \tag{68}$$

Thus every token is evaluated exactly once across the pair, but each prediction observes a realistic partially visible completion. With one complementary pair, the paper combines the two partial-mask losses with the full-mask estimate:

$$\log \widehat{\pi}_\theta(o^k_i \mid c) = \frac{1}{2}\Big[\ell^k_{\theta,t} + \ell^k_{\theta,\hat{t}} + \ell^k_{\theta,T}\Big], \tag{69}$$

where $\ell^k_{\theta,t}$ denotes the position-k log-probability contribution at mask level t. Substituting $\widehat{\pi}_\theta$ into the clipped GRPO ratio yields coupled-GRPO. Complementary masks function as antithetic variates: they increase token coverage and reduce estimator variance without returning to a long Monte Carlo trajectory.

9.1.4 Results and interpretation

DiffuCoder Base reports HumanEval 67.1, HumanEval+ 60.4, MBPP 74.2, and MBPP+ 60.9, comparable to specialized 7–8B AR code models in the paper's protocol. Instruction tuning gives an EvalPlus average of 63.6; coupled-GRPO raises it to 67.9, a 4.4-point gain, and improves the overall reported benchmark average from 53.7 to 56.5. The gains are not uniform: BigCodeBench-Hard slightly decreases in the main coupled-GRPO row. A leave-one-out advantage variant improves several broader code tasks but reduces HumanEval. The evidence therefore supports variance-reduced diffusion-native RL, not universal dominance on every code distribution.

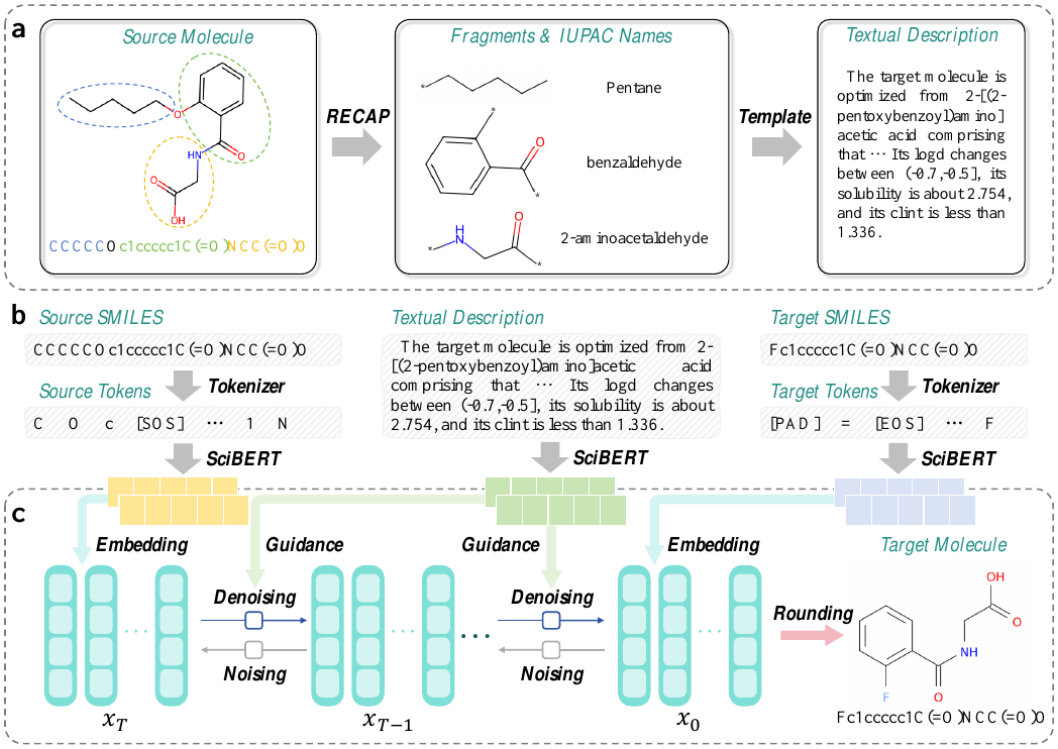


Figure 6: TransDLM framework: RECAP fragments and IUPAC semantics form a textual condition; source and target SMILES are embedded; cross-attention guides continuous denoising; and the final latent is rounded back to SMILES. Cropped from Figure 2 of (Xiong et al., 2026).

After coupled-GRPO, global AR-ness decreases and performance degrades less when the number of decoding steps is halved. This links reward optimization to deployment: RL changes not only which programs are likely, but also how readily several positions can be committed together. The result also warns against treating decoding order as a fixed property of the architecture; it is shaped by data, temperature, SFT, reward, and routing.

9.2 TransDLM: Text-Guided Multi-Property Molecular Optimization

9.2.1 Why molecular optimization is a language-diffusion application

Molecular optimization starts from a source molecule M_{src} and seeks a related molecule M_{tgt} satisfying several desired properties. Predictor-guided search repeatedly estimates properties and moves through chemical or latent space. Errors in an external predictor can therefore compound, especially outside its training distribution. TransDLM instead learns the source-to-target mapping from matched molecular pairs and encodes property requirements directly in a textual condition (Xiong et al., 2026).

This paper is an application of *continuous conditional language diffusion*, not a masked dLLM. Source and target SMILES strings are tokenized with multi-character chemical units preserved. RECAP decomposition extracts chemically meaningful fragments; IUPAC names make fragment roles and attachment information more legible to a pretrained scientific language model. The condition c_{mol} contains the source description, fragment connections, and requested LogD, solubility, and clearance changes. The source molecule supplies the initial latent scaffold, while the text condition guides the denoiser toward the optimized target.

9.2.2 Conditional continuous diffusion and rounding

Let $\mathbf{z}_0 \in \mathbb{R}^{d\times L}$ be the target-molecule embedding and let c_{mol} be the encoded textual description. The Gaussian forward process is

$$q(\mathbf{z}_t \mid \mathbf{z}_{t-1}) = \mathcal{N}\Big(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t\mathbf{I}\Big). \tag{70}$$

The Transformer predicts the clean target latent $\widehat{\mathbf{z}}_0 = f_\theta(\mathbf{z}_t, t, c_{\text{mol}})$. Using $\bar{\alpha}_t = \prod_{u=1}^{t}(1-\beta_u)$, the mean of the learned reverse transition can be written

$$\boldsymbol{\mu}_\theta = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} f_\theta(\mathbf{z}_t, t, c_{\text{mol}}) + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{z}_t. \tag{71}$$

Each Transformer layer uses cross-attention from the molecule state to the projected condition. Training combines clean-latent reconstruction across timesteps with a token-rounding likelihood:

$$\mathcal{L}_{\text{TransDLM}} = \mathbb{E}\left[\sum_{t=1}^{T} \|f_\theta(\mathbf{z}_t, t, c_{\text{mol}}) - \mathbf{z}_0\|_2^2 - \log p_\theta(M_{\text{tgt}} \mid \mathbf{z}_0)\right]. \tag{72}$$

At inference, every latent column is rounded to the nearest or most likely SMILES-token embedding. This restores a discrete molecule but retains the standard continuous-text-diffusion vulnerability: a locally plausible latent may round to an invalid sequence.

9.2.3 Error-suppression claim and its scope

The paper analyzes a clean-latent prediction error $\boldsymbol{\delta}_t = \widehat{\mathbf{z}}_0^{(t)} - \mathbf{z}_0$. Under its DDPM reverse mean, the induced one-step deviation has coefficient

$$\Delta\mathbf{z}_{t-1} = \gamma_t\boldsymbol{\delta}_t, \qquad \gamma_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} < 1. \tag{73}$$

Repeated factors $\prod_{u=1}^{t}\gamma_u$ attenuate an earlier denoising error in the stated schedule, whereas the paper models predictor-guided search error as accumulating across optimization steps. This is a useful explanation for avoiding an external predictor in the loop, but it is not a distribution-free guarantee of chemical correctness. It assumes bounded local errors and does not cover representation bias, rounding failures, inaccurate textual property labels, or errors from the evaluation property predictor.

9.2.4 Empirical molecular evidence

The MMP dataset contains 198,558 source–target pairs with ADMET annotations. TransDLM reports the best result among five baselines on most structure and property metrics: BLEU 0.740, Levenshtein distance 14.838, Morgan fingerprint similarity 0.665, FCD 0.109, and validity 0.993. For requested property-direction accuracy, it reports LogD 0.157, solubility 0.783, clearance 0.757, and an "All" success ratio of 0.110. The strongest baseline reaches 0.075 on "All", so the reported relative improvement is 46.7%. The model also obtains the best hypervolume (0.929) and R2 indicator (0.121) in the paper's QED/SA analysis.

The near-perfect rather than perfect validity is informative. Search methods can reject invalid candidates explicitly; TransDLM produces a continuous latent in one generative pass and pays a small rounding failure rate. Moreover, property-direction accuracy is itself computed with a trained predictor. The architecture removes a predictor from the *generation loop*, but empirical evaluation still inherits predictor uncertainty. Prospective wet-lab validation and stronger out-of-distribution property assessment remain necessary before treating generated molecules as drug candidates.

10 Multimodal Diffusion Language Models

Domain specialization still assumes one output vocabulary. Multimodality adds a harder asymmetry: image features may be fixed evidence, while text or image tokens can be generated and revised. Let $\mathbf{v} = g_\phi(I)$ and

Method	Specialized object	Specialized mechanism	Representative evidence	Critical limitation
DiffuCoder	Executable program tokens	130B-token code adaptation; order/AR-ness analysis; temperature-dependent order diversity; complementary-mask GRPO	Base HumanEval 67.1; EvalPlus 63.6 → 67.9 with coupled-GRPO	Gains vary by benchmark; likelihood ratios remain estimated rather than exact
TransDLM	Source-related molecule satisfying multiple properties	RECAP/IUPAC textual condition; source-latent initialization; cross-attentive continuous diffusion; SMILES rounding	Validity 0.993; "All" ADMET success 0.110; best reported HV 0.929	Continuous-to-token rounding and predictor-based evaluation can still introduce chemical error

Table 6: Two forms of domain specialization. DiffuCoder changes data, decoding analysis, and RL for code; TransDLM changes representation and conditioning for molecular design.

let p be the visible prompt. LLaDA-V learns visual conditioning directly under diffusion; Dimple first aligns vision under an AR objective; MMaDA also diffuses image outputs. Comparisons must account for vision encoder, data mixture, output canvas, and reward, not only the language backbone.

10.1 LLaDA-V: Direct Visual Instruction Tuning

10.1.1 Architecture and response-only objective

LLaDA-V combines a SigLIP2 vision tower, a two-layer MLP projector, and the LLaDA-8B-Instruct language tower (You et al., 2026). The projector maps visual features into the language embedding space. For turn k with response $r_0^{(k)}$, only response tokens are masked; image features and prompts remain visible. Translating the paper's multi-turn expression into the unified notation gives

$$\mathcal{L}_{\text{LLaDA-V}} = \mathbb{E}\left[\frac{1}{t}\sum_{k}\sum_{i\in\mathcal{M}_t^{(k)}} -\log p_\theta\Big(r_0^{(k),i} \mid \mathbf{v}, p^{(\le k)}, r_t^{(\le k)}, t\Big)\right]. \tag{74}$$

Responses are generated turn by turn, but every response is internally denoised from a fully masked sequence. From level t to $s < t$, the model predicts all masked positions and remasks an s/t fraction, preferentially retaining high-confidence predictions.

The language tower uses bidirectional attention across the packed multimodal dialogue. An ablation against a dialogue-causal mask is mixed rather than uniformly positive: bidirectional attention wins on seven of twelve reported benchmarks and improves MuirBench strongly, but causal attention is better on some perception benchmarks. The result supports full-context modeling as a reasonable default, not as a universal superiority claim.

10.1.2 Three-stage data curriculum and evidence

Training proceeds through (i) projector-only alignment on 558K LLaVA-Pretrain samples, (ii) full-model visual instruction tuning on 10M single-image and approximately 2M OneVision samples, and (iii) multimodal reasoning enhancement using 900K VisualWebInstruct pairs followed by balanced reasoning/non-reasoning data. Maximum length reaches 16,384 in the OneVision and balanced stages.

Against an in-house LLaMA3-V baseline with the same vision tower, projector, and training pipeline, LLaDA-V wins six of nine multidisciplinary knowledge and mathematical benchmarks. It reports MMMU 48.6 versus 45.4, MMMU-Pro 35.2 versus 28.3, MMStar 60.1 versus 56.5, and MMBench 82.9 versus 79.8. It trails on MathVista (59.7 versus 62.1), AI2D (77.8 versus 81.1), DocVQA (83.9 versus 86.2), and RealWorldQA (63.2 versus 66.0), while improving multi-image/video benchmarks such as MuirBench and MLVU. The paper also shows monotonic improvements with more visual data and particularly favorable scaling on MMMU-type tasks. Because the underlying language towers are not equally strong on text, the fairest conclusion concerns the viability and data scaling of the diffusion interface, not factorization alone.

10.2 Dimple: AR-Then-Diffusion Multimodal Tuning

10.2.1 Why pure diffusion tuning is unstable

Dimple begins from Dream and a Qwen2.5-VL vision encoder (Yu et al., 2026). The authors identify two supervision gaps in pure diffusion tuning: one sampled mask level supervises only masked answer positions, and it covers only one point on the reverse trajectory. In contrast, teacher-forced AR training supervises every answer token and every left-to-right prediction step in one example. In the paper's experiments, direct diffusion alignment/tuning produces higher instability and severe sensitivity to the prescribed response length.

Dimple therefore uses three operational stages: AR visual alignment, AR instruction tuning, and diffusion instruction tuning. The first two replace Dream's full attention with a causal mask and use the LLaVA-NeXT recipe; the last restores full attention and response-only masked loss. Random numbers of Dream padding tokens replace end-of-sequence markers during diffusion tuning, teaching the model that a fixed response canvas can contain a shorter semantic answer.

10.2.2 Adaptive commitment, approximate prefilling, and structure priors

Let $\gamma_i(t)$ be the confidence from Table 1. Fixed-K confidence decoding commits a predetermined number of positions. Dimple instead selects

$$\mathcal{U}_t = \{i \in \mathcal{M}_t : \gamma_i(t) \geq \tau\}, \quad \mathcal{U}_t \leftarrow \{\arg\max_{i \in \mathcal{M}_t} \gamma_i(t)\} \text{ if } \mathcal{U}_t = \varnothing. \tag{75}$$

This permits many easy tokens to be filled together and guarantees progress when none exceeds threshold τ. In one structured-output example, 22 unknown tokens are completed in seven iterations.

Visual prompts can be long, so Dimple caches prompt keys and values after the first pass. This is exact for causal attention but approximate under full bidirectional attention: changing response states can change prompt-side hidden activations. Empirically, prefilling causes an average 0.8% performance drop across the reported benchmarks while producing a mean 1.79× speedup at batch size 1 and up to roughly 7× at batch size 32.

Structure priors fix selected response positions before denoising. They can impose a JSON-like format, place a final answer near the end of a fixed canvas, or provide intermediate scaffold phrases. This is more direct than prompt-only format instruction because the sampler is prohibited from changing those positions. It is also less flexible: the user must know a compatible position-level structure and response length before generation.

10.2.3 Results, length bias, and comparison to LLaDA-V

Dimple-7B is trained on approximately 1.3M multimodal samples and 0.8B tokens. It reports a 62.4% average across its benchmark suite versus 58.5% for LLaVA-NeXT, a 3.9-point gain, and wins eight of thirteen comparisons among similarly scaled systems. Direct diffusion tuning is much worse than AR-then-diffusion in the ablation. On ChartQA, the pure diffusion model's accuracy falls sharply as response length increases, reaching 8.6 at length 32, while the hybrid model remains much more stable. The result identifies *length modeling* as a multimodal diffusion problem rather than a cosmetic padding choice.

LLaDA-V and Dimple are complementary rather than redundant. LLaDA-V shows that direct diffusion visual instruction tuning can scale with large multimodal corpora and a from-scratch diffusion language tower. Dimple shows that AR supervision can be a data-efficient bridge when the language tower already came from an AR checkpoint, and it develops adaptive decoding and positional control. Their data scales and backbones differ substantially, so cross-paper benchmark numbers should not be treated as a head-to-head ranking.

10.3 MMaDA: One Discrete Diffusion Interface for Understanding and Generation

LLaDA-V and Dimple generate text from fixed visual evidence. MMaDA targets a broader interface in which the output itself may be text or image tokens, while the same backbone performs language reasoning,

multimodal understanding, and text-to-image generation (Yang et al., 2025). Text uses the LLaDA tokenizer. A MAGVIT-v2 quantizer maps a 512×512 image to $32 \times 32 = 1024$ discrete tokens from a codebook of size 8192. After modality-specific tokenization, both token types are trained by the same masked prediction objective:

$$\mathcal{L}_{\text{MMaDA}} = \mathbb{E}_{x_0,t,x_t}\left[\frac{1}{\rho(t)L}\sum_{i\in\mathcal{M}_t} -\log p_\theta(x_0^i \mid x_t, c, t)\right], \tag{76}$$

where $\rho(t) = |\mathcal{M}_t|/L$. This is Eq. 7 with inverse-mask-ratio weighting; the vocabulary and condition identify the modality, not a separate AR or image-diffusion head.

Post-training has two stages. Mixed long-chain-of-thought SFT uses a common format containing a reasoning segment followed by a result, including an image-token result for generation tasks. Prompts remain visible and only response positions are corrupted:

$$\mathcal{L}_{\text{mixed}} = \mathbb{E}_{q,o,t,\tilde{o}_t}\left[\frac{1}{|\mathcal{M}_t|}\sum_{i\in\mathcal{M}_t} -\log p_\theta(o^i \mid q, \tilde{o}_t, t)\right]. \tag{77}$$

The data mix covers textual reasoning, visual reasoning, and world-knowledge-aware image generation. A shared textual rationale may help transfer structure across tasks, but for image generation it is a supervised interface choice rather than evidence that visual synthesis internally follows a faithful verbal chain.

UniGRPO then adapts group-relative policy optimization to masked diffusion. For completion o_i, sample a mask ratio $p_i \sim \mathcal{U}[0,1]$, construct $\tilde{o}_i$, and define the survey's normalized log-policy score

$$\widehat{\ell}_\theta(o_i \mid q, \tilde{o}_i, p_i) = \frac{1}{|\mathcal{M}_i|}\sum_{k\in\mathcal{M}_i} \log p_\theta(o_i^k \mid q, \tilde{o}_i, p_i), \qquad \widehat{\pi}_\theta = \exp(\widehat{\ell}_\theta). \tag{78}$$

This notation resolves an ambiguity in the paper, where a quantity called π'_θ is sometimes written as an average log-likelihood and then used in a ratio. The PPO-style ratio is coherently expressed as

$$r_i(\theta) = \frac{\widehat{\pi}_\theta(o_i \mid q, \tilde{o}_i, p_i)}{\widehat{\pi}_{\text{old}}(o_i \mid q, \tilde{o}_i, p_i)} = \exp\Big(\widehat{\ell}_\theta - \widehat{\ell}_{\text{old}}\Big). \tag{79}$$

UniGRPO applies the usual clipped surrogate with group-normalized advantage and a reference KL penalty. Across inner updates it samples one random starting noise level and then evenly spaces later levels over the remaining reverse trajectory. This remains an approximate completion policy: averaging masked-token scores does not reconstruct the exact any-order sequence likelihood.

The reward $R(o)$ is modality-specific inside a shared optimization template. Textual reasoning uses correctness and format rewards; geometric or visual reasoning adds task correctness and, for captions, a scaled CLIP term; image generation uses scaled CLIP and ImageReward signals. This is unified at the optimizer level, not at the semantics of the verifier. Optimizing CLIP Score or ImageReward and reporting the same family of metrics also creates evaluation coupling that should be distinguished from independent human preference.

The stage ablation is the clearest evidence for the design. From pretraining to mixed-CoT SFT to UniGRPO, GSM8K moves $17.4 \to 65.2 \to 73.4$, MATH500 $4.2 \to 26.5 \to 36.0$, GeoQA $8.3 \to 15.9 \to 21.0$, CLEVR $10.3 \to 27.5 \to 34.5$, CLIP Score $23.1 \to 29.4 \to 32.5$, and ImageReward $0.69 \to 0.84 \to 1.15$. MMaDA-8B also reports MMLU 68.4, ARC-C 57.4, GSM8K 73.4, and MATH 36.0; it improves substantially over LLaDA on several reasoning tasks but does not surpass Qwen2-7B on all language metrics. On image generation it reports CLIP Score 32.46, ImageReward 1.15, WISE-Cultural 0.67, and GenEval overall 0.63. Comparisons mix dedicated and unified baselines with different data and objectives, so they establish breadth rather than controlled architectural dominance.

Sampling results illustrate modality-dependent compute. From a 1024-token image canvas, CLIP Score changes from 32.8 at 1024 steps to 32.0 at 50 and 31.7 at 15. Text MMLU changes from 66.9 at 1024 steps to 65.7 at 256. These are promising Pareto points, but the paper reports steps and task metrics rather than a complete hardware-matched latency comparison. MMaDA makes image tokens part of the diffused output and carries heterogeneous rewards through one approximate policy-gradient interface. This breadth increases the need for modality-specific validity, independent evaluation, and trust analysis.

Model	Language/vision base	Training route	Inference and control	Evidence and caveat
LLaDA-V	LLaDA-8B-Instruct + SigLIP2	Projector alignment; direct full-model diffusion visual tuning; reasoning curriculum	Low-confidence remasking; fixed target response length; bidirectional dialogue attention	Strong data scaling; beats matched LLaMA3-V on 6/9 knowledge/math tasks but trails on several document/scene tasks
Dimple-7B	Dream/Qwen2.5 lineage + Qwen2.5-VL encoder	AR alignment and instruction tuning, then diffusion recovery	Threshold-based confident decoding; approximate prompt prefilling; fixed structure priors	62.4 vs. 58.5 average for LLaVA-NeXT; pure diffusion exhibits instability and length bias
MMaDA-8B	LLaDA tokenizer + MAGVIT-v2 image tokenizer	Unified masked pretraining; mixed long-CoT SFT; UniGRPO with diversified rewards	Text, visual understanding, and image-token denoising; flexible step budgets	Broad three-task evidence and strong stage ablation; heterogeneous data/rewards and partially coupled evaluation

Table 7: Three routes to multimodal diffusion. LLaDA-V directly tunes visual conditioning, Dimple learns vision under AR supervision before restoring diffusion, and MMaDA diffuses both text and image outputs under one token objective.

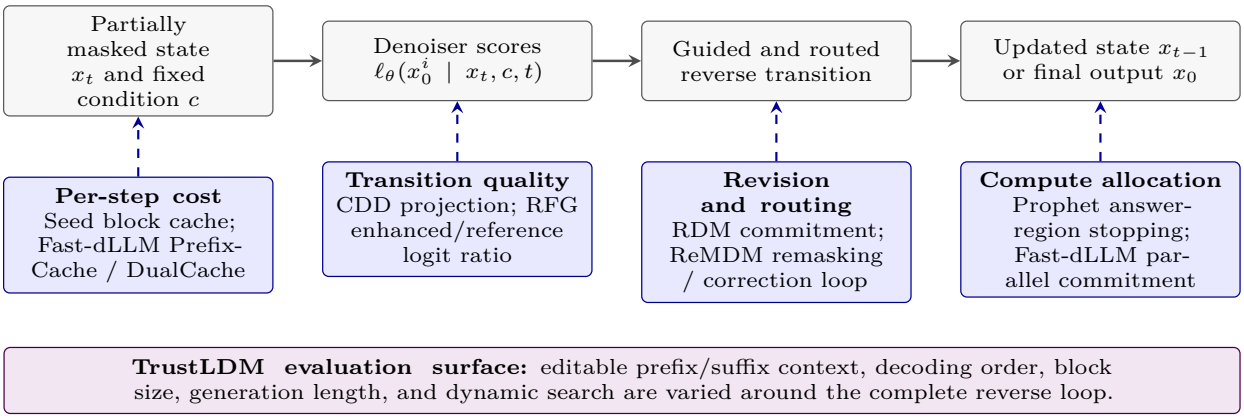


Figure 7: Intervention map for a masked-diffusion reverse step. Methods alter different resources and can therefore be complementary: RFG changes transition scores, ReMDM changes revisability, Prophet changes the stopping time, and Fast-dLLM changes both per-step computation and parallel commitment. TrustLDM probes the surrounding context and decoding configuration rather than one isolated transition.

11 Inference-Time Computation: Correction, Guidance, Stopping, and Systems

Architectural parallelism is not equivalent to low latency. A reverse chain can still require many full-sequence evaluations, and aggressive commitment can damage dependencies. Inference-time methods address different resources: ReMDM buys correction, RFG buys transition quality with an extra model call, Prophet removes late steps, and Fast-dLLM lowers per-step cost and increases safe commitment. Figure 7 separates these interventions so that "acceleration" does not conflate quality scaling, compute allocation, and systems reuse.

11.1 Prophet: Early Answer Convergence as Adaptive Stopping

Full-budget decoding may keep revising a chain of thought after the short answer region has already stabilized. Li et al. (2026) measure when the top-1 predictions over ground-truth answer tokens first become correct, conditional on final correctness. On LLaDA-8B and GSM8K, random remasking yields correct answer tokens by half of the scheduled steps in 97.2% of such samples; with a suffix anchor marking the answer region this becomes 97.3%. Under low-confidence remasking the corresponding proportions are 24.2% without the anchor and 75.8% with it. These are conditional trajectory statistics, not a claim that 97% of the full benchmark is solved halfway.

Prophet converts the observation into a training-free stopping rule. Let $\ell_{t,i}^{(1)}$ and $\ell_{t,i}^{(2)}$ be the largest and second-largest logits at position i. For a known answer region $\mathcal{A}$,

$$g_t^i = \ell_{t,i}^{(1)} - \ell_{t,i}^{(2)}, \qquad \bar{g}_t = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} g_t^i. \tag{80}$$

With descending step index t and budget $T_{\max}$, progress is $u = (T_{\max} - t)/T_{\max}$. The staged threshold is

$$\tau(u) = \begin{cases} \tau_{\text{high}}, & u < 0.33, \\ \tau_{\text{mid}}, & 0.33 \le u < 0.67, \\ \tau_{\text{low}}, & u \ge 0.67. \end{cases} \tag{81}$$

When $\bar{g}_t \ge \tau(u)$, all remaining masks are filled by the current argmax and sampling terminates. The reported default $(\tau_{\text{high}}, \tau_{\text{mid}}, \tau_{\text{low}}) = (7.5, 5.0, 2.5)$ demands strong evidence for an early high-saving exit and relaxes the threshold when little budget remains.

Across LLaDA-8B-Instruct and Dream-7B-Instruct, Prophet reports up to 3.40× fewer decoding steps while generally preserving benchmark accuracy. Examples include LLaDA GSM8K 77.1 → 77.9 at 1.63×, Dream Sudoku 89.0 unchanged at 3.40×, and LLaDA HumanEval 30.5 unchanged at 1.20×. The variation is meaningful: code receives fewer savings because confidence stabilizes later. Prophet also combines with a 2-step-distilled LLaDA model for 3.21× total step reduction on GSM8K, and with Fast-dLLM for 7.66× total speedup versus 6.82× for Fast-dLLM alone.

There are three important caveats. First, a high top-two logit gap is confidence, not calibrated correctness; some positive accuracy changes may reflect avoiding late corruption, but negative changes remain on WinoGrande and several Dream tasks. Second, the answer region and often an "Answer:" suffix must be identifiable in advance, which is natural for short-answer benchmarks but not open-ended generation. Third, the paper reports step reduction as "speedup" in its main table, while wall-clock gain depends on attention cost, block schedule, and hardware. Prophet is best understood as an adaptive stopping layer that can sit above a sampler, not as a replacement for correction, guidance, or cache optimization.

11.2 Fast-dLLM: Lowering Cost per Step and Parallel Commitment

Seed Diffusion learns faster trajectories and freezes completed blocks during training and deployment. Dimple adapts the number of committed tokens and approximately caches a fixed visual prompt. Fast-dLLM asks how far an already trained masked dLLM can be accelerated without changing its weights (Wu et al., 2026). Its answer combines approximate KV reuse with a criterion for when independent marginals are reliable enough for parallel commitment.

11.2.1 Why Parallel Marginals Can Break Token Dependencies

Suppose positions i and j are masked under evidence $E = (c, x_t^{\mathcal{V}_t})$. A standard parallel sampler uses

$$q(x^i, x^j \mid E) = p_\theta(x^i \mid E)\, p_\theta(x^j \mid E), \tag{82}$$

whereas the true joint can be factorized as $p_\theta(x^i, x^j \mid E) = p_\theta(x^i \mid E) p_\theta(x^j \mid E, x^i)$. The difference is consequential for multi-token units such as "high card" versus the invalid recombination "high house", and it becomes more severe as $|\mathcal{U}_t|$ grows. The main speed–quality problem is therefore not merely low token confidence; it is whether several high marginal modes are jointly compatible.

11.2.2 Approximate PrefixCache and DualCache

Bidirectional attention prevents exact AR-style caching because any response update can change hidden states at all positions. Fast-dLLM adopts block-wise generation. A completed prefix block is treated as stable, its keys and values are reused while the active block is denoised, and caches are refreshed at block boundaries. PrefixCache stores the prompt and completed prefix. DualCache additionally stores the still-masked suffix.

The approximation is motivated by measured cosine similarity of adjacent-step key/value activations, not by exact conditional independence.

Cache block size controls a three-way trade-off. Small blocks refresh frequently and reduce staleness but add overhead. Large blocks reuse more computation but allow a greater mismatch between cached and current bidirectional states. A block size of 32 gives the best reported compromise in the main text setup. For LLaDA-V, short blocks are especially harmful: MathVista loses more than eight points when block length is reduced from 96 to 8. Fast-dLLM therefore keeps the full visual response block and refreshes its cache every r steps instead.

11.2.3 Confidence Theorem and Factor-Based Commitment

Let n candidate positions have marginal modes $x^{\star,1:n}$ with $p_\theta(x^{\star,j} \mid E) > 1 - \epsilon$ for every j. The paper proves that if

$$(n+1)\epsilon \leq 1, \tag{83}$$

then the greedy mode of the product of marginals and the greedy mode of the true joint are both $x^{\star,1:n}$. It also bounds total variation by $D_{\mathrm{TV}}(p, q) < (3n-1)\epsilon/2$ and provides a forward-KL bound that grows with n, ϵ, and vocabulary size. These statements do not make parallel decoding exact in general; they identify a high-confidence regime where the greedy decision is protected.

The threshold strategy commits all positions with $\gamma_i(t) \geq \tau$, with a one-token fallback as in Eq. 75. The factor strategy more closely follows Eq. 83. Sort confidences $\gamma_{(1)} \geq \cdots \geq \gamma_{(m)}$ and select the largest n satisfying

$$(n+1)\big(1 - \gamma_{(n)}\big) < f, \tag{84}$$

where f is a deployment knob. Unlike fixed-K decoding, n shrinks automatically when the active distribution is uncertain.

11.2.4 End-to-End Results and What the Speedups Mean

Experiments use an NVIDIA A100 80GB GPU and evaluate LLaDA, LLaDA-1.5, Dream, and LLaDA-V. On LLaDA-Instruct at generation length 512, Fast-dLLM reports 11.0× throughput on GSM8K, 5.9× on MATH, 4.0× on HumanEval, and 9.2× on MBPP, with accuracy changes typically within one or two points. On Dream-Base, the combined method reaches 5.6×, 6.5×, 3.2×, and 7.8× on the same task order. Longer prompts and outputs amortize cache computation more strongly. In an 8-shot LLaDA setting with maximum length 1024, DualCache plus parallel decoding reaches 27.6× relative to the uncached one-token baseline while changing GSM8K accuracy from 77.3 to 76.0.

For LLaDA-V, refresh-based Fast-dLLM reports 9.9× throughput on MathVista with accuracy 56.6 versus 59.2 at full steps, and 8.5× on MathVerse with accuracy 28.6 versus 28.5. These results demonstrate cross-modal applicability, but also show that a headline speedup may correspond to a non-negligible task-specific drop. Every speed number is relative to the paper's own unoptimized diffusion baseline, not to a universal AR implementation.

12 Trustworthiness Under Editable Context and Flexible Decoding

The flexibility used for correction and control also enlarges the attack surface. Full-sequence attention lets an attacker place text *after* a masked answer, constrain both prefix and suffix, or choose a decoding order and response canvas that amplify malicious context. Plain-prompt alignment therefore does not characterize the deployed system. TrustLDM evaluates safety, privacy, and fairness over this configuration space rather than importing an AR benchmark unchanged (Mo et al., 2026).

12.1 Static Evaluation: Context, Order, Block Size, and Length

The static benchmark contains 50 AdvBench harmful prompts and all JailbreakBench misuse behaviors for safety; 500 database-manager scenarios for privacy; and 200 gender-balanced Adult-income examples for

Mechanism	Reused or committed object	Why it can work	Failure mode	Control knob
Seed block decoding	Completed causal prefix blocks	Frozen blocks create an exact stable prefix for later blocks	Restricts global revision and changes effective factorization	Block size and learned trajectory
Dimple prefilling	Long multimodal prompt	Visual/prompt activations change little in measured tasks	Bidirectional response updates can stale prompt states	Enable/disable prefill; response length
Fast PrefixCache	Prompt and completed prefix	Adjacent-step KV activations are highly similar	Staleness grows with block size and state change	Cache block size and refresh interval
Fast DualCache	Prefix plus masked suffix	Masked suffix activations are also similar across nearby steps	Approximation affects both sides of active block	Block size and refresh interval
Adaptive confidence	Several response positions	High marginal confidence bounds greedy joint disagreement	Correlated moderate-confidence tokens can form invalid combinations	Threshold τ or factor f

Table 8: Acceleration mechanisms and their approximation boundaries. The rows differ in what is frozen, reused, or committed; "parallel" is not one uniform operation.

fairness. Safety reports harmful rate (HR) under an LLM judge. Privacy reports exact-string leakage rate (LR). Fairness uses equalized-odds difference,

$$\mathrm{EOD} = \max_{y \in \{0,1\}} \left| \Pr(\widehat{Y} = 1 \mid Y = y, A = \text{male}) - \Pr(\widehat{Y} = 1 \mid Y = y, A = \text{female}) \right|, \tag{85}$$

where lower values are preferred. The paper evaluates LLaDA, LLaDA-1.5, LLaDA-MoE, and Dream under six orders: random, left-to-right, right-to-left, smallest entropy, highest confidence, and largest margin. It also varies block size in $\{16, 32, 64, 128\}$ and generation length in $\{128, 256, 512, 1024\}$.

For safety, a suffix may be a warning, a short inducive statement, or a longer positive framing; an affirmative prefix can be added before the masks. With no injected context, average HR is at most 3% for every model on both safety sets. This apparent alignment collapses under editable context. On TrustLDM-Adv, LLaDA HR rises from 3% to 73% with a short suffix and 90% with short suffix plus prefix; LLaDA-1.5 rises from 2% to 72% and 93%. LLaDA-MoE and Dream are more robust to suffix-only injection but reach 59–60% with the combined short prefix/suffix setting. Order interacts with where the adversarial evidence is placed: right-to-left is particularly vulnerable to suffix-only context, while several confidence-based orders become highly vulnerable once both sides are constrained.

Privacy is already weak for some plain prompts in this role-playing setup: average empty-context LR is 51.5% for LLaDA, 54.6% for LLaDA-1.5, 40.7% for LLaDA-MoE, and 30.8% for Dream. Joint prefix–suffix injection drives every model to roughly 90–96% leakage. Fairness shows a similarly sharp context effect. Empty-context EOD ranges from 12.0% to 33.1%, but misleading suffixes produce 88.0–98.7%, and combined contexts approach 100%. Across fairness configurations, EOD and prediction accuracy have Pearson correlation -0.863, so the observed disparity is not a utility-improving trade-off; lower accuracy and worse fairness tend to occur together.

Longer context, larger blocks, or longer generation are not monotonically stronger attacks. Very long suffixes can distract a weaker instruction follower, and a 1024-token response sometimes becomes repetitive or meaningless, lowering measured harmfulness or leakage because the model no longer produces a coherent violation. Such decreases are not improved alignment. They expose a general evaluation confound: trust metrics must be paired with output quality and task completion, otherwise generation failure can look safe or private.

12.2 TrustLDM-Auto: Searching the Decoding Configuration

Static strings can be memorized or tuned against, so TrustLDM-Auto uses an attack model to revise prefix and suffix, an LDM to generate the candidate response, and a judge to score violations. Decoding order and response length are search variables. Hierarchical search first evaluates many configurations with few denoising steps, keeps the top-K, and regenerates them at higher fidelity. Space shrinking progressively

reduces the active sets $\mathcal{G}_t$ of lengths and $\mathcal{O}_t$ of orders:

$$|\mathcal{G}_t| = \left\lceil \left(1 - \frac{t}{T_{\max}}\right) |\mathcal{G}_0| \right\rceil, \qquad |\mathcal{O}_t| = \left\lceil \left(1 - \frac{t}{T_{\max}}\right) |\mathcal{O}_0| \right\rceil. \tag{86}$$

With a maximum of 20 iterations, dynamic evaluation reaches 98–100% HR on open models and 97.2–99.4% privacy leakage. Hierarchical search plus shrinking cuts average successful-query time by roughly an order of magnitude relative to vanilla search. On the closed-source Mercury Edit 2 API, static context changes safety HR from 0 to 70–74%, privacy from 5.43% to 71.96%, and EOD from 16.0% to 90.0%. Automatic search increases them further to 83–94%, 97.83%, and 93.1%.

These results establish a diffusion-specific evaluation surface, not a complete safety comparison with AR models. The benchmark covers three trust dimensions, several English datasets, four open models and one closed model; safety depends on an LLM judge, privacy on exact-string matching, and fairness on one tabular-income task. Some favorable order results are explicitly attributed to worse output quality. Moreover, the automatic framework is dual-use because it searches for high-violation configurations. A defensible conclusion is that plain-prompt alignment does not transfer uniformly to editable contexts, and that deployment configuration must be included in threat modeling, red-teaming, and safety regression tests.

12.3 Trust as a Cross-Layer Property

TrustLDM links design layers that are usually evaluated separately. ReMDM may reopen a refusal, and RFG may amplify helpful or harmful post-training behavior. Prophet may freeze an answer before later correction. Fast-dLLM may preserve stale activations associated with an injected suffix, while MMaDA adds image-token outputs and heterogeneous rewards. TrustLDM does not test these interactions directly. Its results nevertheless show why safety cannot be attached only to the final checkpoint. The deployment object includes the model, context layout, routing rule, stopping policy, cache, length controller, and judge.

13 Comparison and Discussion

13.1 From Local Denoising Decisions to a Capability Stack

Evaluation is distributed across this literature; there is no single paper whose benchmark isolates every source of progress. Representation studies measure likelihood or control, reasoning studies emphasize exact answers and verifiers, systems papers report hardware-specific throughput, scientific applications add validity metrics, and TrustLDM changes the threat model. The synthesis must therefore compare evidence at the layer where the intervention occurs.

At the *modeling layer*, representation and corruption define uncertainty, while the metric and validity model determine whether success means token fluency, exact sequence correctness, executable syntax, chemical validity, or grounded output. Conditioning and scheduling decide which evidence remains stable. The reverse policy and solver then decide what changes, whether earlier decisions can reopen, and how numerical error accumulates.

At the *capability layer*, the scaling route determines whether knowledge is inherited or learned from scratch. Post-training decides whether rewards act on a completion, complementary corruptions, or the whole path. Specialized representations and modality interfaces must preserve program, molecular, source, or visual semantics; length and structure models define the response canvas and padding behavior.

At the *deployment layer*, an inference system converts safe commitment and reusable computation into wall-clock throughput. Its trust configuration includes editable context, order, length, stopping, cache, and trajectory-level safeguards. These layers are coupled: an apparently local sampler change can alter both the measured speed and the effective policy exposed to users. This organization exposes cross-layer dependencies. LLaDA scales an MDLM-style objective, while Dream combines AR transfer with position-dependent weighting. Seed restricts the any-order factorization for systems efficiency. d1, DCoLT, DiffuCoder, and MMaDA optimize different policy units. ReMDM makes commitment reversible; RFG converts post-training differences into guidance; Prophet detects answer convergence; and Fast-dLLM approximates states

that full attention would recompute. TrustLDM evaluates the surrounding configuration rather than only the checkpoint.

13.2 Parallel Prediction Is Not Equivalent to Fast Generation

If a model commits K tokens per reverse step, an optimistic serial-depth measure is N/K. Wall-clock latency is closer to

$$T_{\text{decode}} \approx S\, C_{\text{forward}}(L, B, \mathcal{K}) + C_{\text{refresh}}(\mathcal{K}) + C_{\text{routing}} + C_{\text{verification}}, \tag{87}$$

where S is the number of denoising steps, L the active sequence length, B a block configuration, and $\mathcal{K}$ a cache/refresh policy. ReMDM can increase S to buy correction; RFG can approximately double model work per step to buy transition quality; Prophet reduces S adaptively; Seed restricts the order space and caches completed blocks; and Fast-dLLM lowers forward cost with approximate $\mathcal{K}$ while lowering S through dependency-aware commitment. A fair comparison must therefore report not only throughput but model calls per step, reference/guidance models, remasking loops, stop frequency, hardware, precision, batch size, prompt/output length, block size, cache refreshes, effective tokens, accuracy, and verification time.

13.3 Three Scaling Routes Make Different Errors

AR adaptation is data-efficient but imports the source model's feature geometry and left-to-right bias. From-scratch diffusion avoids that inherited factorization but pays the full pretraining cost and may require architecture changes that complicate inference. Dream defines a hybrid route: preserve an AR-compatible prediction interface while assigning a diffusion-specific token-level learning signal. The DiffuLLaMA ablation shows that the shift is valuable; LLaDA shows that performance scales from scratch; Dream shows that transfer plus rescheduling can outperform the earlier from-scratch dLLM with fewer tokens in the reported protocol. None isolates factorization from checkpoint quality, data, tokenizer, alignment, optimizer, and systems choices. The correct conclusion is that all three are credible routes with different cost and bias profiles.

13.4 Specialization Reopens the Representation Question

The foundational continuous-versus-discrete distinction returns in a less abstract form. DiffuCoder needs executable token identities and uses masked discrete diffusion; TransDLM needs gradients and semantic proximity in a scientific embedding space and uses Gaussian latent diffusion followed by rounding. The appropriate representation is therefore task-dependent. For code, a syntactically wrong token is directly observable and testable. For molecules, a valid SMILES string may still encode the wrong scaffold or property profile, and a nearby latent does not guarantee a nearby chemical. The right comparison includes validity and downstream semantics, not only sequence likelihood.

13.5 Multimodality Turns Length into a Modeling Variable

AR multimodal models stop at an end-of-sequence token. LLaDA-V and Dimple instead allocate a response canvas before denoising; MMaDA additionally allocates a 1024-token canvas when the response is an image. Dimple's pure-diffusion length bias shows that padding and termination semantics can dominate benchmark accuracy even when the visual encoder is unchanged. MMaDA shows that image quality can remain stable under far fewer steps, but its text, understanding, and image branches have different acceptable budgets. A complete multimodal dLLM therefore needs calibrated output type, length, and compute distributions, not only a better visual connector.

13.6 Training and Inference Approximation Are Coupled

Recent methods invalidate a clean separation between "model" and "sampler." Coupled-GRPO and UniGRPO change the distribution of mask contexts used for policy ratios. ReMDM applies a different posterior to a pretrained denoiser; RFG uses the difference between two trained models as a guidance direction; Prophet assumes answer confidence is calibrated enough for stopping; and Fast-dLLM assumes hidden states vary

slowly enough for reuse. The strongest future designs should report cross-combinations: whether an RL-trained model remains cache-stable, whether remasking invalidates a stopping signal, whether RFG amplifies unsafe context dependence, and whether faster multimodal trajectories preserve grounding.

13.7 Self-Correction Requires an Intervention Test

Qualitative trajectories often show an incorrect token becoming correct later. This is necessary but not sufficient evidence of robust self-correction. Scheduled sampling in DoT trains recovery from model-induced states; Seed's edit corruption and ReMDM explicitly reopen visible tokens; d1 inherits verification traces from SFT; and DCoLT rewards trajectories that eventually succeed. ReMDM's controlled remasking probability provides the cleanest intervention among these, but its largest language result is distributional MAUVE rather than exact reasoning recovery. Stronger evaluation should inject known errors, compare irreversible and reversible samplers at matched compute, and measure recovery probability, calibration, and correction cost.

13.8 Trustworthiness Is a Property of the Decoding Configuration

TrustLDM rules out a checkpoint-only view of alignment. A model that refuses a harmful plain prompt can comply when the answer is placed between an affirmative prefix and an inducive suffix. The most vulnerable order changes with model and context, and apparent safety at extreme lengths can result from incoherent output. This finding generalizes beyond the benchmark: any method that changes order, revisability, guidance, stopping, or cache freshness changes the effective policy exposed to users. Safety, privacy, fairness, and utility should therefore be reported jointly over a deployment-relevant grid of configurations.

13.9 Synthesis Across Representative Methods

14 Open Problems and Future Directions

14.1 Matched Scaling Laws

Future scaling studies should match architecture width/depth, tokenizer, data mixture, source checkpoint, training tokens, optimizer, precision, and FLOPs. Likelihood bounds, downstream accuracy, calibration, and wall-clock sampling must be reported together. LLaDA, DiffuLLaMA, and Dream establish credible routes, but none supplies a complete from-scratch-versus-transfer diffusion/AR scaling law under controlled compute. CART also needs an ablation that holds checkpoint and training corpus fixed at 7B scale.

14.2 Metric-Matched Theory for Natural Language

The TER/SER separation gives the first rigorous efficiency boundary, but it is proved for n-gram and HMM targets. Future work should connect conditional denoising error to exact-answer accuracy, executable correctness, semantic equivalence classes, and verifier acceptance under realistic Transformers. The key question is not whether the worst-case lower bound is true, but which structural assumptions on natural data, adaptive routing, or verification permit sublinear steps without hiding sequence errors inside a token-average metric.

14.3 Exact or Controlled-Bias Policy Gradients

d1 gains efficiency by approximating the completion distribution; DCoLT gains fidelity to the reverse process but expands the action horizon; DiffuCoder reduces estimator variance through complementary masks; and UniGRPO averages token scores under structured mask levels. A central theoretical problem is to construct estimators with explicit bias–variance guarantees between these extremes. Useful targets include exact accounting of coupled masks, Rao–Blackwellization across patterns, off-policy correction for stored trajectories, and credit assignment that exploits only verified conditional independencies among simultaneously updated tokens.

Method	Primary layer	Core mechanism	Empirical contribution	Main unresolved issue
Diffusion-LM	Representation / control	Gaussian latent diffusion with gradient guidance	Demonstrates differentiable semantic and structural control	Rounding mismatch and slow reverse sampling
D3PM / MDLM	Discrete base model	Categorical/absorbing corruption and variational masked loss	Principled token-space likelihood modeling	Scaling and decoding policy remain separate problems
Metric-dependent theory	Inference limits	Constant-step near-optimal TER versus linear-step worst-case SER	Formal and synthetic evidence that the metric changes the speed conclusion	HMM/formal-language assumptions and worst-case scope
DiffuSeq family	Conditional denoising	Partial noising, encoder-decoder caching, adaptive schedules	Strong global seq2seq refinement	Long trajectories and MBR/meta-training cost
RDM / jump-process theory	Reverse policy/solver	Explicit routing, confidence commitment, CTMC simulation	Separates token prediction, routing, and numerical error	Theory assumptions and scalable implementation
ReMDM	Correction / test-time scale	Non-Markovian remasking posterior with capped, confidence, switch, or loop schedules	MAUVE 0.042→0.403 at 1024 steps; improved molecule guidance frontier	Extra compute; plug-in posterior differs from training; modest task gains
CDD / SEPO	Control/post-training	Reverse projection or reward-based policy update	Hard constraints and persistent preferences	Projection cost and high-variance iterative RL
DiffuLLaMA	Scaling by transfer	Shift-preserving AR-to-diffusion adaptation	Converts 127M–7B checkpoints with limited continual pretraining	Incomplete base-capability recovery
LLaDA	Scaling from scratch	8B masked generative model with SFT and remasking	General, math, code, ICL, and instruction-following evidence	Expensive full-context decoding; imperfect matched baselines
Seed Diffusion	Deployment	Edit curriculum, order distillation, on-policy shortening, block caching	High-throughput code generation	Undisclosed size and non-standardized speed comparisons
DoT	Supervised reasoning	Rationale diffusion over test-time steps	Reasoning-time trade-off, diversity, and correction cases	Small/task-specific evaluation
d1	Completion-level RL	One-step approximate log probability and diffu-GRPO	First large masked-dLLM policy-gradient recipe	Approximate ratios and costly online generation
DCoLT	Trajectory-level RL	Reward every reverse step and learn unmask order	Strong math/code gains; direct UPM ablation	Reward scope, compute, and thought faithfulness
RFG	Test-time guidance	Enhanced/reference log-density ratio as implicit step reward	Consistent math/code gains; up to +9.2 HumanEval points	Two model calls per step and reward-optimum assumption

Table 9: Integrated comparison from foundational modeling and theory to correction, scale, and reasoning. Each method is placed at its primary intervention layer while retaining its main unresolved dependency.

14.4 Adaptive Length, Step Count, and Block Size

Current models often choose response length, denoising steps, tokens per step, remasking budget, guidance strength, cache refresh interval, and block size globally. Reasoning results show that more tokens can help MATH yet harm Sudoku; ReMDM benefits from extra correction; Prophet finds many redundant late steps; Dimple shows length bias; and Seed/Fast-dLLM show that larger blocks improve reuse but can degrade state fidelity. A unified controller should allocate canvas length and reverse computation per example, with calibrated stopping, correction, padding semantics, and a verifier-aware budget.

14.5 Caching Under Bidirectional Dependence

Standard KV caching is exact because an AR prefix never changes. In diffusion, a token update can alter both earlier and later hidden states. Block diffusion avoids this by freezing completed blocks, whereas PrefixCache and DualCache reuse approximately stable activations. Future work should bound output error as a function of activation drift, layer depth, block size, and refresh interval. Cosine similarity is useful evidence but is not by itself a bound on final token probability or task accuracy.

14.6 Dependency-Aware Parallel Decoding Beyond Marginal Confidence

Fast-dLLM proves a greedy equivalence result in a high-confidence regime, but correlated moderate-confidence tokens remain difficult. Future samplers could detect dependency clusters, jointly decode multi-token units, or estimate a low-rank correction to the product of marginals. Code delimiters, chemical tokens, named entities, and visual OCR spans provide natural test cases because their errors are structured and measurable.

Method	Primary layer	Core mechanism	Representative evidence	Main unresolved issue
Dream 7B	Scaling / objective	AR initialization, preserved shift, CART token-specific context weighting	Stronger general/math/code results than LLaDA in the paper's protocol; large planning gains	Checkpoint, data, and CART contributions are not fully isolated
DiffuCoder	Code / RL	Code adaptation, AR-ness analysis, temperature-dependent order diversity, coupled-GRPO	EvalPlus 63.6 → 67.9; smaller loss under half-step decoding	Estimated policy ratios and heterogeneous gains across code suites
TransDLM	Scientific application	RECAP/IUPAC condition, source-latent initialization, cross-attentive continuous diffusion	Validity 0.993 and best reported multi-property "All" ratio 0.110	Rounding error, predictor-based evaluation, and lack of prospective validation
LLaDA-V / Dimple	Multimodality	Direct diffusion tuning versus AR-then-diffusion; response-only masking; adaptive routing/priors	Competitive matched multimodal results and favorable data scaling	Fixed response canvases, length bias, and unmatched backbone/data comparisons
MMaDA	Unified multimodality / RL	Shared discrete text/image objective, mixed long-CoT SFT, and UniGRPO	Stage-wise gains across text, visual reasoning, and image generation	Approximate likelihood, heterogeneous rewards, and evaluation coupling
Prophet	Adaptive stopping	Answer-region top-two logit gap with staged exit threshold	Up to 3.4× step reduction; combines with Fast-dLLM to 7.66×	Requires identifiable answer region and calibrated confidence
Fast-dLLM	Deployment	Approximate Prefix-Cache/DualCache plus threshold- or factor-based parallel commitment	Up to 27.6× relative speedup; 8.5–9.9× on LLaDA-V	Cache staleness and task-dependent accuracy loss; speed is baseline-specific
TrustLDM	Trust evaluation	Static and searched prefix/suffix contexts with order and length variation	Near-perfect automatic violation rates; closed-source evaluation	Limited dimensions/models; judge and quality confounds; dual-use search

Table 10: Synthesis of specialized, multimodal, inference-time, and trust-evaluation systems. The comparison makes domain validity, adaptive compute, cache staleness, and configuration-dependent risk explicit scientific objects.

14.7 Rewards Beyond Verifiable Mathematics and Code

d1 and DCoLT rely on exact answers, symbolic checks, formatting rules, or unit tests; DiffuCoder uses executable code tests. Factual dialogue, multimodal grounding, summarization, safety, and creative generation lack equally reliable verifiers. Reward models must avoid favoring superficial traces, exploiting weak tests, or collapsing the order diversity that motivates diffusion. Combining CDD-style hard guarantees with learned preferences is a promising but computationally difficult direction.

14.8 Scientific Validation Without Evaluation Closure

TransDLM removes an external property predictor from the optimization loop but uses a predictor to score generated ADMET changes. ReMDM's QM9 experiment uses RDKit-computed validity, novelty, ring count, and QED rather than biological assays. These are appropriate algorithmic tests but not evidence of drug efficacy. Future molecular and biological studies should report uncertainty-calibrated ensembles, scaffold splits, out-of-distribution assays, structure-aware validity, docking or physics-based checks where appropriate, and prospective laboratory measurements. Chemical validity, synthetic accessibility, novelty, target affinity, and multi-property success should remain separate metrics.

14.9 Multimodal Grounding Under Global Revision

LLaDA-V and Dimple establish visual instruction tuning, while MMaDA adds image-token generation and heterogeneous reward optimization. Global revision can change claims that were previously grounded in an image, and a text-derived reward can improve one branch while distorting another. Evaluation should track when an object, number, OCR span, or relation appears; whether later steps correct or hallucinate it; how cache staleness changes that trajectory; and whether image rewards generalize beyond the metrics used for training. Variable image tokens and multi-image/video evidence also require adaptive compute.

14.10 Faithful Reasoning and Causal Self-Correction

Intermediate diffusion states are editable but not automatically explanatory. Benchmarks should distinguish final accuracy, recovery from injected errors, counterfactual dependence on intermediate states, and human-readable faithfulness. DCoLT deliberately permits ungrammatical latent states, while DoT exposes explicit rationales; these should be evaluated with different interpretability criteria.

14.11 Standardized Speed-Quality Reporting

Tokens/s without context is insufficient. A minimum report should identify hardware, model size, precision, batch size, prompt/output length, effective tokens, denoising steps, and model calls per step. It should also give remasking loops, guidance strength, stopping frequency, block size, cache policy, refresh interval, average commitment, verifier cost, and matched task accuracy. Step-reduction factors such as Prophet's are not wall-clock speedups unless measured end to end. Pareto curves are more informative than one maximum-throughput point.

14.12 Configuration-Aware Trust and Mitigation

TrustLDM diagnoses prefix/suffix and decoding vulnerabilities but does not yet provide trajectory-preserving safeguards. Future work should evaluate adaptive refusal constraints, CDD-style projection, remasking, RFG, early stopping, and cache reuse under the same attacks. Benchmarks should expand beyond three dimensions and one language, pair violation metrics with output quality, report judge uncertainty, and disclose the search budget. The defensive goal is a constraint that remains valid under every allowed context layout and reverse configuration, not one prompt template that happens to trigger refusal.

14.13 Compositional Compatibility Across the Stack

A mature DLM may combine AR initialization, token-adaptive training, scientific or visual conditioning, learned routing, correction, guidance, stopping, caching, and trajectory RL. These components can conflict. Remasking can invalidate a stop signal or cache; guidance can amplify unsafe context dependence; RL can collapse diverse orders; and multimodal reuse can preserve stale grounding. Future work should report compatibility matrices, trust regressions, and end-to-end ablations rather than isolated gains.

15 Conclusion

Four findings emerge from the evidence. First, token-space formulations have largely resolved the original representation mismatch, so the main bottleneck has moved from defining corruption to acquiring capability and controlling the reverse policy. Second, current scaling gains are not attributable to architecture alone: data, initialization, supervised or reinforcement post-training, decoding, and hardware change together in most comparisons. Third, parallelism is conditional rather than absolute. A method can improve token-level error or model calls while losing on exact-sequence accuracy, wall-clock latency, cache fidelity, or verifier cost. Fourth, bidirectional editability is dual-use: the same interface enables infilling, correction, and guidance, but also exposes suffix-, order-, and length-dependent trust failures.

These findings lead directly to the agenda in Section 14. Matched scaling laws are needed to identify architectural gains; metric-matched theory and dependency-aware decoding are needed to connect token parallelism to sequence correctness; and joint controllers should allocate length, denoising steps, remasking, block size, and cache refresh per example. Scientific and multimodal systems additionally require domain-valid evaluation, while trust studies must test the same sampler, cache, guidance, and stopping configurations used at deployment. The central object of future study is therefore not a checkpoint in isolation, but a complete reverse-process stack with explicit resources, guarantees, and failure boundaries.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

ByteDance Seed, Institute for AI Industry Research, Tsinghua University, and SIA-Lab of Tsinghua AIR and ByteDance Seed. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

Michael Cardei, Jacob K. Christopher, Bhavya Kailkhura, Thomas Hartvigsen, and Ferdinando Fioretto. Constrained discrete diffusion. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0415.

Tianlang Chen, Minkai Xu, Jure Leskovec, and Stefano Ermon. RFG: Test-time scaling for diffusion large language model reasoning with reward-free guidance. In *International Conference on Machine Learning*, 2026.

Yun-Yen Chuang, Hung-Min Hsu, Kevin Lin, Chen-Sheng Gu, Ling Zhen Li, Ray-I Chang, and Hung-yi Lee. Meta-DiffuB: A contextualized sequence-to-sequence text diffusion model with meta-exploration. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical benefit and limitation of diffusion language model. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0825.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. DiffuSeq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations*, 2023.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Hao Peng, Jiawei Han, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. In *International Conference on Learning Representations*, 2025.

Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation. In *International Conference on Learning Representations*, 2026.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4521–4534, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Zemin Huang, Zhiyang Chen, Zijun Wang, Tiancheng Li, and Guo-Jun Qi. Reinforcing the diffusion chain of lateral thought with diffusion language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Yi Liang, Soroush Vosoughi, and Shiwei Liu. Diffusion language models know the answer before decoding. In *International Conference on Learning Representations*, 2026.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4328–4343, 2022.

Yichuan Mo, Yukun Jiang, Yanbo Shi, Mingjie Li, Michael Backes, Yang Zhang, and Yisen Wang. TrustLDM: Benchmarking trustworthiness in language diffusion models. In *Workshop on Trustworthy AI at the International Conference on Learning Representations*, 2026.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yinuo Ren, Haoxuan Chen, Grant M. Rotskoff, and Lexing Ying. How discrete and continuous diffusion meet: Comprehensive analysis of discrete diffusion models via a stochastic integral framework. In *International Conference on Learning Representations*, 2025.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184, 2024.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dLLM: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. In *International Conference on Learning Representations*, 2026.

Yida Xiong, Kun Li, Jiameng Chen, Hongzhi Zhang, Di Lin, Yan Che, and Wenbin Hu. Text-guided multi-property molecular optimization with a diffusion language model. *Information Fusion*, 127:103907, 2026. doi: 10.1016/j.inffus.2025.103907.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal large diffusion language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and Lingpeng Kong. Diffusion of thought: Chain-of-thought reasoning in diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Zebin You, Shen Nie, Xiaolu Zhang, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. LLaDA-V: Large language diffusion models with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10093–10105, 2026.

Runpeng Yu, Qi Li, and Xinchao Wang. Discrete diffusion in large language and multimodal models: A survey. *arXiv preprint arXiv:2506.13759*, 2025.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. In *International Conference on Learning Representations*, 2026.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. SeqDiffuSeq: Text diffusion with encoder-decoder transformers. *arXiv preprint arXiv:2212.10325*, 2022.

Oussama Zekri and Nicolas Boulle. Fine-tuning discrete diffusion models with policy gradient methods. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5112.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1898.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. In *Proceedings of the 1st Conference on Language Modeling*, 2024.