PaRDi: Learning to Preserve and Revise with Discrete Diffusion

Anonymous authors
Paper under double-blind review

Abstract

Existing discrete diffusion methods for image editing often encode the source as a reference sequence and generate a separate target from a fully masked canvas—a paradigm we call *reference-based regeneration*. This forces even content meant to remain unchanged to be generated again, potentially introducing unwanted drift, such as background changes during foreground edits or facial changes during clothing edits. We introduce PaRDi, a discrete diffusion framework that starts from the source canvas and models preservation and repeated revision within the same dynamics. Each token can retain its value or pass through `<MASK>` before taking a new value, and newly generated tokens remain eligible for further revision. These asynchronous updates produce partially revised canvases in which preserved source tokens and completed revisions jointly guide subsequent generation, supporting both local and image-wide edits. Endpoint-conditioned bridges, combined with constant paths at unchanged positions, analytically provide training-state distributions and transition targets, yielding training canvases and transition-matching objectives without hand-designed remasking trajectories. PaRDi achieves 6.98 on GEdit-EN, 3.95 on ImgEdit, and 92.27% background retention on PIE-Bench. Extending in-place editing to visual reasoning, PaRDi attains 87.29% average accuracy, outperforming DiffThinker's 75.36%.

1 Introduction

Motivated by the prospect of modeling text and images jointly as token sequences, recent work has made rapid progress in discrete visual generation (AI et al., 2026; Peng et al., 2026; Cui et al., 2025). Within this paradigm, discrete diffusion offers parallel updates with bidirectional attention, making it attractive for long, spatially structured visual sequences compared with autoregressive next-token prediction (Chang et al., 2022; Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024). For image editing, widely used absorbing-state discrete diffusion (Bond-Taylor et al., 2022; Shi et al., 2026) formulations generate a separate target sequence from a fully masked canvas while encoding the source as a reference sequence—a paradigm we call *reference-based regeneration* (Figure 1a).

Generating a separate target requires reconstructing even content meant to remain unchanged, exposing it to potential drift. Recent remasking methods enable iterative refinement of generated tokens (Wang et al., 2025; Zhao et al., 2026; Li et al., 2026), but in this setup they revise the separate target sequence rather than the source itself. By contrast, editing the source canvas in place lets retained tokens keep their values and provide clean context for modifications (Figure 1c).

Editing in place (Chang et al., 2023; Kulikov et al., 2025; Fang et al., 2026) still requires defining how revisions unfold and how to learn them. The recently proposed Reflective Masking (Zhang et al., 2026) uses hand-designed revision trajectories and token-wise cross-entropy to learn keep, remask, and reveal decisions. It uses mask-all-then-refill inference for image editing and introduces History Reference to retain information from earlier revisions. We seek a source-to-target diffusion that provides training states and transition supervision from the same process, without a separate history mechanism.

Discrete diffusion evolves as a Markov process (Gu et al., 2022; Austin et al., 2021; Hoogeboom et al., 2021; Lou et al., 2024); conditioning it on paired source and target endpoints yields a Markov bridge between them. We introduce *PaRDi*, a framework that uses this construction to model preservation and repeated revision on the source canvas. Source and generated tokens can retain their values or

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

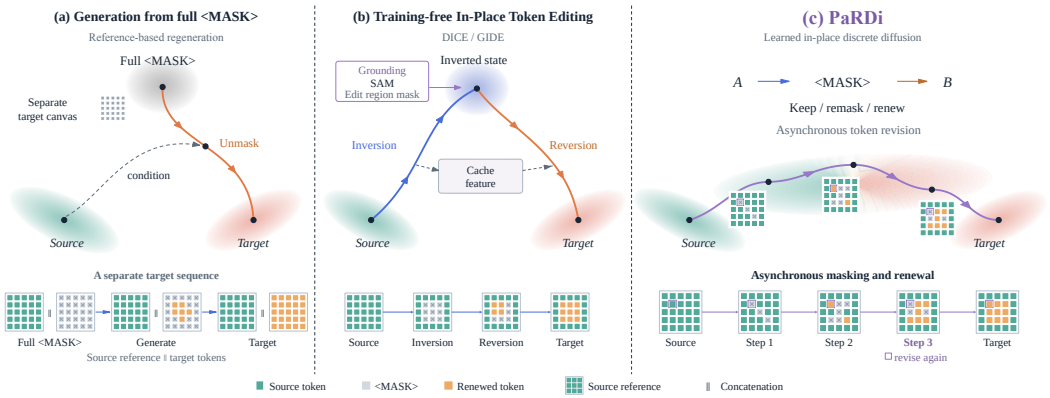

Figure 1: Image-editing trajectories. (a) Reference-based regeneration generates a separate target from <MASK>, using the source as a reference sequence. (b) Training-free in-place methods reuse source information: GIDE (Zhu et al., 2026) masks its grounded edit region before refilling it, whereas DICE (He et al., 2024) follows a mask schedule and reuses inversion residuals. (c) PaRDi revises the source canvas in place, allowing source, <MASK>, and renewed tokens to coexist.

pass through <MASK> to take new values (Figure 1c). Different positions enter and leave <MASK> at different times, allowing edits to proceed without first clearing all positions that need to change. Retained source tokens and newly revised tokens jointly provide context for subsequent updates. The model updates this single canvas in parallel for local or image-wide edits, without a separate fully masked target sequence or history mechanism.

Paired source-target images specify endpoints but not the intermediate canvases or revisions connecting them. PaRDi combines MASK-hub bridges for changed endpoints with constant paths at unchanged positions. The resulting process analytically supplies training canvases and transition targets for discrete-time posterior matching (Austin et al., 2021) and continuous-time rate matching (Campbell et al., 2022). This supervision covers how the canvas evolves toward the target, including preserving and remasking visible tokens. The model learns these dynamics through target-token prediction, without hand-designed remasking trajectories or action labels. At inference, the model predicts target-token probabilities, which the bridge analytically converts into transition probabilities or rates for sampling keep, remask, and renew updates.

Editing experiments show improved quality and source preservation with lower latency than reference-based regeneration. Ablations under the shared PaRDi sampler show that bridge-derived training states and transition matching jointly improve performance. We further test these dynamics on visual reasoning tasks (He et al., 2025), where the revised canvas must satisfy the problem's constraints. Revision-budget ablations show that repeated revision improves reasoning accuracy under the same update sampling schedule.

Our contributions are threefold.

1. **Discrete Diffusion for In-Place Token Editing.** We introduce PaRDi, a discrete diffusion framework that jointly models token preservation and repeated revision on the source canvas. Asynchronous transitions support local and image-wide edits while retaining source content and completed revisions as visual context, without a separate source-token memory or a history of previous canvases.
2. **Bridge-Derived Training States and Objectives.** We analytically derive intermediate-state distributions and target-directed transitions by conditioning the MASK-hub Markov process on paired source–target endpoints. Together with constant paths at unchanged positions, these bridges provide training canvases and discrete-time posterior-matching and continuous-time rate-matching objectives.
3. **Image Editing and Controlled Validation of Revision.** We demonstrate improved image-editing quality and source preservation, with controlled ablations supporting bridge-derived training states and transition-matching objectives. Further evaluation on visual reasoning tasks shows that repeated token revision improves accuracy under the same 30-update sampling schedule.

2 Related Work

Image editing with discrete generative models. A central distinction in discrete image editing is whether source tokens serve as a reference sequence or as the state being revised. Models such as Lumina-DiMOO and LLaDA2.0-Uni support editing through conditional target generation (Xin et al., 2025; AI et al., 2026). Inversion-based methods instead recover an editing trajectory from the source using a pretrained generator (Figure 1b). DICE (He et al., 2024) records noise sequences and masking patterns for reconstruction and editing, while GIDE (Zhu et al., 2026) combines grounding (Ravi et al., 2025; Carion et al., 2026), discrete noise inversion (Huberman-Spiegelglas et al., 2024), and refinement. These approaches enable editing through inversion and sampling; PaRDi learns preservation and revision as source-to-target dynamics from paired images, starting each edit from the source tokens.

Token revision and remasking. The standard reverse process of absorbing-state diffusion reveals masked tokens while leaving visible values unchanged (Sahoo et al., 2024). ReMDM derives a remasking sampler for pretrained masked diffusion models, enabling repeated refinement within a diffusion formulation (Wang et al., 2025). Reflective Masking learns revision through tokenwise cross-entropy on prescribed labels and hand-designed training trajectories (Zhang et al., 2026). Its History Reference mechanism retains information from previous states to guide subsequent revisions. PaRDi instead supervises the transition probabilities or rates induced by target-token predictions, using training states and transition targets from the same endpoint-conditioned process. Its predictor reads the current canvas, instruction, and time without a separate history representation. We develop the relationship between revision-label CE and transition matching in Sections B.4 and B.5.

Diffusion bridges and transition learning. Continuous-state diffusion bridges have been studied for distribution transport (De Bortoli et al., 2021). I^2SB develops a continuous-state bridge for image restoration, connecting degraded and clean images through analytic intermediate distributions (Liu et al., 2023). In discrete state spaces, Campbell et al. (2022) extend denoising models such as D3PM (Austin et al., 2021) to continuous time, using Markov jump processes to learn noise-to-data generation. Discrete Flow Matching specifies probability paths and derives their sampling dynamics from learned endpoint posteriors (Gat et al., 2024). PaRDi constructs a discrete source-to-target bridge for instruction-guided image editing. Its MASK-hub transitions explicitly model preservation and repeated revision on the source canvas, while conditioning on paired endpoints determines both intermediate training states and transition supervision. This yields posterior- and rate-matching objectives for the same editing process.

3 PaRDi for In-Place Editing

PaRDi first defines how tokens can change, then conditions this process on paired images to obtain training canvases and transition targets. At inference, predicted target probabilities replace the observed target in the transition map (Figure 2). We develop the continuous-time formulation below; its discrete-time counterpart and posterior-matching objective are given in Sections A.1 and B, with the small-step connection established in Proposition 2.

3.1 Problem Formulation

A visual tokenizer represents the source and target images as $\mathbf{s}, \mathbf{y} \in \mathcal{V}^N$, with N spatial tokens from a visible vocabulary $\mathcal{V}$ of size V. At position i, s_i and y_i denote the paired source and target values. The current canvas $\mathbf{X}_\tau \in \mathcal{V}_{\texttt{<MASK>}}^N$, where $\mathcal{V}_{\texttt{<MASK>}} = \mathcal{V} \cup \{\texttt{<MASK>}\}$, occupies the same spatial positions throughout editing. Training uses paired endpoints and an instruction c; inference starts from $\mathbf{s}$ with only the source image and instruction available.

Time $\tau \in [0, 1]$ increases from target to source, with $\mathbf{X}_0 = \mathbf{y}$ and $\mathbf{X}_1 = \mathbf{s}$; generation runs in decreasing τ. For a single-token derivation, write s, y for the endpoints and X_τ for the current value.

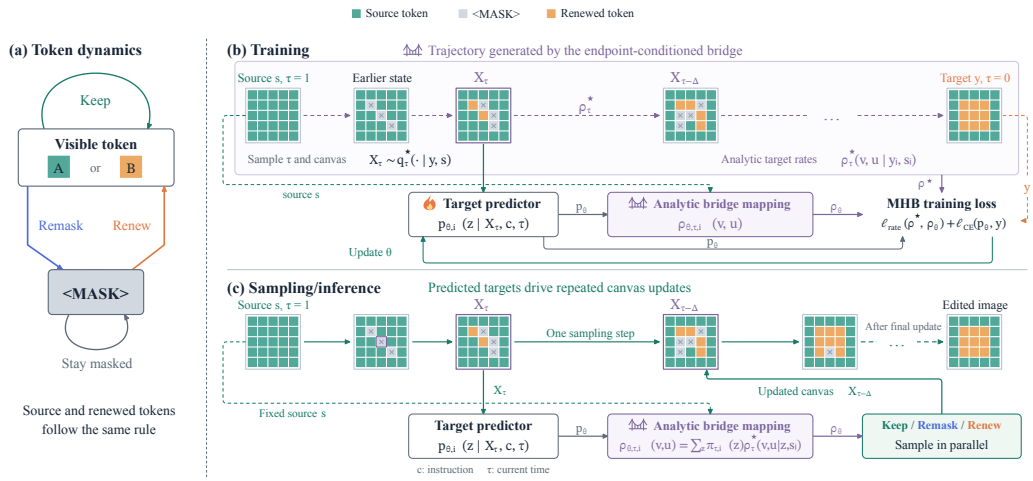

Figure 2: PaRDi dynamics, training, and sampling. (a) Visible tokens can be kept or remasked, while masked tokens can remain masked or be renewed; source and renewed tokens follow the same rule. (b) Paired endpoints provide intermediate training canvases and analytic transition targets; transition matching and Target CE train the same target-token predictor. (c) Predicted target probabilities induce transition probabilities or rates for parallel sampled updates. The source initializes the canvas and enters the analytic mapping; the predictor reads the current canvas, instruction, and time.

3.2 The MASK-Hub Markov Process

We define a MASK-hub Markov process that connects distinct visible values $A, B \in \mathcal{V}$ through MASK (Figure 2a),

$$\text{visible } A \longrightarrow \texttt{<MASK>} \longrightarrow \text{visible } B.$$

A single jump cannot replace one visible value directly with another. A visible token can retain its value or become masked, and a masked token can remain masked or reveal a value. Let $\lambda_\tau, \mu_\tau \geq 0$ be the masking and reveal rates, and let r_s be a full-support distribution on $\mathcal{V}$, fixed across time and target hypotheses for a given s. Our implementation uses $r_s(v) = 1/V$. Ordering visible states before <MASK>, the single-token MASK-hub process has the transition-rate matrix (generator)

$$R_\tau = \begin{pmatrix} -\lambda_\tau I_V & \lambda_\tau \mathbf{1}_V \\ \mu_\tau \boldsymbol{r}_s^\top & -\mu_\tau \end{pmatrix}, \qquad \tau \in [0,1]. \tag{1}$$

Here I_V is the identity, $\mathbf{1}_V$ is the all-ones column, and $\boldsymbol{r}_s = (r_s(v))_{v\in\mathcal{V}}$. For $u \neq v$, $R_\tau(u,v)$ is the instantaneous rate of the jump $u \to v$ as τ increases; diagonal entries make each row sum to zero. Renewal leaves a token subject to the same transition rule, allowing further revision. The canvas-level Markov process is the product of these single-token processes, with independent masking and renewal times across positions. A position can therefore be renewed while others retain their current values, without a canvas-wide mask-then-refill order.

3.3 Endpoint-Conditioned Bridges

Conditioning the MASK-hub Markov process on a source–target pair yields a bridge that provides intermediate training states and target-directed transitions. Writing $\Pr_R$ for probabilities under the MASK-hub process defined by Equation (1), define the forward message $F_\tau^y(v) = \Pr_R(X_\tau = v \mid X_0 = y)$ and source-side message $h_\tau^s(v) = \Pr_R(X_1 = s \mid X_\tau = v)$. They measure the probability of reaching v from the target and continuing from v to the source, respectively. Conditioning on both endpoints gives

$$q_\tau^{\mathrm{br}}(v \mid y, s) = \frac{F_\tau^y(v) h_\tau^s(v)}{Z(y,s)}, \qquad Z(y,s) = F_1^y(s) > 0. \tag{2}$$

Sampling this distribution supplies intermediate training states; $Z(y,s)$ is the probability of reaching s from y under the MASK-hub Markov process. At a state with positive bridge mass, the target-directed jump $v \to u$ has rate (Campbell et al., 2022)

$$\rho_\tau^{\mathrm{br}}(v, u \mid y, s) = R_\tau(u,v) \frac{F_\tau^y(u)}{F_\tau^y(v)}, \qquad u \neq v. \tag{3}$$

This rate specifies how the sampled state should change toward the target, providing supervision per unit decrease in τ. The source-side message shapes the training-state distribution and cancels from the local reverse rate once the current state is given.

A bridge with identical endpoints may still change a token and later restore it. To supervise continuous preservation at unchanged positions, we select a constant path when $y = s$. Writing δ_a for a point mass at state a, the resulting training marginal is

$$q_\tau^\star(v \mid y, s) = \begin{cases} \delta_s(v), & y = s, \\ q_\tau^{\mathrm{br}}(v \mid y, s), & y \neq s. \end{cases} \tag{4}$$

Its outgoing rates $\rho_\tau^\star$ are zero for $y = s$ and otherwise given by Equation (3). Sampling these bridge and identity-path marginals supplies the training canvas and its local transition targets (Figure 2b).

The hub structure makes the bridge quantities computable through a few scalar coefficients. Extending $r_s(\texttt{<MASK>}) = 0$, the forward message has the form

$$F_\tau^y(v) = A_\tau \delta_y(v) + C_\tau r_s(v) + M_\tau \delta_{\texttt{<MASK>}}(v), \qquad A_\tau + C_\tau + M_\tau = 1. \tag{5}$$

The nonnegative coefficients track mass that has not left the target, mass drawn from the visible proposal, and MASK mass. They follow three scalar differential equations, with an analogous three-coefficient representation for the source-side message. Propagating these coefficients has constant size; evaluating a visible distribution costs $O(V)$ per token without a dense $(V + 1) \times (V + 1)$ transition table. Section A gives the recurrences, differential equations, and proofs.

3.4 Learning Bridge-Induced Transitions

For each training pair $(\mathbf{s}, \mathbf{y}, c)$, we sample a time τ and a canvas from $q_\tau^\star(\mathbf{x} \mid \mathbf{y}, \mathbf{s}) = \prod_i q_\tau^\star(x_i \mid y_i, s_i)$. The model, parameterized by θ, predicts $p_{\theta,i}(z \mid \mathbf{X}_\tau, c, \tau)$ over candidate target values $z \in \mathcal{V}$. At the current value $v = X_{\tau,i}$, retain only endpoint hypotheses whose bridge or identity path can occupy v. Renormalizing the prediction over this compatible set $\mathcal{C}_\tau(v; s_i)$ gives $\pi_{\tau,i}(z)$ (Equation (25)). Each candidate endpoint supplies analytic transition rates; averaging them with these predicted probabilities gives

$$\rho_{\theta,\tau,i}(v, u) = \sum_{z \in \mathcal{C}_\tau(v; s_i)} \pi_{\tau,i}(z) \rho_\tau^\star(v, u \mid z, s_i), \qquad u \neq v. \tag{6}$$

The target predictor in Figure 2c estimates target-token probabilities from the current canvas, instruction, and time; the analytic bridge mapping converts them into revision rates without a separate action predictor. These rates govern the parallel keep, remask, and renew updates in Figure 2c, making each token's preservation and revision depend on the full visual context. At a source-valued position, greater probability assigned to the unchanged endpoint reduces the remasking rate.

We match the induced rates to those conditioned on the observed target. For nonnegative rates a, b, define $D_{\mathrm{rate}}(a, b) = a \log(a/b) - a + b$, with zero-rate limits specified in Section B. The ***PaRDi rate loss*** is

$$\ell_{\mathrm{rate},i} = \sum_{u \in \mathcal{V}_{\texttt{<MASK>}} \setminus \{v\}} D_{\mathrm{rate}}(\rho_\tau^\star(v, u \mid y_i, s_i), \rho_{\theta,\tau,i}(v, u)) \,. \tag{7}$$

At a visible state, MASK is the only outgoing destination, so the loss supervises when to withdraw the current value and how likely it is to remain unchanged over an interval. At MASK, target and model total rates agree, and the loss supervises which value to reveal. The intensity–destination decomposition in Equation (31) makes these two roles explicit. Auxiliary ***Target CE***, $\ell_{\mathrm{CE},i} = -\log p_{\theta,i}(y_i \mid \mathbf{X}_\tau, c, \tau)$, directly supervises the endpoint value. The joint objective is

$$\mathcal{L}(\theta) = \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} (w_{\mathrm{dyn}} \ell_{\mathrm{rate},i} + w_{\mathrm{CE}} \ell_{\mathrm{CE},i})\right]. \tag{8}$$

The expectation is over training pairs, times sampled uniformly from $[\varepsilon_{\mathrm{tr}}, 1 - \varepsilon_{\mathrm{tr}})$ with an endpoint cutoff $0 < \varepsilon_{\mathrm{tr}} < 1/2$, and canvases drawn from $q_\tau^\star$. We use $w_{\mathrm{dyn}} = w_{\mathrm{CE}} = 1$ and average over all N tokens, including unchanged positions. Together, transition matching and Target CE train the predictor to assign probabilities to target values that both match the desired content and induce appropriate preservation and revision dynamics.

Table 1: General-purpose image editing. Methods are grouped into reference-based regeneration and in-place editing, with model families shown within each group. PIE keep is the fraction of non-edited tokens that remain unchanged; PIE leak is the fraction of changed tokens that lie in the non-edited.

	Method	GEdit-EN ↑	ImgEdit ↑	PIE keep (%) ↑	PIE leak (%) ↓
Ref.-based Regen.	▼ *Autoregressive*				
	UniAR (Peng et al., 2026)	5.31	3.73	0.00	<u>31.62</u>
	▼ *Masked Generative Models*				
	EditMGT (Chow et al., 2026)	5.87	2.94	33.60	44.60
	▼ *Discrete Diffusion Models*				
	Lumina-DiMOO (Xin et al., 2025)	3.91	2.77	87.39	<u>31.62</u>
	▼ *Block Discrete Diffusion Models*				
	LLaDA2.0-Uni (AI et al., 2026)	<u>6.61</u>	<u>3.92</u>	59.72	45.97
In-place Edit	▼ *Training-Free In-Place Token Editing*				
	DICE (He et al., 2024)	2.89	2.22	<u>88.87</u>	42.31
	GIDE (Zhu et al., 2026)	4.49	2.89	69.40	36.12
	▼ *Discrete Diffusion Bridges*				
	PaRDi	**6.98**	**3.95**	**92.27**	**14.19**

3.5 Inference on the Source Canvas

Starting from $\mathbf{X}_1 = \mathbf{s}$, each step applies classifier-free guidance (CFG) (Schiff et al., 2025) and compatibility normalization to target predictions, then forms rates $b_i(u) = \rho_{\theta,\tau,i}(X_i, u)$ via Equation (6). For a time decrement Δ and total rate $\Lambda_i = \sum_{u \neq X_i} b_i(u)$, token i stays unchanged with probability $e^{-\Lambda_i \Delta}$ and otherwise draws a destination from b_i/Λ_i; zero-rate tokens remain unchanged. This frozen-rate approximation uses the same canvas for all rates before parallel updates, permitting at most one jump per token per step and further revision in later steps. We use 30 uniform updates to $\varepsilon = 10^{-6}$, followed by conditional MASK completion (Section C.5). Algorithm 4 in Section E gives the complete numerical sampler.

4 Experiments

We evaluate editing quality and preservation, isolate the contributions of transition structure and training, and test repeated revision on visual reasoning tasks. All baseline methods use their officially released weights and the inference settings recommended by their authors.

4.1 General-Purpose Image Editing

Setup. We evaluate instruction following on the English split of GEdit-Bench (Liu et al., 2025) and the Basic split of ImgEdit-Bench (Ye et al., 2026), and use PIE-Bench (Brooks et al., 2023) to measure changes outside the annotated edit region. PaRDi adapts LLaDA2.0-Uni on the image-editing datasets listed in Section C.3. Preprocessing, endpoint construction, and inference settings appear in Section C.

Results. PaRDi combines effective instruction following with stronger preservation of unrelated source content (Table 1). Compared with its LLaDA2.0-Uni backbone, GEdit-EN improves from 6.61 to 6.98 while PIE background token retention rises from 59.72% to 92.27%. The comparisons in Figure 3 make this joint improvement visible across changes to text, appearance, objects, and backgrounds. PaRDi carries out the requested modifications while better retaining scene composition and fine details outside the intended change, consistent with its lower token leakage. In the final row, the enlarged foreground details show PaRDi preserving the knitted pattern on the cup sleeve while replacing the background. The PIE-Target CLIP (Radford et al., 2021) results in Section D.2 further show that this preservation is accompanied by strong semantic alignment within the edited region. Category-level results and additional qualitative comparisons appear in Sections D.1 and D.3.

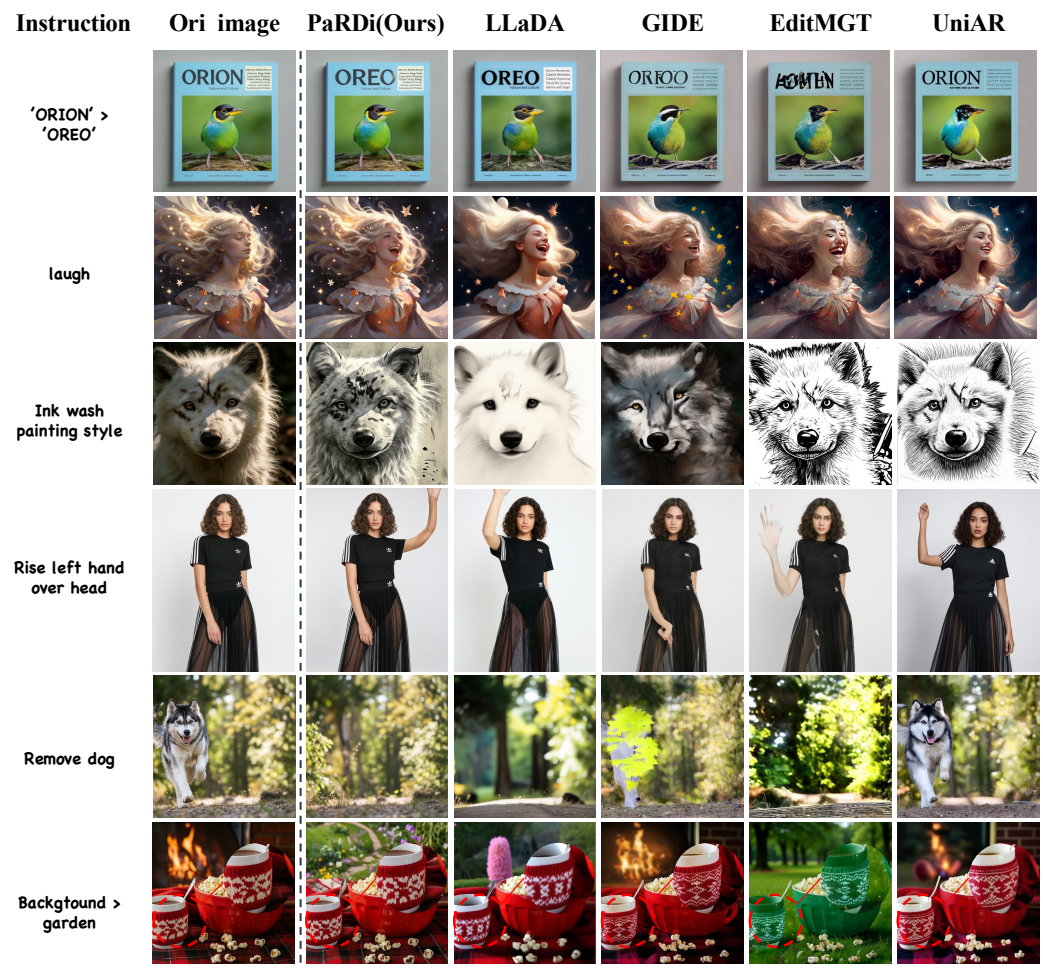

Figure 3: Qualitative image-editing comparisons across text, expression, style, pose, object removal, and background edits. Each row shows an instruction, the source image, and outputs from PaRDi and four baseline methods. The enlarged details in the final row highlight the foreground cup sleeve's pattern, which should remain unchanged during background replacement.

4.2 In-Place Editing and Transition Structure

Table 2 compares the canvas interfaces in Figure 1 and the transition topology in Figure 2a. Absorbing-state diffusion starts from a separate fully masked canvas; the uniform-transition bridge starts from the source and allows direct visible-value replacement; PaRDi starts from the source and routes replacement through MASK. We present their Markov transition matrices in Section A.2. We match training data, trainable parameter count, sample exposure, and evaluation settings across variants.

The source-to-target bridges improve preservation and reduce latency relative to separate-target absorbing generation (Table 2). Among these bridges, the MASK hub provides further gains with source initialization and endpoint conditioning fixed, raising ImgEdit from 3.63 to 3.95 and reducing leakage from 18.51% to 14.19%. This joint improvement shows that the MASK hub supports more selective editing, improving the requested changes while better retaining unrelated content. PaRDi also runs in approximately half the time of the absorbing baseline.

4.3 Bridge-Derived States and Transition Matching

We separately test the intermediate training canvases and transition supervision supplied by the bridge (Figure 2b). The revised hand-designed control (Zhang et al., 2026) matches PaRDi's full-codebook uniform proposal and its mixture weight at every training time, while retaining a manually specified source–MASK–target schedule (Equation (49)). For each state distribution, Table 3 compares Target

Table 2: Effects of canvas initialization and transition topology. Sampling arrows show allowed transitions between visible values $A, B, C, \ldots$ and MASK. Relative time is normalized to PaRDi.

Process	Initial canvas	Sampling transitions	GEdit-EN ↑	ImgEdit ↑	Keep (%) ↑	Leak (%) ↓	Time (%) ↓
Absorbing-state diffusion	Full <MASK>	<MASK> $\rightarrow \{A, B, C, \ldots\}$	6.61	3.52	84.52	31.84	193.60
Uniform-transition bridge	Source	$A \rightarrow \{A, B, C, \ldots\}$	<u>6.78</u>	<u>3.63</u>	<u>91.38</u>	<u>18.51</u>	<u>126.00</u>
PaRDi bridge (MASK-hub)	Source	$\{A, B, C, \ldots\} \leftrightarrow$ <MASK>	**6.98**	**3.95**	**92.27**	**14.19**	**100.00**

Table 3: Training-state and objective ablation. The revised control matches proposal weights over time and shares the target-token head and PaRDi sampler. Target CE and rate divergence use shared held-out PaRDi bridge states, times, and token averaging. Rate divergence diagnoses how closely model-induced rates match endpoint-conditioned bridge rates.

Training states	Objective	GEdit-EN ↑	ImgEdit ↑	Keep (%) ↑	Leak (%) ↓	Target CE ↓	Rate divergence ↓
Hand-designed edit states	Target CE	1.88	1.77	82.48	22.87	1.36	1.4724
Hand-designed edit states	Target CE + Keep/remask CE	2.21	1.88	85.28	20.46	1.34	1.4653
Hand-designed edit states	Target CE + PaRDi rate loss	2.57	1.93	85.45	21.49	1.27	1.4273
PaRDi bridge states	Target CE	6.33	3.44	87.43	<u>17.85</u>	0.96	<u>1.1446</u>
PaRDi bridge states	Target CE + Keep/remask CE	<u>6.74</u>	<u>3.79</u>	<u>89.65</u>	18.75	<u>0.95</u>	1.1451
PaRDi bridge states	Target CE + PaRDi rate loss	**6.98**	**3.95**	**92.27**	**14.19**	**0.83**	**1.1364**

CE alone and its combinations with Keep/remask CE or the PaRDi rate loss, using the same target-token head and sampler. Keep/remask CE tests classification supervision within this shared interface; Sections B.4 and B.5 relates it to the revision-label CE of Reflective Masking (Zhang et al., 2026). State construction and diagnostic settings appear in Section C.2.

Matching the proposal distribution and its time-dependent weight controls both the candidate values and their probability mass. The updated state comparison therefore tests how the remaining mass is allocated among source values, MASK, and target values, under the same Target CE and PaRDi sampler. Repeating this comparison with each augmented objective tests whether the effect of state construction persists across supervision choices.

Transition matching further improves how the model revises these canvases. On bridge states, adding Keep/remask CE improves editing scores but increases leakage from 17.85% to 18.75%; adding the PaRDi rate loss improves editing while reducing leakage to 14.19%. Among the bridge-state variants, the full objective also gives the lowest held-out Target CE and rate divergence, reflecting both improved target prediction and a closer fit to the bridge dynamics. On these states, transition matching improves editing quality while preserving structure.

4.4 Visual Reasoning

DiffThinker frames visual reasoning as image-to-image generation (He et al., 2025), while EndoCoT iteratively refines latent thought states to guide diffusion denoising (Dai et al., 2026). Here, editing requires preserving the given problem while constructing a solution that satisfies task constraints.

Setup. The model receives a rendered problem and solves the same canvas. We evaluate Maze pathfinding, Euclidean TSP, Sudoku, and VSP using deterministic task checks. Instances are disjoint across splits and stratified by difficulty values in Table 4. The DiffThinker baseline (He et al., 2025) uses its official checkpoint and recommended inference settings.

Results. PaRDi raises average accuracy from 75.36% to 87.29% (Table 4). Largest gains occur on Sudoku, where grid decisions must satisfy joint constraints; accuracy improves by 21–54 points across three settings. Every Maze setting improves, while TSP gains are modest and VSP reaches full accuracy from a strong baseline. These results extend in-place revision from appearance changes to constructing solutions governed by canvas-wide relationships.

Inference-time revision budget. We limit the repeated revisions permitted by the sampling loop in Figure 2c by setting a per-token cap K on <MASK>-to-visible renewals. $K = 1$ locks a token after its first reveal, whereas $K = \infty$ is the unrestricted sampler used for the main results (30 steps; at most 15 renewals per token). All settings share the checkpoint, 30-update time grid, guidance, and terminal completion rule; only the per-token revision cap changes.

Table 4: Visual reasoning accuracy (%). Avg. is the unweighted mean over the 14 difficulty settings.

Model	Maze ↑				TSP ↑			Sudoku ↑			VSP ↑				Avg. ↑
	4	5	6	7	9	12	15	30	35	40	4	8	12	16	
DiffThinker	96	97	81	87	**81**	64	49	36	24	44	**100**	**100**	99	97	75.36
PaRDi (Ours)	**99**	**100**	**100**	**98**	**81**	**71**	**54**	**57**	**78**	**84**	**100**	**100**	**100**	**100**	**87.29**

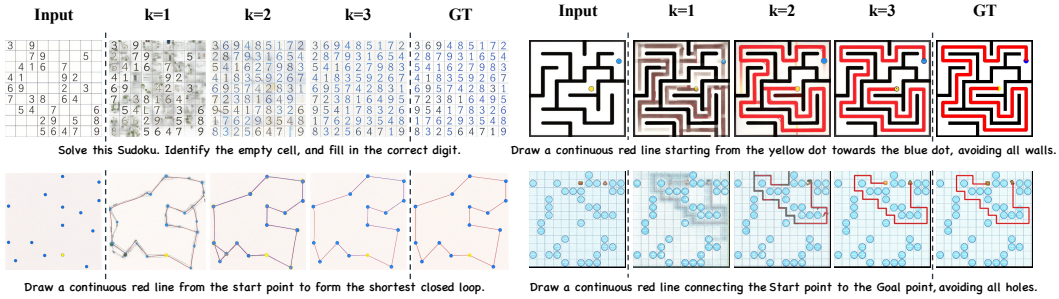

Figure 4: Visual-reasoning examples under per-token revision caps $K = 1, 2, 3$. Sudoku, Maze, TSP, and VSP are shown from top left to bottom right, each with the input, PaRDi outputs, and a ground-truth solution (GT). Additional examples with $K = 1, 2, 3, 4$ appear in Section D.4.

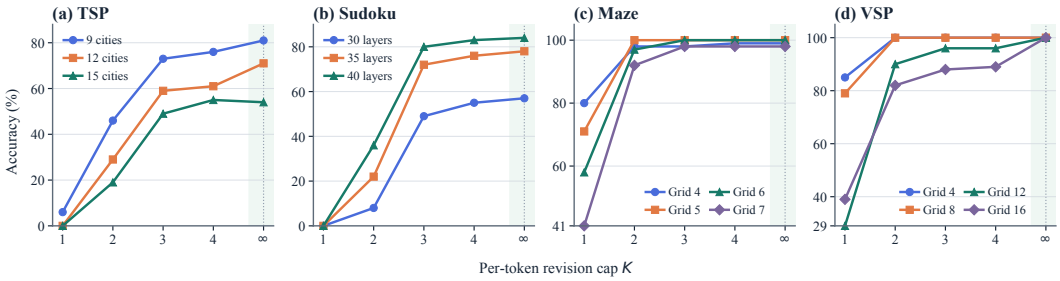

Figure 5: Effect of the per-token revision cap K on visual reasoning. All caps use the same 30-update time grid and terminal completion rule. Each line represents one difficulty setting, and the shaded $K = \infty$ column reproduces the unrestricted results from Table 4.

Allowing revealed tokens to change improves final accuracy under this shared schedule (Figure 5). Increasing K from 1 to 3 raises TSP accuracy from 2.00% to 60.33% and Sudoku accuracy from 0.00% to 67.00%. Maze and smaller VSP settings approach saturation early, while TSP, Sudoku, and larger VSP settings continue benefiting from additional revision. Outputs in Figure 4 and Section D.4 connect these gains to solution structure, with more complete paths and more consistent digit assignments at higher caps. Together, these results show local transition supervision supports effective repeated revision without explicit multi-round solution trajectories. The GEdit-EN sweep in Figure 6 shows a similar trend, rising from 3.93 at $K = 1$ to 6.98 without a cap.

5 Conclusion

PaRDi makes preservation and revision part of the diffusion dynamics on the source canvas. Endpoint-conditioned bridges and constant paths at unchanged positions jointly provide training canvases and transition supervision. Editing experiments demonstrate gains in quality and preservation, while visual reasoning shows the value of repeated revision under the same integration schedule.

PaRDi depends on the quality of the discrete visual tokenizer and on paired source–target supervision. Preservation at inference remains learned, and continuous-time sampling numerically approximates the analytic rates.

Reproducibility Statement

The appendices provide bridge derivations, learning objectives and their estimators, and training and sampling algorithms. Section C specifies the training populations, endpoint preprocessing, ablation controls, and inference and scoring settings.

Ethics Statement

Image editing can be used to manipulate depictions of people and events. Use of image-editing systems requires attention to consent, privacy, and the provenance of edited content. The reported evaluations measure editing quality and preservation, not misuse resistance.

AI use statement

We used AI tools for translation, language polishing, and grammar correction. The authors reviewed and take responsibility for all content.

References

Inclusion AI, Tiwei Bie, Haoxing Chen, Tieyuan Chen, Zhenglin Cheng, Long Cui, Kai Gan, Zhicheng Huang, Zhenzhong Lan, Haoquan Li, et al. Llada2. 0-uni: Unifying multimodal understanding and generation with diffusion large language model. *arXiv preprint arXiv:2604.20796*, 2026.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P Breckon, and Chris G Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. In *European Conference on Computer Vision*, pp. 170–188. Springer, 2022.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402. IEEE, 2023.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11305–11315. IEEE, 2022.

Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, José Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, 2023.

Wei Chow, Linfeng Li, Lingdong Kong, Zefeng Li, Qi Xu, Hang Song, Tian Ye, Xian Wang, Jinbin Bai, Shilin Xu, et al. Editmgt: unleashing potentials of masked generative transformers in image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38038–38048, 2026.

Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, et al. Emu3. 5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

Xuanlang Dai, Yujie Zhou, Long Xing, Jiazi Bu, Xilin Wei, Yuhong Liu, Beichen Zhang, Kai Chen, and Yuhang Zang. Endocot: Scaling endogenous chain-of-thought reasoning in diffusion models supplementary material. In *Computer Vision - ECCV 2026 - 19th European Conference, Malmö, Sweden, September 8-12, 2026, Proceedings, Part IX*, volume 17009 of *Lecture Notes in Computer Science*, pp. 621–638. Springer, 2026.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in neural information processing systems*, 34:17695–17709, 2021.

Hongyi Fang, Chuwen Xie, Benjia Zhou, Yu-Xuan Qiu, Chenggong Hu, Zhibin Wang, Chao Chen, Jianbin Qin, and Rui Mao. Model the edit, not the image: Visual autoregressive editing from a source-centric perspective. *arXiv preprint arXiv:2608.09057*, 2026.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10686–10696. IEEE, 2022.

Xiaoxiao He, Quan Dao, Ligong Han, Song Wen, Minhao Bai, Di Liu, Han Zhang, Martin Renqiang Min, Felix Juefei-Xu, Chaowei Tan, et al. Dice: Discrete inversion enabling controllable editing for multinomial diffusion and masked generative models. *arXiv preprint arXiv:2410.08207*, 2024.

Zefeng He, Xiaoye Qu, Yafu Li, Tong Zhu, Siyuan Huang, and Yu Cheng. Diffthinker: Towards generative multimodal reasoning with diffusion models. *arXiv preprint arXiv:2512.24165*, 2025.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021.

Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12469–12478. IEEE, 2024.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19721–19730. IEEE, 2025.

Shufan Li, Greg Heinrich, Hanrong Ye, Yonggan Fu, Aditya Grover, Jan Kautz, and Pavlo Molchanov. Nemotron-labs-diffusion-image: Advancing masked discrete diffusion for high-resolution image synthesis. *arXiv preprint arXiv:2606.29814*, 2026.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. I^2sb: Image-to-image schrödinger bridge. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22042–22062. PMLR, 2023.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 32819–32848, 21–27 Jul 2024.

Wujian Peng, Lingchen Meng, Yuxuan Cai, Xianwei Zhuang, Yuhuan Yang, Rongyao Fang, Chenfei Wu, Junyang Lin, Zuxuan Wu, and Shuai Bai. Unified multimodal autoregressive modeling with shared context—visual tokenizer is key to unification. In *Forty-third International Conference on Machine Learning*, 2026.

Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-banana-400k: A large-scale dataset for text-guided image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37226–37235, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

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

Yair Schiff, Subham Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-Torre, Bernardo Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 43776–43821, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Qingyu Shi, Jinbin Bai, Zhuoran Zhao, Wenhao Chai, Kaidong Yu, Jianzong Wu, Shuangyong Song, Yunhai Tong, Xiangtai Li, Xuelong Li, et al. Beyond text-to-image: Liberating generation with a unified discrete diffusion model. In *The Fourteenth International Conference on Learning Representations*, 2026.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.

Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Keqi Wang, Yibin Wang, et al. Lumina-dimoo: An omni diffusion large language model for multi-modal generation and understanding. *arXiv preprint arXiv:2510.06308*, 2025.

Peilin Xiong, Honghui Yuan, Junwen Chen, and Keiji Yanai. Bridge: Background routing and isolated discrete gating for coarse-mask local editing. *arXiv preprint arXiv:2605.07846*, 2026.

Yang Ye, Xianyi He, Zongjian Li, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, Li Yuan, et al. Imgedit: A unified image editing dataset and benchmark. *Advances in Neural Information Processing Systems*, 38, 2026.

Yanming Zhang, Yihan Bian, Jingyuan Qi, Yuguang Yao, Lifu Huang, and Tianyi Zhou. Multi-turn reflective masking elicits reasoning in mask diffusion models. *arXiv preprint arXiv:2606.16700*, 2026.

Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models. *Advances in Neural Information Processing Systems*, 38: 125510–125538, 2026.

Zifeng Zhu, Jiaming Han, Jiaxiang Zhao, Minnan Luo, and Xiangyu Yue. Gide: Unlocking diffusion llms for precise training-free image editing. In *European Conference on Computer Vision*, pp. 464–482. Springer, 2026.

A MASK-HUB BRIDGE DERIVATIONS

We expand the Markov transition matrices, derive compact messages for endpoint conditioning, and obtain the intermediate-state distributions and reverse transitions used in Section 3. The final subsection specifies how unchanged endpoints and compatible target predictions enter the sampling map.

A.1 DISCRETE-TIME MASK-HUB MARKOV PROCESS

On a discrete clock $t \in \{0, \dots, T\}$, place the target at $X_0 = y$ and source at $X_T = s$. Let $Q_t(u, v)$ be the transition probability of moving from u at step $t-1$ to v at step t. With masking and reveal probabilities $\beta_t, \gamma_t \in [0, 1]$, the MASK-hub kernel is

$$Q_t = \begin{pmatrix} (1-\beta_t) I_V & \beta_t \mathbf{1}_V \\ \gamma_t \boldsymbol{r}_s^\top & 1-\gamma_t \end{pmatrix}. \tag{9}$$

The upper block implements keep or remask; the lower block implements renew or remain masked. Each row sums to one, and generation traverses the endpoint-conditioned process in decreasing t.

A.2 COMPARISON OF DISCRETE TRANSITION MATRICES

We first contrast the one-step transition kernels of uniform diffusion, absorbing-state diffusion, and PaRDi. For this example, take $\mathcal{V} = \{A, B, C, D\}$, so $V = 4$, and add the special state <MASK> where required. All matrices are row-stochastic, with rows indexing the current state and columns the next state in increasing t. In contrast, the arrows in Table 2 describe sampling in the reverse direction. Uniform diffusion uses only visible states; the other two kernels add <MASK>.

Uniform diffusion. Uniform diffusion (Austin et al., 2021) uses only the four visible states and resamples uniformly with mass β_t,

$$Q_t^{\text{unif}} = (1-\beta_t) I_4 + \frac{\beta_t}{4} \mathbf{1}\mathbf{1}^\top = \begin{bmatrix} 1-\frac{3}{4}\beta_t & \frac{1}{4}\beta_t & \frac{1}{4}\beta_t & \frac{1}{4}\beta_t \\ \frac{1}{4}\beta_t & 1-\frac{3}{4}\beta_t & \frac{1}{4}\beta_t & \frac{1}{4}\beta_t \\ \frac{1}{4}\beta_t & \frac{1}{4}\beta_t & 1-\frac{3}{4}\beta_t & \frac{1}{4}\beta_t \\ \frac{1}{4}\beta_t & \frac{1}{4}\beta_t & \frac{1}{4}\beta_t & 1-\frac{3}{4}\beta_t \end{bmatrix}. \tag{10}$$

Every visible value can change directly into every other visible value in one step; for example, $A \to A, B, C$, or D.

Absorbing-state diffusion. Adding <MASK> gives the ordered state space $(A, B, C, D, \texttt{<MASK>})$. An absorbing kernel moves a visible value to <MASK> with probability β_t and keeps it there in subsequent steps,

$$Q_t^{\text{abs}} = \begin{bmatrix} 1-\beta_t & 0 & 0 & 0 & \beta_t \\ 0 & 1-\beta_t & 0 & 0 & \beta_t \\ 0 & 0 & 1-\beta_t & 0 & \beta_t \\ 0 & 0 & 0 & 1-\beta_t & \beta_t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{11}$$

Thus $A \to A$ or <MASK>, whereas <MASK> $\to$ <MASK> only. In the reverse sampling direction, this kernel permits reveals from MASK and keeps revealed tokens fixed.

MASK-Hub diffusion. PaRDi keeps the same visible-to-mask transitions but makes <MASK> non-absorbing. A masked token is revealed from the proposal r_s with probability γ_t,

$$Q_t^{\text{PaRDi}} = \begin{bmatrix} 1-\beta_t & 0 & 0 & 0 & \beta_t \\ 0 & 1-\beta_t & 0 & 0 & \beta_t \\ 0 & 0 & 1-\beta_t & 0 & \beta_t \\ 0 & 0 & 0 & 1-\beta_t & \beta_t \\ \gamma_t r_s(A) & \gamma_t r_s(B) & \gamma_t r_s(C) & \gamma_t r_s(D) & 1-\gamma_t \end{bmatrix}, \qquad \sum_{v \in \mathcal{V}} r_s(v) = 1. \tag{12}$$

A visible token can stay or return to <MASK>, while <MASK> can stay masked or reveal any visible value supported by r_s. Consequently, changing A into B follows $A \to$ <MASK> $\to B$ rather than a direct $A \to B$ transition.

Table 5: One-step topology of the three transition kernels. Here $j, k \in \mathcal{V}$.

Kernel	From a visible token j	From <MASK>
Uniform	$j \to k$ for every $k \in \mathcal{V}$	not present
Absorbing-state	$j \to j$ or <MASK>	<MASK> $\to$ <MASK>
PaRDi	$j \to j$ or <MASK>	<MASK> $\to$ <MASK> or $k \sim r_s$

The parameter β_t follows the conventional interpretation of each family. In the uniform kernel, the total probability of changing to a different visible value is $3\beta_t/4$ because uniform resampling can return the current value. In the absorbing and PaRDi kernels, β_t is directly the probability of moving from a visible value to <MASK>.

A.3 Compact discrete-time messages

Fix a source token s. The derivation allows the proposal r_s to depend on s, provided it stays fixed across time and target hypotheses; the uniform implementation is a special case. The discrete forward and source-side messages are $F_t^y(v) = [Q_1 \cdots Q_t]_{y,v}$ and $h_t^s(v) = [Q_{t+1} \cdots Q_T]_{v,s}$, with empty products interpreted as identities. Extend the proposal by $r_s(\texttt{<MASK>}) = 0$. For any state a, let $\delta_a(v) = \mathbf{1}\{v = a\}$, where the indicator is one when its condition holds and zero otherwise. The forward message decomposes into a component that has not left the target, a visible proposal component, and MASK mass.

Proposition 1 (Hub-structured messages)**.** *For the kernel in Equation* (9)*,*

$$F_t^y(v) = A_t \delta_y(v) + C_t r_s(v) + M_t \delta_{\texttt{<MASK>}}(v), \qquad A_t + C_t + M_t = 1, \tag{13}$$

with nonnegative component masses satisfying

$$\begin{aligned} A_t &= (1-\beta_t) A_{t-1}, \\ C_t &= (1-\beta_t) C_{t-1} + \gamma_t M_{t-1}, \\ M_t &= \beta_t (A_{t-1} + C_{t-1}) + (1-\gamma_t) M_{t-1}. \end{aligned} \tag{14}$$

Initially $(A_0, C_0, M_0) = (1, 0, 0)$*. The backward message uses three additional scalar coefficients, with* $h_t^s(v) = D_t \mathbf{1}\{v = s\} + E_t r_s(s)$ *for visible* v *and* $h_t^s(\texttt{<MASK>}) = H_t r_s(s)$*.*

Proof. The forward representation holds at $t = 0$ with $(A_0, C_0, M_0) = (1, 0, 0)$. For the induction step, a visible point mass retains fraction $1 - \beta_t$ and sends fraction β_t to MASK. The proposal component obeys the same rule, while MASK sends fraction γ_t to r_s. Substituting these three updates into $F_t^y(v) = \sum_{u \in \mathcal{V}_{\texttt{<MASK>}}} F_{t-1}^y(u) Q_t(u, v)$ gives Equation (14). Nonnegativity and total mass one follow by induction.

The terminal backward coefficients are $(D_T, E_T, H_T) = (1, 0, 0)$. Substituting into $h_t^s(v) = \sum_{u \in \mathcal{V}_{\texttt{<MASK>}}} Q_{t+1}(v, u) h_{t+1}^s(u)$ yields

$$\begin{aligned} D_t &= (1-\beta_{t+1}) D_{t+1}, \\ E_t &= (1-\beta_{t+1}) E_{t+1} + \beta_{t+1} H_{t+1}, \\ H_t &= \gamma_{t+1}(D_{t+1} + E_{t+1}) + (1-\gamma_{t+1}) H_{t+1}. \end{aligned} \tag{15}$$

These updates preserve the stated backward representation from its terminal values, completing the proof. □

For different endpoints, the bridge mass can be written as a mixture of $\delta_y, \delta_s, r_s, \delta_{\texttt{<MASK>}}$. The proposal component can return either endpoint, so contributions are summed wherever they assign mass to the same token.

A.4 Target-directed discrete transitions

The discrete counterpart of Equation (2) is

$$q_t^{\mathrm{br}}(v \mid y, s) = \frac{F_t^y(v) h_t^s(v)}{Z(y, s)}, \qquad Z(y, s) = F_T^y(s) > 0. \tag{16}$$

The normalizer is evaluated at the terminal time of the corresponding clock. For a pair $(X_{t-1}, X_t) = (u, v)$, the unnormalized bridge mass is $F^y_{t-1}(u)Q_t(u, v)h^s_t(v)$. Dividing by the marginal mass of v cancels $h^s_t(v)$ and yields

$$q^{\mathrm{br}}_{t-1|t}(u \mid v, y, s) = \frac{F^y_{t-1}(u)Q_t(u, v)}{F^y_t(v)}. \tag{17}$$

This cancellation requires positive conditioning mass. The identity convention in Equation (4) applies on the discrete clock as well, giving $q^\star_t$ and the reverse kernel $q^\star_{t-1|t}$ specified in Section A.7.

A.5 Continuous messages and reverse rates

The decomposition in Equation (13) also holds at continuous time. With a dot denoting differentiation with respect to τ, its coefficients obey

$$\dot{A}_\tau = -\lambda_\tau A_\tau, \qquad \dot{C}_\tau = -\lambda_\tau C_\tau + \mu_\tau M_\tau, \qquad \dot{M}_\tau = \lambda_\tau(A_\tau + C_\tau) - \mu_\tau M_\tau. \tag{18}$$

The initial values are $(A_0, C_0, M_0) = (1, 0, 0)$. These equations follow from the Kolmogorov forward equation $\partial_\tau F^y_\tau(v) = \sum_{u \in \mathcal{V}_{<\texttt{MASK}>}} F^y_\tau(u)R_\tau(u, v)$. The source-side coefficients solve

$$\dot{D}_\tau = \lambda_\tau D_\tau, \qquad \dot{E}_\tau = \lambda_\tau(E_\tau - H_\tau), \qquad \dot{H}_\tau = \mu_\tau(H_\tau - D_\tau - E_\tau), \tag{19}$$

with terminal $(D_1, E_1, H_1) = (1, 0, 0)$ and integration toward decreasing τ. For a positive constant rate $\lambda_\tau = \mu_\tau = \omega$,

$$A_\tau = e^{-\omega\tau}, \quad C_\tau = \frac{(1 - e^{-\omega\tau})^2}{2}, \quad M_\tau = \frac{1 - e^{-2\omega\tau}}{2}.$$

To obtain the reverse rates, let $\mathsf{P}_{\sigma,\tau}(u, v)$ be the transition probability from time σ to τ under the MASK-hub Markov process. Under the bridge conditioned on (y, s), Bayes' rule at a state with positive bridge mass gives

$$\Pr_{\mathrm{br}}(X_{\tau-\Delta} = u \mid X_\tau = v) = \frac{F^y_{\tau-\Delta}(u)\mathsf{P}_{\tau-\Delta,\tau}(u, v)}{F^y_\tau(v)}. \tag{20}$$

The joint and marginal masses both contain $h^s_\tau(v)$, which cancels as in the discrete posterior. For $u \neq v$, at a continuity point of the masking and reveal rates, $\mathsf{P}_{\tau-\Delta,\tau}(u, v) = \Delta R_\tau(u, v) + o(\Delta)$. Dividing by the interval length and taking its limit therefore yields

$$\rho^{\mathrm{br}}_\tau(v, u \mid y, s) = \lim_{\Delta \downarrow 0} \frac{\Pr_{\mathrm{br}}(X_{\tau-\Delta} = u \mid X_\tau = v)}{\Delta} = R_\tau(u, v)\frac{F^y_\tau(u)}{F^y_\tau(v)}. \tag{21}$$

Substituting the hub generator gives the only nonzero off-diagonal rates,

$$\begin{aligned}
\rho^{\mathrm{br}}_\tau(v, \texttt{<MASK>} \mid y, s) &= \frac{\mu_\tau r_s(v) M_\tau}{A_\tau \mathbf{1}\{v = y\} + C_\tau r_s(v)}, \quad v \in \mathcal{V}, \\
\rho^{\mathrm{br}}_\tau(\texttt{<MASK>}, u \mid y, s) &= \frac{\lambda_\tau}{M_\tau}\big(A_\tau \mathbf{1}\{u = y\} + C_\tau r_s(u)\big), \quad u \in \mathcal{V}.
\end{aligned} \tag{22}$$

For visible v at an interior time with positive coefficients, equivalently write

$$\rho^{\mathrm{br}}_\tau(v, \texttt{<MASK>} \mid y, s) = a_\tau\big(1 - \kappa_{\tau,v}\mathbf{1}\{y = v\}\big), \quad a_\tau = \frac{\mu_\tau M_\tau}{C_\tau}, \quad \kappa_{\tau,v} = \frac{A_\tau}{A_\tau + C_\tau r_s(v)}.$$

Here a_τ is a baseline remasking rate and $\kappa_{\tau,v} \in (0, 1)$ is the fractional reduction when the current value equals the target. The identity convention below is applied before averaging rates over endpoint beliefs in Equation (6).

A.6 Discrete-to-continuous limit

Proposition 2 (Relationship between the two clocks)**.** *For fixed r_s, bounded Lipschitz rate schedules, and a partition $0 = \tau_0 < \cdots < \tau_T = 1$, let $\Delta\tau_t = \tau_t - \tau_{t-1}$ and $\Delta_{\max} = \max_t \Delta\tau_t$. Set $\beta_t = \lambda_{\tau_{t-1}}\Delta\tau_t$ and $\gamma_t = \mu_{\tau_{t-1}}\Delta\tau_t$. For sufficiently small $\Delta_{\max}$, the discrete transition products and endpoint-conditioned marginals converge to their continuous-time counterparts wherever the limiting conditioning mass is positive. Scaling an interior discrete reverse probability by $1/\Delta\tau_t$ then recovers the corresponding reverse rate.*

Proof. For the stated scaling, $Q_t = I + \Delta\tau_t R_{\tau_{t-1}}$ exactly. Bounded Lipschitz generators on a finite state space have local transition error $O(\Delta\tau_t^2)$ relative to this Euler factor. A telescoping expansion of the transition products, with bounded operator norms, accumulates error of order $\sum_t (\Delta\tau_t)^2 \le \Delta_{\max}$ over the unit interval. Forward and backward messages therefore converge.

Dividing their products by a limiting strictly positive endpoint mass preserves convergence of bridge marginals. For a fixed interior state of positive limiting mass, the relevant forward-message denominator is bounded away from zero on sufficiently fine partitions. For an interior off-diagonal reverse transition, $Q_t(u,v)/\Delta\tau_t = R_{\tau_{t-1}}(u,v)$; forward-message ratios converge wherever the denominator remains positive. The discrete posterior in Equation (17), divided by $\Delta\tau_t$, consequently tends to Equation (3). □

A.7 Unchanged endpoints and compatible predictions

A stochastic bridge with $y = s$ can leave the source and return; the selected identity path in Equation (4) fixes $X_t = s$ at every time. On states with positive selected marginal, the corresponding dynamics are

$$q^\star_{t-1|t}(u \mid v, y, s) = \begin{cases} \delta_s(u), & y = s \quad (v = s), \\ q^{\mathrm{br}}_{t-1|t}(u \mid v, y, s), & y \neq s, \end{cases}$$
$$\rho^\star_\tau(v, u \mid y, s) = \begin{cases} 0, & y = s \quad (v = s), \\ \rho^{\mathrm{br}}_\tau(v, u \mid y, s), & y \neq s, \end{cases} \qquad (u \neq v). \tag{23}$$

The first line is a complete discrete probability kernel; the second defines the off-diagonal jump rates.

Only different-endpoint pairs with $Z(z,s) > 0$ define stochastic bridges. Let $\mathcal{V}_s^{\mathrm{adm}} = \{s\} \cup \{z \in \mathcal{V} \setminus \{s\} : Z(z,s) > 0\}$ be the admissible endpoint set. The compatible set at a current value v is

$$\mathcal{C}_t(v; s) = \{z \in \mathcal{V}_s^{\mathrm{adm}} : q^\star_t(v \mid z, s) > 0\}. \tag{24}$$

Thus both forward reachability and continuation to the source are included. For a non-source endpoint z, compatibility requires $F_t^z(v) h_t^s(v) > 0$. When $h_t^s(v) > 0$ and $Z(z,s) > 0$ are already known, the check reduces to $F_t^z(v) > 0$. Continuous time uses the same definition with t replaced by τ and normalizer $F_1^z(s)$.

For a full canvas, the compatible prediction is

$$\pi_{\theta,t,i}(z \mid \mathbf{X}_t, \mathbf{s}, c) = \frac{p_{\theta,i}(z \mid \mathbf{X}_t, c, t) \mathbf{1}\{z \in \mathcal{C}_t(X_{t,i}; s_i)\}}{\sum_{z' \in \mathcal{C}_t(X_{t,i}; s_i)} p_{\theta,i}(z' \mid \mathbf{X}_t, c, t)}. \tag{25}$$

We abbreviate this belief as $\pi_{t,i}(z)$, or $\pi_{\tau,i}(z)$ in continuous time. This is defined for a nonempty compatible set and a positive denominator. A full-support predictor has a positive denominator on any state reachable under the selected process. In discrete time, mixtures of its valid reverse kernels preserve this support when initialized at the source. Zero-weight incompatible hypotheses are omitted from the sums, so no undefined ratio is evaluated. At a continuous boundary, the relevant rates are interpreted by one-sided limits.

With positive continuous rates and full-support r_s, every non-source endpoint is compatible at interior times. The identity endpoint $y = s$ is compatible exactly when the current value is $v = s$, including after a return to that value. This rule depends on the current state, not on its history. The learned model may still leave a source position, so preservation at inference is learned rather than guaranteed by the training identity labels.

B Transition-Matching Objectives and Classification Surrogates

We derive continuous rate matching and its discrete posterior-matching counterpart, then separate the supervision of jump timing from that of destination values. We introduce the revision-label CE used by Reflective Masking before comparing classification and rate matching under a shared target-token predictor. The final subsections establish objective non-equivalence, discuss calibration, and describe the vocabulary-sampling estimator.

B.1 PATH-SPACE RATE RISK AND ITS POPULATION OPTIMUM

Fix endpoints $(\mathbf{s}, \mathbf{y})$ and instruction c. Let $\mathbb{Q}^\star_{[a,b]}$ and $\mathbb{P}_{\theta,[a,b]}$ be the target and learned full-canvas path laws on $0 < a < b < 1$, traversed in decreasing τ from b to a. Their initial laws at b are $\nu^\star_b$ and $\nu_{\theta,b}$. The total exit intensity of the target process at canvas $\mathbf{x}$ is $\Lambda^\star_\tau(\mathbf{x}) = \sum_i \sum_{u \neq x_i} \rho^\star_\tau(x_i, u \mid y_i, s_i)$; replacing the target rates by $\rho_{\theta,\tau,i}$ defines $\Lambda_{\theta,\tau}(\mathbf{x})$. Assume nonexplosion, absolute continuity of $\mathbb{Q}^\star_{[a,b]}$ with respect to $\mathbb{P}_{\theta,[a,b]}$, and integrability of the rate terms below.

For a sampled path $\mathbf{X}_{[a,b]}$, let $\mathcal{J}$ record each jump as (τ, i, v, u), meaning that token i moves from v to u at time τ in the decreasing-time direction. All model rates at a jump are evaluated on the canvas just before that jump in the sampling direction. A path likelihood is its initial probability times the jump rates and the exponential survival factors between jumps. Its log-density ratio is therefore

$$\begin{aligned}\log \frac{d\mathbb{Q}^\star_{[a,b]}}{d\mathbb{P}_{\theta,[a,b]}}(\mathbf{X}_{[a,b]}) = \log \frac{\nu^\star_b(\mathbf{X}_b)}{\nu_{\theta,b}(\mathbf{X}_b)} + \sum_{(\tau,i,v,u)\in\mathcal{J}} \log \frac{\rho^\star_\tau(v, u \mid y_i, s_i)}{\rho_{\theta,\tau,i}(v, u)} \\ + \int_a^b \big[\Lambda_{\theta,\tau}(\mathbf{X}_\tau) - \Lambda^\star_\tau(\mathbf{X}_\tau)\big]\, d\tau.\end{aligned} \tag{26}$$

The sum compares the probabilities of the observed jumps, while the integral compares the probabilities of remaining in each state between jumps. Under the target process, a possible jump contributes in expectation at its target intensity. Writing $v_i = X_{\tau,i}$ inside the integral gives

$$\begin{aligned}&\mathbb{E}_{\mathbb{Q}^\star} \sum_{(\tau,i,v,u)\in\mathcal{J}} \log \frac{\rho^\star_\tau(v, u \mid y_i, s_i)}{\rho_{\theta,\tau,i}(v, u)} \\ &\quad = \mathbb{E}_{\mathbb{Q}^\star} \int_a^b \sum_{i=1}^N \sum_{u \neq v_i} \rho^\star_\tau(v_i, u \mid y_i, s_i) \log \frac{\rho^\star_\tau(v_i, u \mid y_i, s_i)}{\rho_{\theta,\tau,i}(v_i, u)}\, d\tau.\end{aligned} \tag{27}$$

Combining this term with the exit-intensity difference yields $D_{\mathrm{rate}}(\rho^\star, \rho_\theta)$ for each transition. Thus the Kullback–Leibler (KL) divergence between the path laws is

$$\mathrm{KL}(\mathbb{Q}^\star_{[a,b]} \| \mathbb{P}_{\theta,[a,b]}) = \mathrm{KL}(\nu^\star_b \| \nu_{\theta,b}) + \mathbb{E}_{\mathbb{Q}^\star} \int_a^b \sum_{i=1}^N \ell_{\mathrm{rate},i}\, d\tau. \tag{28}$$

The local loss is evaluated at $\mathbf{X}_\tau$ as in Equation (7). The sum over positions accounts for every single-token jump of the full-canvas process. Truncated bridge intervals retain their initial-law term. We use $D_{\mathrm{rate}}(0, b) = b$, $D_{\mathrm{rate}}(0, 0) = 0$, and $D_{\mathrm{rate}}(a, 0) = +\infty$ for $a > 0$, including identity paths whose outgoing target rates vanish.

At a fixed canvas, source, instruction, and time, average over the training distribution of target endpoints. For a free predicted rate b, the expected local loss has model-dependent part $-\mathbb{E}[a] \log b + b$, minimized by $b = \mathbb{E}[a]$. This motivates the endpoint mixture in Equation (6); attaining this optimum requires its learned weights to represent the conditional mean rate. Compatibility normalization enforces support but does not itself identify the true endpoint posterior. The auxiliary CE, token averaging, and truncated uniform-time sampling in Equation (8) define the implemented risk; Equation (28) gives its underlying unweighted path-space identity.

B.2 DISCRETE VARIATIONAL OBJECTIVE

For a fixed example $(\mathbf{s}, \mathbf{y}, c)$, let $q^\star$ be the selected path distribution over $\mathbf{X}_{1:T-1}$, with $\mathbf{X}_0 = \mathbf{y}$ and $\mathbf{X}_T = \mathbf{s}$. Its reverse kernels factorize across positions as $q^\star_{t-1|t}(\mathbf{u} \mid \mathbf{v}, \mathbf{y}, \mathbf{s}) = \prod_i q^\star_{t-1|t}(u_i \mid v_i, y_i, s_i)$. The model predicts $p_{\theta,i}(z \mid \mathbf{X}_t, c, t)$, normalized over compatible endpoints as in Equation (25) to obtain $\pi_{t,i}(z)$. For current value $v = X_{t,i}$, its discrete endpoint mixture is

$$P_{\theta,t,i}(u) = \sum_{z \in \mathcal{C}_t(v; s_i)} \pi_{t,i}(z) q^\star_{t-1|t}(u \mid v, z, s_i). \tag{29}$$

The learned full-canvas kernel is $P_{\theta,t}(\mathbf{u} \mid \mathbf{X}_t, \mathbf{s}, c) = \prod_i P_{\theta,t,i}(u_i)$, with each factor conditioned on the full current canvas through the predictor. Let p^{gen}_θ denote the endpoint distribution of this reverse

chain initialized at $\mathbf{s}$. Jensen's inequality gives

$$\log p_\theta^{\text{gen}}(\mathbf{y} \mid \mathbf{s}, c) \geq \mathbb{E}_{q^\star} \log \frac{\prod_{t=1}^{T} P_{\theta,t}(\mathbf{X}_{t-1} \mid \mathbf{X}_t, \mathbf{s}, c)}{\prod_{t=1}^{T} q^\star_{t-1|t}(\mathbf{X}_{t-1} \mid \mathbf{X}_t, \mathbf{y}, \mathbf{s})}.$$

Grouping the expected log ratios gives the posterior-matching bound

$$\log p_\theta^{\text{gen}}(\mathbf{y} \mid \mathbf{s}, c) \geq -\mathcal{L}_{\text{DT}},$$

$$\mathcal{L}_{\text{DT}} = -\mathbb{E}_{q_1^\star} \log P_{\theta,1}(\mathbf{y} \mid \mathbf{X}_1, \mathbf{s}, c) + \sum_{t=2}^{T} \mathbb{E}_{q_t^\star} \sum_{i=1}^{N} d_{t,i}, \tag{30}$$

$$d_{t,i} = \text{KL}\Big(q^\star_{t-1|t}(\cdot \mid X_{t,i}, y_i, s_i) \,\Big\|\, P_{\theta,t,i}(\cdot) \Big).$$

Each expectation samples the full-canvas marginal $q_t^\star(\mathbf{x} \mid \mathbf{y}, \mathbf{s}) = \prod_i q_t^\star(x_i \mid y_i, s_i)$. At $t = 1$, the posterior is a point mass at the target, so $\sum_i d_{1,i}$ is exactly the reconstruction term. Both chains start at the same fixed source, eliminating an initial-distribution KL term. Insufficient model support incurs infinite KL. Discrete training samples $t \sim \text{Unif}\{1, \ldots, T\}$ and replaces $\ell_{\text{rate},i}$ in Equation (8) by $T d_{t,i}$, with Target CE evaluated at t. The factor T estimates the time sum in Equation (30). Each posterior KL equals cross-entropy on that posterior minus its model-independent entropy.

B.3 Separating jump timing from destination

At one position and time, let a_j, b_j be target and model outgoing rates indexed by the allowed destinations j, with positive totals $\Lambda_a = \sum_j a_j$ and $\Lambda_b = \sum_j b_j$. For positive rates, writing each as its total intensity times a destination probability separates the logarithm,

$$\log \frac{a_j}{b_j} = \log \frac{\Lambda_a}{\Lambda_b} + \log \frac{a_j/\Lambda_a}{b_j/\Lambda_b}.$$

Multiplying by a_j, summing over destinations, and adding $-\Lambda_a + \Lambda_b$ gives

$$\sum_j D_{\text{rate}}(a_j, b_j) = D_{\text{rate}}(\Lambda_a, \Lambda_b) + \Lambda_a \,\text{KL}\left(\frac{\boldsymbol{a}}{\Lambda_a} \,\middle\|\, \frac{\boldsymbol{b}}{\Lambda_b} \right). \tag{31}$$

Here $\boldsymbol{a} = (a_j)_j$ and $\boldsymbol{b} = (b_j)_j$; dividing each by its total gives a destination distribution. Zero-rate cases follow by limits when support permits. For a visible hub state there is only one destination, so its learning signal is entirely about the exit intensity. At MASK the target and model total intensities agree, leaving a destination-distribution loss. Thus the two branches supervise when a visible token is withdrawn and which value replaces a masked token.

B.4 Revision-label cross-entropy

Reflective Masking supervises revision through categorical cross-entropy on prescribed token labels (Zhang et al., 2026). Write $v = X_{\tau,i}$ for the current value and y_i for its target, using our state notation. Let $p^{\text{cls}}_{\theta,i}(u)$, $u \in \mathcal{V}_{\texttt{<MASK>}}$, denote its categorical output at a sampled training input, with the history representation and instruction omitted from the notation. Its core per-token objective can then be written as

$$\ell_{\text{cls},i} = \begin{cases} -\log p^{\text{cls}}_{\theta,i}(y_i), & v = \texttt{<MASK>} \quad \text{(reveal)}, \\ -\log p^{\text{cls}}_{\theta,i}(\texttt{<MASK>}), & v \in \mathcal{V},\ v \neq y_i \quad \text{(remask)}, \\ -\log p^{\text{cls}}_{\theta,i}(y_i), & v = y_i \quad \text{(keep)}. \end{cases} \tag{32}$$

Summing over the visual positions gives the core revision-label objective

$$\mathcal{L}_{\text{cls}}(\theta) = \mathbb{E}\left[\sum_{i=1}^{N} \ell_{\text{cls},i} \right], \tag{33}$$

where the expectation uses that method's data, synthetic trajectories, and sampled training steps. At a visible position, the label specifies whether to keep its value or replace it with MASK; at a masked position, it specifies the target value to reveal. PaRDi instead supervises the transition rates supplied by the endpoint-conditioned bridge in Equation (7). Its predictor assigns probabilities to target values in $\mathcal{V}$, and the analytic bridge map converts these probabilities into rates. Below we compare these revision labels with rate matching under a shared correctness probability, and distinguish them from a binary surrogate constructed using the bridge coefficients.

B.5 RATE MATCHING AND CLASSIFICATION AT EACH STATE

For the following local calculations, fix a position i, its current full context, and an interior time τ. Suppress time indices on $A_\tau, C_\tau, M_\tau, \lambda_\tau, \mu_\tau$, and write $\pi_j = \pi_{\tau,i}(j)$ for the compatible endpoint belief. Assume these coefficients and rates are positive and r_s is model-independent with full support. We write $\mathrm{BCE}(q,p) = -q\log p - (1-q)\log(1-p)$ for binary cross-entropy with label probability q and prediction p.

Visible state distinct from the source. Consider a visible state $v \neq s$ of positive training occupancy under the deterministic-identity convention. Its compatible endpoint distribution excludes s. Set

$$a_0 = \frac{\mu M}{C} > 0, \qquad \kappa = \frac{A}{A + Cr_s(v)} \in (0,1), \qquad p = \pi_v, \qquad \zeta = \mathbf{1}\{y = v\}.$$

The target and model remasking rates are $a = a_0(1-\kappa\zeta)$ and $b = a_0(1-\kappa p)$. Thus

$$\ell_v(p;\zeta) = a_0\left[(1-\kappa\zeta)\log\frac{1-\kappa\zeta}{1-\kappa p} + \kappa(\zeta - p)\right]. \tag{34}$$

When $v = y \neq s$, the remasking rate is $a_0(1-\kappa) > 0$, so reaching the target value at an intermediate time does not lock that token for the rest of the bridge. This differs from the keep label in Equation (32); the separate $y = s$ identity path remains constant by Equation (4).

Keep/remask CE at visible states. To compare classification with rate matching using the same endpoint belief p, set $b^\star = \kappa\zeta$ and $b_\theta = \kappa p$ and define

$$\ell_{\mathrm{keep/remask}} = a_0\,\mathrm{BCE}(b^\star, b_\theta). \tag{35}$$

These Bernoulli parameters are constructed from the bridge coefficients and endpoint belief, rather than taken from the categorical output in Equation (32). Expanding the two Bernoulli terms gives

$$a_0\,\mathrm{KL}(\mathrm{Bern}(b^\star)\|\,\mathrm{Bern}(b_\theta)) = \ell_v + D_{\mathrm{rate}}(a_0 b^\star, a_0 b_\theta). \tag{36}$$

Here $\mathrm{Bern}(b)$ is the Bernoulli distribution with success probability b. The parameters describe a classification surrogate, not the finite-step jump probability of the sampler. After subtracting its weighted target entropy, Keep/remask CE equals the remasking-rate loss plus a nonnegative rate penalty for the complementary branch. If training uses a sampled hard branch label with conditional mean $\kappa\zeta$, its expected BCE equals the soft-label BCE.

Source state with deterministic identity. At $v = s$, let $p_s = \pi_s$ be the compatible source endpoint probability and $\zeta_{\mathrm{id}} = \mathbf{1}\{y = s\}$. Then $a = a_0(1-\zeta_{\mathrm{id}})$ and $b = a_0(1-p_s)$. For identity and edit endpoints respectively,

$$\ell_s = \begin{cases} a_0(1-p_s), & \zeta_{\mathrm{id}} = 1, \\ a_0[-\log(1-p_s) - p_s], & \zeta_{\mathrm{id}} = 0. \end{cases}$$

As above, $a_0\,\mathrm{KL}(\mathrm{Bern}(\zeta_{\mathrm{id}})\|\,\mathrm{Bern}(p_s)) - \ell_s = D_{\mathrm{rate}}(a_0\zeta_{\mathrm{id}}, a_0 p_s)$.

MASK-state destination loss. Here $a_j = (\lambda/M)(A\mathbf{1}\{j = y\} + Cr_s(j))$ and $b_j = (\lambda/M)(A\pi_j + Cr_s(j))$. Their shared total is $\Lambda = \lambda(A+C)/M$. Set $\alpha = A/(A+C)$, $d^\star(j) = \alpha\delta_y(j) + (1-\alpha)r_s(j)$, and $d_\theta(j) = \alpha\pi_j + (1-\alpha)r_s(j)$. For distributions d, e, define $\mathrm{CE}(d,e) = -\sum_j d(j)\log e(j)$ and $H(d) = -\sum_j d(j)\log d(j)$. Then

$$\ell_{\texttt{<MASK>}} = \Lambda\,\mathrm{KL}(d^\star\|d_\theta) = \Lambda\,\mathrm{CE}(d^\star, d_\theta) - \Lambda H(d^\star). \tag{37}$$

This is CE on smoothed destinations, not ordinary Target CE. The proposal floor keeps it finite as $\pi_y \to 0$, unlike $-\log\pi_y$.

To decompose these destination mixtures, let B identify their target or proposal component. The target components are $q_{\mathrm{mark}}(B = \mathrm{target}, j) = \alpha\delta_y(j)$ and $p_{\mathrm{mark}}(B = \mathrm{target}, j) = \alpha\pi_j$, while both proposal branches equal $(1-\alpha)r_s(j)$. The KL chain rule gives

$$\frac{\lambda A}{M}(-\log\pi_y) - \ell_{\texttt{<MASK>}} = \Lambda\sum_j d^\star(j)\,\mathrm{KL}\big(q_{\mathrm{mark}}(B \mid j)\|p_{\mathrm{mark}}(B \mid j)\big) \geq 0. \tag{38}$$

The gap measures the additional component information retained by the weighted Target CE and marginalized out by the destination-rate loss.

B.6 OBJECTIVE NON-EQUIVALENCE AND CALIBRATION

The preceding decompositions identify the extra terms introduced by classification under the shared endpoint predictor. We now show that fixed-weight CE terms on affine probabilities cannot reproduce the visible-state rate loss exactly, while distinguishing this functional difference from their population optima. Returning to the visible-state loss in Equation (34), fix $v \neq s$ and the corresponding a_0, κ, p, ζ. Its first two derivatives are

$$\ell'_v(p;\zeta) = \frac{a_0\kappa^2(p-\zeta)}{1-\kappa p}, \qquad \ell''_v(p;\zeta) = \frac{a_0\kappa^2(1-\kappa\zeta)}{(1-\kappa p)^2} > 0. \tag{39}$$

Proposition 3 (Finite affine-probability CE separation)**.** *For the visible-state loss in Equation* (34)*, assume at least two compatible endpoints and the positive coefficients specified in Section B.5. On no nonempty open interval of $p \in (0,1)$ can $\ell_v(p;\zeta)$ equal, up to a model-independent constant, an objective*

$$G(p) = -\sum_{\ell=1}^{L_{\mathrm{CE}}} c_\ell \log(d_\ell + e_\ell p), \tag{40}$$

where $L_{\mathrm{CE}} < \infty$, all coefficients are independent of p, and each logarithm argument is positive on the interval. The statement allows arbitrary fixed real weights c_ℓ.

Proof. If equality held, the derivatives would agree there. Every term of G' is either zero or $-c_\ell e_\ell/(d_\ell + e_\ell p)$. Their finite sum is a proper rational function with limit zero at infinity. In contrast,

$$\ell'_v(p;\zeta) = -a_0\kappa + \frac{a_0\kappa(1-\kappa\zeta)}{1-\kappa p} \tag{41}$$

has limit $-a_0\kappa \neq 0$. Agreement on an open interval makes the rational functions identical, since clearing denominators gives a polynomial difference that vanishes throughout the interval. The different limits contradict this identity. □

The class includes fixed-weight binary CE and categorical CE after restricting the compatible distribution to $\pi_v = p,\ \pi_j = (1-p)\xi_j$ for $j \neq v, s$, with fixed positive ξ_j summing to one. It also includes fixed affine aggregation or smoothing of those probabilities. Any identity with fixed-weight categorical CE under this parameterization would hold on this path and contradict the proposition. Endpoint normalization is handled by fixing the raw source probability, so that its removal has a constant normalizer.

Corollary 1 (Revision-label CE under a shared correctness probability)**.** *Under the assumptions of Proposition 3, fix the input and a visible value $v \neq s$. Let $w_{\mathrm{cls}} \in (0,1]$ be a fixed total probability mass assigned to the two possible revision labels v and <MASK>. Identify their conditional keep probability with the endpoint belief $p = \pi_v$ by setting*

$$p^{\mathrm{cls}}_{\theta,i}(v) = w_{\mathrm{cls}} p, \qquad p^{\mathrm{cls}}_{\theta,i}(\texttt{<MASK>}) = w_{\mathrm{cls}}(1-p). \tag{42}$$

For either label $\zeta = \mathbf{1}\{y = v\}$, the visible-state revision-label CE in Equation (32)*, or any fixed positive multiple of it, cannot equal $\ell_v(p;\zeta)$ up to a model-independent constant on a nonempty open interval of $p \in (0,1)$.*

Proof. Substituting Equation (42) into the two visible-state cases of Equation (32) gives

$$\begin{aligned} \ell_{\mathrm{cls},i} &= -\zeta \log(w_{\mathrm{cls}} p) - (1-\zeta)\log\big(w_{\mathrm{cls}}(1-p)\big) \\ &= \mathrm{BCE}(\zeta, p) - \log w_{\mathrm{cls}}. \end{aligned} \tag{43}$$

The last term is independent of p, while the BCE is a finite sum of logarithms of affine probabilities, with fixed weights ζ and $1-\zeta$. Any fixed positive rescaling remains in the class of Proposition 3, which proves the claim. □

The case $w_{\mathrm{cls}} = 1$ gives a binary keep/remask classifier; $w_{\mathrm{cls}} < 1$ allows probability mass on other vocabulary values while keeping the two revision labels' total mass fixed. This hard-label CE differs from the bridge-derived surrogate $a_0\,\mathrm{BCE}(\kappa\zeta, \kappa p)$ in Equation (35); the proposition covers both under their stated parameterizations.

Conditional expectation and calibration. To compare their population optima, use the same conditioning information and conditional endpoint distribution for both losses, and let $\eta \in (0,1)$ be the probability that the target equals v. Averaging Equation (34) replaces ζ by η in its model-dependent part. The same nonzero affine term proves the same separation. Under Equation (42), the expected revision-label CE is $\mathrm{BCE}(\eta, p) - \log w_{\mathrm{cls}}$. Both this objective and the expected rate loss have the unique optimum $p = \eta$. Their derivatives satisfy

$$\frac{d}{dp}\mathbb{E}[\ell_v] = \frac{a_0 \kappa^2 p(1-p)}{1-\kappa p}\frac{d}{dp}\,\mathrm{BCE}(\eta, p). \tag{44}$$

The scaling is positive but depends on the model prediction. Thus the non-equivalence in Corollary 1 concerns loss values and gradients, even though these losses share the same optimal correctness probability under this common conditional distribution.

Scope of the comparison. The proposition concerns finite CE sums on affine probabilities of a fixed predictor. For a nonlinear representation, take $g = b/a_0 = 1 - \kappa p \in (0,1)$,

$$D_{\mathrm{rate}}(a,b) = a\ \mathrm{BCE}(1,g) + a_0\ \mathrm{BCE}(1, e^{-g}) + a\log(a/a_0) - a.$$

The nonlinear probability e^{-g} lies outside this class, as do model-dependent weights and new prediction heads. An infinitesimal Bernoulli construction also recovers the rate divergence,

$$\lim_{\Delta \downarrow 0} \frac{\mathrm{KL}(\mathrm{Bern}(a\Delta) \| \,\mathrm{Bern}(b\Delta))}{\Delta} = D_{\mathrm{rate}}(a,b).$$

Replacing KL by CE without subtracting the target entropy does not give this limit.

B.7 Monte Carlo estimation of the PARDi rate loss

At MASK, the target and model total rates are equal, so the linear terms in the rate divergence cancel. Using the local notation above, set $g_j^\star = A\mathbf{1}\{j = y\} + Cr_s(j)$ and $g_j^\theta = A\pi_j + Cr_s(j)$ for the target and model-induced visible masses. Separating the target entry gives

$$\ell_{\texttt{<MASK>}} = \frac{\lambda}{M}\left[g_y^\star \log \frac{g_y^\star}{g_y^\theta} + C\sum_{j \neq y} r_s(j) \log \frac{Cr_s(j)}{g_j^\theta} \right]. \tag{45}$$

The implementation uses the uniform codebook proposal $r_s(j) = 1/V$ with $V = 16384$. Draw $U_1, \ldots, U_{S_{\mathrm{MC}}}$ independently with replacement from this same distribution, where $S_{\mathrm{MC}} = 8$. The estimator is

$$\widehat{\ell}_{\texttt{<MASK>}} = \frac{\lambda}{M}\left[g_y^\star \log \frac{g_y^\star}{g_y^\theta} + \frac{C}{S_{\mathrm{MC}}}\sum_{\ell=1}^{S_{\mathrm{MC}}} \mathbf{1}\{U_\ell \neq y\} \log \frac{Cr_s(U_\ell)}{g_{U_\ell}^\theta} \right]. \tag{46}$$

The target term is exact. A draw equal to the target contributes zero to the sampled sum and is neither redrawn nor removed from its denominator of eight. Source tokens are not excluded, and repeated draws are retained. Taking the expectation of the sampled term recovers the sum in Equation (45), so the estimator is unbiased before any nonlinear numerical clipping. MC8 estimates a vocabulary sum; the model still predicts the full codebook, and time is sampled separately. Individual MC8 estimates can be negative even though their expectation, the exact rate divergence, is nonnegative.

Training uses global-token-mean reduction with identity weight one. Dynamical and Target CE weights are both one. Training times follow the truncated uniform distribution in Equation (8) with $\varepsilon_{\mathrm{tr}} = 0.001$, so the training risk averages over that interval. The comparison protocol in Section 4.3 fixes these operations when isolating a learning-objective effect.

C Experimental Details

We specify the controlled training-state construction, data and endpoint preparation, and evaluation settings used in Section 4.

C.1 Experimental configuration

Table 6 summarizes the training and inference configuration.

Table 6: Training and inference configuration.

Parameter	Setting
Training	
Base model	LLaDA2.0-Uni
Proposal	Uniform over the full codebook, $V = 16{,}384$
Masking / reveal rates	$\lambda_\tau = \mu_\tau = 4$
LoRA rank / alpha / dropout	128 / 256 / 0.05
LoRA target modules	`query_key_value`, `dense`, `lm_head`
Optimizer	Fused AdamW
Adam parameters	$\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon_{\text{Adam}} = 10^{-8}$
Weight decay	0.0
Learning rate	10^{-4}
Learning-rate schedule	Warmup over the first 3% of steps, then constant
Training hardware	24 A100 GPUs across 3 nodes
Global batch size	144
Training duration	5 epochs
Distributed training	AutoEP8 + DeepSpeed ZeRO-2
Training seed	42
Time encoding	Fourier features + MLP, injected into the time-marker residual
Fourier dimension	64
Time MLP dimension	512
Maximum Fourier frequency	1000
Inference	
Text CFG scale / rescale	1.5 / 0
Stochastic-update temperature	1.0
Sampling grid	30 uniform updates from 1 to 10^{-6}
Residual MASK completion	Argmax excluding the original source token
NFE (CFG branches counted separately)	60; 62 with terminal completion

C.2 HAND-DESIGNED EDIT STATES

The revised hand-designed control retains manually specified masking and reveal times inspired by Reflective Masking (Zhang et al., 2026), while matching PaRDi's uniform-proposal mass at each training time. For an edited position $s_i \neq y_i$, ordering two independent $\mathrm{Unif}(0,1)$ times gives a source–MASK–target schedule with marginal

$$q_\tau^{\text{sched}}(v \mid y_i, s_i) = \tau^2 \delta_{s_i}(v) + 2\tau(1-\tau)\delta_{\texttt{<MASK>}}(v) + (1-\tau)^2 \delta_{y_i}(v). \tag{47}$$

Expanding the bridge marginal in Equation (2) for distinct endpoints and $\lambda_\tau = \mu_\tau = 4$ gives the uniform-component weight

$$w_{\text{prop}}(\tau) = \frac{C_\tau C_{1-\tau}}{C_1}, \qquad C_\tau = \frac{(1-e^{-4\tau})^2}{2}. \tag{48}$$

Using the same full-codebook proposal $r_{s_i}(v) = 1/V$, with $V = 16{,}384$, the updated training-state distribution is

$$q_\tau^{\text{hand}}(v \mid y_i, s_i) = \begin{cases} \delta_{s_i}(v), & y_i = s_i, \\ [1 - w_{\text{prop}}(\tau)] q_\tau^{\text{sched}}(v \mid y_i, s_i) + w_{\text{prop}}(\tau) r_{s_i}(v), & y_i \neq s_i. \end{cases} \tag{49}$$

The proposal can return either endpoint; its contribution is added to the corresponding point mass. Since $w_{\text{prop}}(0) = w_{\text{prop}}(1) = 0$, the construction preserves both endpoints and keeps unchanged positions fixed. It matches the bridge's proposal component while retaining the hand-designed allocation among source, MASK, and target. As for PaRDi states, training samples $\tau \sim \mathrm{Unif}[\varepsilon_{\text{tr}}, 1 - \varepsilon_{\text{tr}})$ with $\varepsilon_{\text{tr}} = 0.001$ and supplies τ to the model.

The controls share data, initialization, sample exposure, target-token head, time interface, deterministic identity convention, token averaging, and inference budget. Using the same PaRDi sampler compares how effectively each training configuration supports these inference dynamics. On hand-designed states, the rate loss also uses the PaRDi target transition rates.

Both diagnostic losses use the same held-out PaRDi bridge states and times with global-token-mean reduction. For every model, we apply the common analytic bridge map to its predicted target probabilities and evaluate rate divergence using Equation (7).

C.3 Training data

Image-editing training uses ImgEdit (Ye et al., 2026), Pico-Banana-400K (Qian et al., 2026), and BRIDGE (Xiong et al., 2026).

We construct visual-reasoning data for Maze, TSP, Sudoku, and VSP following the task types used in DiffThinker (He et al., 2025).

C.4 Endpoint preprocessing

For image editing, let $\mathbf{y}^{\mathrm{raw}}$ be the raw target canvas and let $e_i^{\mathrm{ann}} \in \{0, 1\}$ indicate whether position i lies inside the annotation-derived edit region. Following common practice in discrete image editing, we construct the training target as

$$y_i = \begin{cases} y_i^{\mathrm{raw}}, & e_i^{\mathrm{ann}} = 1, \\ s_i, & e_i^{\mathrm{ann}} = 0. \end{cases} \tag{50}$$

All ablations, including the reference-based regeneration baseline, use the same training data and target construction. Inference requires only the source image and instruction.

C.5 Inference and scoring

PaRDi uses 30 integration updates on the uniform grid

$$\tau_k = 1 - \frac{k}{30}(1 - \varepsilon), \qquad k = 0, \ldots, 30, \qquad \varepsilon = 10^{-6}. \tag{51}$$

Each update predicts at τ_k and advances to τ_{k+1}; the time module receives the raw τ_k. If MASK tokens remain at ε, one additional prediction on the final canvas supplies CFG logits $\ell_i^{\mathrm{CFG}}(v)$, and each remaining MASK at position i is replaced by $\arg\max_{v \in \mathcal{V} \setminus \{s_i\}} \ell_i^{\mathrm{CFG}}(v)$. Visible tokens remain unchanged; this endpoint projection completes the numerical sampler. CFG uses scale 1.5 and rescale 0; stochastic updates use temperature 1.0. Each CFG prediction uses two model forwards, giving 60 function evaluations (NFE), or 62 when terminal completion is required. Image decoding follows the default settings of LLaDA2.0-Uni (AI et al., 2026).

GEdit-Bench and ImgEdit evaluations follow their official default settings. Visual-reasoning evaluation follows the official default settings of DiffThinker (He et al., 2025).

For metric computation, each method uses its own VQ-reconstructed source images as the source reference, reducing the influence of VQ reconstruction differences and focusing evaluation on editing performance. Qualitative comparisons display the original source images as a common reference. PIE token metrics use native-code equality and are averaged over valid images with background tokens; UniAR uses full BSQ-code equality. Unchanged outputs receive 100% Keep and 0% Leak.

D ADDITIONAL RESULTS

D.1 EDITING CATEGORY BREAKDOWNS

Category-level scores complement the aggregate editing results in Table 1. The GEdit-EN revision-budget sweep further examines how limiting repeated renewal affects different edit types.

Table 7: GEdit-EN category scores.

Method	Background ↑	Color ↑	Material ↑	Motion ↑	Portrait ↑	Style ↑	Add ↑	Remove ↑	Replace ↑	Text ↑	Tone ↑	Overall ↑
▼ *Autoregressive*												
UniAR	6.86	5.14	4.38	5.56	6.74	5.56	5.82	3.80	5.85	2.23	**6.42**	5.31
▼ *Masked Generative Models*												
EditMGT	**7.69**	**7.71**	5.77	3.84	5.13	6.53	6.13	5.24	5.56	4.53	**6.42**	5.87
▼ *Discrete Diffusion Models*												
Lumina-DiMOO	3.61	3.28	5.00	4.59	5.07	4.70	4.20	2.10	5.17	2.56	3.45	3.91
▼ *Block Discrete Diffusion Models*												
LLaDA2.0-Uni	7.27	7.29	5.82	6.45	6.14	**6.70**	**7.27**	6.63	7.14	6.07	6.41	6.61
▼ *Training-Free In-Place Token Editing*												
DICE	1.61	1.41	2.02	3.97	5.00	2.58	1.93	3.21	3.57	2.28	3.64	2.89
GIDE	4.59	5.92	5.00	3.90	4.30	3.84	3.89	6.11	5.79	2.81	5.11	4.49
▼ *Discrete Diffusion Bridges*												
PaRDi	7.44	6.83	**7.06**	**6.65**	**7.26**	5.93	7.09	**8.32**	**7.64**	**6.28**	6.26	**6.98**

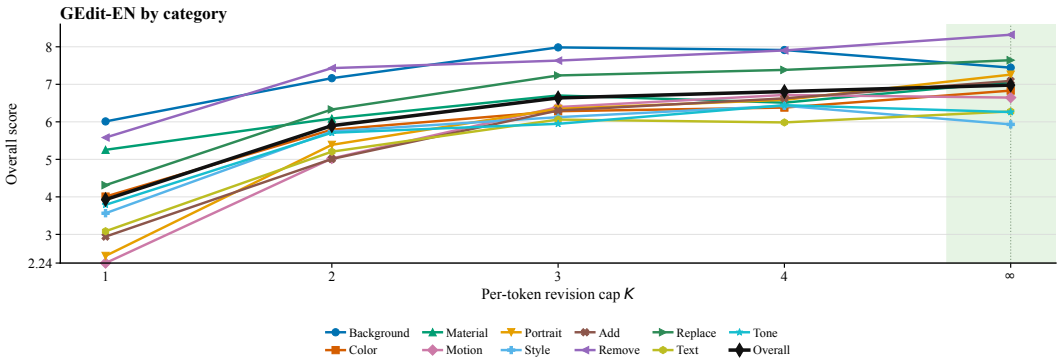

Figure 6: GEdit-EN under different per-token revision caps. Each colored line shows one editing category and the thicker black line shows Overall. All caps use the same 30-update time grid and terminal completion rule. The shaded $K = \infty$ column reproduces the unrestricted PaRDi results in the preceding category table.

Table 8: ImgEdit category scores.

Method	Add ↑	Adjust ↑	Extract ↑	Replace ↑	Remove ↑	Background ↑	Style ↑	Hybrid ↑	Action ↑	Overall ↑
▼ *Autoregressive*										
UniAR	**3.91**	3.48	2.75	3.94	3.64	3.74	4.27	3.06	**4.70**	3.73
▼ *Masked Generative Models*										
EditMGT	3.08	3.11	1.41	3.48	2.89	2.95	4.33	1.81	2.58	2.94
▼ *Discrete Diffusion Models*										
Lumina-DiMOO	3.41	2.38	1.90	3.26	2.21	2.11	4.19	2.26	3.17	2.77
▼ *Block Discrete Diffusion Models*										
LLaDA2.0-Uni	3.76	**4.16**	2.40	4.04	**3.82**	**4.07**	**4.60**	**3.97**	4.42	3.92
▼ *Training-Free In-Place Token Editing*										
DICE	2.79	2.41	1.81	2.60	1.67	1.61	2.61	1.52	2.84	2.22
GIDE	3.20	3.03	1.79	3.53	2.98	2.92	3.22	2.32	2.92	2.89
▼ *Discrete Diffusion Bridges*										
PaRDi	**3.91**	3.64	**4.05**	**4.25**	3.69	**4.07**	4.07	3.86	3.71	**3.95**

D.2 TARGET ALIGNMENT AND SOURCE PRESERVATION

PaRDi retains more source content at comparable semantic alignment within the edited region (Table 9). PaRDi and GIDE attain nearly identical PIE-Target CLIP scores of 27.19 and 27.18, while PaRDi preserves 92.27% of background tokens compared with 69.40% for GIDE and produces less leakage. Its lower MSE and higher SSIM than GIDE outside the edit region show that this preservation advantage extends to decoded appearance and structure. The same combination of strong target alignment and improved preservation holds against the absorbing-state and uniform-transition controls. The updated hand-designed control extends this comparison to state construction with the proposal distribution and its time-dependent weight held fixed (Section C.2).

Table 9: Target alignment and source preservation on PIE. PIE-Target CLIP (Edit region) follows the original PIE settings; MSE and SSIM measure source preservation outside the edit region. Token Keep and Leak are reproduced from Tables 1 to 3. Values are reported to four significant digits.

Method / training objective	PIE-Target CLIP ↑	Token metrics (%)		Outside edit region	
		Keep ↑	Leak ↓	MSE ↓	SSIM ↑
▼ *Baseline methods*					
UniAR-RL	26.27	0.000	31.62	0.01326	0.7429
EditMGT	26.80	33.60	44.60	0.01710	0.7813
Lumina-DiMOO	25.93	87.39	31.62	0.005054	0.8040
LLaDA2.0-Uni	26.96	59.72	45.97	0.02432	0.6810
DICE	26.18	88.87	42.31	<u>0.005025</u>	**0.8968**
GIDE	<u>27.18</u>	69.40	36.12	0.006589	0.8518
▼ *Transition structure*					
Absorbing-state diffusion	27.11	84.52	31.84	0.008905	0.8577
Uniform-transition bridge	27.01	<u>91.38</u>	18.51	0.006037	0.8810
▼ *Hand-designed edit states*					
Target CE	22.21	82.48	22.87	0.01650	0.7955
Target CE + Keep/remask CE	22.57	85.28	20.46	0.01518	0.8082
Target CE + PaRDi rate loss	22.97	85.45	21.49	0.01329	0.8123
▼ *PaRDi bridge states*					
Target CE	27.05	87.43	<u>17.85</u>	0.007740	0.8736
Target CE + Keep/remask CE	27.14	89.65	18.75	0.006742	0.8806
Target CE + PaRDi rate loss (**Full**)	**27.19**	**92.27**	**14.19**	**0.004709**	<u>0.8861</u>

D.3 ADDITIONAL IMAGE-EDITING EXAMPLES

The comparisons in Figures 7 and 8 show PaRDi's ability to retain scene composition and unrelated appearance details across changes in pose, expression, and object identity. In scenes with multiple objects, the requested subject can change while neighboring objects remain close to the source, illustrating the selective editing behavior reflected in the preservation metrics.

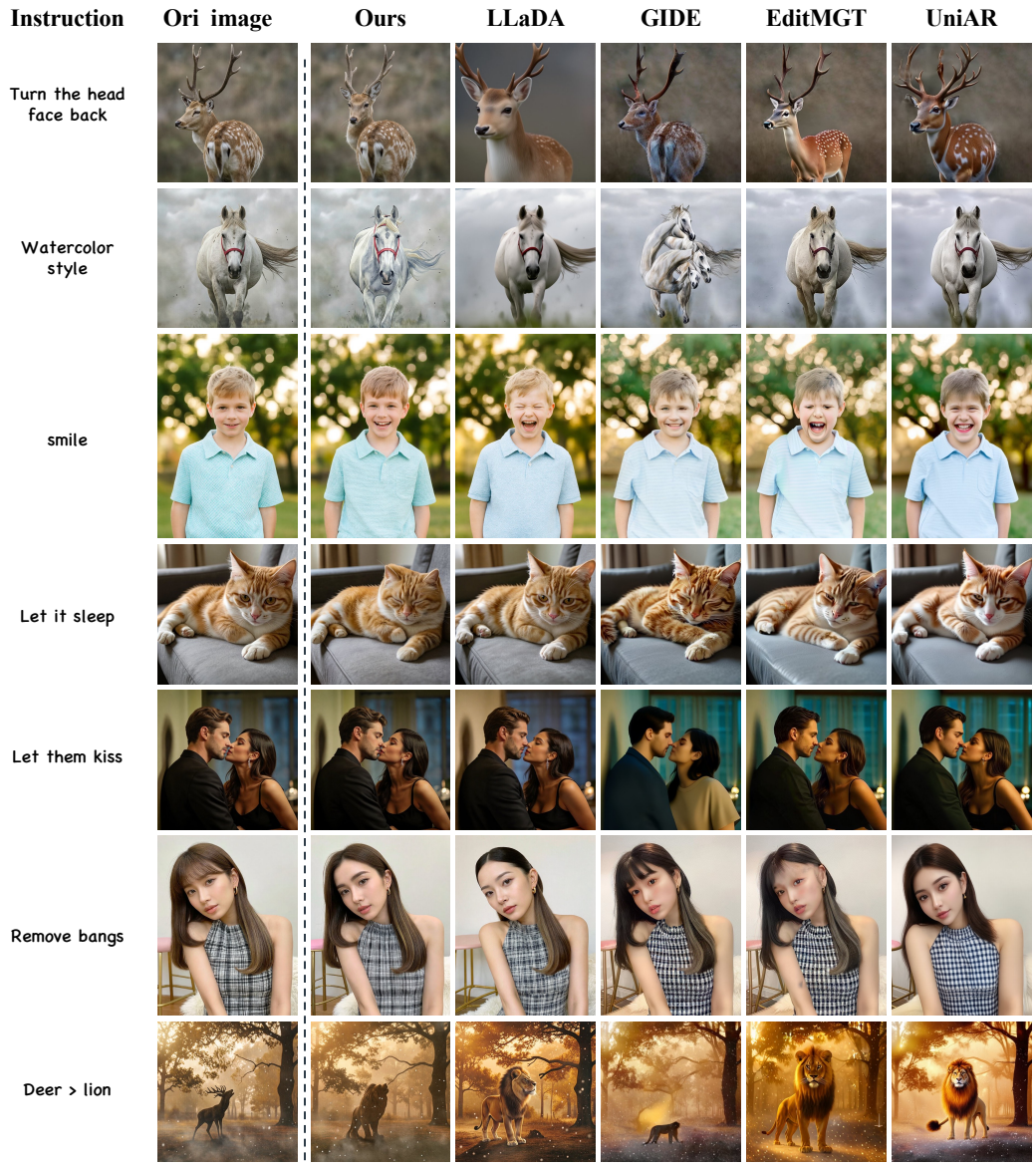

Figure 7: Additional image-editing comparisons across pose, style, expression, appearance, and object replacement. Each row shows an instruction, the source image, and outputs from PaRDi (Ours), LLaDA2.0-Uni, GIDE, EditMGT, and UniAR, supplementing Figure 3.

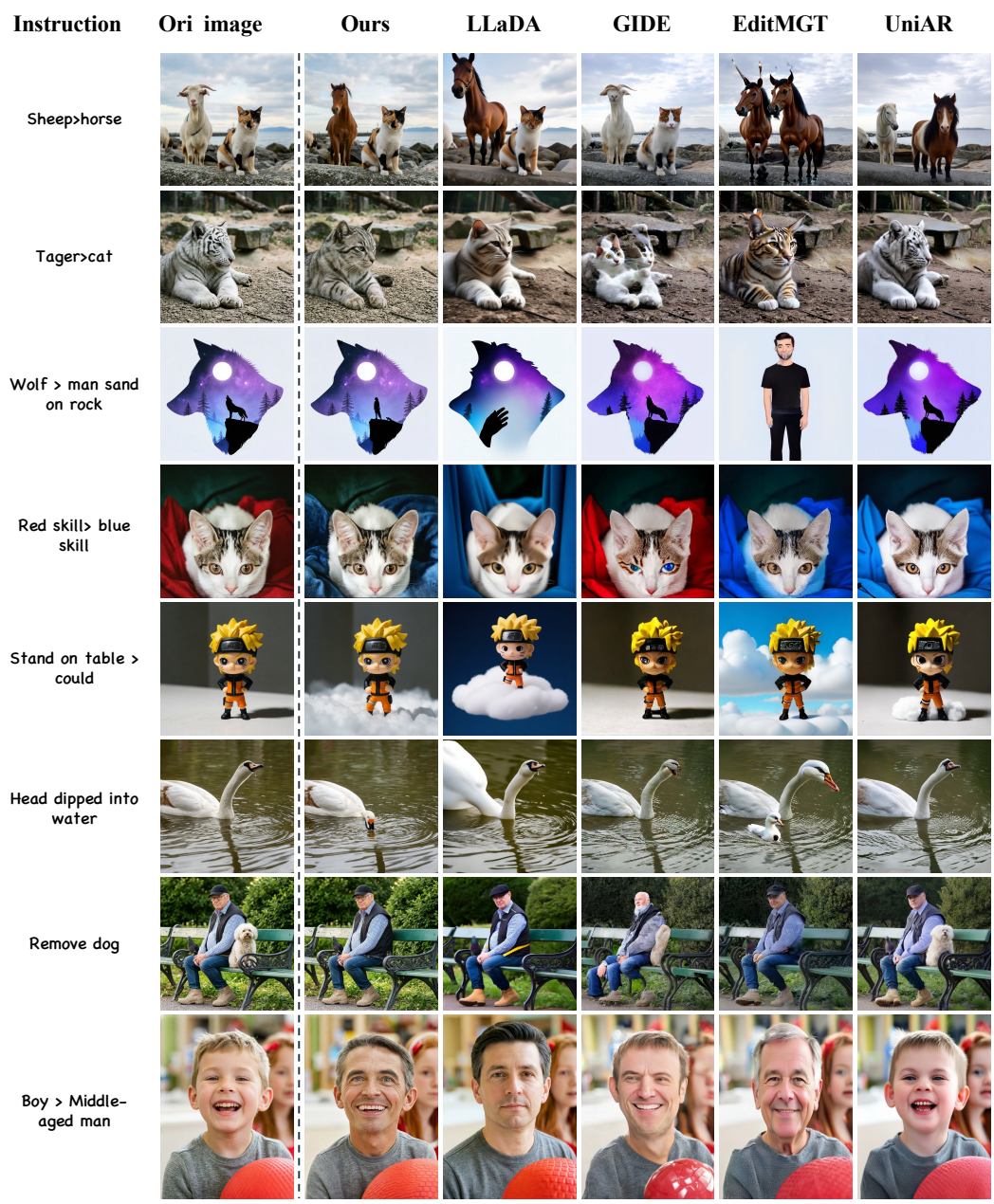

Figure 8: Additional image-editing comparisons across object replacement, color, background, action, object removal, and age changes. Columns show the instruction, source image, PaRDi (Ours), LLaDA2.0-Uni, GIDE, EditMGT, and UniAR.

D.4 ADDITIONAL VISUAL-REASONING EXAMPLES

The final outputs in Figures 9 and 10 reveal how the revision cap affects solution structure. Higher revision caps yield more consistent digit assignments and more complete path connections in these examples. These final-output comparisons illustrate how allowing repeated renewal improves solution quality under the shared sampling schedule.

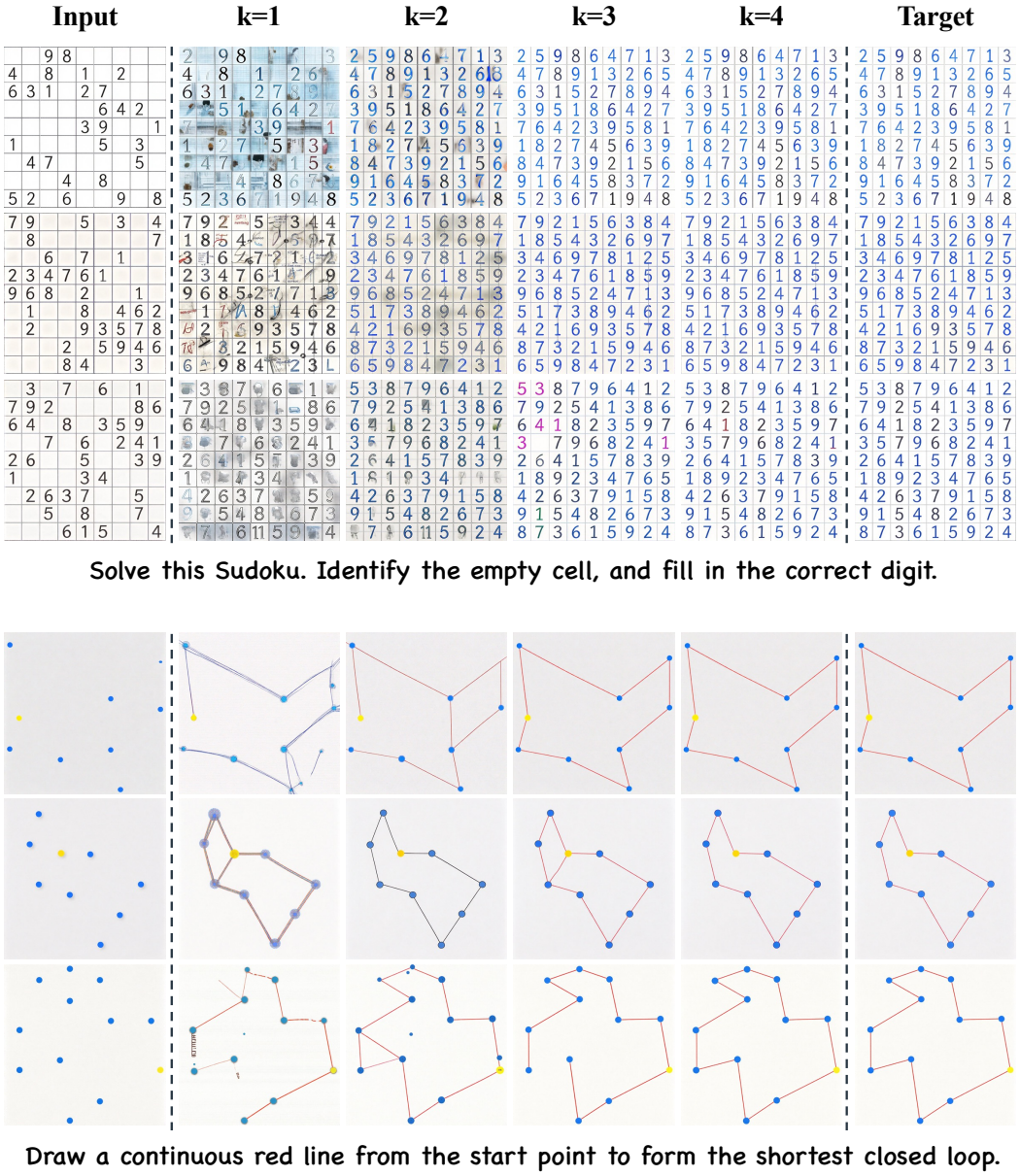

Figure 9: Additional Sudoku (top) and TSP (bottom) examples under per-token revision caps $K = 1, 2, 3, 4$. Columns show the input, PaRDi's final output for each cap, and a ground-truth solution (Target). All caps use the same 30-update time grid and terminal completion rule.

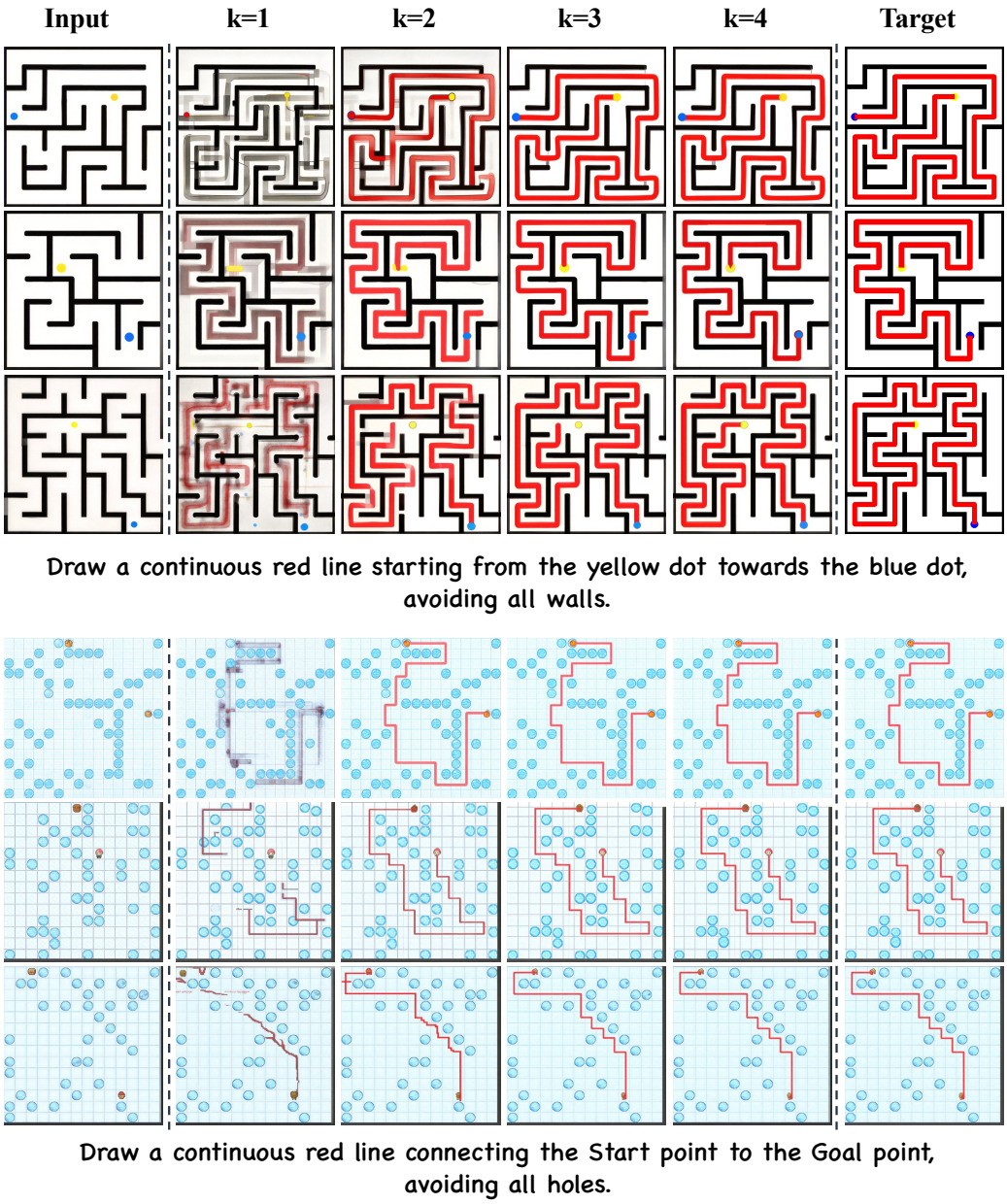

Figure 10: Additional Maze (top) and VSP (bottom) examples under per-token revision caps $K = 1, 2, 3, 4$. Columns show the input, PaRDi's final output for each cap, and a ground-truth solution (Target). All caps use the same 30-update time grid and terminal completion rule.

D.5 THEORETICAL ATTENTION COST

We count attention interactions for the inference settings listed below, using 256 text tokens and 1,024 tokens per image. Counts are normalized to one Transformer attention block and include the listed guidance branches. For fully bidirectional sequences, the count is steps $\times$ branches $\times L^2$, where L is the sequence length. Editing a single canvas reduces L from 2,304 to 1,280 by avoiding a separate target sequence. At fixed steps and guidance branches, this uses 30.86% as many attention interactions as fully bidirectional source-plus-target generation, reducing the cost of every revision update. PaRDi uses 30 CFG prediction rounds for integration, or 31 including terminal completion (Section C.5), giving $30 \times 2 \times 1280^2 \approx 98.30\text{M}$ or $31 \times 2 \times 1280^2 \approx 101.58\text{M}$ interactions. The counts describe attention computation under the listed configurations; measured latency is reported separately in Table 2. Projection and feed-forward operations, codec costs, and hardware effects are excluded.

Table 10: Theoretical attention interactions with $n_{\text{txt}} = 256$ text tokens and $n_{\text{src}} = n_{\text{tgt}} = 1024$ source and target image tokens. L is the total sequence length; n_q, n_k count attention queries and keys. Costs are normalized to one Transformer attention block. B2/B3 denote two/three guidance branches; M/B denote millions/billions of interactions in the cost column. Forward depth counts prediction rounds before multiplying by guidance branches. These values are not measured model FLOPs.

Method / path	Attention sequence	Inference setting	Forward depth ↓	Normalized attention interactions ↓
▼ *Autoregressive*				
UniAR-RL	Prefix $n_{\text{txt}} + n_{\text{src}} = 1280$; generate $n_{\text{tgt}} = 1024$ causally	1,024 NTP, CFG B2	1024	3.38B
▼ *Masked Generative Models*				
EditMGT	$L = n_{\text{txt}} + n_{\text{src}} + n_{\text{tgt}} = 2304$; fully bidirectional	36 steps, CFG B2	<u>36</u>	382.21M
▼ *Discrete Diffusion Models*				
Lumina-DiMOO	$L = n_{\text{txt}} + n_{\text{src}} + n_{\text{tgt}} = 2304$; fully bidirectional	64 steps, text/image CFG B3	64	1.02B
▼ *Block Discrete Diffusion Models*				
LLaDA2.0-Uni	Block-causal; $n_{\text{txt}} + n_{\text{src}} = 1280$ prefix; generate $n_{\text{tgt}} = 1024$	32 blocks $\times$ 8 steps, CFG B3	256	1.31B
▼ *Training-Free In-Place Token Editing*				
DICE	$n_q = 1024$, $n_k = n_{\text{txt}} + n_{\text{src}} = 1280$	Inversion 63 + edit 62	125	<u>163.84M</u>
GIDE add/replace, no repair	$L = n_{\text{txt}} + n_{\text{src}} = 1280$; fully bidirectional	Inversion 65 + edit 64 + refine 1, CFG B2	130	425.98M
GIDE replace with source-minus-target repair	$L = 1280$; fully bidirectional	Inversion 65 + edit 64 + refine 1 + repair 64, CFG B2	194	635.70M
GIDE remove	$L = 1280$; fully bidirectional	Inpaint 64, CFG B2	64	209.72M
▼ *Discrete Diffusion Bridges*				
PaRDi	$L = n_{\text{txt}} + n_{\text{src}} = 1280$; fully bidirectional	30 updates + optional completion, B2	**30–31**	**98.30–101.58M**

D.6 COMPARISON OF EDITING TRAJECTORIES

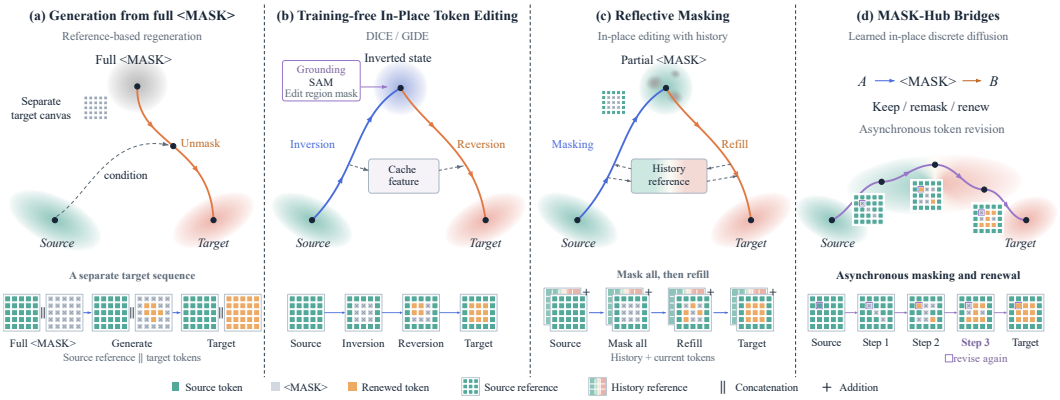

Figure 11: Extended comparison of image-editing trajectories. (a) Reference-based regeneration generates a separate target from <MASK>, using the source as a reference sequence. (b) Training-free in-place methods reuse source information: GIDE masks its grounded edit region before refilling it, whereas DICE follows a mask schedule and reuses inversion residuals. (c) Reflective Masking uses mask-all-then-refill editing with History Reference. (d) PaRDi revises the source canvas in place, allowing source, <MASK>, and renewed tokens to coexist. The purple outline follows a token as it is renewed and remasked.

E TRAINING AND SAMPLING ALGORITHMS

The training algorithms implement the paired-endpoint construction in Figure 2b; the sampling algorithms implement the prediction and revision loop in panel (c). Training uses observed targets to construct states and transition supervision, while inference uses predicted target probabilities in the same analytic transition map. Source and target token canvases are $\mathbf{s}, \mathbf{y} \in \mathcal{V}^N$, and c is the instruction. Time increases from target to source; sampling runs from T to 0 (discrete) or 1 to 0 (continuous). All token updates use the same frozen current canvas. With the deterministic-identity convention, $y_i = s_i$ gives a constant path, posterior δ_{s_i}, and zero outgoing rates. Otherwise use the stochastic bridge formulas, as specified by $q^\star$ and $\rho^\star$ in Equations (4) and (23). COMPATIBLE renormalizes the predictor over the positive-support endpoint set using Equation (25). All endpoint sums below run only over that set. The predictor receives the current canvas, instruction, and time. Classifier-free guidance (CFG), when used, modifies the predicted logits before the softmax and compatibility normalization. Each $\widehat{\mathcal{L}}$ below is a stochastic estimate of the training risk. DECODE maps the final token canvas to an image using the visual decoder; $\mathrm{Categorical}(P)$ draws a token with probabilities P.

E.1 DISCRETE-TIME TRAINING AND SAMPLING

Algorithm 1 Discrete PaRDi training with posterior matching

Require: Paired data; kernels $Q_{1:T}$; weights $w_{\mathrm{dyn}}, w_{\mathrm{CE}}$
while training **do**
  Sample $(\mathbf{s}, \mathbf{y}, c)$ and $t \sim \mathrm{Unif}\{1, \dots, T\}$.
  Compute the structured messages for each position and candidate endpoint.
  Sample $X_{t,i} \sim q_t^\star(\cdot \mid y_i, s_i)$ independently across positions.
  Predict $p_{\theta,i}(\cdot \mid \mathbf{X}_t, c, t)$ for all positions.
  for each token i, with $v = X_{t,i}$ **do**
    $\pi_{t,i} \leftarrow$ COMPATIBLE$(p_{\theta,i}, v, s_i, t)$.
    Form $P_{\theta,t,i}$ by Equation (29) over $\mathcal{C}_t(v; s_i)$.
    $d_{t,i} \leftarrow \mathrm{KL}(q^\star_{t-1|t}(\cdot \mid v, y_i, s_i) \| P_{\theta,t,i})$.
    $\ell_{\mathrm{CE},i} \leftarrow -\log p_{\theta,i}(y_i \mid \mathbf{X}_t, c, t)$.
  end for
  $\widehat{\mathcal{L}} \leftarrow N^{-1} \sum_i (w_{\mathrm{dyn}} T d_{t,i} + w_{\mathrm{CE}} \ell_{\mathrm{CE},i})$; update θ.
end while

Uniform timestep sampling uses T to estimate the sum in Equation (30). Structured message closure avoids dense $(V+1) \times (V+1)$ transition storage.

Algorithm 2 Discrete PaRDi inference with reverse-posterior sampling

Require: Source $\mathbf{s}$, instruction c, trained predictor, kernels $Q_{1:T}$
$\mathbf{X}_T \leftarrow \mathbf{s}$.
for $t = T, T-1, \dots, 1$ **do**
  Predict target-token logits from $(\mathbf{X}_t, c, t)$; apply CFG if enabled, then softmax.
  for each token i, with $v = X_{t,i}$ **do**
    $\pi_{t,i} \leftarrow$ COMPATIBLE$(p_{\theta,i}, v, s_i, t)$.
    Form $P_{\theta,t,i}$ by Equation (29) over $\mathcal{C}_t(v; s_i)$.
    Sample $X_{t-1,i} \sim \mathrm{Categorical}(P_{\theta,t,i})$.
  end for
  Commit all token updates simultaneously.
end for
return Decode$(\mathbf{X}_0)$.

The exact $t = 1$ posterior has a visible endpoint as its destination, so this full discrete schedule requires no additional MASK projection. For skipped steps, the reverse posterior uses the corresponding multi-step transition of the MASK-hub Markov process.

E.2 CONTINUOUS-TIME TRAINING AND SAMPLING

Algorithm 3 Continuous PaRDi training with reverse-rate matching

Require: Paired data; masking and reveal rates λ_τ, μ_τ; proposal r_s; loss weights
while training **do**
  Sample $(\mathbf{s}, \mathbf{y}, c)$ and $\tau \sim \mathrm{Unif}[\varepsilon_{\mathrm{tr}}, 1-\varepsilon_{\mathrm{tr}})$, $\varepsilon_{\mathrm{tr}} = 0.001$.
  Obtain forward/backward coefficients from the MASK-hub Markov process (Equations (18) and (19)).
  Sample $X_{\tau,i} \sim q^\star_\tau(\cdot \mid y_i, s_i)$ for each token.
  Predict $p_{\theta,i}(\cdot \mid \mathbf{X}_\tau, c, \tau)$ for all positions.
  for each token i, with $v = X_{\tau,i}$ **do**
    $\pi_{\tau,i} \leftarrow \textsc{Compatible}(p_{\theta,i}, v, s_i, \tau)$.
    $a_i(u) \leftarrow \rho^\star_\tau(v, u \mid y_i, s_i)$, $u \neq v$, with the identity rule.
    $b_i(u) \leftarrow \rho_{\theta,\tau,i}(v, u)$ from Equation (6).
    Evaluate $\ell_{\mathrm{rate},i} = \sum_{u \neq v} D_{\mathrm{rate}}(a_i(u), b_i(u))$; use MC8 at MASK.
    $\ell_{\mathrm{CE},i} \leftarrow -\log p_{\theta,i}(y_i \mid \mathbf{X}_\tau, c, \tau)$.
  end for
  $\widehat{\mathcal{L}} \leftarrow N^{-1} \sum_i (w_{\mathrm{dyn}} \ell_{\mathrm{rate},i} + w_{\mathrm{CE}} \ell_{\mathrm{CE},i})$; update θ.
end while

At a visible state, the rate sum has only the MASK destination. At MASK, the displayed dense sum specifies the objective; the implementation uses the exact target term and the uniform-proposal MC8 estimate in Equation (46). The time distribution and token average match Equation (8).

Algorithm 4 Continuous PaRDi inference with frozen-rate single-jump sampling

Require: Source $\mathbf{s}$, instruction c, predictor, generator R_τ, grid from Equation (51)
$\mathbf{X} \leftarrow \mathbf{s}$.
for $k = 0, \ldots, 29$ **do**
  $\tau \leftarrow \tau_k$; $\Delta \leftarrow \tau_k - \tau_{k+1}$.
  Predict target-token logits from $(\mathbf{X}, c, \tau)$; apply CFG if enabled, then softmax.
  Form compatible beliefs $\pi_{\tau,i}$ and rates $b_i(u) = \rho_{\theta,\tau,i}(X_i, u)$ via Equation (6).
  Evaluate boundary rates by their one-sided limits.
  Freeze $\mathbf{X}$ and all rates.
  for each token i **do**
    $\Lambda_i \leftarrow \sum_{u \neq X_i} b_i(u)$; $X'_i \leftarrow X_i$.
    if $\Lambda_i > 0$ and $U_i < 1 - \exp(-\Lambda_i \Delta)$, where $U_i \sim \mathrm{Unif}(0,1)$ **then**
      Sample $X'_i \sim \mathrm{Categorical}(\{b_i(u)/\Lambda_i : u \neq X_i\})$.
    end if
  end for
  $\mathbf{X} \leftarrow \mathbf{X}'$.
end for
if MASK tokens remain **then**
  Recompute CFG logits $\ell_i^{\mathrm{CFG}}(v)$ on $(\mathbf{X}, c, \varepsilon)$.
  Fill each MASK at i with $\arg\max_{v \in \mathcal{V} \setminus \{s_i\}} \ell_i^{\mathrm{CFG}}(v)$.
end if
return Decode$(\mathbf{X})$.

Here Λ_i is the total outgoing rate and Δ is the decrement in τ. Each leap permits at most one jump per token; it approximates the CTMC, which may make multiple jumps over a finite interval. The conditional endpoint projection leaves visible tokens unchanged; Section C.5 specifies guidance and NFE counting.

F VISUALIZATIONS OF ACTUAL SAMPLING TRAJECTORIES

Figures 12 to 14 visualize actual PaRDi sampling trajectories at six selected times. Snapshots are ordered left to right and top to bottom, with displayed times $\tau = 0.833, 0.667, 0.500, 0.333, 0.167, 0.000$ rounded to three decimal places. Each snapshot pairs the decoded canvas with a revision overlay: green marks keep, red remask, and blue renew. The top row shows the source, instruction, final output, and a source–output token-difference map, with retained tokens in cyan and changed tokens in yellow. These snapshots illustrate the evolution of the canvas during the sampling process described in Section 3.5 and Algorithm 4.



Figure 12: Actual sampling trajectory for replacing the goat with a horse. Six selected τ snapshots show the decoded canvas and keep/remask/renew overlays as the requested replacement develops alongside the neighboring cat. The top row provides the source and final output for comparison.

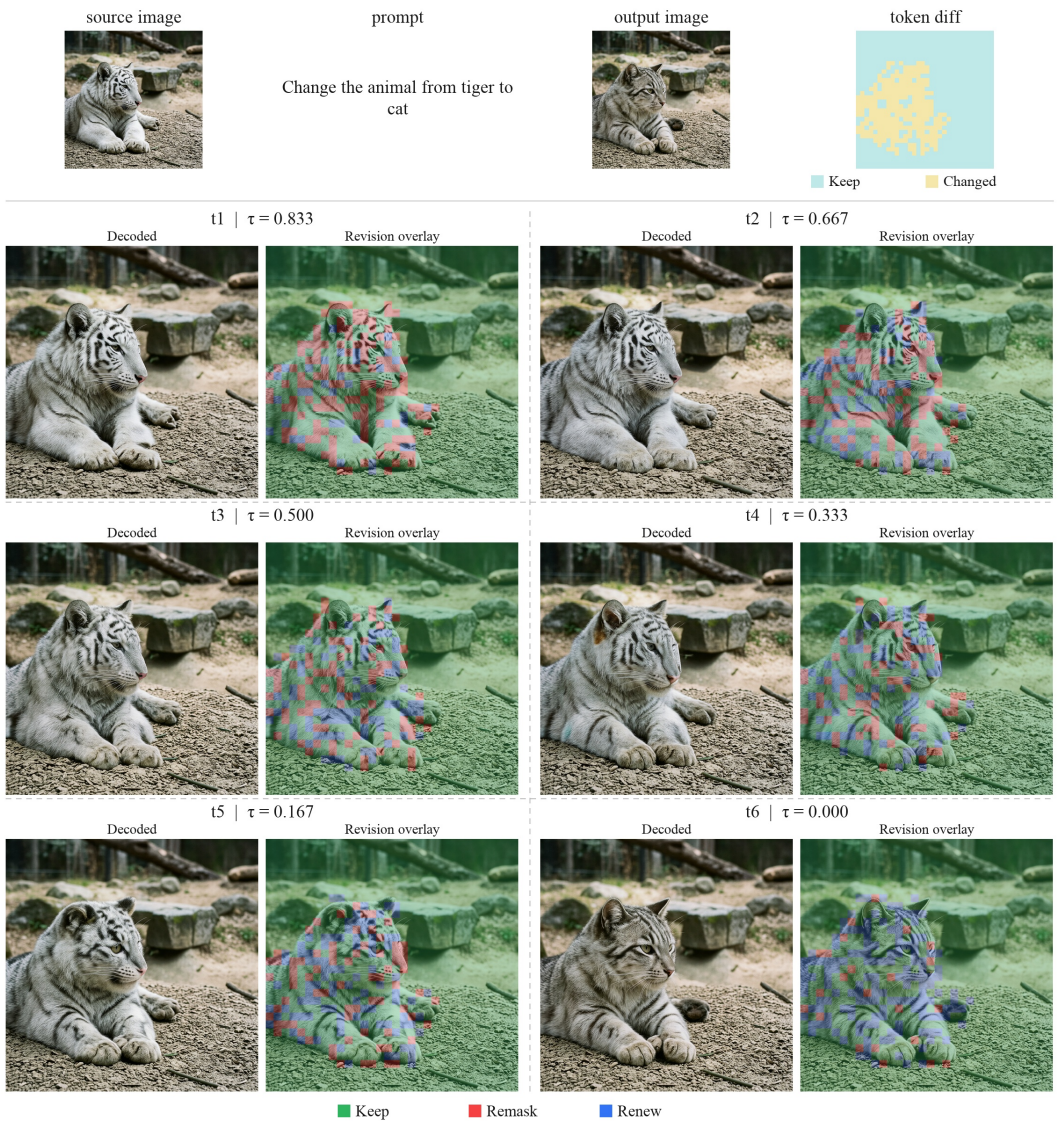

Figure 13: Actual sampling trajectory for changing the animal from a tiger to a cat. The six snapshots follow decreasing τ from left to right and top to bottom, at the times specified in Section F. Decoded canvases are paired with revision overlays; the top row shows the source, instruction, final output, and token differences.



Figure 14: Actual sampling trajectory for changing the figure's standing surface to a cloud. Six selected τ snapshots show how the supporting surface changes during sampling, together with keep/remask/renew overlays. Snapshots follow the same order and displayed times as in Figure 12.