Concurrent Image Understanding and Generation: Self-Correcting Coupled Markov Jump Processes

Minh-Quan Le[1,3†] **Armand Comas**[1] **Alexandros Lattas**[1]
Stylianos Moschoglou[1] **Pedro Vélez**[2] **Amit Raj**[2] **Aaron Germuth**[1]
Thabo Beeler[1] **Dimitris Samaras**[3] **Di Qiu**[1]

[1]Google [2]Google DeepMind [3]Stony Brook University

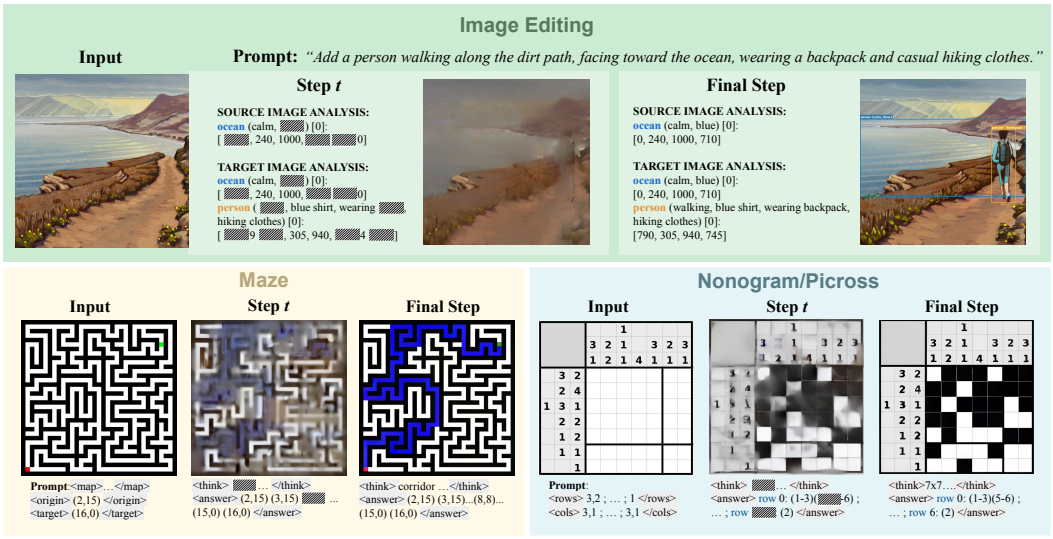


Figure 1: $\texttt{CO}_2\texttt{Jump}$ **in action: text and image co-author the answer.** Three trajectories of $\texttt{CO}_2\texttt{Jump}$ on image editing, maze, and nonogram solving, showing the joint state at an intermediate step t and at the final step. The image-editing panel highlights the core mechanism: at step t the text branch has already begun committing a *target*-image bounding box in text for the new object *person*; by the final step the image branch has placed the hiker *exactly* inside the finalized box (we overlay the bounding boxes from generated text on edited image). The text branch *plans* where the edit should land, and the image branch *executes* that plan within the same denoising trajectory — no second forward pass, no external grounder. Maze and Nonogram show the same coupled-refinement pattern: partial-path and partial cell-fill commitments at step t converge with their text-side answers by the final step.

Abstract

Human cognition does not separate understanding and generation. A teacher at a whiteboard speaks and draws *together*, each modality reshaping the other. In this paper, we bring this coupled loop to artificial systems. Masked Diffusion Models (MDMs) are ideally suited to this task, yet existing samplers either interleave text and image decoding or update them independently in parallel branches that share only previous-step history, but not the other modality's latest decisions *within* the same step; combined with MDMs' inability to remask, cross-modal contradictions are neither detected nor repaired. We introduce **Self-Correcting Coupled Markov Jump Processes (SC-CMJP)**, a framework in which one modality's transition rates are functionals of the other modality's confidence score, as weighted by

†This work was done while Minh-Quan was a Student Researcher at Google.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

cross-modal attention. Furthermore, a remasking jump retracts commitments the moment cross-modal evidence turns against them. In conjunction with SC-CMJP, we introduce $\texttt{CO}_2\texttt{Jump}$ (Self-COrrecting COupled Jump), a novel training-free single-pass sampler for joint multimodal generation. For training and evaluation purposes, we have created and will release three large-scale joint multimodal generation corpora: JEDIT-1M, JMAZE-200K, JNONO-200K, with matching in- and out-of-distribution benchmarks. $\texttt{CO}_2\texttt{Jump}$ achieves best joint performance for image understanding and editing as well as visual reasoning (maze and nonogram solving). The performance of the sampler scales monotonically with the number of denoising steps, evidence that the benefits of cross-modal coupling *compound* across the trajectory. Project page: https://coupled-jump.github.io

1 Introduction

When a teacher explains an idea at a whiteboard, language and drawing take place *together*: each utterance affects the sketch, and each new mark affects the next sentence. Understanding and generation are not separate stages but a tightly coupled loop, with each modality continuously informing and revising the other as the explanation unfolds. We aim to operationalize this loop for artificial systems, producing text and image content concurrently rather than sequentially, with one modality shaping the other *as* it is generated. While our method is modality-agnostic in principle, in this paper we focus on pairing text for understanding, with images for generation.

Masked Diffusion Models (MDMs) [2, 29, 23, 30] are well-suited for parallel multimodal generation. Unlike autoregressive sequential pipelines [10, 8, 36], which first decode the entire textual reasoning trace and then condition image synthesis on it – a unidirectional flow that cannot retract early reasoning errors – MDMs predict all masked tokens simultaneously, admit a clean continuous-time formulation as Markov Jump Processes [5, 4], and scale naturally to multiple modalities under a unified vocabulary spanning text and image tokens [37, 38]. Their parallel structure makes them, in principle, a natural framework for joint text/image generation in a single decoding loop.

In practice, existing MDM samplers [29, 34, 31, 7, 26] fall short of *true* concurrent joint multimodal generation. Even samplers that nominally decode both modalities in parallel [7, 31] factorize each denoising step so that the text and image updates depend only on the previous joint state and not *within* the same step; concurrency reduces to interleaving over a shared history. The resulting trajectories might drift: text might commit to descriptions the image cannot illustrate, the image might render content the text never described. Compounding this, standard masked diffusion is unable to remask [2, 5, 29], once a token is committed it cannot be revised. Cross-modal contradictions introduced by an uncoupled parallel decoder persist for the rest of sampling.

We address both problems jointly. We introduce **Self-Correcting Coupled Markov Jump Processes (SC-CMJP)**, a general framework for concurrent joint multimodal generation in which the two modalities actively cross-weigh their commitments *within* every denoising step. One modality's transition rates become functionals of the other modality's unmasking confidence score, weighted by cross-modal attention extracted from the same backbone forward pass. The unmasking schedule of one modality adapts to the confidence of the other modality. Combined with a remasking jump in the spirit of ReMDM [34], this lifts decoding from one-way unmasking to a bidirectional birth-death (unmask-remask) process that can both reveal new tokens and retract earlier ones whenever cross-modal evidence turns against them.

Along with SC-CMJP, we design a single-pass training-free sampler $\texttt{CO}_2\texttt{Jump}$ (Self-COrrecting COupled Jump) that highlights two core ideas: *Coupling* the per-modality transition rates through cross-modal attention, and *Correcting* earlier commitments via a remasking jump. $\texttt{CO}_2\texttt{Jump}$ runs on a frozen MDM with no architectural change, no auxiliary evaluator, and a single forward pass per step. Figure 1 illustrates $\texttt{CO}_2\texttt{Jump}$ on all three of our benchmarks; in the image-editing trajectory, the text branch commits a bounding box for the inserted object and the image branch fills it in. In the *same* step, the text *plans* where the edit should land, the image branch *executes* that plan.

To validate $\texttt{CO}_2\texttt{Jump}$, we instantiate it on three concurrent text-and-image tasks of increasing semantic difficulty: image editing on an extended ImgEditBench [40] protocol with mAP-style [22] grounded-understanding metrics, and two new visual-reasoning tasks, a maze and a nonogram (JMAZE and JNONO) where text and image are logically interlocked and jointly verifiable against algorithmic

ground truth. To enable training and evaluation, we curate three corpora: JEDIT-1M, JMAZE-200K, and JNONO-200K, all of which we plan to release. Across all three tasks, $\texttt{CO}_2\texttt{Jump}$ consistently improves both modalities, beats existing sampling methods [29, 34, 31] on concurrent joint image understanding and generation, especially joint performance metrics. In addition, $\texttt{CO}_2\texttt{Jump}$ sampler's performance scales monotonically with the number of denoising steps. In summary, our main contributions are:

- We introduce the first approach to model simultaneous image understanding and generation as a single, unified stochastic process. To achieve this, we propose Self-Correcting Coupled Markov Jump Processes, that integrate parallel joint multimodal generation with built-in self-correction.
- We design a novel coupled sampler $\texttt{CO}_2\texttt{Jump}$ for joint multimodal sampling, running in a single forward pass per step.
- We curated three large-scale joint-generation corpora (JEDIT-1M, JMAZE-200K, JNONO-200K) along with matching benchmarks that probe both in-distribution and out-of-distribution performance. We have a plan to release model checkpoints, code, datasets to community.
- $\texttt{CO}_2\texttt{Jump}$ improves the state-of-the-art on concurrent image understanding and editing, as well as visual reasoning tasks against existing sampling methods. Our sampler's performance scales monotonically with the number of denoising steps – empirical evidence that the benefits of cross-modal coupling *compound* across the trajectory.

2 Related Work

Masked diffusion. Absorbing-state discrete diffusion was introduced in D3PM [2] and refined into score-entropy [23], simplified-ELBO [29, 30], and any-order autoregressive [25] formulations. Scaling to LLMs has been driven by LLaDA [24], Dream [39], and SDAR [9]. Decoding improvements include block-wise generation [1], Top-K confidence selection [24, 20], and length-adaptive scheduling [25]. Multimodal extensions tokenize images with VQ-VAE [32] and operate over a unified vocabulary spanning text and image tokens, e.g. Lumina-DiMOO [37] and MMaDA [38].

Self-correction via remasking. Three families of method address the failure-to-remask problem. *Predictor-corrector samplers* [5, 21, 11, 6] reduce τ-leaping error via corrector steps without an explicit remask jump. *Training-based remasking* modifies the model: GIDD [33] generalizes forward and reverse processes; Zhao et al. [41] train a separate hollow-transformer evaluator; Kim et al. [19], Huang et al. [17] fine-tune the pretrained MDM to estimate per-token quality. *Training-free remasking* keeps the backbone frozen: ReMDM [34] adds a heuristic σ_t schedule, Peng et al. [27] explore path-following corrections, and Ouyang et al. [26] use cumulative-confidence signals. $\texttt{CO}_2\texttt{Jump}$ sits within the training-free family, but is the first to couple the remasking signal across modalities and treat cross-modal contradiction as the self-correction trigger.

Coupled MJPs for Multimodal Generation. A Markov Jump Process (MJP) [5, 4] is a continuous-time Markov process on a discrete state space specified by a rate matrix $R_t(z, z')$; the Continuous-Time Markov Chain (CTMC) view [2, 5, 29, 26] of MDM is exactly such an MJP over $\mathcal{V}^L$ with rate matrix in Eq. (10). Exact reverse simulation via Gillespie's algorithm [12, 13] updates one position per jump and is prohibitive at modern sequence lengths, so practical samplers use τ-leaping [14] – a parallel approximation that updates all positions simultaneously, of which the standard MDM reverse step is the first-order discretization [5].

Coupled jump processes are classical in chemistry [13] and Glauber dynamics [15], but their coupling structure is hand-specified and they target physical simulation; standard MJP-based diffusion likewise treats per-position rates as conditionally independent given $\mathbf{x}_\theta(\mathbf{z}_t)$ even in multimodal settings. We define a *Coupled Markov Jump Process* (CMJP) over $\mathbf{z}_t = (\mathbf{z}_t^{\texttt{text}}, \mathbf{z}_t^{\texttt{image}})$ for joint multimodal generation. Combined with a ReMDM-style σ_t jump (Eq. (12)), birth (unmask) and death (remask) rates of one modality are informed by the current commitments of the other, enabling localized cross-modal self-correction at sampling time.

Concurrent multimodal samplers. UD-VLA [7] factorizes the joint into independent per-modality terms (an instance of MDM [29] sampling); MMaDA-Parallel [31] interleaves text and image updates across steps but samples them *independently within each step* – both modality updates condition

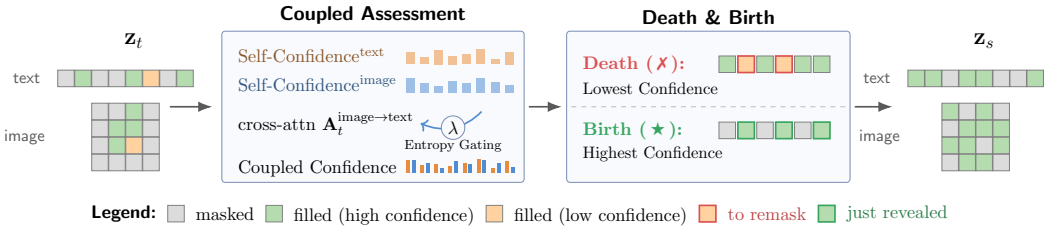


Figure 2: $\texttt{CO}_2\texttt{Jump}$ **sampler.** A single denoising step from $\mathbf{z}_t$ to $\mathbf{z}_s$. From one forward pass, the model produces per-token Self-Confidence for both modalities; cross-modal attention $\mathbf{A}_t^{\text{image}\to\text{text}}$ propagates text confidence to image positions, and an entropy-based gate λ mixes self and cross signals into Coupled Confidence. The *Death* jump remasks the lowest-confidence committed tokens, and the *Birth* jump reveals the highest-confidence masked tokens under the noise schedule.

only on the previous joint state, with no cross-modal feedback inside the step itself, so any coupling is realized only through shared history rather than instantaneous negotiation. Our experiments compare $\texttt{CO}_2\texttt{Jump}$ against three representative samplers covering these regimes: MDM [29], MMaDA-Parallel [31], and ReMDM [34]. Coupled jump processes in chemistry and statistical physics [13, 15] use hand-specified couplings for physical simulation, neither method transfers to our setting.

3 Self-Correcting Coupled Markov Jump Processes

Standard multimodal diffusion samplers often suffer from *modality-drift*, where the generated text and image diverge because their unmasking schedules are independent and irreversible. We address this by reformulating joint generation as a unified system where modalities actively negotiate their commitment through a shared birth-death jump process whose intensities are coupled across modalities. Figure 2 illustrates a single denoising step of the resulting sampler.

3.1 Continuous-Time Likelihood Bounds for Joint Generation

Following MDM derivation [29], for $t \in [0,1]$ and a monotonically decreasing noise schedule $\alpha_t \in [0,1]$, the per-position marginal of the forward process is

$$q(\mathbf{z}_t \mid \mathbf{x}) = \mathrm{Cat}\big(\mathbf{z}_t;\ \alpha_t\,\mathbf{x} + (1-\alpha_t)\,\boldsymbol{m}\big), \tag{1}$$

factorized across positions. We treat text and image as a single sequence over the union vocabulary $\mathcal{V} = \mathcal{V}_{\texttt{text}} \cup \mathcal{V}_{\texttt{image}}$ and share a single absorbing token $\boldsymbol{m}$ across both modalities, so that the per-position marginal in Eq. (1) applies uniformly to text and image positions; this is what allows the joint NELBO in Eq. (2) below to factor cleanly without a cross-modality term. A joint clean sample is $\mathbf{x} = (\mathbf{x}^{\texttt{text}}, \mathbf{x}^{\texttt{image}}) \in \mathcal{V}^L$ with total sequence length $L = L^{\texttt{text}} + L^{\texttt{image}}$, and is corrupted by $q(\mathbf{z}_t \mid \mathbf{x})$, applied position-wise across both modalities under a shared schedule α_t.

Because the per-position marginal factorizes and both modalities use the same α_t, the joint NELBO inherits the similar form in MDM [29] with no cross-modality term in the objective:

$$\mathcal{L}_{\text{NELBO}}^{\infty} \;=\; \mathbb{E}_{q,\,t} \int_0^1 \frac{\dot{\alpha}_t}{1-\alpha_t} \sum_{\ell=1}^{L} \log \big\langle \mathbf{x}_\theta^\ell(\mathbf{z}_t, t),\ \mathbf{x}^\ell \big\rangle \,\mathrm{d}t. \tag{2}$$

A crucial property is implicit in the conditioning: the network output $\mathbf{x}_\theta^\ell(\mathbf{z}_t, t)$ at any position ℓ has access to the *entire* joint state $\mathbf{z}_t = (\mathbf{z}_t^{\texttt{text}}, \mathbf{z}_t^{\texttt{image}})$, including the complementary modality. Training under Eq. (2) therefore implicitly forces $\mathbf{x}_\theta$ to learn cross-modal denoising signals – to reconstruct a masked image patch from surrounding pixels *and* concurrent textual clues, and conversely to reconstruct a masked text token from surrounding context *and* the partially decoded image. The NELBO never references this dependency explicitly, yet at convergence it is encoded in the network's hidden representations and attention patterns. Our coupled sampler $\texttt{CO}_2\texttt{Jump}$ extracts these latent cross-modal signals at inference time via a single forward pass through $\mathbf{x}_\theta$, with no architectural changes to the trained model and no auxiliary cross-modal evaluator.

3.2 Asymmetric Sequential Sampling via Chain-Rule Decomposition

Let c denote the task conditioning context (e.g., the editing prompt together with the source image for image editing, or the puzzle specification for visual reasoning). A naive sampler for the joint reverse posterior $p_\theta(\mathbf{z}_s^{\texttt{text}}, \mathbf{z}_s^{\texttt{image}} \mid \mathbf{z}_t, c)$ updates both modalities independently from the same $\mathbf{z}_t$, ignoring that the latest text decision $\mathbf{z}_s^{\texttt{text}}$ provides immediate context for inferring the corresponding image state $\mathbf{z}_s^{\texttt{image}}$. We instead exploit the chain-rule factorization

$$p(\mathbf{z}_s^{\texttt{text}}, \mathbf{z}_s^{\texttt{image}} \mid \mathbf{z}_t, c) \;=\; \underbrace{p(\mathbf{z}_s^{\texttt{text}} \mid \mathbf{z}_t, c)}_{\text{(I) text update}} \cdot \underbrace{p(\mathbf{z}_s^{\texttt{image}} \mid \mathbf{z}_t, \mathbf{z}_s^{\texttt{text}}, c)}_{\text{(II) image update conditioned on text at step } s}, \tag{3}$$

which makes explicit that the text update needs no information beyond the current $\mathbf{z}_t$, while the image update would ideally condition on the latest sampled $\mathbf{z}_s^{\texttt{text}}$.

Evaluating Term (II) exactly demands a second forward pass through $\mathbf{x}_\theta$ at every diffusion step – prohibitive at inference time. We avoid this cost by approximating $\mathbf{z}_s^{\texttt{text}}$ with quantities already produced by the single forward pass at $\mathbf{z}_t$: the image side reads the model's self-belief over $\mathbf{z}_t^{\texttt{text}}$, propagates it through cross-modal attention, and uses the result as a low-cost surrogate for the unavailable $\mathbf{z}_s^{\texttt{text}}$. This yields an *asymmetric scoring rule*: text positions are scored by their pure self-confidence, while image positions are scored by an entropy-gated mixture of self-confidence and a cross-modal signal – the *Coupled Confidence*. Both scores then drive the death-and-birth dynamics.

3.3 Self-Confidence and Coupled Confidence

Self-Belief via Gumbel-Max. At each timestep, the network produces beliefs $p_\theta(\mathbf{x}^\ell \mid \mathbf{z}_t, c)$ over each position's clean token. To escape local minima, we apply Gumbel-max sampling, $\hat{\mathbf{x}}^\ell = \arg\max_{v \in \mathcal{V}} \big(\log p_\theta(v \mid \mathbf{z}_t, c) + \gamma_v\big)$ with i.i.d. Gumbel noise γ_v. The *Self-Confidence* of position ℓ is the model's probability for its sampled choice: $\texttt{SelfConf}_\ell = p_\theta(\hat{\mathbf{x}}^\ell \mid \mathbf{z}_t, c)$.

Text Scoring. Because Term (I) of Eq. (3) has no cross-modal dependency, the per-position text score is the self-confidence directly: $\texttt{Score}_{\texttt{text},\ell} \;=\; \texttt{SelfConf}_{\texttt{text},\ell}$. (4)

Cross-Modal Negotiation. For Term (II), we approximate the conditioning on $\mathbf{z}_s^{\texttt{text}}$ using $\mathbf{z}_t^{\texttt{text}}$ via cross-attention. Let $\mathbf{H}_t^{\texttt{image}} \in \mathbb{R}^{L^{\texttt{image}} \times D}$ and $\mathbf{H}_t^{\texttt{text}} \in \mathbb{R}^{L^{\texttt{text}} \times D}$ be the hidden representations of the two modalities, extracted post-hoc from the model's final layer during the same forward pass that produced $\texttt{SelfConf}$ – no architectural modification or auxiliary attention head is introduced. We compute the image→text cross-attention

$$\mathbf{A}_t^{\texttt{image}\to\texttt{text}} \;=\; \texttt{Softmax}\left(\frac{\mathbf{H}_t^{\texttt{image}} (\mathbf{H}_t^{\texttt{text}})^\top}{\sqrt{D}} + \mathbf{B}_t\right), \tag{5}$$

where $\mathbf{B}_t$ is a mask-aware bias that downweights attention toward currently masked positions. The cross-modal negotiation is implemented as the cross signal for image position ℓ, which is the attention-weighted text self-confidence:

$$\texttt{CrossSignal}_\ell \;=\; \sum_{j=1}^{L^{\texttt{text}}} \mathbf{A}_{t,\ell j}^{\texttt{image}\to\texttt{text}} \cdot \texttt{SelfConf}_{\texttt{text},j}. \tag{6}$$

Dynamic Trust through Entropy-Based Gating. The weight given to the cross signal versus the self signal on the image side is a *single scalar gate* $\lambda \in [0, 1]$, computed once per denoising step and applied uniformly to every image position. We derive λ from per-modality predictive uncertainty: letting $\bar{\mathcal{H}}_a \;=\; \frac{1}{L^a} \sum_{\ell=1}^{L^a} \big[-\sum_v p_\theta(v \mid \mathbf{z}_t, c)_{a,\ell} \log p_\theta(v \mid \mathbf{z}_t, c)_{a,\ell}\big]$ denote the mean token-level Shannon entropy in modality a, averaged over *all* positions of that modality,

$$\lambda \;=\; \frac{\bar{\mathcal{H}}_{\texttt{image}}}{\bar{\mathcal{H}}_{\texttt{image}} + \bar{\mathcal{H}}_{\texttt{text}} + \epsilon}. \tag{7}$$

When the image is locally chaotic (high $\bar{\mathcal{H}}_{\texttt{image}}$), $\lambda \to 1$ and the image modality defers to the text for evidentiary support; when the text is the uncertain side, $\lambda \to 0$ and the image relies on its own self-belief. The single-gate formulation follows directly from Eq. (3) – only the image term carries a cross-modal dependency to be resolved.

Shared Percentile Rank Space. Because the text and image vocabularies differ in size ($|\mathcal{V}_{\texttt{text}}| \gg |\mathcal{V}_{\texttt{image}}|$), absolute confidence values are not directly comparable across modalities. We therefore project each scalar score into its empirical percentile rank within its own modality, ensuring the image-side mixture below is well-conditioned regardless of vocabulary scale.

Coupled Confidence for Image Scoring. Combining the rank-normalized self and cross signals:

$$\texttt{Score}_{\texttt{image},\ell} \;=\; \texttt{CoupledConf}_{\ell} \;=\; (1-\lambda)\cdot \text{Rank}(\texttt{SelfConf}_{\texttt{image},\ell}) + \lambda \cdot \text{Rank}(\texttt{CrossSignal}_{\ell}). \quad (8)$$

3.4 The $\texttt{CO}_2\texttt{Jump}$ Sampler

We interpret the joint reverse generation as a modality-specific *birth-death* jump process. Standard masked diffusion only "births" tokens (unmask) and never updates them. $\texttt{CO}_2\texttt{Jump}$ additionally allows tokens of either modality to "die" (remask) when their scores fall below the schedule-driven death rate, allowing both intra-modal errors and cross-modal contradictions to be retracted.

Death Step (Remasking). At each discretization step $(t \to s)$ we set the death rate σ_t to satisfy the valid-posterior constraint $\sigma_t \le \min(\eta, (1-\alpha_s)/\alpha_t)$. For each modality a, the idealized unmasked count is $U_t^a = \alpha_t L^a$ and the idealized masked count is $M_t^a = (1-\alpha_t)L^a$ (we use $\lfloor\cdot\rfloor$ to obtain integer token counts). The death quota is $N^a_{\text{remask}} = \lfloor U_t^a \cdot \sigma_t \rfloor$, and the N^a_{remask} unmasked tokens with the *lowest* $\texttt{Score}_a$ are reverted to absorbing token $\boldsymbol{m}$. Per the asymmetric rule (Eq. (3)), remasking uses $\texttt{SelfConf}$ for text (Eq. (4)) and $\texttt{CoupledConf}$ for image (Eq. (8)).

Birth Step (Unmasking). To keep the global signal-to-noise progression matched to the schedule, the birth quota $N^a_{\text{unmask}} = \lfloor \frac{\alpha_s - \alpha_t}{1-\alpha_t} \cdot M_t^a \rfloor + \Delta N^a$ combines the base unmask count with an extra-unmask compensation $\Delta N^a = \lfloor \sigma_t \cdot \frac{\alpha_t}{1-\alpha_t} \cdot M_t^a \rfloor$ derived from the ReMDM [34] posterior. Among the post-remask masked tokens of modality a, we reveal the N^a_{unmask} with the *highest* $\texttt{Score}_a$, again with the asymmetric definition. Algorithm 1 in the Appendix gives the full pseudocode.

4 Datasets and Benchmarks for Joint Multimodal Generation

To train and evaluate $\texttt{CO}_2\texttt{Jump}$ across both photographic and logical multimodal tasks, we curate three joint-generation datasets: JEDIT-1M for image editing, JMAZE-200K for maze solving, and JNONO-200K for nonogram solving, each paired with a held-out benchmark. All three corpora share the same record schema (prompt, source image, target image, structured understanding, thinking trace); Figure 3 shows the curation pipeline.

4.1 JEDIT-1M: Joint Image Editing and Understanding

Training corpus. JEDIT-1M combines a 724k-pair subset of ImgEdit [40] with a 368k-pair subset of OmniEdit [35], yielding 1M (prompt, source, target) tuples. Because raw editing data lacks the joint understanding and thinking supervision that joint multimodal generation requires, we augment each pair with two fields synthesized by Qwen3-VL-235B [3]: (i) a pixel-aligned SOURCE/TARGET scene-graph extraction with per-category indices preserved across panels, and (ii) a logic-based thinking trace conditioned on both images, the prompt, and the scene graph from (i).

Benchmark and metrics. We extend ImgEditBench [40] with two metric families that score joint understanding + generation behavior. **Text quality** is generative perplexity under a frozen GPT-2 Large [28] plus token-distribution entropy as a mode-collapse safeguard, following MDLM [29] and ReMDM [34]. **Image-edit fidelity** is the standard ImgEditBench oracle score from Gemini 3 Flash [16]. **Image-grounded understanding** is COCO-style mAP@0.5:0.95 [22] against a *pseudo* scene graph constructed per-sample by Gemini on the model's own (source, generated-target) pair: pseudo grounding is necessary because each model produces a different target image. We report mAP separately for source and target sections plus their average.

4.2 JMAZE-200K and JNONO-200K: Visual Reasoning

To stress-test joint generation on tasks where text and image are *logically interlocked*, we add two synthetic visual-reasoning corpora. In both, the input fully specifies a unique solution and the two modalities are independently verifiable against algorithmic ground truth.

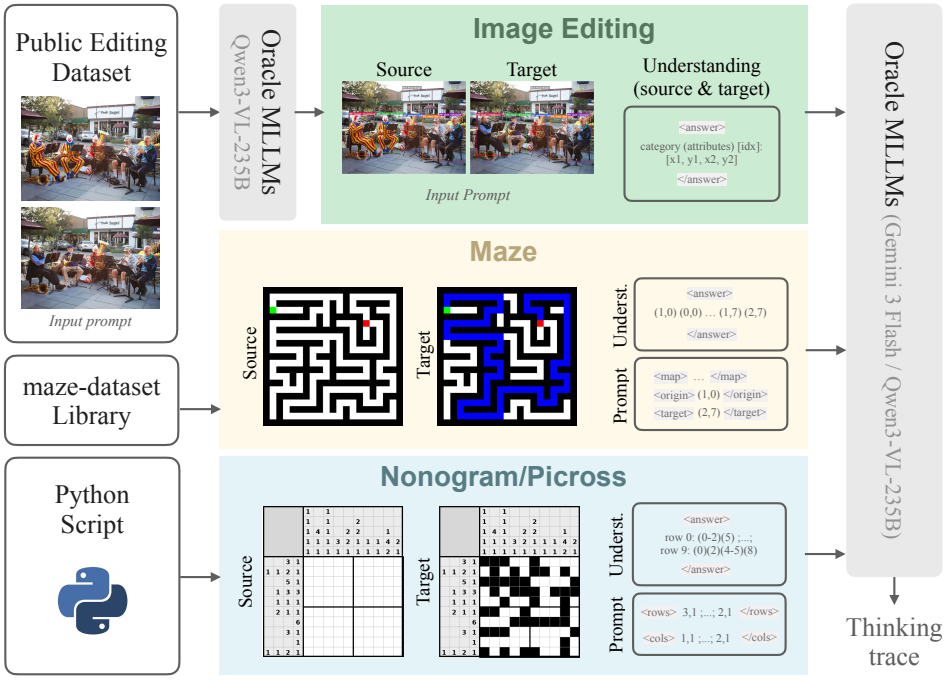


Figure 3: **Dataset curation pipeline.** For JEDIT-1M, raw editing pairs are augmented by an oracle Qwen3-VL-235B that produces both the per-image scene-graph understanding and the thinking trace. For JMAZE-200K and JNONO-200K, source/target images and the structured understanding are produced algorithmically.

Corpora. We use the `maze-dataset` [18] library to generate JMAZE-200K comprising 200k DFS-perfect mazes with grid sizes uniformly sampled from $\{6, \ldots, 20\}$; the input shows walls plus a green start and red end, the output overlays the solution path in blue, and the text answer is the path as an (r, c) sequence. JNONO-200K is 200k nonogram puzzles with grid sizes $\{5, \ldots, 25\}$, generated by a mixture of pattern synthesizers biased toward multi-run row/column clues; the input renders the empty $N \times N$ grid with marginal clue numbers, and the output fills cells black to satisfy every clue. For both corpora, the thinking trace is synthesized by an oracle Gemini 3 Flash [16] conditioned on the source/target images and the algorithmic ground-truth answer.

Parallel-form supervision. Both the structured understanding and the thinking trace are written in a form native to parallel decoding rather than autoregressive ordering: maze solutions use global (r, c) coordinates instead of relative moves (*Left/Right/Up/Down*), and nonogram supervision emphasizes bidirectional row/column constraint propagation instead of row-by-row solving. A relative direction is only well-defined once the previous coordinate is committed, while a global coordinate keeps its meaning regardless of which other tokens have been unmasked, aligning the supervision with the parallel denoising the sampler actually performs.

Benchmarks and metrics. For each task we hold out a 500-sample test set whose grid sizes *exceed* the training range on both ends: JMAZE-TEST500 spans $\{3, \ldots, 22\}$ (in-dist $\{6, \ldots, 20\}$, OOD $\{3, 4, 5, 21, 22\}$); JNONO-TEST500 spans $\{3, \ldots, 27\}$ (in-dist $\{5, \ldots, 25\}$, OOD $\{3, 4, 26, 27\}$). Our headline metric is **joint accuracy**, a sample counts correct only when both the text answer (judged by a tolerant Gemini extractor against the algorithmic ground truth) and the generated image (compared side-by-side with the ground-truth output by Gemini 3 Flash) are correct. **Text accuracy** and **image accuracy** are reported separately as diagnostics; on in-distribution and OOD splits.

5 Experiments

We evaluate $\texttt{CO}_2\texttt{Jump}$ against three sampler families across the three tasks of Section 4. To ensure a strictly controlled comparison: we initialize from Lumina-DiMOO [37], fine-tune on the corresponding corpus, and swap only the inference-time sampler. For each task, we fine-tune the model on $64\times$ H100 80 GB GPUs with a total batch size of 512 and learning rate 2×10^{-5}. Baselines include:

Table 1: **Image-editing results on the extended ImgEditBench protocol.** Bottom block: apple-to-apple sampler comparison; above the rule: pre-trained Lumina-DiMOO and Qwen3-VL-8B references. **Bold**/underline = best/second best across the four fine-tuned samplers.

		Text Generation		Image Generation	Image Understanding (mAP@0.5:0.95 ↑)		
Method	# turns ↓	GPT-2L PPL ↓	Entropy ↑	ImgEditBench ↑	Source	Target	Overall
Qwen3-VL-8B	2 (I2I → I2T)	11.9128	4.7523	–	0.391	0.330	0.360
Lumina-DiMOO	3 (I2I + 2×I2T)	3.1647	3.0513	1.68	0.019	0.019	0.019
MDM	1	8.5780	<u>4.5160</u>	1.78	0.386	0.322	0.354
ReMDM	1	**8.4863**	4.4929	1.73	0.388	0.318	0.353
MMaDA-Parallel	1	8.8363	**4.5353**	1.44	0.366	0.304	0.335
$\texttt{CO}_2\texttt{Jump}$ **(Ours)**	1	<u>8.5649</u>	4.5005	**1.93**	**0.392**	**0.346**	**0.369**

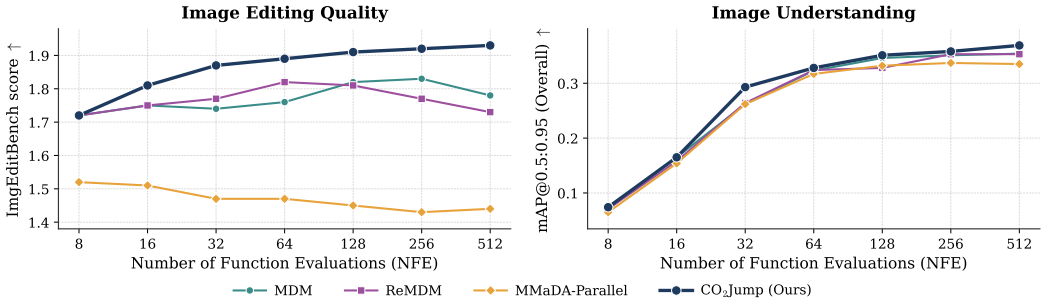


Figure 4: **Scaling sampling steps.** ImgEditBench (left) and overall mAP (right) vs. NFE. $\texttt{CO}_2\texttt{Jump}$ is the only sampler with monotonically rising curves; the gap to baselines widens with NFE.

MDM [29], independent modalities with no remasking; **ReMDM** [34], independent modalities with remasking; and **MMaDA-Parallel** [31], modalities interleave and are independent with no remasking. $\texttt{CO}_2\texttt{Jump}$ is the only entry that generates joint modalities concurrently with cross-modal coupling and self-correction. All samplers use a cosine α_t schedule; for the remasking variants (ReMDM and $\texttt{CO}_2\texttt{Jump}$), we use a remasking schedule with $\sigma_t = 0.01$ on $t \in [0.25, 0.75]$ and 0 elsewhere.

5.1 Joint Image Editing and Understanding

Table 1 reports results on extended ImgEditBench. Our headline metric is per-section mAP@0.5:0.95 from the pseudo-grounding pipeline; target-mAP is the cleanest test of cross-modal coupling because it requires the image branch to place each edited object inside the bounding boxes that the text branch declares *and* the text branch to describe what the image branch actually produces.

Coupled generation beats a strong sequential grounder. Qwen3-VL-8B is a stringent reference: same scale as our backbone, and the supervision used to fine-tune all four samplers comes from the Qwen3-VL family. Because Qwen3-VL-8B itself cannot generate images, we evaluate its grounding on the same (source, generated-target) image pairs produced by our $\texttt{CO}_2\texttt{Jump}$ sampler, effectively giving it the generated target image as input for free. Yet $\texttt{CO}_2\texttt{Jump}$ produces grounded text *concurrently* with the image and still exceeds Qwen3-VL-8B on the same images, on both target mAP (0.346 vs. 0.330) and overall mAP (0.369 vs. 0.360).

Better understanding leads to better generation. Pre-trained Lumina-DiMOO has near-zero understanding mAP (0.019); fine-tuning unlocks it across all samplers, and the sampler with the strongest understanding ($\texttt{CO}_2\texttt{Jump}$, 0.369 overall) also achieves the best ImgEditBench score (1.93), while the weakest (MMaDA-Parallel, 0.335) is worst on ImgEditBench (1.44). This is the empirical signature of the chain-rule decomposition (Section 3.2): once text decisions inform image decisions within a single step, sharper text grounding tightens the image-side score distribution.

Scaling sampling steps. Sweeping NFE from 8 to 512 (Figure 4), $\texttt{CO}_2\texttt{Jump}$ is the only sampler that improves *monotonically* on both metrics: ImgEditBench rises from 1.72 to 1.93 and overall mAP from 0.074 to 0.369. Single-modality baselines plateau and regress at high NFE, and MMaDA-Parallel *degrades* from 1.52 to 1.44, its uncoupled schedule cannot productively use additional steps. The gap also widens: at 8 NFE the four methods sit within 0.009 mAP, but at 512 NFE $\texttt{CO}_2\texttt{Jump}$ is $+0.015/+0.016/+0.034$ ahead of MDM / ReMDM / MMaDA-Parallel. Coupling *compounds* across steps rather than saturating.

Table 2: **Joint accuracy on JMAZE-TEST500 and JNONO-TEST500.** A sample is correct only when text and image both match the algorithmic ground truth. **Bold**/underline = best/second best.

	Maze Solving (JMAZE-TEST500)			Nonogram Solving (JNONO-TEST500)		
Method	In-Distribution $6\times6-20\times20$	OOD $\{3,4,5,21,22\}^2$	Total $3\times3-22\times22$	In-Distribution $5\times5-25\times25$	OOD $\{3,4,26,27\}^2$	Total $3\times3-27\times27$
MDM	<u>0.461</u>	<u>0.312</u>	<u>0.424</u>	0.110	0.050	0.100
ReMDM	0.229	0.032	0.180	0.002	0.000	0.002
MMaDA-Parallel	0.419	0.304	0.390	<u>0.143</u>	<u>0.113</u>	<u>0.138</u>
CO_2Jump **(Ours)**	**0.469**	**0.320**	**0.432**	**0.167**	**0.175**	**0.168**

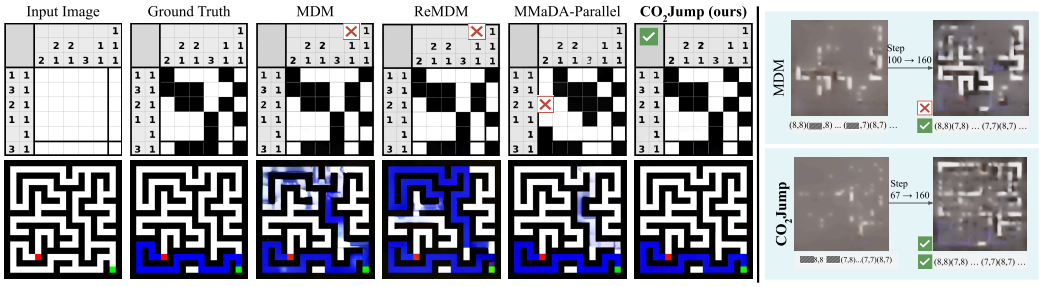


Figure 5: **Qualitative comparison.** Top left: Nonogram (red ✗ marks clue violations); Bottom left: Maze; Right: per-step MDM vs. CO_2Jump trajectory. Only CO_2Jump satisfies all clues and paths, and its joint trajectory stays consistent across both modalities.

5.2 Visual Reasoning: Maze and Nonogram Solving

Table 2 reports joint accuracy on visual reasoning benchmarks on in-distribution and OOD grid sizes.

CO_2Jump **outperforms other samplers in all six columns.** Improvements over the best baseline range from $+0.008$ (Maze, In-Dist) to $+0.062$ (Nonogram, OOD). No baseline is the runner-up everywhere — MDM is second on Maze but fourth on Nonogram, and MMaDA-Parallel is the reverse — so consistent joint correctness across both task structures is itself evidence that the coupling mechanism is generic rather than benchmark-specific.

Out-of-distribution generalization. CO_2Jump also holds up best on grid sizes outside the training range. On Maze it leads on the OOD split (0.320 vs. 0.312 for the runner-up), and on Nonogram it widens its lead substantially (0.175 vs. 0.113). MDM drops 55% from In-Dist to OOD on Nonogram and MMaDA-Parallel 21%, while CO_2Jump stays roughly flat. It shows that cross-modal coupling transfers solution-style information across grid sizes more robustly than uncoupled samplers.

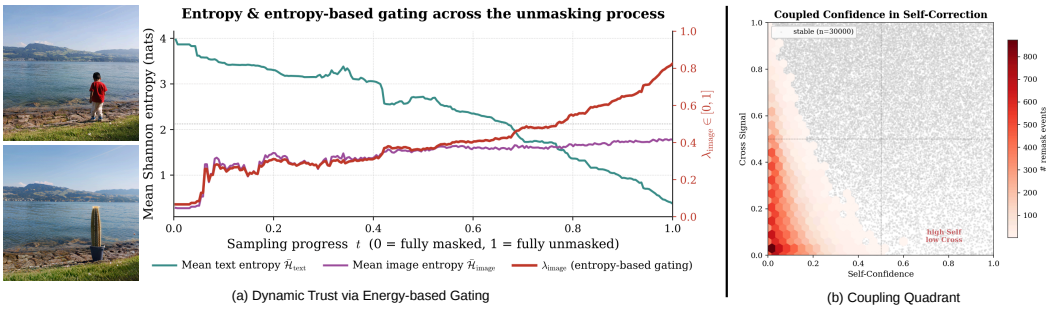
Qualitative trajectories expose the uncoupling failure mode. In Figure 5, every baseline violates at least one Nonogram clue (red ✗), while CO_2Jump satisfies all clues; on the maze, baselines wander into wrong corridors or cut through walls and CO_2Jump reproduces the ground-truth path. The right column exposes *why*: by step 160 of MDM's trajectory the text answer is correct but the image has drifted onto a different path, because under MDM's independence factorization the image branch never sees the latest committed text. CO_2Jump's chain-rule decomposition propagates each text commitment into image scoring at the same step, so both modalities converge on the same solution.

5.3 Ablation Study and Analysis

Ablation. Table 3 removes each mechanism in isolation. Removing Shared Percentile Rank causes the largest drop in image-edit fidelity ($1.93 \rightarrow 1.87$): without a common scale, one of Self-Confidence and Cross Signal arbitrarily dominates the blend. Removing Entropy-Based Gating leaves source-mAP unchanged but cuts target-mAP by 0.030 and overall mAP by 0.015, the gate's main role is in late-stage sampling where the target image is committed under freshly grounded text. Removing Self-Correction is the most damaging change for understanding (-0.032 on overall mAP) and drops editing by 0.01: without remasking, late-discovered cross-modal contradictions cannot be repaired.

Table 3: **Ablation on $\texttt{CO}_2\texttt{Jump}$.** Each row removes one mechanism: Shared Percentile Rank, Entropy-Based Gating (λ_{image}), or Self-Correction. **Bold**/underline = best/second best.

	Text Generation		Image Generation	Image Understanding (mAP@0.5:0.95 ↑)		
Method	GPT-2L PPL ↓	Entropy ↑	ImgEditBench ↑	Source	Target	Overall
$\texttt{CO}_2\texttt{Jump}$ **(full)**	**8.5649**	4.5005	**1.93**	**0.392**	**0.346**	**0.369**
– Shared Percentile Rank	8.7117	4.4957	1.87	0.374	0.315	0.344
– Entropy-Based Gating	8.7078	4.4984	1.91	**0.392**	0.316	0.354
– Self-Correction	8.8299	**4.5180**	1.92	0.363	0.310	0.337



(a) Dynamic Trust via Energy-based Gating

(b) Coupling Quadrant

Figure 6: **(a)** Per-step text/image entropy and the gate λ_{image}. The image starts confident (source prior), text starts uncertain; λ_{image} rises from ≈ 0.05 to ≈ 0.85 as text commits. **(b)** Image-side remask events in (Self-Confidence, Cross Signal) space: most fire in the lower-left region; a smaller hotspot at high-Self / low-Cross corresponds to coupling-driven revocations.

Dynamic trust via entropy-based gating. Figure 6 (a) traces per-step entropies and λ_{image} for a representative editing sample. The image branch starts confident (source-image prior) while text starts uncertain (only the editing prompt as context), so $\lambda_{\text{image}} \approx 0.25$ and the image relies more on self-confidence. As text commits, its entropy falls and the image moves into regions outside its prior; λ_{image} rises to ≈ 0.85, and the image branch increasingly leans on the text-grounded cross signal — the empirical signature of the chain-rule decomposition.

Coupling quadrant. Figure 6 (b) shows every image-side remask event in (Self-Confidence, Cross Signal) space. Most events concentrate in the lower-left region where both signals are low, so the coupled confidence $(1 - \lambda) \cdot \text{Self} + \lambda \cdot \text{Cross}$ is small and the death jump is triggered. A smaller hotspot in the high-Self / low-Cross quadrant corresponds to coupling-driven remasking — the image was locally confident but the text disagreed — and these account for the gap between the full $\texttt{CO}_2\texttt{Jump}$ and the "– Self-Correction" ablation row in Table 3.

6 Conclusion

We presented **Self-Correcting Coupled Markov Jump Processes (SC-CMJP)**, a framework in which the two modalities of a unified MDM negotiate their commitments *within* every denoising step, and $\texttt{CO}_2\texttt{Jump}$, a training-free single-pass sampler that instantiates it on a frozen backbone. By coupling one modality's transition rates to the other's emerging confidence via cross-modal attention, and by allowing committed tokens to be retracted through a remasking jump triggered by cross-modal contradictions, $\texttt{CO}_2\texttt{Jump}$ closes the concurrent-generation loop that prior parallel samplers leave open. $\texttt{CO}_2\texttt{Jump}$ delivers best performance across joint image understanding and editing, as well as visual reasoning tasks and it scales monotonically with denoising steps, evidence that cross-modal coupling compounds across the trajectory. Our experiments instantiate the framework on the canonical text-and-image pair; the SC-CMJP recipe is modality-agnostic, and extending it to audio, video, or structured outputs is left to future work.

References

[1] Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Jiaqi Han, Zhihan Yang, Zhixuan Qi, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Interpolating autoregressive and discrete denoising diffusion language models. In *ICLR*, 2025.

[2] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 2021.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

[4] David Berghaus, Kostadin Cvejoski, Patrick Seifner, Cesar Ojeda, and Ramses J Sanchez. Foundation inference models for markov jump processes. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=f4v7cmm5sC.

[5] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *NeurIPS*, 2022.

[6] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *ICML*, 2024.

[7] Jiayi Chen, Wenxuan Song, Pengxiang Ding, Ziyang Zhou, Han Zhao, Feilong Tang, Donglin Wang, and Haoang Li. Unified diffusion VLA: Vision-language-action model via joint discrete denosing diffusion process. In *ICLR*, 2026.

[8] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

[9] Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and Bowen Zhou. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *ArXiv*, abs/2510.06303, 2025.

[10] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining, 2025. URL https://arxiv.org/abs/2505.14683.

[11] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *NeurIPS*, 2024.

[12] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434, 1976.

[13] Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361, 1977.

[14] Daniel T Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of chemical physics*, 115(4):1716–1733, 2001.

[15] Roy J Glauber. Time-dependent statistics of the ising model. *Journal of mathematical physics*, 4(2):294–307, 1963.

[16] Google. Gemini 3 flash, 2026. URL https://gemini.google.com/.

[17] Zemin Huang, Yuhang Wang, Zhiyang Chen, and Guo-Jun Qi. Don't settle too early: Self-reflective remasking for diffusion language models. *ArXiv*, abs/2509.23653, 2025.

[18] Michael Igorevich Ivanitskiy, Rusheb Shah, Alex F. Spies, Tilman Räuker, Dan Valentine, Can Rager, Lucia Quirke, Chris Mathwin, Guillaume Corlouer, Cecilia Diniz Behn, and Samy Wu Fung. A configurable library for generating and manipulating maze datasets, 2023. URL http://arxiv.org/abs/2309.10498.

[19] Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z. Pan, Hyeji Kim, Sham M. Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction. *ArXiv*, abs/2510.01384, 2025.

[20] Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *ICML*, 2025.

[21] José Lezama, Tim Salimans, Lu Jiang, Huiwen Chang, Jonathan Ho, and Irfan Essa. Discrete predictor-corrector diffusion models for image synthesis. In *International Conference on Learning Representations*, 2023.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pp. 740–755. Springer, 2014.

[23] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *ICML*, 2024.

[24] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *NeurIPS*, 2025. URL https://openreview.net/forum?id=KnqiC0znVF.

[25] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *ICLR*, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

[26] Yidong Ouyang, Panwen Hu, Zhengyan Wan, Zhe Wang, Liyan Xie, Dmitriy Bespalov, Ying Nian Wu, Guang Cheng, Hongyuan Zha, and Qiang Sun. Training-free self-correction for multimodal masked diffusion models, 2026. URL https://arxiv.org/abs/2602.02927.

[27] Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. *ArXiv*, abs/2502.03540, 2025.

[28] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[29] 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. *NeurIPS*, 2024.

[30] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *NeurIPS*, 37:103131–103167, 2024.

[31] Ye Tian, Ling Yang, JiongFan Yang, Anran Wang, Yu Tian, Jiani zheng, Haochen Wang, Zhiyang Teng, Zhuochen Wang, Yinjie Wang, Yunhai Tong, Mengdi Wang, and Xiangtai Li. MMaDA-parallel: Multimodal large diffusion language models for thinking-aware editing and generation. In *ICLR*, 2026. URL https://openreview.net/forum?id=mkQAd11ovn.

[32] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017.

[33] Dimitri von Rutte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Scholkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. *ICML*, 2025.

[34] Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *NeurIPS*, 2025. URL https://openreview.net/forum?id=IJryQAOy0p.

[35] Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. Omniedit: Building image editing generalist models through specialist supervision. In *ICLR*, 2025. URL https://openreview.net/forum?id=Hlm0cga0sv.

[36] Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. In *NeurIPS*, 2025. URL https://openreview.net/forum?id=7VMg7Jb7AL.

[37] Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Ke Wang, Yibin Wang, Jinbin Bai, Qian Yu, Dengyang Jiang, Yuandong Pu, Haoxing Chen, Le Zhuo, Junjun He, Gen Luo, Tian-Xin Li, Ming Hu, Jin Ye, Shenglong Ye, Bo Zhang, Chang Xu, Wenhai Wang, Hongsheng Li, Guangtao Zhai, Tianfan Xue, Bin Fu, Xiaohong Liu, Yu Qiao, and Yihao Liu. Lumina-dimoo: An omni diffusion large language model for multi-modal generation and understanding. *ArXiv*, abs/2510.06308, 2025.

[38] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal large diffusion language models. In *NeurIPS*, 2025. URL https://openreview.net/forum?id=wczmXLuLGd.

[39] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *ArXiv*, abs/2508.15487, 2025.

[40] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. In *NeurIPS*, 2025. URL https://openreview.net/forum?id=uUCSrMlfD3.

[41] Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models. In *NeurIPS*, 2026. URL https://openreview.net/forum?id=HDeIb67lJe.

A Broader Impacts

Concurrent multimodal generation lowers the cost of producing tightly aligned text and image pairs in a single pass, which has plausible positive applications: illustrated tutoring and explanation, accessibility tools that pair narration with diagrams or maps, faster iteration in design and storyboard workflows, and more data-efficient visual-reasoning systems for tasks where text and image are logically interlocked (mazes, puzzles, scene graphs). Within machine learning research, the framework also offers a concrete training-free recipe for closing the cross-modal loop on top of a frozen backbone, which we hope will encourage further work on inference-time methods for joint multimodal models.

Like any image-editing or multimodal generative model, the technology carries risk of misuse for misleading or fabricated visual content (e.g., manipulated photographs paired with consistent textual descriptions), and the same coupling that makes the output internally consistent could make such fabrications harder to detect by inspecting either modality in isolation. JEDIT-1M, JMAZE-200K, and JNONO-200K can be released through a request-based interface with usage terms that prohibit deceptive applications, and the synthetic visual-reasoning corpora carry no real-world likeness risk. Downstream users who fine-tune the backbone on the released data are encouraged to follow community-standard safeguards.

B Background

B.1 Masked Diffusion Models and Remasking

Masked discrete diffusion models [2, 29, 23, 30] corrupt a clean sample $\mathbf{x} \in \mathcal{V}^L$ by gradually replacing tokens with a special absorbing state $\boldsymbol{m}$, and learn to invert this corruption. We adopt the continuous-time formulation of Campbell et al. [5] as our default view because it admits the cross-modal coupling and remasking extensions developed in this paper.

Forward process and CTMC equivalence. For $t \in [0, 1]$ and a monotonically decreasing noise schedule $\alpha_t \in [0, 1]$ with $\alpha_0 \approx 1$, $\alpha_1 \approx 0$, the per-position marginal of the forward process is

$$q(\mathbf{z}_t \mid \mathbf{x}) = \mathrm{Cat}\big(\mathbf{z}_t;\ \alpha_t\, \mathbf{x} + (1 - \alpha_t)\, \boldsymbol{m}\big), \tag{9}$$

factorized across positions. The same dynamics admit an equivalent Continuous-Time Markov Chain (CTMC) description: states evolve by stochastic jumps between $\mathbf{x}$ and $\boldsymbol{m}$ at the infinitesimal rate

$$\mathbf{R}_t \;=\; -\frac{\dot{\alpha}_t}{\alpha_t}\big(\mathbf{I} - \boldsymbol{m}\mathbf{1}^\top\big), \tag{10}$$

under which the integrated transition probability $q(\mathbf{z}_t = \mathbf{x} \mid \mathbf{x}) = \alpha_t$ recovers Eq. (9) exactly. Sahoo et al. [29] establish that the discrete-time absorbing-state formulation of MDLM and the CTMC formulation of Campbell et al. [5] parameterize the same family of marginals, posteriors, and likelihood bounds; we use the two views interchangeably.

Training objective. A denoising network $\mathbf{x}_\theta$ is trained to predict the clean state $\mathbf{x}$ from $\mathbf{z}_t$, and the resulting NELBO collapses to a position-wise weighted cross-entropy [29, 30]:

$$\mathcal{L}^{\infty}_{\text{NELBO}} \;=\; \mathbb{E}_{q,\,t} \int_0^1 \frac{\dot{\alpha}_t}{1 - \alpha_t} \sum_{\ell=1}^{L} \log\big\langle \mathbf{x}_\theta^\ell(\mathbf{z}_t, t),\ \mathbf{x}^\ell \big\rangle \,\mathrm{d}t. \tag{11}$$

The corresponding reverse posterior of standard MDLM has a well-known inability to remask [2, 5, 29]: once a token is unmasked, no subsequent reverse step can revisit it, so any error committed at a timestep t persists for the rest of the trajectory.

Remasking via σ_t. ReMDM [34] repairs this by allowing committed tokens to revert to $\boldsymbol{m}$ with per-step probability $\sigma_t \in [0, 1]$, yielding the modified reverse posterior

$$q_\sigma(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x}) = \begin{cases} \mathrm{Cat}\big(\mathbf{z}_s;\ (1 - \sigma_t)\, \mathbf{x} + \sigma_t\, \boldsymbol{m}\big), & \mathbf{z}_t \neq \boldsymbol{m} \\ \mathrm{Cat}\left(\mathbf{z}_s;\ \dfrac{\alpha_s - (1 - \sigma_t)\alpha_t}{1 - \alpha_t}\, \mathbf{x} + \dfrac{1 - \alpha_s - \sigma_t \alpha_t}{1 - \alpha_t}\, \boldsymbol{m}\right), & \mathbf{z}_t = \boldsymbol{m}, \end{cases} \tag{12}$$

which preserves the marginal in Eq. (9) when $\sigma_t \leq \min(1, (1 - \alpha_s)/\alpha_t)$ and recovers MDLM at $\sigma_t = 0$. The remask jump makes the forward process non-Markovian, but the reverse stays Markovian and tractable [34]. Existing σ_t schedules [34] are *modality-agnostic* – they score remasks from intra-modal likelihoods alone, leaving cross-modal contradictions undetected in joint generation.

B.2 Markov Jump Processes and Coupled Multimodal Generation

A Markov Jump Process (MJP) is a continuous-time Markov process on a discrete state space specified by a rate matrix $R_t(z, z')$; the CTMC view of MDM is exactly such an MJP over $\mathcal{V}^L$ with rate matrix Eq. (10). Exact reverse simulation via Gillespie's algorithm [12, 13] updates one position per jump and is prohibitive at modern sequence lengths, so practical samplers use τ-leaping [14] – a parallel approximation that updates all positions simultaneously, of which the standard MDM reverse step is the first-order discretization [5].

Coupled MJPs for Multimodal Generation. Coupled jump processes are classical in chemistry [13] and Glauber dynamics [15], but their coupling structure is hand-specified and they target physical simulation; standard MJP-based diffusion likewise treats per-position rates as conditionally independent given $\mathbf{x}_\theta(\mathbf{z}_t)$ even in multimodal settings. We instead define a *Coupled Markov Jump Process* (CMJP) over $\mathbf{z}_t = (\mathbf{z}_t^{\texttt{text}}, \mathbf{z}_t^{\texttt{image}})$ with modality-specific rate matrices $\mathbf{R}_t^a$ whose intensities depend on the hidden representations and instantaneous confidence of the complementary modality through learned cross-modal attention. Combined with a ReMDM-style σ_t jump (Eq. (12)), birth (unmask) and death (remask) rates of one modality are informed by the current commitments of the other, enabling localized cross-modal self-correction at sampling time.

Algorithm 1 $\texttt{CO}_2\texttt{Jump}$ Sampler (Asymmetric Scoring)

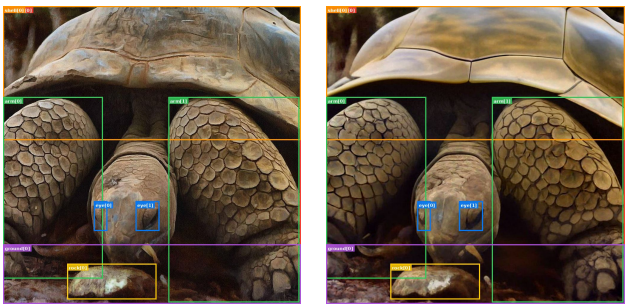
Input: Pretrained $\mathbf{x}_\theta$, schedule α_t, remasking ratio η, steps T, modality lengths $L^{\texttt{text}}, L^{\texttt{image}}$, conditioning c.
Initialize $\mathbf{z}_1 \leftarrow \{\boldsymbol{m}\}^{L^{\texttt{text}}+L^{\texttt{image}}}$.
for $i = T$ **down to** 1 **do**
  $t \leftarrow i/T,\ \ s \leftarrow (i-1)/T$
  $\mathbf{z}_s \leftarrow \mathbf{z}_t$ // initialize next-step state; death/birth jumps below write into $\mathbf{z}_s$
  // Single forward pass
  Run $\mathbf{x}_\theta(\mathbf{z}_t, c)$; extract $\mathbf{H}_t^{\texttt{text}}, \mathbf{H}_t^{\texttt{image}}$ and mean entropies $\bar{\mathcal{H}}_{\texttt{text}}, \bar{\mathcal{H}}_{\texttt{image}}$ over all positions
  Sample $\hat{\mathbf{x}}^\ell$ via Gumbel noise; record $\texttt{SelfConf}_\ell$ for both modalities
  // Asymmetric scoring (Eq. (3))
  Text $\texttt{Score}_{\texttt{text},\ell} \leftarrow \texttt{SelfConf}_{\texttt{text},\ell}$ // Term (I): no cross-modal dependence
  Image compute $\mathbf{A}_t^{\texttt{image}\to\texttt{text}}$, $\texttt{CrossSignal}_\ell$ (Eqs. 5 and 6), scalar gate λ (Eq. 7)
    $\texttt{Score}_{\texttt{image},\ell} \leftarrow \texttt{CoupledConf}_\ell$ via Eq. (8) // Term (II): approx. $\mathbf{z}_s^{\texttt{text}}$ via attention
  // Death & Birth Jumps
  $\sigma_t \leftarrow \min\big(\eta,\ (1-\alpha_s)/\alpha_t\big)$ // valid-posterior constraint
  for $a \in \{\texttt{text}, \texttt{image}\}$ **do**
    $U_t^a \leftarrow \lfloor \alpha_t L^a \rfloor,\ M_t^a \leftarrow \lfloor (1-\alpha_t) L^a \rfloor$ // idealized counts, floored at runtime
    $N^a_{\text{remask}} \leftarrow \lfloor U_t^a \sigma_t \rfloor,\ N^a_{\text{unmask}} \leftarrow \left\lfloor \frac{\alpha_s - \alpha_t}{1-\alpha_t} M_t^a \right\rfloor + \left\lfloor \sigma_t \frac{\alpha_t}{1-\alpha_t} M_t^a \right\rfloor$
    Death: in $\mathbf{z}_s$, remask the N^a_{remask} tokens of modality a with *lowest* $\texttt{Score}_a$ to $\boldsymbol{m}$
    Birth: in $\mathbf{z}_s$, reveal the N^a_{unmask} masked tokens of modality a with *highest* $\texttt{Score}_a$
  end for
end for
Output: Unified multimodal sample $\mathbf{z}_0$

C Qualitative Example: Joint Image Editing and Understanding

Figure 7 shows a complete $\texttt{CO}_2\texttt{Jump}$ sample from JEDIT-1M: prompt, source image, generated target image, the per-panel scene-graph understanding, and the thinking trace, all produced concurrently in a single denoising loop. Bounding boxes overlaid on each image are exactly those emitted by the model's own text branch (no external grounder), normalized to a 1001×1001 canvas. The example illustrates the planning-then-execution pattern that drives the gain on extended ImgEditBench: the text branch isolates `shell` as the only dynamic object and keeps every other entity (`tortoise`, `arm`, `eye`, `ground`, `rock`) at identical bounding boxes between source and target, while changing only the shell's attributes from `(scaly)` to `(smooth)`; the image branch realizes precisely that delta, removing the high-frequency scaly pattern on the shell while leaving body, limbs, and surroundings untouched.

<think> The transition from the source to the target image follows a precise, localized logic based on the prompt: "change the tortoise's shell texture to a smooth surface."
Object Categorization:
- **Static Objects:** The `tortoise` [0,0 to 1000,1000], `arm` [0, 305, 335, 915; 555, 305, 995, 995], `eye` [305, 655, 350, 755; 445, 655, 525, 755], `ground` [0, 800, 1000, 1000], and `rock` [215, 865, 515, 985] remain constant in both spatial coordinates and visual attributes.
- **Dynamic Objects:** The `shell` [0, 0, 1000, 450] is the sole dynamic element. While its bounding box remains identical, its internal attributes undergo a significant transformation.
Quantifying Change:
The visual delta is confined to the `shell` region. In the source state, the shell exhibits a high, high-frequency "scaly" texture characterized by sharp, irregular ridges and Inlay-like patterns. In the state state, the texture is smoothed into a "smooth" finish. The color palette shifts from a dark, desaturated brown to a lighter, uniform tan and tan,. and the surface geometry transitions from a complex, segmented scaly pattern to a continuous,,smooth,, smooth surface.
Critical Alignment Check:
The edit perfectly aligns with the instruction. The "smooth surface" is achieved by removing the high-frequency details of the sc scales while preserving the underlying lighting and lighting interactions on the tortoise's skin, such as the subtle highlights on the eyes and the soft shadows of the snout. There are no hallucinations or unintended alterations to the tortoise's anatomy or the complex textures of the surrounding environment.
Connection to Intent:
The intent was a specific texture modification. By maintaining the static nature of the tortoise and environment while exclusively altering the surface properties of the shell, the edit successfully fulfills the user's intent while compromising the integrity of the the original composition </think>
<answer>
SOURCE IMAGE ANALYSIS:
tortoise (brown, textured, scaly) [0]: [0, 0, 1000, 1000]
shell (textured, brown, tan, scaly) [0]: [0, 0, 1000, 450]
arm (scaly, brown, textured) [0]: [0, 305, 335, 915]
arm (scaly, brown, textured) [1]: [555, 305, 995, 995]
eye (dark, round) [0]: [305, 655, 350, 755]
eye (dark, round) [1]: [445, 655, 525, 755]
ground (dark, earthy) [0]: [0, 800, 1000, 1000]
rock (brown, textured) [0]: [215, 865, 515, 985]

TARGET IMAGE ANALYSIS:
tortoise (brown, textured, scaly) [0]: [0, 0, 1000, 1000]
shell (smooth, brown, tan, smooth) [0]: [0, 0, 1000, 450]
arm (scaly, brown, textured) [0]: [0, 305, 335, 915]
arm (scaly, brown, textured) [1]: [555, 305, 995, 995]
eye (dark, round) [0]: [305, 655, 350, 755]
eye (dark, round) [1]: [445, 655, 525, 755]
ground (dark, earthy) [0]: [0, 800, 1000, 1000]
rock (brown, textured) [0]: [215, 865, 515, 985]
</answer>



Prompt: Change the tortoise's shell texture to a smooth surface.

Figure 7: **Joint image editing with concurrent understanding on JEDIT-1M.** Prompt, model-generated thinking trace, scene-graph source/target analyses, and the source/generated images with predicted bounding boxes overlaid (boxes are the model's own text-side predictions, not ground truth). The text branch declares only the `shell` as a dynamic object — every other entity has identical bounding boxes in the source and target analyses — and the image branch realizes exactly this localized texture edit: scaly $\rightarrow$ smooth on the shell, with body, limbs, ground and rock preserved.

D Qualitative Example: Maze Solving

Figure 8 shows a $\texttt{CO}_2\texttt{Jump}$ sample from JMAZE: the input maze (with green start and red end), the model-generated thinking trace and coordinate path, and the solved maze with the blue path overlaid. Both the textual coordinate sequence and the rendered solution path are produced concurrently in a single denoising loop. The thinking trace reasons globally over the maze — identifying the upper half as a dead-end region, the lower corridors as the main artery, and the central zig-zag through the bottom row as the southern bypass — before committing the explicit (r, c) sequence in `<answer>`, and the image branch traces exactly that sequence, demonstrating that the path drawn on the grid agrees position-by-position with the coordinates the text branch declares.

E Qualitative Example: Nonogram Solving

Figure 9 shows a $\texttt{CO}_2\texttt{Jump}$ sample from JNONO: the input nonogram (an empty 7×7 grid with row and column clue numbers), the model-generated thinking trace and per-row filled-cell answer, and the solved grid with the corresponding cells filled black. The thinking trace performs bidirectional constraint propagation rather than row-by-row solving: it identifies the most globally-constraining lines first (rows whose largest clue forces a 4-cell block, columns whose clue sums match the grid width), explains how the row and column constraints tighten each other concurrently, and only then commits the row-by-row filled-cell ranges in `<answer>`; the image branch fills precisely those ranges on the grid, satisfying every clue.

F Dataset Curation and Pipeline Details

In this section, we detail the visual and textual data generation pipelines illustrated in Figure 3, as well as the structural format of the curated samples shown in Figure 10.

<think>
The maze is a 12x12 grid with the start at (8, 1) and the target at (6, 10). Analyzing the global structure, the upper half of the maze (rows 0–5) is largely disconnected from the target, acting as a massive dead-end region. The primary challenge must occur through the lower quadr, the long horizontal corridors between rows 8 and 11 serve as the main artery connecting the start and right sides.

The path begins by a descent from from the start at (8, 1) through the bottom row 11 via (8, 1) and (9, 1) to reach the, albeit,, bottom-left corner at (9, 0) and (10, 0). Simultaneously, we can see the target at (6, 10) requires be reached from the left through (6, 9) and (6, 8). This creates a clear objective: connect the bottom-left corner (11, 0) to the vertical column on the right..
The central forms involves a series of vertical zig-zags navigate the internal walls. From the bottom corner at (11, 0), the path moves through (11, 1), (11, 2), (10, 2), and (9, 2) to bypass a wall at (9, 3). It then wevers through a vertical at (8, 2) through (8, 3), (9, 3), (10, 3) and (11, 3). The final stretch is a rhythmic sequence along the bottom row: (11, 3) to (11, 4) to (11, 5) to (10, 5) to (10, 6) to (10, 7) to (11, 7) to (11, 8) to (11, 9) to (10, 9) to (9, 9) to (9, 8) to (8, 8). This southern bypass effectively bridges the northern traps and the central barriers. The is completed by moving from (7,8) through (6,8) and (6,9) directly to the target at (6,10).
</think>
<answer> (8,1) (9,1) (9,0) (10,0) (11,0) (11,1) (11,2) (10,2) (9,2) (8,2) (8,3) (9,3) (10,3) (11,3) (11,4) (11,5) (10,5) (10,6) (10,7) (11,7) (11,8) (11,9) (10,9) (9,9) (9,8) (8,8) (7,8) (6,8) (6,9) (6,10) </answer>

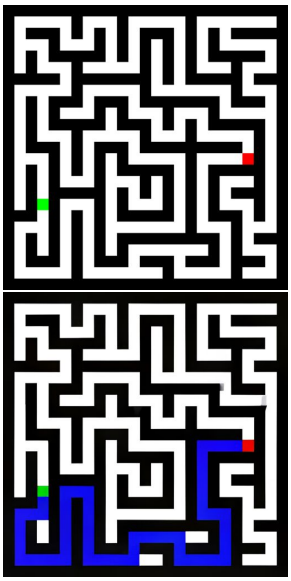


Figure 8: **Joint maze solving on JMAZE.** $\texttt{CO}_2\texttt{Jump}$'s thinking trace and answer (left) alongside the input maze (top right; green start, red end) and the solved maze with the blue path overlaid (bottom right). The text branch first reasons globally about which corridors are dead-ends and which form the main artery, then commits the path as an explicit (r, c) coordinate sequence; the image branch traces precisely that sequence on the grid in the same denoising trajectory.

F.1 Curation Pipelines

The curation process splits into two distinct paradigms: learning from natural image-editing corpora with VLM-augmented annotations, and generating deterministic reasoning tasks via programmatic scripts.

- **JEDIT-1M Augmentation Pipeline:** We ingest raw triplets of (Source Image, Target Image, Input Prompt) from public image-editing benchmarks. We pass these inputs to the oracle MLLM (`Qwen3-VL-235B`) to produce pixel-aligned, cross-panel bounding boxes and category attributes. The model outputs a structured tag `<answer>...</answer>` preserving category and tracking-indices for corresponding objects in both the source and target frames. Finally, both panels, the original prompt, and the generated scene-graph understanding are fed back to the `Qwen3-VL-235B` model to generate the final natural-language thinking trace.

- **JMAZE-200K Algorithmic Generation:** We use the `maze-dataset` Python library to procedurally generate maze geometries. The pipeline automatically renders a black-and-white maze (Source Image) marking a green starting point and red goal point, and a target image overlaying the unique solution path in solid blue. The script outputs a structured text prompt specifying the maze grid map alongside a step-by-step path sequence written in global (r, c) coordinates (`<answer>...</answer>`). Finally, these algorithmic outputs are processed by `Gemini 3 Flash` to synthesize the parallel-reasoning trace.

- **JNONO-200K Nonogram Pipeline:** A customized Python pattern-generation script constructs solvable $N \times N$ nonogram puzzles. The script renders a source image depicting an empty canvas grid flanked by marginal constraint clues, and a target image containing the algorithmically solved grid where valid cells are filled with black pixels. The prompt contains sequential row and column clues (e.g., `<rows>...</rows>`), while the corresponding understanding tag contains the coordinate-mapped cell filling rules. Similar to the maze task, the full algorithmic output is fed to `Gemini 3 Flash` to generate a holistic thinking trace.

<think>
The 7x7 grid presents a compact puzzle that reveals a staggered, blocky shape. Solving begins by identifying the most constrained lines, specifically 5 and R 6, where the clues (3,3) and (2,3) respectively with the mandatory gap gap account the entire 7. This immediately fixes the bottom-left corner (0-2 and 4-6) and the bottom-right corner (0-1 and 4-6). Simultaneously, Columns 0, 1, 2, 4, and 5, 6 all have clues of 4. which, when combined with the forced fills from R 5 and R 6, create a solid 4x4 block in the lower-left quadrant.

As these vertical and horizontal, the central " (R, 3, 4) 5" clues in Rows 3 and 4 are almost the only way to satisfy a 5 5 clue in a 7x7 space, as they must be the the intersection of Columns 0- 4 and Columns 5- 6. This central "5"5 forces the columns to fills in R R R R R in Columns 5 and 6. Consequently, the "2" clues (Row 02) are pushed to the far right, which aligns perfectly with the (2,2) clues in Columns 5 and 6. This leaves a solid 2x2 block in the top-right corner, completing the image..
row 0: (5-6) ; row 1: (5-6) ; row 2: (2-3) ; row 3: (0-4) ; row 4: (0-4) ; row 5: (0-4) ; row 6: (0-1)
</think>
<answer>
row 0: (5-6) ; row 1: (5-6) ; row 2: (2-3) ; row 3: (0-4) ; row 4: (0-4) ; row 5: (0-2)(4-6) ; row 6: (0-1)(4-6) </answer>

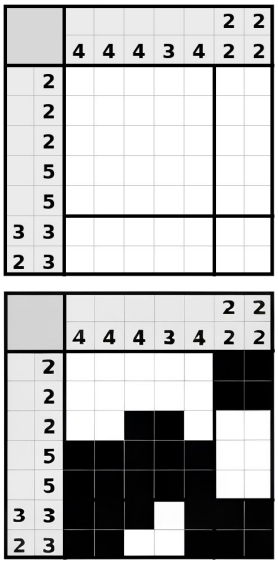


Figure 9: **Joint nonogram solving on JNONO.** CO_2Jump's thinking trace and answer (left) alongside the input puzzle (top right; row clues at left margin, column clues at top margin) and the solved grid with cells filled black (bottom right). The text branch reasons in a bidirectional, constraint-propagation style — starting from the most globally-constrained rows and columns and using each forced fill to tighten the other dimension — before committing the per-row filled-cell ranges; the image branch fills exactly those ranges on the grid in the same denoising trajectory, satisfying every clue.

F.2 Sample Structure and Thinking Traces

Figure 10 provides a concrete example of a training sample, in particular of the JEDIT-1M dataset. Generally, each sample across our three datasets is formatted into a unified schema consisting of five key fields:

1. **Input Prompt:** The natural language edit instruction or task specification.
2. **Source Image:** The initial conditioning visual input (e.g., the unedited image, the unsolved maze, or the empty nonogram grid).
3. **Target Image:** The ground-truth generation output target (e.g., the edited image, the solved maze path, or the completed nonogram pattern).
4. **Structured Understanding:** Pixel-aligned scene graphs (for image editing) or coordinate/line constraints (for visual reasoning) wrapped in parsing tags (such as `<answer>...</answer>`). This acts as an intermediate symbolic representation.
5. **Thinking Trace:** Synthesized by oracle MLLMs. For JEDIT-1M, this trace is generated by `Qwen3-VL-235B` to explain the regional changes needed to transition between panels. For JMAZE-200K and JNONO-200K, the trace is synthesized by `Gemini 3 Flash` based on the algorithmically generated structured layout, providing reasoning that justifies constraint propagation and coordinates.

<think>
The transition from the source to the target image executes a targeted semantic swap of subjects while maintaining environmental and compositional integrity.

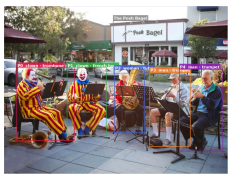
Static Objects: The background architecture (The Posh Bagel building [487, 85, 804, 401]), parked vehicles (silver sedan [0] and dark gray car [1]), and burgundy umbrellas [0, 1] remain spatially invariant. Crucially, the three musicians on the right (Persons [2, 3, 4]) and all five black metal music stands [0-4] remain fixed in their respective coordinates, anchoring the scene's layout.
Dynamic/Deleted & Added Objects:

- **Person [0]:** The clown in a red and yellow striped costume playing a trombone at [56, 363, 315, 804] is removed. He is replaced by an older man with glasses, wearing a blue shirt and black sweater vest [48, 371, 321, 803], who continues to play the trombone.
- **Person [1]:** The second clown in striped attire at [281, 363, 458, 775] is removed. He is replaced by an older man with a white beard and blue cap wearing a tan jacket [290, 372, 457, 779], playing the French horn.

Quantifying the Change: The transformation involves a precise substitution of the leftmost figures. The bounding box for Person [0] shifts marginally (-8px left, +8px top) to accommodate the new musician's posture, while Person [1] shifts (+9px left, +9px top). The core change is attribute-based: costumes and face paint are replaced by "casual clothing" (sweater vests, jackets, and caps).

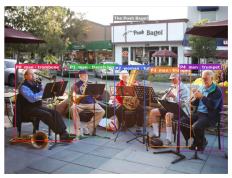
Critical Alignment Check: The edit strictly adheres to the prompt. The two clowns are successfully swapped for "older musicians" in "casual clothing." Both new subjects are depicted playing "brass instruments" (trombone and French horn), matching the musical context of the scene. There are no hallucinations; the replacement is seamless and logical, transforming the ensemble into a stylistically unified group of seniors.
</think>
<answer>
SOURCE IMAGE ANALYSIS:

building (white facade, "The Posh Bagel" sign)	[0]: [487, 85 , 804, 401]
person (clown, red and yellow striped costume, face paint, playing trombone)	[0]: [56 , 363, 315, 804]
person (clown, red and yellow striped costume, mask, playing French horn)	[1]: [281, 363, 458, 775]
person (older woman, white hair, red jacket, playing tuba)	[2]: [493, 378, 629, 755]
person (older man, tan hat, grey jacket, khaki shorts, playing trumpet)	[3]: [651, 395, 834, 839]
person (older man, glasses, blue shirt, dark navy jacket, playing trumpet)	[4]: [783, 376, 980, 866]



TARGET IMAGE ANALYSIS:

building (white facade, "The Posh Bagel" sign)	[0]: [487, 85 , 804, 401]
person (older man, glasses, blue shirt, black sweater vest, playing trombone)	[0]: [48 , 371, 321, 803]
person (older man, white beard, blue cap, tan jacket, playing French horn)	[1]: [290, 372, 457, 779]
person (older woman, white hair, red jacket, playing tuba)	[2]: [493, 378, 629, 755]
person (older man, tan hat, grey jacket, khaki shorts, playing trumpet)	[3]: [651, 395, 834, 839]
person (older man, glasses, blue shirt, dark navy jacket, playing trumpet)	[4]: [783, 376, 980, 866]



</answer>

"Swap the two clowns in red and yellow striped costumes with two older musicians wearing casual clothing and playing brass instruments."

Figure 10: **Editing sample with bounding-box understanding and thinking trace.** For JMAZE-200K and JNONO-200K, the thinking trace is synthesized by an oracle MLLM based on the algorithmically generated structured layout.

F.3 System Prompts for Dataset Curation

F.3.1 Understanding Prompt: Pixel-Aligned Scene Graph Extraction

The following system prompt was used to extract the structured understanding (bounding boxes and attributes) from the source and target images.

```
Task: Pixel-Aligned Scene Graph Extraction for Image Editing Pairs.

### Objective:
Analyze the transition from the Source Image to the Target Image. You
must extract a high-fidelity grounding dataset that maps how visual
tokens change in response to the Editing Prompt. Focus on maintaining
spatial "anchors" for static regions while precisely detailing the
"delta" in edited regions.

### Inputs:
1. Source Image: (The original state)
2. Target Image: (The modified state)
3. Editing Prompt: "{editing_prompt_var}"

### Strict Constraints:
THINKING BEFORE ANSWERING: In your <think> tags, you MUST
explicitly compare the two images. List:
1. Static entities (Background/Context).
2. Transformed entities (Attribute/State changes).
3. New or Deleted entities (Structural changes).

FIDELITY OVER INTENT: Describe what is visually rendered in the
Target Image. If the prompt asks for a "red car" but the Target Image
contains a "pink car," you MUST label it as "pink car."

COMPLETENESS: Annotate the main subject, all objects involved in
the edit, and key background anchors (e.g., floor, sky, walls) to
provide global context.

COORDINATES: Use normalized integer bounding boxes [x1, y1, x2, y2]
on a [0-1000] scale. Bounding boxes must be tight.

IDENTITY CONSISTENCY: For any entity present in both images, the
'category_name' and '[index]' MUST be identical.

PER-CATEGORY INDEXING: Every new object category MUST start its
index at [0]. For example, the first 'car' is 'car [0]', and the
first 'person' is 'person [0]'. Do NOT use a single global
incrementing index for the whole scene.

STRUCTURAL LOGIC: If an object is deleted in the Target, do not
list it in the TARGET IMAGE ANALYSIS. If an object is added, list it
only in the TARGET section.

### Prohibited Actions:
DO NOT use generic placeholder strings (e.g., "category_name",
"attribute_list").
DO NOT include the brackets around the category name or attributes
in the final answer; only the index should remain in brackets as per
format.

### Output Format:
<answer>
SOURCE IMAGE ANALYSIS:
category_name (attribute_list) [index]: [x1, y1, x2, y2]

TARGET IMAGE ANALYSIS:
category_name (attribute_list) [index]: [x1, y1, x2, y2]
</answer>

### Reference Example:
<answer>
SOURCE IMAGE ANALYSIS:
mountain (snowy, jagged) [0]: [0, 50, 1000, 450]
person (black hoodie, walking) [0]: [420, 550, 580, 910]
person (blue jeans, standing) [1]: [600, 550, 700, 910]

TARGET IMAGE ANALYSIS:
mountain (snowy, jagged) [0]: [0, 50, 1000, 450]
person (yellow raincoat, walking) [0]: [420, 550, 580, 910]
person (blue jeans, standing) [1]: [600, 550, 700, 910]
</answer>
```

F.3.2 JEDIT-1M Thinking Prompt: Reasoning Trace Synthesis

The following system prompt was used with Qwen3-VL-235B to synthesize the logic-based reasoning trace for the JEDIT-1M dataset.

```
Task: Synthesize a Logic-Based Reasoning Trace for Image Editing.

### Inputs:
1. Source and Target Images: Visual evidence.
2. Editing Prompt (The intended transformation): {editing_prompt_var}
3. Image Understanding (Bounding box coordinates and object labels
for both states): {answer}

### Objective:
Synthesize a logic-based reasoning trace that analyzes the visual
transition relative to the Editing Prompt. Critically evaluate
whether the target state successfully fulfills the prompt's
instructions or deviates from the intended logic.

### Strict Reasoning Guidelines:
Categorize Objects: Group objects into: Static (persistent
coordinates/attributes), Dynamic (modified coordinates/attributes),
and Deleted/Added (present in only one state).

Quantify Change: Reference bounding box shifts or attribute
updates as direct evidence of the transition.

Critical Alignment Check: Explicitly compare the visual delta
against the Editing Prompt. Identify hallucinations or instruction
mismatches.

Connect to Intent: Explain how the detections fulfill-or fail
to fulfill-the specific requirements of the Editing Prompt.

Constraint: Keep the response dense and information-rich, within
a max of 350 words. Avoid generic phrases. Start with the direct
logical flow.

### Output Format:
<think> [Your concise logical trace here] </think>
```

F.3.3 Thinking Prompt: Maze Solving Parallel Reasoning

The following system prompt was used to synthesize the parallel-reasoning trace for the maze-solving problem in the JMAZE-200K dataset.

```
Task: Synthesize a parallel-reasoning trace for a maze-solving problem.

### Inputs:
1. Input maze image: walls, a green start cell, and a red end cell. No path drawn.
2. Output maze image: the same maze with the correct solution path drawn in blue.
3. Maze structure as text (adjacency list + start + end):
{prompt}
4. Ground-truth solution path:
{answer}

### Objective:
Write a natural reasoning trace that reflects a global-to-local refinement process, illustrating how
the maze is solved simultaneously rather than purely sequentially.
Briefly describe the maze boundaries (grid size, start position, end position).
Identify the global structure first: note major bottlenecks, large dead-end regions to mask out, or
key anchor points/corridors between the start and end.
Describe the path formation as a concurrent process-e.g., establishing a
bridge through a central corridor while simultaneously connecting the start
and end regions to that main artery, verifying local wall-openings via the
adjacency list.
Write in natural prose, as a holistic chain of thought. No bullet lists.
Keep it under 350 words.

### Output format (strict - respond with exactly this tag and nothing else):
<think>
{}[Your reasoning trace]
</think>
```

F.3.4 Thinking Prompt: Nonogram Parallel Reasoning

The following system prompt was used to synthesize the parallel-reasoning trace for the nonogram puzzle in the JNONO-200K dataset.

```
Task:  Synthesize a parallel-reasoning trace for a nonogram (picross) puzzle.

### Inputs:
1.  Source image:  empty NxN grid with row clue numbers along the left margin and column clue numbers
along the top margin.
2.  Target image:  the same grid with cells filled to satisfy every clue, revealing a pixel-art
picture.
3.  Puzzle structure as text (size + row clues + column clues):
{prompt}
4.  Ground-truth solution (filled cell ranges per row):
{answer}

### Objective:
Write a holistic reasoning trace that reflects parallel constraint propagation,
NOT row-by-row sequential solving.
Briefly describe the puzzle (grid size, rough shape of the revealed picture).
Identify the most globally-constraining lines first:  rows or columns where
the sum of clues plus required gaps equals N (full-line forces), or lines
whose single largest clue produces "definite-fill" cells via the
left-most / right-most overlap argument.
Describe how column constraints and row constraints tighten each other
concurrently:  a forced fill produced by a row clue confirms a position in
some column's clue list, which can in turn unlock more rows.
Avoid sequential "first solve row 0, then row 1" prose.  Emphasize the
bidirectional, global interplay between row and column constraints.
Write in natural prose, no bullet lists.  Keep it under 250 words.

### Output format (strict - respond with exactly this tag and nothing else):
<think>
{}[Your reasoning trace]
</think>
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the four contributions — the SC-CMJP framework, the $\texttt{CO}_2\texttt{Jump}$ sampler, the three released corpora (JEDIT-1M, JMAZE-200K, JNONO-200K), and the headline empirical findings (best joint performance on image editing and visual reasoning, monotone scaling with denoising steps). Each claim is supported by Sections 3–5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Conclusion (Section 6) explicitly notes that our experiments instantiate the SC-CMJP framework only on the text-and-image modality pair, and that extension to audio, video, or structured outputs is left to future work. We also note that the cross-modal attention used at inference is read post-hoc from the frozen backbone, so its quality is bounded by what the backbone has already learned to attend to.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not state new theorems requiring proofs. Equations in Sections B and 3 (the MDLM marginal/posterior, the ReMDM remasking posterior, the CMJP rate matrices, the joint NELBO) restate or specialize established results from the cited prior work and are used to motivate the sampler design.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 fully specifies the $\texttt{CO}_2\texttt{Jump}$ sampler (asymmetric scoring rule, entropy gate, remasking jump) and Algorithm 1 in the appendix gives the pseudocode. Section 4 documents the three corpora and benchmarks; Section 5 specifies the controlled comparison protocol (frozen Lumina-DiMOO backbone, fine-tune per corpus, swap only the inference-time sampler), the shared cosine α_t schedule, and the linear remasking schedule $\sigma_t = 0.01$ on $t \in [0.25, 0.75]$ following ReMDM.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The three new corpora (JEDIT-1M, JMAZE-200K, JNONO-200K, totaling 1.4M joint samples) and the matching held-out benchmarks (JMAZE-TEST500, JNONO-TEST500) are released alongside the paper, and the $\texttt{CO}_2\texttt{Jump}$ sampler implementation together with the evaluation pipeline will be released upon publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 specifies the train/test splits (1.4M training samples; held-out 500-sample test sets per reasoning task; ID and OOD grid-size partitions). Section 5 specifies the backbone (Lumina-DiMOO), the four-sampler comparison protocol, the cosine α_t schedule, the ReMDM-style remasking schedule ($\sigma_t = 0.01$ on $t \in [0.25, 0.75]$), and the evaluation pipeline (Gemini-3-Flash oracle judge, mAP@0.5:0.95 against pseudo-grounding from the same Qwen3-VL family used for supervision).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Tables report single-run point estimates per sampler and benchmark; full-corpus fine-tuning of the backbone and 500-sample Gemini-judged evaluation per model make multi-seed runs prohibitively expensive. To compensate, the controlled-comparison protocol holds the backbone, training corpus, and decoding budget constant across the four samplers, varying only the inference-time sampler, and the ImgEditBench NFE-scaling sweep (Figure 4) covers a wide grid of decoding budgets to surface trends rather than single-point comparisons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources for fine-tuning each model from the Lumina-DiMOO backbone is mentioned in the Experiments section. Inference of $\texttt{CO}_2\texttt{Jump}$ adds no extra forward passes over a baseline MDM sampler; the cross-attention is read post-hoc from the same forward pass, and the remasking jump uses indexing operations only, so the per-step compute is dominated by the backbone forward pass.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The work does not involve human subjects; the released corpora are derived from publicly available image-editing datasets (ImgEdit, OmniEdit) augmented with synthetic annotations from oracle MLLMs, and from algorithmically generated maze and nonogram puzzles, with no scraped personal data and no sensitive content.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Concurrent multimodal generation enables positive applications such as illustrated tutoring, accessibility tools that pair narration with diagrams, and data-efficient visual-reasoning systems. Like any image-editing or multimodal generative model, it also carries risk of misuse for misleading or fabricated visual content; we discuss this in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: We do not release a new pretrained generative model; $\texttt{CO}_2\texttt{Jump}$ is a training-free inference-time sampler that runs on top of the publicly released Lumina-DiMOO backbone. The released corpora are derived from public image-editing datasets and algorithmically generated puzzles, and contain no scraped or sensitive content beyond what is already public.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All used assets are credited in-text and in the bibliography: ImgEdit [40], OmniEdit [35], Lumina-DiMOO [37], MMaDA [38], ReMDM [34], MMaDA-Parallel [31], UD-VLA [7], Qwen3-VL-235B [3], Gemini 3 Flash [16], GPT-2 Large [28], COCO [22], and `maze-dataset` [18]. We respect each asset's original license and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The three new corpora (JEDIT-1M, JMAZE-200K, JNONO-200K) and the two held-out benchmarks (JMAZE-TEST500, JNONO-TEST500) are documented in Section 4, including the curation pipeline, record schema (prompt, source image, target image, structured understanding, thinking trace), source corpora, oracle MLLMs used for synthetic supervision, train/test splits, and evaluation protocol.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All annotations in our released corpora are produced by oracle MLLMs (Qwen3-VL-235B for JEDIT-1M understanding and thinking traces; Gemini 3 Flash for JMAZE/JNONO thinking traces) or by algorithmic generators.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects research, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

Justification: We use oracle multimodal LLMs as substantive components of dataset construction and evaluation, and we declare these uses explicitly in Sections 4 and 5: Qwen3-VL-235B synthesizes the per-image scene-graph understanding and thinking-trace supervision for JEDIT-1M; Gemini 3 Flash synthesizes thinking traces for JMAZE/JNONO, scores ImgEditBench fidelity, and judges text/image correctness on the visual-reasoning benchmarks; pseudo-grounding labels for the mAP metric are obtained by feeding (source, generated-target) image pairs to Gemin 3 Flash. The $\texttt{CO}_2\texttt{Jump}$ sampler itself does not call an external LLM at inference.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.