Visual-Redundancy-Controlled Parallel Decoding for Diffusion-Based Multimodal Large Language Models

Yulin Yuan
Zhejiang University
yulinyuan@zju.edu.cn

Hongshuo Zhao
ZJU-UIUC Institute,
Zhejiang University
hongshuo.24@intl.zju.edu.cn

Xiangming Meng*
ZJU-UIUC Institute,
Zhejiang University
xiangmingmeng@intl.zju.edu.cn

Abstract

Diffusion-based multimodal large language models (dMLLMs) decode by iteratively predicting tokens at multiple masked positions in parallel. This turns each decoding step into a position-selection problem: the model must choose not only which predictions are reliable in isolation, but also which positions should be committed together as context for later decoding steps. Confidence-based unmasking ranks masked positions independently and commits the top-K positions, largely ignoring whether the committed tokens provide complementary visual grounding. We identify a step-level limitation of this strategy in multimodal settings: high-confidence tokens selected in the same step can rely on overlapping visual grounding, introducing visual redundancy among the committed tokens and leaving less complementary visual grounding available for later decoding. To quantify this effect, we introduce the Visual Redundancy Index (VRI), which measures visual grounding overlap among tokens committed in parallel. To control this redundancy during decoding, we propose Visual-Redundancy-Controlled Decoding (VRCD), a training-free inference-time decoding method that uses token-to-image attention to prioritize visually complementary positions. On M^{3}CoT, VRCD reduces visual redundancy and remaining-position entropy with modest runtime overhead. It improves accuracy over confidence-based unmasking on all five benchmarks, with relative gains of up to 18.8% on M^{3}CoT and 6.9% on MMBench in longer decoding. Code is available at `https://github.com/infiniteYuanyl/VRCD`.

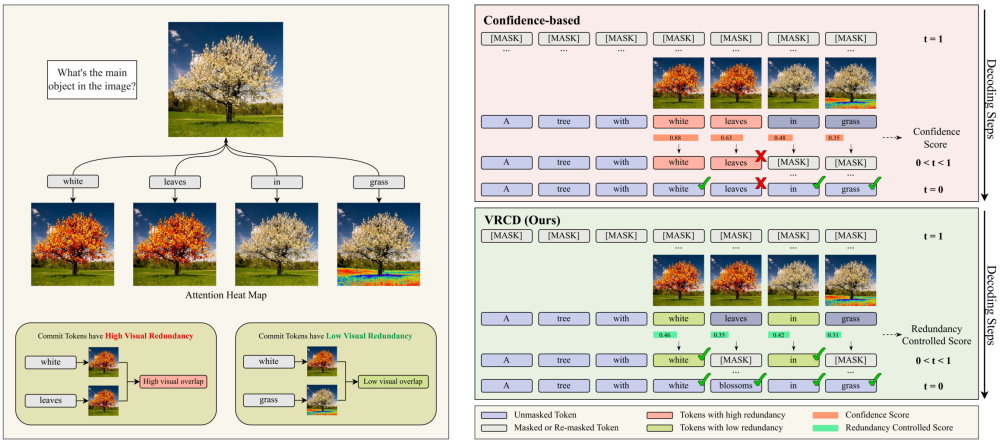
1 Introduction

Diffusion-based multimodal large language models (dMLLMs) are emerging as an alternative to autoregressive multimodal generation [Alayrac et al., 2022, Liu et al., 2023, Zhu et al., 2023, Bai et al., 2023]. Instead of extending a prefix one token at a time, dMLLMs generate by iteratively decoding masked textual positions conditioned on visual input [Yu et al., 2025, Li et al., 2025d, You et al., 2025, Yang et al., 2025b]. This formulation naturally enables parallel decoding: at each decoding step, the model predicts all remaining masked positions and commits only a subset of them. Parallelism can reduce the number of decoding iterations, but it also changes the nature of decoding. The central decision is no longer only which token should come next, but which masked positions should be committed together and become context for the remaining generation process.

Most existing decoding policies make this decision through token-level reliability. Confidence-based unmasking commits the top-K positions with the highest predicted probabilities, while recent variants refine the process through remasking criteria, dilated schedules, revocable commitments, uncertainty-aware search, or information-gain objectives [Wang et al., 2025, Luxembourg et al., 2025, Hong et al., 2025, Chen et al., 2025, Yang et al., 2026]. These methods improve the reliability or efficiency of diffusion decoding, but they largely preserve an implicit assumption: positions that are reliable in isolation are also suitable to commit in the same step. In multimodal decoding, this assumption can

*Corresponding author.

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



(a) Visual redundancy **(b)** Confidence-based vs. VRCD

Figure 1: Overview of visual redundancy and VRCD. (a) Visual redundancy arises when tokens committed in the same decoding step are associated with overlapping visual grounding. (b) VRCD prioritizes lower-redundancy tokens for parallel decoding, which provides more visually complementary context for later masked positions.

fail because textual predictions are not only statistically confident; they are also visually grounded. Thus, a parallel decoding step should be judged not only by the confidence of individual tokens, but also by whether the tokens committed together rely on complementary visual grounding.

We identify an overlooked phenomenon in dMLLM parallel decoding: high-confidence textual tokens committed in the same step can rely on overlapping visual grounding. For example, several candidate tokens may attend to the same object or local image region. Although each token is reliable on its own, committing them together can repeatedly introduce similar visual grounding rather than complementary context. We call this pattern *visual redundancy within the same decoding step*. It is distinct from semantic repetition in generated text and from redundancy among input visual tokens: here, redundancy refers specifically to overlap in the visual grounding of textual tokens unmasked in the same decoding step. Fig. 1(a) illustrates this phenomenon.

This visual redundancy can shape later decoding because unmasked tokens become conditioning context for subsequent steps. Tokens with complementary visual grounding can provide more diverse cues for the remaining masked positions, whereas visually redundant tokens repeatedly ground the same region. We quantify this property with the *Visual Redundancy Index* (VRI), which measures token-to-image grounding overlap among tokens committed in parallel; lower values indicate less repeated grounding within the current step.

VRI measures visual redundancy after tokens are committed, but reducing such redundancy requires a lightweight signal during decoding. Directly searching for a high-confidence, low-redundancy subset would require joint subset selection at every decoding step, which is impractical for inference-time decoding. We therefore propose *Visual-Redundancy-Controlled Decoding* (VRCD), a training-free reranking method for dMLLM parallel decoding. As shown in Fig. 1(b), at each step, VRCD builds a high-confidence candidate window, estimates pairwise visual overlap from token-to-image attention, and converts this overlap into a token-level redundancy score for confidence reweighting. This simple reweighting preserves confidence as the base reliability signal while making visually overlapping candidates less likely to be committed together. Our experiments show that VRCD lowers visual redundancy across decoding steps, improves task performance on multiple benchmarks, and retains near-baseline throughput. These results suggest that visual redundancy is a useful signal for parallel decoding in visually grounded dMLLMs.

Our contributions are summarized as follows:

- We observe visual redundancy among textual tokens committed together in the same decoding step, an overlooked phenomenon in dMLLM parallel decoding where individually confident tokens can rely on overlapping visual grounding.

- We introduce the *Visual Redundancy Index* (VRI), a metric that quantifies visual grounding overlap among tokens unmasked in parallel and enables systematic analysis of redundancy across decoding steps.
- We propose *Visual-Redundancy-Controlled Decoding* (VRCD), a lightweight training-free reranking method that uses attention-derived visual overlap to prioritize high-confidence but visually complementary positions under the same decoding schedule.

2 Related Work

From autoregressive decoding to masked diffusion language models. Autoregressive language models generate tokens under a left-to-right factorization, whereas masked diffusion language models generate by iteratively decoding masked textual positions. Recent work also studies block-structured generation with Block Diffusion [Arriola et al., 2025]. More centrally for masked diffusion language modeling, Masked Discrete Diffusion for Discrete Data (MD4) [Shi et al., 2024] and Masked Diffusion Language Models (MDLM) [Sahoo et al., 2024] are particularly important: the former provides a simplified and general masked-diffusion formulation for discrete data, while the latter shows that masked diffusion can be made simple and effective for language modeling. Recent systems such as LLaDA [Nie et al., 2025] and Dream [Ye et al., 2025] further scale masked diffusion language modeling to large language models through modern pretraining, instruction tuning, or adaptation from autoregressive checkpoints. This shift makes decoding order a controllable choice rather than a left-to-right convention. Recent analyses study generation order [Zhang et al., 2026], parallelism and generation order [Zhong et al., 2026], and dependency-guided decoding [Ringel et al., 2026]. Together, they show that decoding choices affect both efficiency and decoding quality.

Parallel decoding for diffusion language models. A major advantage of diffusion language models is that they can decode multiple positions in one step, reducing the number of decoding iterations and improving inference speed. However, the standard masked-token objective factorizes over masked tokens, so parallel decoding uses marginal predictions for several positions at once rather than a joint distribution over those tokens. For dependent positions, this approximation can introduce bias; recent analyses of truly parallel decoding [Li et al., 2026] and masked diffusion parallelism [Zhong et al., 2026] also report that high parallelism can reduce performance in current dLLMs. Decoding schedules therefore try to manage the tradeoff between inference speed and quality by changing which positions are unmasked. Existing methods adapt the number or layout of parallel updates, as in APD [Israel et al., 2025] and Plan for Speed [Luxembourg et al., 2025]. Others choose positions using Lookahead Unmasking [Lee et al., 2025], information-gain sampling [Yang et al., 2026], Bits2Rounds [Fu et al., 2025], or learned unmasking policies [Jazbec et al., 2025]. These methods show that the parallel decoding strategy is important, but they still rely mainly on textual signals.

Multimodal diffusion models and visual redundancy. Autoregressive MLLMs are important for multimodal understanding and generation and have been used in downstream tasks [Wu et al., 2026, Wang et al., 2026b,a]. Recent work also extends diffusion language modeling to multimodal understanding and generation. Dimple [Yu et al., 2025] introduces a discrete diffusion multimodal LLM with parallel decoding. LLaDA-V [You et al., 2025] equips language diffusion models with visual instruction tuning, and LaViDa [Li et al., 2025d] adapts large diffusion language models for multimodal understanding. MMaDA [Yang et al., 2025b] and Lumina-DiMOO [Xin et al., 2025] further broaden diffusion models toward unified multimodal reasoning and generation. In parallel, visual redundancy in multimodal LLMs has mostly been studied as redundancy in the visual input or in visual-token processing. Visual Context Compression [Chen et al., 2024a] and PyramidDrop [Xing et al., 2024] reduce or remove redundant visual context or tokens, while RedundancyLens [Li et al., 2025b] and FALCON [Zhang et al., 2025] analyze or mitigate redundant visual processing in decoder-only or high-resolution MLLMs. Beyond Intermediate States [Yang et al., 2025a] explains redundant visual tokens through textual descriptions. For dMLLMs, a recent visual-token redundancy study [Li et al., 2025a] analyzes how visual token redundancy evolves across architectures and decoding steps. Sparse-LaViDa [Li et al., 2025c] accelerates multimodal diffusion sampling by truncating unnecessary masked tokens, while RedVTP [Xu et al., 2025] prunes visual tokens using masked text-token attention during diffusion VLM inference. Our focus is different: rather than pruning or compressing visual tokens themselves, we study whether *text tokens committed together* in a parallel decoding step are associated with overlapping visual grounding.

3 Method

3.1 Problem Setup

We consider dMLLM decoding conditioned on an image I, a text prompt P, and a partially decoded token sequence $\mathbf{x}^{(t)}$ at step t. Let C_t denote the set of positions that remain masked at this step. For each position $i \in C_t$, the decoder predicts a token distribution $p_i(v) = p(x_i = v \mid I, P, \mathbf{x}^{(t)})$, from which we obtain the most likely token $\hat{x}_i = \arg\max_v p_i(v)$ and its confidence score $c_i = p_i(\hat{x}_i)$. Let K_t denote the number of tokens to commit at step t, as specified by the decoding schedule. Confidence-based unmasking then commits the K_t highest-confidence predictions:

$$S_t = \mathrm{top}_{K_t}(\{(i, c_i) \mid i \in C_t\}). \tag{1}$$

Confidence-based unmasking is efficient, but it ranks candidate tokens independently by confidence and does not account for how much visual grounding is repeated among the tokens selected for the same decoding step.

3.2 Visual Redundancy Within Decoding Steps

We formalize the visual redundancy illustrated in Fig. 1(a) as an analysis metric for a realized decoding step. Given the positions S_t committed at decoding step t, let $v_i(u) \geq 0$ denote how strongly textual token i is associated with image token u, where $u \in \{1, \ldots, N\}$ and $\sum_{u=1}^{N} v_i(u) = 1$. We instantiate v_i with normalized token-to-image attention as a proxy for this association [Kim et al., 2026]. Let $m = |S_t|$. We define the *Visual Redundancy Index* (VRI) of S_t as

$$\mathrm{VRI}(S_t) = \begin{cases} 0, & \text{if } m \leq 1, \\ \dfrac{\sum_{u=1}^{N} \left(\sum_{i \in S_t} v_i(u) - \max_{i \in S_t} v_i(u) \right)}{m - 1}, & \text{if } m \geq 2. \end{cases} \tag{2}$$

With normalized v_i, $\mathrm{VRI}(S_t) \in [0, 1]$. For each image token u, $\sum_{i \in S_t} v_i(u)$ is the total attention weight that the committed tokens assign to u, and subtracting $\max_{i \in S_t} v_i(u)$ leaves the weight from all committed tokens except the one that attends to u most. A smaller VRI therefore indicates lower visual redundancy among the tokens in S_t. VRCD, introduced next, reduces this redundancy with a token-level redundancy-controlled score computed before the tokens are committed.

3.3 Visual-Redundancy-Controlled Decoding

We propose *Visual-Redundancy-Controlled Decoding* (VRCD), a decoding method that controls visual redundancy during parallel decoding. VRCD keeps confidence as the base reliability signal and adjusts the ranking only within a window of high-confidence candidates, so it retains the efficiency of confidence-based unmasking. As shown in Fig. 2, each decoding step builds this candidate window, extracts visual saliency from token-to-image attention, and reweights confidence by a redundancy score derived from pairwise visual overlap.

Candidate window. Estimating visual redundancy over all masked positions is unnecessary because many low-confidence tokens are unlikely to be committed at the current step. We therefore build a candidate window

$$U_t = \mathrm{top}_{M_t}(\{(i, c_i) \mid i \in C_t\}), \qquad M_t = \min(|C_t|,\ \max(K_t, \lceil \lambda K_t \rceil)). \tag{3}$$

The parameter λ controls the size of this window. This focuses redundancy estimation on positions likely to be committed at the current step and reduces the computation required by VRCD.

Visual saliency extraction. For each $i \in U_t$, we use the token-to-image attention from the final decoder layer and average it across heads. We denote the resulting normalized attention weight on image token u by $a_i(u)$. To reduce attention noise, we first subtract $1/N$ from each image-token attention weight and keep only positive residual attention weights:

$$\bar{a}_i(u) = \max\left(a_i(u) - \frac{1}{N},\ 0 \right). \tag{4}$$

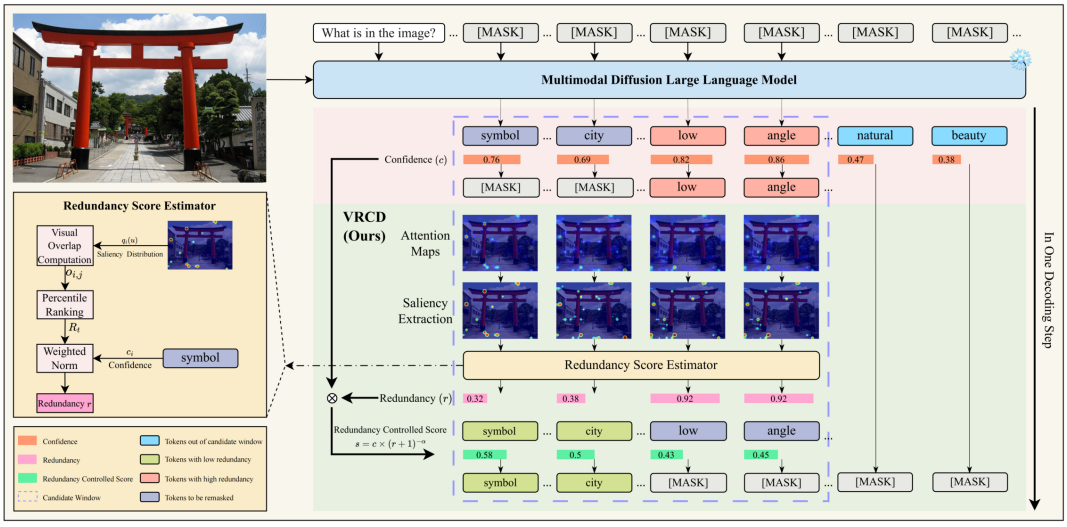


Figure 2: **Overview of VRCD:** at each decoding step, VRCD extracts visual saliency from token-to-image attention, estimates pairwise visual overlap, converts the overlaps into redundancy scores, and reweights confidence to favor visually complementary positions under the same decoding schedule. In the illustrated step, confidence-based unmasking commits the two most confident candidates, *low* and *angle* (red), whose visual saliency overlaps strongly; VRCD commits *symbol* and *city* (green), which overlap less with the other candidates. **Example output of VRCD:** The image captures a striking red torii gate, a **symbol** of welcome in Shinto culture, located in a Japanese **city**. The gate, with its vibrant red color, stands out against the backdrop of the cityscape. The gate is framed by lush **green trees**, adding a touch of natural beauty to the scene. Green words mark where VRCD selects different tokens from confidence-based unmasking.

We then renormalize the positive residual attention when at least one image token has positive residual mass:

$$q_i(u) = \begin{cases} \dfrac{\bar{a}_i(u)}{\sum_{u'=1}^{N} \bar{a}_i(u')}, & \text{if } \sum_{u'=1}^{N} \bar{a}_i(u') > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Thus, q_i is a normalized visual saliency distribution over image tokens after subtracting the uniform component. If all residual attention weights are non-positive, q_i becomes a zero vector and the candidate is excluded from the pairwise overlap computation. This extraction highlights visual overlap, helping the redundancy score identify visual redundancy among candidates.

Redundancy-controlled score. Given $\{q_i\}_{i \in U_t}$, let $U_t^{\text{vis}} = \{i \in U_t \mid \|q_i\|_1 > 0\}$. For each pair $i, j \in U_t^{\text{vis}}$, we compute their pairwise visual overlap as $o_{ij} = \sum_{u=1}^{N} \sqrt{q_i(u)\, q_j(u)}$. Let $\mathcal{O}_t$ collect all pairwise overlaps in the current candidate window. We convert each overlap to $R_t(i,j) = \text{PctRank}(o_{ij}; \mathcal{O}_t)$, where PctRank returns its percentile among $\mathcal{O}_t$. Thus, a larger $R_t(i,j)$ means that the two candidates share stronger visual saliency than most candidate pairs in the same step. This percentile rank makes pairwise overlaps comparable within the current candidate window. We define a token-level *redundancy score* for each candidate as

$$r_i = \begin{cases} 0, & \text{if } i \notin U_t^{\text{vis}} \text{ or } B_i = \emptyset \text{ or } Z_i = 0, \\ \sum_{j \in B_i} \hat{c}_{j|i} R_t(i,j), & \text{otherwise.} \end{cases} \tag{6}$$

Here $B_i = U_t^{\text{vis}} \setminus \{i\}$, $Z_i = \sum_{\ell \in B_i} c_\ell$, and $\hat{c}_{j|i} = c_j / Z_i$ when $Z_i > 0$. The confidence weight $\hat{c}_{j|i}$ reflects that token i should receive a larger redundancy score when it strongly overlaps with high-confidence candidates, since these candidates are more likely to be selected for unmasking in the current decoding step. In this way, r_i measures how strongly token i's visual grounding overlaps with the visual grounding of other likely candidates in the current window.

We then reweight the original confidence using this redundancy score and define the redundancy-controlled score as

$$s_i = c_i \, (r_i + 1)^{-\alpha}, \tag{7}$$

where $\alpha \geq 0$ controls how strongly r_i changes the confidence score. The term $r_i + 1$ keeps the reweighting factor bounded and leaves $s_i = c_i$ when $r_i = 0$. A large r_i indicates that the visual grounding associated with token i strongly overlaps with that of other likely candidates in the current window. VRCD ranks candidates by s_i and selects the K_t positions with the highest redundancy-controlled scores for unmasking, so candidates with stronger visual redundancy are less likely to be committed together in the same decoding step.

3.4 Understanding VRCD from an Information-Theoretic Perspective

This subsection explains why controlling visual redundancy can improve parallel decoding. Let X_i denote the token at masked position $i \in C_t$, and write $\mathcal{I}$ for mutual information to avoid confusion with the image I; we omit the conditioning on I, P, and $\mathbf{x}^{(t)}$. Parallel decoding commits the tokens in S_t from their per-position predictions and thus treats them as independent. Even if each prediction equals its true marginal, the KL divergence between their product and the joint distribution of the committed tokens equals the dependence among them [Ben-Hamu et al., 2025, Zhang et al., 2026]. This dependence also enters the uncertainty left for the remaining masked positions:

$$\mathcal{E}(S_t) = \mathrm{KL}\left(p(x_{S_t}) \,\middle\|\, \textstyle\prod_{i \in S_t} p(x_i)\right) = \sum\nolimits_{i \in S_t} H(X_i) - H(X_{S_t}), \tag{8}$$

$$H\left(X_{C_t \setminus S_t} \,\middle|\, X_{S_t}\right) = H(X_{C_t}) - \sum\nolimits_{i \in S_t} H(X_i) + \mathcal{E}(S_t). \tag{9}$$

Confidence-based unmasking favors tokens with low individual entropy, which keeps the bound $\mathcal{E}(S_t) \leq \sum_{i \in S_t} H(X_i) - \max_{i \in S_t} H(X_i)$ small [Ben-Hamu et al., 2025]. The confidence score, however, is computed for each position separately and cannot account for $\mathcal{E}(S_t)$ itself.

Visual redundancy as dependence. Tokens committed in the same step are dependent when they are predicted from the same uncertain visual content. Let V be a latent visual fact that the text has not yet fixed at step t, such as an attribute of an object. Decomposing $\mathcal{E}(S_t)$ with respect to V gives

$$\mathcal{E}(S_t) = \mathcal{E}(S_t \mid V) + \big[\, \sum\nolimits_{i \in S_t} \mathcal{I}(X_i; V) - \mathcal{I}(X_{S_t}; V) \big], \tag{10}$$

where $\mathcal{E}(S_t \mid V)$ is the dependence that remains once V is known, such as grammatical agreement. When positive, the bracketed term is visual information that the committed tokens convey more than once. It is large when several tokens state the same visual fact, such as the color of one object, and vanishes when the tokens depend on different regions whose contents are independent. Visual redundancy within a decoding step therefore contributes to $\mathcal{E}(S_t)$: the step commits the same visual fact more than once, and the committed tokens resolve less uncertainty than their individual entropies indicate.

Redundancy control in VRCD. VRCD acts on the visual overlap among tokens committed in the same step. Its redundancy-controlled score s_i down-weights candidates whose extracted visual saliency overlaps with that of other likely candidates, so the tokens committed together draw on more distinct visual evidence. We take a lower pairwise visual overlap as an indication that the bracketed term in Eq. (10) is smaller. The committed tokens then share less visual information, and their dependence $\mathcal{E}(S_t)$ decreases. By Eq. (8), this reduces the error of committing the tokens in parallel; by Eq. (9), for a comparable $\sum_{i \in S_t} H(X_i)$, it also leaves less uncertainty to the remaining masked positions, so later predictions draw on more informative context. We empirically verify these effects in Sec. 4.3 using VRI and remaining-position entropy.

3.5 Further Discussion of Visual Redundancy

Sec. 3.4 reads a lower attention overlap as less shared visual information. This raises two questions: whether attention overlap reflects shared visual grounding, and when that grounding is redundant.

Token-to-image attention is a noisy proxy for visual grounding. Nearly uniform attention, common for format tokens (e.g., numeric indices or punctuation) and function words (e.g., "and" or "of"), overlaps strongly while carrying little visual information. VRCD therefore computes overlap on

the extracted saliency q_i (Eqs. (4) and (5)), which removes this uniform component. Sec. 4.5 shows that the resulting pairwise overlap tracks whether the annotated image regions of two tokens coincide, and the ablation in Sec. 4.4 shows that the extraction makes the overlap more useful for reranking.

Shared visual grounding is redundant when several tokens state the same visual fact: the bracketed term in Eq. (10) is then large, and under the same per-step commit size K_t these tokens displace candidates that would add other evidence. It is necessary when tokens grounded in the same region state different facts, such as an identity and an attribute of one object, or the parts of one time value read from a table; the bracketed term is then small despite large overlap (Appendix A.8). The redundancy score r_i measures overlap and cannot separate these cases, so VRCD uses it as a soft signal: confidence remains the base score, and α sets how strongly r_i reweights it. Larger α favors candidates with more complementary visual grounding at the cost of admitting candidates with lower confidence, while smaller α keeps the ranking closer to confidence-based unmasking; Sec. 4.4 examines this trade-off. The full pseudocode, implementation details, and complexity analysis are provided in Appendices A.1 and A.2.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate on five benchmarks that cover complementary multimodal abilities. **M^{3}CoT** [Chen et al., 2024b] tests multi-domain, multi-step multimodal reasoning with multiple-choice answers. **MMBench** [Liu et al., 2024] measures general vision-language understanding across perception and reasoning skills. **SQA-IMG** [Lu et al., 2022] denotes the image-context subset of ScienceQA, requiring models to answer multiple-choice science questions grounded in diagrams or other visual context. **DocVQA** [Mathew et al., 2021] evaluates document question answering over text-rich page images, and **InfoVQA** [Mathew et al., 2022] evaluates question answering over infographics with mixed text, layout, and visual elements. We report each benchmark's primary score, using ANLS for DocVQA and InfoVQA. All benchmarks use their official prompt formats; answer extraction and judging details are described in Appendix A.2.

Models and baselines. We use `lavida-llada-reason` [Li et al., 2025d] as the default backbone in the main experiments. All methods in the main comparison use the same model weights, prompt format, generation length limits, decoding schedules, and evaluation pipeline. In dMLLM decoding, each step predicts all currently masked positions, unmasks a subset according to the decoding schedule, and leaves the remaining positions masked for later refinement. We therefore describe local baselines as remasking policies. **Confidence** denotes confidence-based unmasking, which commits the K_t positions with the highest predicted probabilities. **Margin** selects positions with the largest gap between the two most likely tokens, while **Entropy** selects positions with the lowest predictive entropy; both are local uncertainty criteria computed from the masked-diffusion predictive distribution [Sahoo et al., 2024]. **IG** denotes InfoGain [Yang et al., 2026], which uses information gain to choose update positions. **VRCD** is our method, which reweights confidence using visual redundancy.

Forward ratio and decoding length. We measure inference cost by the forward ratio (FR). For maximum decoding length L and T model forward passes, $\mathrm{FR} = T/L$. The main comparison uses $L \in \{192, 384\}$ and fixed decoding schedules with $\mathrm{FR} \in \{0.125, 0.25, 0.5\}$, corresponding to default per-step commit sizes of 8, 4, and 2 tokens.

Metrics and hyperparameters. We report VRI by decoding step and remaining-position entropy. The mechanism analysis is run on M^{3}CoT for $L \in \{192, 384\}$ and $\mathrm{FR} \in \{0.25, 0.5\}$; the mean-VRI table reports both lengths, and Fig. 3 shows the curves for $L = 192$ and $\mathrm{FR} = 0.25$. Unless otherwise stated, VRCD uses $\lambda = 2.0$ in the candidate-window size definition in Eq. (3) and $\alpha = 1.5$ in the redundancy-controlled score in Eq. (7) across benchmarks and FR settings.

4.2 Main Comparison

We vary FR to test the setting where dMLLMs commit multiple tokens in parallel: lower FR values use larger per-step commit sizes, making the choice of which positions are committed together more

Table 1: Main comparison with decoding length 192. Best values are shown in bold.

	M^{3}CoT			MMBench			SQA-IMG			DocVQA			InfoVQA		
Method / FR	0.125	0.25	0.5	0.125	0.25	0.5	0.125	0.25	0.5	0.125	0.25	0.5	0.125	0.25	0.5
Confidence	36.43	37.68	34.81	53.95	51.14	44.52	55.68	57.11	50.04	46.81	50.43	53.68	28.95	31.70	32.93
Margin	35.86	36.20	36.83	55.77	51.40	51.21	56.24	57.28	54.37	46.85	50.71	53.90	29.09	31.69	32.70
Entropy	32.95	33.48	27.91	49.68	41.68	34.86	51.69	48.24	32.04	46.72	49.95	53.00	29.18	31.30	32.69
IG	36.66	37.39	34.56	54.06	51.16	45.68	55.53	57.31	50.67	46.81	50.70	53.80	29.02	31.69	**33.26**
VRCD	**37.94**	**40.38**	**39.60**	**58.79**	**59.37**	**57.50**	**60.09**	**63.38**	**58.64**	**49.27**	**52.30**	**54.34**	**29.93**	**32.61**	33.06

Table 2: Main comparison with decoding length 384. Best values are shown in bold.

	M^{3}CoT			MMBench			SQA-IMG			DocVQA			InfoVQA		
Method / FR	0.125	0.25	0.5	0.125	0.25	0.5	0.125	0.25	0.5	0.125	0.25	0.5	0.125	0.25	0.5
Confidence	28.82	37.16	38.35	57.29	63.04	66.91	53.93	62.96	66.06	46.45	50.41	53.74	29.01	31.83	32.81
Margin	34.11	38.06	38.47	60.64	66.43	68.48	**63.38**	**64.98**	65.54	46.54	50.57	53.74	29.03	31.87	32.96
Entropy	26.33	34.57	35.86	50.30	57.48	69.02	42.44	54.47	63.56	46.39	49.93	52.93	28.89	31.57	32.79
IG	28.72	37.26	39.23	57.54	63.35	67.73	54.12	62.30	66.57	46.48	50.51	54.05	28.94	31.81	**33.37**
VRCD	**34.24**	**39.82**	**41.99**	**61.24**	**66.73**	**69.87**	56.62	64.49	**68.43**	**49.22**	**52.24**	**54.33**	**30.37**	**32.55**	32.94

consequential. VRCD improves over confidence-based unmasking at every FR setting and at both decoding lengths (Tabs. 1 and 2).

The gains are largest on M^{3}CoT, MMBench, and SQA-IMG, whose answers often involve longer multimodal reasoning. Tokens committed early in this reasoning become context for later steps, so the choice of which tokens are committed together carries over to the rest of decoding. Gains on DocVQA and InfoVQA are smaller, since their prompts tend to elicit short answers and leave fewer later positions for this effect. VRCD also improves over the MMaDA baseline in most settings (Appendix A.4).

Throughput. Tab. 3 reports throughput on full M^{3}CoT with decoding length $L = 192$ and $\mathrm{FR} = 0.5$. VRCD adds about 1.5% runtime overhead over confidence-based unmasking. In contrast, IG [Yang et al., 2026] incurs a much larger slowdown under the same setting. These results show that VRCD remains a lightweight decoding method. Complexity and measurement details are deferred to Appendix A.1.

Table 3: Throughput on M^{3}CoT ($L = 192$, $\mathrm{FR} = 0.5$).

Method	tokens/s
LaViDa	46.982
LaViDa+IG	24.884
LaViDa+VRCD	46.264

4.3 Redundancy and Predictive Certainty

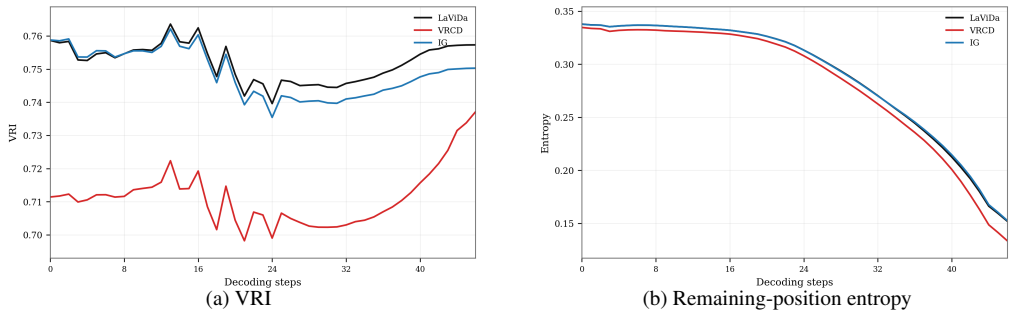
We study the two effects described in Sec. 3.4 on M^{3}CoT: whether tokens committed in the same step share less visual grounding, measured by VRI, and whether the remaining masked positions are predicted with less uncertainty, measured by remaining-position entropy.

Reducing VRI during decoding. In all four settings of Tab. 4, VRCD has a lower mean VRI than both LaViDa and IG [Yang et al., 2026]. Fig. 3(a) tracks VRI by decoding step at $L = 192$ and $\mathrm{FR} = 0.25$. VRCD stays below LaViDa throughout decoding, while IG, which also changes the selection criterion, stays close to LaViDa. Reweighting confidence by visual overlap thus changes which tokens are committed together, and the tokens committed in the same step repeat less visual grounding.

Table 4: Mean VRI on M^{3}CoT.

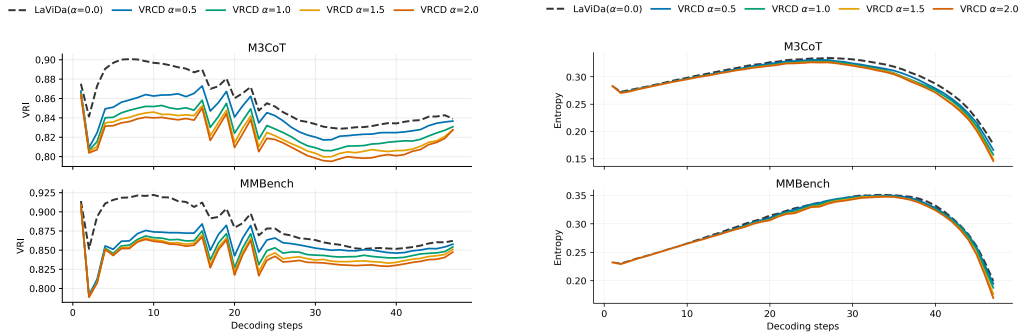
L	FR	LaViDa	IG	VRCD
192	0.25	0.752	0.749	**0.712**
192	0.5	0.685	0.679	**0.588**
384	0.25	0.784	0.783	**0.743**
384	0.5	0.720	0.718	**0.623**

Increasing later predictive certainty. VRCD also leaves less uncertainty for the remaining masked positions. In Fig. 3(b), its remaining-position entropy stays below that of LaViDa and IG throughout decoding, and the gap widens in later steps. Together with the lower VRI, this is in line with the second effect in Sec. 3.4: when tokens committed in the same step share less visual information, later predictions are made with greater certainty.



(a) VRI (b) Remaining-position entropy

Figure 3: M^{3}CoT VRI and entropy curves ($L = 192$, FR $= 0.25$).



(a) VRI (b) Remaining-position entropy

Figure 4: Effect of α on M^{3}CoT and MMBench ($L = 384$, FR $= 0.125$).

4.4 Ablations

We ablate the three design choices that determine VRCD's reranking behavior: the exponent α in the redundancy-controlled score in Eq. (7), redundancy score aggregation, and visual saliency extraction. The α sweep uses M^{3}CoT and MMBench at decoding length $L = 384$ and FR $= 0.125$, the setting with the largest per-step commit size. The component ablations for redundancy score aggregation and visual saliency extraction use the same FR setting and report both $L = 192$ and $L = 384$.

Effect of α. We evaluate $\alpha \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$ under the default configuration. Here $\alpha = 0.0$ corresponds to confidence-based unmasking, while larger α gives the redundancy score r_i more influence in Eq. (7). Tab. 5 reports accuracy for each value; parenthesized values denote gains over $\alpha = 0.0$. On M^{3}CoT, $\alpha = 1.5$ is best in this grid, while on MMBench $\alpha = 2.0$ is best. We therefore keep $\alpha = 1.5$ as a fixed moderate setting across benchmarks and FR settings. Fig. 4 reports the corresponding VRI and remaining-position entropy curves.

Table 5: Effect of α.

α	M^{3}CoT	MMBench
0.0	28.82	57.29
0.5	29.97 (+1.15)	59.03 (+1.74)
1.0	31.54 (+2.72)	59.79 (+2.50)
1.5	**34.24 (+5.42)**	61.24 (+3.95)
2.0	33.45 (+4.63)	**62.69 (+5.40)**

Redundancy score aggregation. To test whether r_i should account for how likely an overlapping neighbor is to be committed in the same decoding step, we compare uniform average aggregation with the default confidence-weighted aggregation. The average variant averages $R_t(i, j)$ over $j \in B_i$, while the weighted variant uses the normalized confidence weights in Eq. (6). Tab. 6 reports each entry as $L = 192/384$ and shows that confidence-weighted aggregation performs better on both benchmarks.

Table 6: Redundancy score aggregation.

Agg.	M^{3}CoT (192/384)	MMBench (192/384)
Average	37.35/32.17	58.10/60.63
Weighted	**37.94/34.24**	**58.79/61.24**

Table 7: Visual saliency extraction (VSE).

Variant	M^{3}CoT (192/384)	MMBench (192/384)
Confidence	36.43/28.82	53.95/57.29
VRCD (w/o VSE)	37.59/34.13	58.25/61.00
VRCD	**37.94/34.24**	**58.79/61.24**

Visual saliency extraction (VSE). We compare VRCD with VRCD (w/o VSE), a variant that computes pairwise visual overlap from raw token-to-image attention without this module. Tab. 7 reports both decoding lengths. Confidence and VRCD values are taken from Tabs. 1 and 2. VRCD (w/o VSE) is consistently below VRCD, indicating that VSE makes the overlap estimate more useful in these settings.

4.5 Attention Overlap as a Visual-Grounding Proxy

The redundancy score of VRCD is useful only if attention overlap reflects shared visual grounding (Sec. 3.5). A proxy for shared grounding should have two properties: tokens grounded in the same image region should overlap more than tokens grounded in different regions, and this difference should come from the image content. We test both on Flickr30K Entities [Plummer et al., 2015], which links caption phrases to annotated image boxes. In each of 307 test images, we choose two phrases whose boxes overlap little, mask two tokens in each phrase, and let the model predict them with the other caption words visible; the boxes are used only for evaluation. This yields same-phrase pairs, which refer to one annotated region, and cross-phrase pairs, which refer to two largely separate regions. For each pair, we compute the pairwise visual overlap o_{ij} as VRCD does (Sec. 3.3) and keep the 510 pairs in which both tokens are predicted correctly.

Attention overlap separates the two kinds of pairs. The mean o_{ij} of same-phrase pairs is 0.148 higher than that of cross-phrase pairs, and o_{ij} is positively correlated with the overlap between the annotated regions of the two tokens (Spearman $\rho = 0.484$); the 95% bootstrap intervals of both lie above zero. To check that this separation comes from the image, we replace each image with an unrelated one and keep the caption and mask positions unchanged. The gap then falls from 0.148 to 0.073, so the image content accounts for about half of the separation; the correlation ρ drops as well. Pairwise attention overlap is therefore an effective proxy for shared visual grounding: it distinguishes tokens grounded in the same region from tokens grounded in different regions, and the distinction depends on the image. This is the property that the redundancy score of VRCD relies on. Appendix A.6 provides more details.

5 Conclusion

We studied visual redundancy within decoding steps in diffusion-based multimodal large language models. Our analysis shows that confidence-based unmasking can commit several individually confident tokens whose attention-derived visual saliency overlaps within the same step. To quantify this behavior, we introduced the Visual Redundancy Index (VRI), and we proposed VRCD, a lightweight inference-time decoding method that estimates pairwise visual overlap from token-to-image attention and reweights confidence under the same decoding schedule.

Empirically, VRCD reduces visual redundancy across decoding steps and is accompanied by lower entropy over remaining masked positions, consistent with higher predictive certainty for later predictions. It also improves decoding quality while retaining near-baseline throughput. Together, these experiments demonstrate that visual redundancy serves as a useful signal for deciding which positions are committed in parallel. More broadly, this work shifts the decoding question from whether individual predictions are confident to whether a parallel decoding step introduces complementary visual grounding for the remaining generation process. Future work can explore adaptive multimodal decoding methods that use visual redundancy to guide both which positions are committed together and the per-step commit size within the decoding schedule.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62306277. We declare no competing interests.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian

Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. URL https://arxiv.org/abs/2204.14198.

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. URL https://arxiv.org/abs/2308.12966.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas S. Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. In *Advances in Neural Information Processing Systems*, volume 38, pages 55981–56007, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/510e22f4a2da5212a64f1591736e2eaf-Paper-Conference.pdf.

Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. Efficient large multi-modal models via visual context compression. *arXiv preprint arXiv:2406.20092*, 2024a.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M^3CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8199–8221. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.acl-long.446.

Ziyu Chen, Xinbei Jiang, Peng Sun, and Tao Lin. Optimizing decoding paths in masked diffusion models by quantifying uncertainty. *arXiv preprint arXiv:2512.21336*, 2025.

Hengyu Fu, Baihe Huang, Virginia Adams, Charles Wang, Venkat Srinivasan, and Jiantao Jiao. From bits to rounds: Parallel decoding with exploration for diffusion language models. *arXiv preprint arXiv:2511.21103*, 2025.

Feng Hong, Geng Yu, Yushi Ye, Haicheng Huang, Huangjie Zheng, Ya Zhang, Yanfeng Wang, and Jiangchao Yao. Wide-in, narrow-out: Revokable decoding for efficient and effective DLLMs. *arXiv preprint arXiv:2507.18578*, 2025.

Daniel Israel, Guy Van den Broeck, and Aditya Grover. Accelerating diffusion LLMs via adaptive parallel decoding. *arXiv preprint arXiv:2506.00413*, 2025.

Metod Jazbec, Theo X. Olausson, Louis Bethune, Pierre Ablin, Michael Kirchhof, Joao Monteiro, Victor Turrisi, Jason Ramapuram, and Marco Cuturi. Learning unmasking policies for diffusion language models. *arXiv preprint arXiv:2512.09106*, 2025.

Bumjun Kim, Dongjae Jeon, Moongyu Jeon, and Albert No. Dependency-aware parallel decoding via attention for diffusion LLMs. *arXiv preprint arXiv:2603.12996*, 2026.

Sanghyun Lee, Seungryong Kim, Jongho Park, and Dongmin Park. Lookahead unmasking elicits accurate decoding in diffusion language models. *arXiv preprint arXiv:2511.05563*, 2025.

Duo Li, Zuhao Yang, Xiaoqin Zhang, Ling Shao, and Shijian Lu. A comprehensive study on visual token redundancy for discrete diffusion-based multimodal large language models. *arXiv preprint arXiv:2511.15098*, 2025a.

Hongliang Li, Jiaxin Zhang, Wenhui Liao, Dezhi Peng, Kai Ding, and Lianwen Jin. RedundancyLens: Revealing and exploiting visual token processing redundancy for efficient decoder-only MLLMs. *arXiv preprint arXiv:2501.19036*, 2025b.

Pengxiang Li, Dilxat Muhtar, Lu Yin, Tianlong Chen, and Shiwei Liu. Why diffusion language models struggle with truly parallel (non-autoregressive) decoding? *arXiv preprint arXiv:2602.23225*, 2026.

Shufan Li, Jiuxiang Gu, Kangning Liu, Zhe Lin, Zijun Wei, Aditya Grover, and Jason Kuen. SparseLaViDa: Sparse multimodal discrete diffusion language models. *arXiv preprint arXiv:2512.14008*, 2025c.

Shufan Li, Konstantinos Kallidromitis, Hritik Bansal, Akash Gokul, Yusuke Kato, Kazuki Kozuka, Jason Kuen, Zhe Lin, Kai-Wei Chang, and Aditya Grover. LaViDa: A large diffusion language model for multimodal understanding. *arXiv preprint arXiv:2505.16839*, 2025d.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pages 216–233. Springer, 2024.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 2507–2521. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/11332b6b6cf4485b84afadb1352d3a9a-Paper-Conference.pdf.

Omer Luxembourg, Haim Permuter, and Eliya Nachmani. Plan for speed: Dilated scheduling for masked diffusion language models. *arXiv preprint arXiv:2506.19037*, 2025.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

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. *arXiv preprint arXiv:2502.09992*, 2025.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2641–2649, 2015.

Liran Ringel, Ameen Ali, and Yaniv Romano. Dependency-guided parallel decoding in discrete diffusion language models. *arXiv preprint arXiv:2604.02560*, 2026.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.04329.

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

Ziyi Wang, Peiming Li, Xinshun Wang, Yang Tang, Kai-Kuang Ma, and Mengyuan Liu. Universal skeleton understanding via differentiable rendering and MLLMs. *arXiv preprint arXiv:2603.18003*, 2026a. URL https://arxiv.org/abs/2603.18003.

Ziyi Wang, Xinshun Wang, Shuang Chen, Yang Cong, and Mengyuan Liu. UniMotion: A unified framework for motion-text-vision understanding and generation. *arXiv preprint arXiv:2603.22282*, 2026b. URL https://arxiv.org/abs/2603.22282.

Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, Chengxiang Zhai, and Klara Nahrstedt. VTool-R1: VLMs learn to think with images via reinforcement learning on multimodal tool use. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Idst6X6gmy.

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

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

Jingqi Xu, Jingxi Lu, Chenghao Li, Sreetama Sarkar, Souvik Kundu, and Peter A. Beerel. RedVTP: Training-free acceleration of diffusion vision-language models inference via masked token-guided visual token pruning. *arXiv preprint arXiv:2511.12428*, 2025.

Dingchen Yang, Bowen Cao, Anran Zhang, Weibo Gu, Winston Hu, and Guang Chen. Beyond intermediate states: Explaining visual redundancy through language. *arXiv preprint arXiv:2503.20540*, 2025a.

Kaisen Yang, Jayden Teoh, Kaicheng Yang, Yitong Zhang, and Alex Lamb. Improving sampling for masked diffusion models via information gain. *arXiv preprint arXiv:2602.18176*, 2026.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal large diffusion language models. In *Advances in Neural Information Processing Systems*, volume 38, pages 138867–138907, 2025b. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/caa934a507a952698d54efb24845fc4b-Paper-Conference.pdf.

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

Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. LLaDA-V: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. MM-Vet: Evaluating large multimodal models for integrated capabilities. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57730–57754. PMLR, 2024. URL https://proceedings.mlr.press/v235/yu24o.html.

Renshan Zhang, Rui Shao, Gongwei Chen, Miao Zhang, Kaiwen Zhou, Weili Guan, and Liqiang Nie. FALCON: Resolving visual redundancy and fragmentation in high-resolution multimodal large language models via visual registers. *arXiv preprint arXiv:2501.16297*, 2025.

Shaorong Zhang, Longxuan Yu, Rob Brekelmans, Luhan Tang, Salman Asif, and Greg Ver Steeg. Generation order and parallel decoding in masked diffusion models: An information-theoretic perspective. *arXiv preprint arXiv:2602.00286*, 2026.

Yangyang Zhong, Yanmei Gu, Zhengqing Zang, Xiaomeng Li, Yuqi Ding, Xibei Jia, et al. Parallelism and generation order in masked diffusion language models: Limits today, potential tomorrow. *arXiv preprint arXiv:2601.15593*, 2026.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. URL https://arxiv.org/abs/2304.10592.

A Appendix

A.1 Full VRCD Procedure and Implementation Details

This section provides the full pseudocode of VRCD and specifies the implementation details that are summarized in the main paper.

Algorithm 1 Visual-Redundancy-Controlled Decoding (VRCD) at decoding step t

Require: masked positions C_t, per-step commit size K_t, predicted tokens $\{\hat{x}_i\}_{i\in C_t}$, confidences $\{c_i\}_{i\in C_t}$, window multiplier λ, exponent α

$M_t \leftarrow \min(|C_t|, \max(K_t, \lceil \lambda K_t \rceil))$
$U_t \leftarrow$ the M_t positions in C_t with the largest confidences c_i
for each $i \in U_t$ **do** ▷ extract visual saliency
    Extract normalized head-averaged final-layer attention $a_i(u)$ over image tokens
    Compute $q_i(u)$ according to Eq. (5)
end for
$U_t^{\text{vis}} \leftarrow \{i \in U_t \mid \|q_i\|_1 > 0\}$ ▷ candidates with nonzero visual saliency
for each $i \in U_t$ **do**
    $R_t(i,i) \leftarrow 0$
end for
for each pair (i,j) with $i,j \in U_t^{\text{vis}}$ and $i<j$ **do** ▷ compute all pairwise overlaps
    $o_{ij} \leftarrow \sum_{u=1}^{N} \sqrt{q_i(u)\, q_j(u)}$
end for
$\mathcal{O}_t \leftarrow \{o_{ab} \mid a,b \in U_t^{\text{vis}},\ a<b\}$
for each pair (i,j) with $i,j \in U_t^{\text{vis}}$ and $i<j$ **do** ▷ rank overlaps after all pairs are available
    $R_t(i,j) \leftarrow \mathrm{PctRank}(o_{ij}; \mathcal{O}_t)$
    $R_t(j,i) \leftarrow R_t(i,j)$
end for
for each $i \in U_t$ **do** ▷ compute normalized confidence-weighted redundancy scores
    $B_i \leftarrow U_t^{\text{vis}} \setminus \{i\}$ and $Z_i \leftarrow \sum_{\ell \in B_i} c_\ell$
    $r_i \leftarrow 0$ if $i \notin U_t^{\text{vis}}$, $B_i = \emptyset$, or $Z_i = 0$; otherwise $r_i \leftarrow \sum_{j\in B_i} (c_j/Z_i) R_t(i,j)$
    $s_i \leftarrow c_i (r_i + 1)^{-\alpha}$
end for
Commit the K_t predicted tokens in U_t with the largest redundancy-controlled scores s_i

Implementation details. In all experiments, the candidate window is rebuilt at every decoding step using the current masked positions and the current K_t. Candidates with no positive residual attention have $q_i = \mathbf{0}$. They are excluded from the pairwise overlaps and from every B_i, receive $r_i = 0$ (Eq. (6)), and remain eligible for selection with $s_i = c_i$. The percentile rank $\mathrm{PctRank}(o_{ij}; \mathcal{O}_t)$ is the rank of o_{ij} in $\mathcal{O}_t$, counted from 0 and divided by $|\mathcal{O}_t| - 1$. Tied values share their average rank, and all ranks are 0 when $\mathcal{O}_t$ has fewer than two distinct values. VRCD does not add extra model-evaluation forward passes; the additional computation is limited to attention post-processing, pairwise overlap calculation, and confidence reweighting inside the current candidate window. The main experiments were run on NVIDIA L40 GPUs, and the other experiments were run on NVIDIA GeForce RTX 4090 GPUs.

Attention-map processing. At each decoding step, VRCD takes the final-layer attention from each position $i \in U_t$ to the N image tokens, averages it over heads, and normalizes it to sum to one, which gives $a_i(u)$. This attention comes from the forward pass that produces the confidences. The number N depends on the aspect ratio of the image (Tab. 8). VRCD subtracts $1/N$ from each $a_i(u)$, keeps the positive part, and renormalizes it to obtain q_i (Eqs. (4) and (5)). The pairwise visual overlap $o_{ij} = \sum_{u=1}^{N} \sqrt{q_i(u)\, q_j(u)}$ is the Bhattacharyya coefficient between two extracted visual saliency distributions. It is 1 when they coincide and 0 when they share no image token. VRI applies Eq. (2) to $v_i = a_i$ without the $1/N$ subtraction. The redundancy score r_i and the proxy validation (Appendix A.6) use q_i and o_{ij}. The proxy validation restricts q_i to the spatial image tokens, where the boxes are defined, and renormalizes it.

Computational cost and throughput. Let $M_t = |U_t|$ and let N be the number of image tokens. After the model forward pass produces token-to-image attention, visual saliency extraction costs $O(M_t N)$. Computing pairwise overlaps over the current candidate window costs $O(M_t^2 N)$, and ranking the overlaps and computing r_i add $O(M_t^2 \log M_t)$. Since $M_t = O(K_t)$ by construction, the additional post-processing scales with the candidate window rather than with all masked positions. Tab. 3 reports decoding throughput, the number of tokens committed per second of decoding time, which includes this post-processing.

A.2 Detailed Experimental Configuration

The default model checkpoint is `lavida-llada-reason`, i.e., LaViDa-Reason with LLaDA-8B as the diffusion language backbone [Li et al., 2025d, Nie et al., 2025]. The vision encoder is SigLIP-400M within LaViDa's any-resolution pipeline, which encodes three to five views per image (Tab. 8).

Table 8: Fixed implementation configuration used across experiments.

Setting	Value
Model checkpoint	`lavida-llada-reason` (LaViDa-Reason)
Diffusion language backbone	LLaDA-8B [Nie et al., 2025]
Vision encoder	SigLIP-400M [Li et al., 2025d]
Image preprocessing	any-resolution: a global view plus two to four 384×384 crops, each bilinearly downsampled to 14×14 tokens
Visual sequence	depends on the aspect ratio; 1008 positions for a square image (980 spatial tokens, 28 row separators)
Attention source for VRCD	final decoder layer, averaged over attention heads
Evaluation implementation	official prompts and scoring code where available; `qwen3:8b` for final-answer extraction on M^{3}CoT, MMBench, and SQA-IMG; GPT-5.6 as judge for the open-ended benchmarks

All methods in the main comparison share the same backbone, prompt format, decoding schedule, per-step commit-size schedule, generation limits, and evaluation pipeline. We report $L = 192$ and $L = 384$. Unless ablated, VRCD fixes $\lambda = 2.0$ in Eq. (3) and $\alpha = 1.5$ in Eq. (7) across all datasets and FR settings, without per-benchmark tuning. The supplementary λ sweep uses M^{3}CoT at $L = 192$ and $\mathrm{FR} \in \{0.125, 0.25, 0.5\}$. The α sweep uses M^{3}CoT and MMBench at $L = 384$ and $\mathrm{FR} = 0.125$, and component ablations use both datasets at both lengths under the same FR setting.

Local remasking scores and remaining-position entropy. For every decoding step t and masked position $i \in C_t$, all local remasking policies use the same predictive distribution $p_i(v) = p(x_i = v \mid I, P, \mathbf{x}^{(t)})$ over the vocabulary $\mathcal{V}$. Let $v_i^{(1)}$ and $v_i^{(2)}$ be the most likely and second most likely tokens under p_i. The confidence score is

$$c_i = \max_{v \in \mathcal{V}} p_i(v) = p_i(v_i^{(1)}), \tag{11}$$

and confidence-based unmasking selects the K_t positions with the largest c_i. The margin score is

$$m_i = p_i(v_i^{(1)}) - p_i(v_i^{(2)}), \tag{12}$$

and the margin baseline selects the K_t positions with the largest m_i. The entropy baseline uses normalized predictive entropy

$$h_i = -\frac{1}{\log |\mathcal{V}|} \sum_{v \in \mathcal{V}} p_i(v) \log p_i(v), \tag{13}$$

and selects the K_t positions with the smallest h_i. The normalization in Eq. (13) only rescales entropy values and does not change the ranking within a decoding step.

For the remaining-position entropy curves, after a method chooses which positions to unmask at step t, let $C_{n,t}^{\mathrm{rem}}$ be the positions that remain masked for sample n. We compute this quantity on

non-terminal decoding steps, so $|C_{n,t}^{\mathrm{rem}}| > 0$. The per-sample remaining-position entropy is

$$e_{n,t} = \frac{1}{|C_{n,t}^{\mathrm{rem}}|} \sum_{i \in C_{n,t}^{\mathrm{rem}}} h_{n,t,i}, \tag{14}$$

where $h_{n,t,i}$ is Eq. (13) for position i in sample n at step t. The plotted value at decoding step t is the average of $e_{n,t}$ over samples at that step.

Benchmark prompting and answer extraction. All benchmarks use official prompt formats, and we use official scorers when available. For M^{3}CoT, MMBench, and SQA-IMG, we use `qwen3:8b` only to map generated text to a final multiple-choice option before scoring, with instructions not to infer a new answer from the question, image, or outside knowledge. GPT-5.6 judges the open-ended benchmarks in Appendix A.7.2.

A.3 Limitations, Societal Impact, and Assets

Limitations. VRCD changes which masked positions are committed together under a given decoding schedule, but it does not adapt the decoding schedule or the number of positions committed at each step. If the schedule commits too many or too few positions for a task, VRCD can only rerank the available candidates within that schedule. Dynamically choosing how many tokens to commit could require searching over candidate subsets, whose cost can grow combinatorially with the candidate window size. We leave adaptive commit-size selection to future work. VRCD also relies on the quality of token-to-image attention: when attention is diffuse or misaligned with the relevant image regions, attention overlap is a weaker proxy for shared visual grounding, and our test of this proxy against human region annotations checks correspondence rather than causal attribution (Appendix A.6). In addition, the gains are small in some open-ended settings with stochastic sampling, and VRCD with $\alpha = 1.5$ scores below confidence-based unmasking in one deterministic and one sampled setting (Appendix A.7.2).

Ethics and broader impacts. This work proposes an inference-time decoding method for existing dMLLMs and does not introduce a new dataset, train a new model, or collect new human-subject data. A positive impact is that VRCD can improve multimodal decoding quality without additional model training and with near-baseline throughput in our measurement. A potential negative impact is that better multimodal decoding can also improve systems used in settings where visual or document understanding errors affect users, including sensitive document analysis. The method should therefore be evaluated in the target domain before use in high-stakes settings, and it should be paired with the safety practices and usage restrictions of the underlying model and datasets.

Existing assets and licenses. Tab. 9 summarizes the public license or terms information we found for the main assets used in the experiments. We use all datasets only for evaluation and do not redistribute benchmark data.

A.4 Additional Backbone Evaluation on MMaDA

We apply VRCD to MMaDA-8B-MixCoT [Yang et al., 2025b] and compare it with the MMaDA baseline under the same prompt format, decoding schedule, per-step commit-size schedule, generation limits, and evaluation pipeline. VRCD uses the same $\alpha = 1.5$ and $\lambda = 2.0$ as with LaViDa. Prompts follow the released MMaDA MixCoT evaluation setting, which adds the official chain-of-thought instruction to multiple-choice and VQA prompts. The evaluation covers M^{3}CoT, MMBench, SQA-IMG, DocVQA, and InfoVQA, with $L \in \{192, 384\}$ and $\mathrm{FR} \in \{0.125, 0.25, 0.5\}$. VRCD improves over the MMaDA baseline in most settings (Tab. 10).

A.5 Candidate Window Size and Position Changes

Candidate window size. We vary the window multiplier λ on M^{3}CoT at decoding length $L = 192$. The grid uses $\lambda \in \{1.5, 2.0, 2.5, 3.0\}$ with $\mathrm{FR} \in \{0.125, 0.25, 0.5\}$. We exclude $\lambda = 1.0$, which would leave the candidate window identical to the positions selected by confidence-based unmasking. Tab. 11 reports M^{3}CoT accuracy. In this grid, $\lambda = 2.5$ gives the best value at all three FR settings.

Table 9: Existing assets used in the experiments and public license or terms information. "Not specified" means that we did not find an explicit license field on the public page used for that asset during manuscript preparation.

Asset	Use in this paper	Public license or terms noted
`lavida-llada-reason`	Backbone checkpoint [Li et al., 2025d]	License not specified on the public Hugging Face checkpoint page.
LaViDa codebase [Li et al., 2025d]	Base of the released VRCD code	Apache-2.0 license on the public LaViDa GitHub repository.
LLaDA-8B [Nie et al., 2025]	Diffusion language backbone used by LaViDa	MIT license on the public LLaDA-8B model page.
SigLIP-400M	Vision encoder used by LaViDa	Apache-2.0 license on the public SigLIP model page.
MMaDA-8B-MixCoT [Yang et al., 2025b]	Backbone for Appendix A.4	MIT license on the public MMaDA-8B-MixCoT model page.
`qwen3:8b`	Final-answer extraction for M^{3}CoT, MMBench, and SQA-IMG	Apache-2.0 license on the public Qwen3-8B model page.
GPT-5.6	Automatic judge for the open-ended benchmarks	Proprietary OpenAI model without public weights; accessed through OpenAI's Codex service and used under OpenAI's terms of use.
M^{3}CoT [Chen et al., 2024b]	Multimodal reasoning benchmark	MIT license on the public M^{3}CoT dataset page.
MMBench [Liu et al., 2024]	General vision-language benchmark	Apache-2.0 license on the public MMBench repository.
ScienceQA [Lu et al., 2022]	Multimodal science question answering benchmark	The public ScienceQA repository states that the code is MIT licensed and that the dataset is licensed under CC BY-NC-SA 4.0.
DocVQA [Mathew et al., 2021]	Document question answering benchmark	The public LMMs-Lab mirror lists Apache-2.0; the official DocVQA page states that task images are from the UCSF Industry Documents Library.
InfoVQA [Mathew et al., 2022]	Infographic question answering benchmark	The official InfographicVQA page describes challenge access and states that images are collected from the Internet; no separate license field is listed there.
MM-Vet [Yu et al., 2024]	Open-ended benchmark (Appendix A.7.2)	The public MM-Vet repository states that the code is under the Apache-2.0 license and the dataset under the CC BY-NC 4.0 license.
LLaVA-Bench-in-the-Wild [Liu et al., 2023]	Open-ended benchmark (Appendix A.7.2)	Apache-2.0 license on the public Hugging Face dataset page.
Flickr30K Entities [Plummer et al., 2015]	Proxy validation (Appendix A.6)	The public repository states that use of the images must abide by the Flickr Terms of Use and that the images are provided for non-commercial research and educational purposes; no separate license is listed for the annotations.

Table 10: Additional MMaDA backbone evaluation. MMaDA denotes the native confidence-based unmasking baseline; MMaDA+VRCD applies the same visual-redundancy-controlled decoding criterion under the matched MMaDA setting. Values are task scores, with ANLS used for DocVQA and InfoVQA. Best values in each matched setting are bolded.

	M^{3}CoT			MMBench			SQA-IMG			DocVQA			InfoVQA		
Method / FR	0.125	0.25	0.5	0.125	0.25	0.5	0.125	0.25	0.5	0.125	0.25	0.5	0.125	0.25	0.5
$L = 192$															
MMaDA	24.42	27.61	28.42	52.09	53.45	50.66	44.75	43.86	**49.22**	8.49	10.18	11.19	7.65	10.08	12.40
MMaDA+VRCD	**26.01**	**29.09**	**29.31**	**53.45**	**54.84**	**52.87**	**47.51**	**45.10**	46.79	**9.09**	**11.38**	**11.88**	**8.40**	**12.05**	**13.33**
$L = 384$															
MMaDA	24.68	25.50	26.85	50.90	50.82	44.65	47.08	43.10	45.51	9.27	11.23	11.64	8.43	10.87	11.80
MMaDA+VRCD	**26.19**	**28.24**	**28.03**	**52.81**	**53.52**	**51.32**	**48.29**	**46.47**	**47.43**	**9.58**	**11.74**	**12.38**	**8.86**	**12.05**	**14.20**

Table 11: M^{3}CoT accuracy under different λ values ($L = 192$).

FR	Confidence	$\lambda = 1.5$	$\lambda = 2.0$	$\lambda = 2.5$	$\lambda = 3.0$
0.125	36.43	37.17	37.94	**38.09**	37.59
0.25	37.68	38.05	40.38	**40.81**	40.38
0.5	34.81	33.49	39.60	**40.51**	39.69

Position-change rate. We measure how often VRCD changes which positions are committed in parallel relative to confidence-based unmasking. For sample n and decoding step t of a VRCD run, let $S_{n,t}^{\mathrm{VRCD}}$ be the positions that VRCD commits and $S_{n,t}^{\mathrm{conf}}$ the top-$K_{n,t}$ positions by confidence in

the same decoding state. We aggregate over non-terminal decoding steps $\mathcal{T} = \{(n,t) : 0 < K_{n,t} < |C_{n,t}|\}$, where more candidate positions are available than can be committed at that step.

$$D_{n,t} = K_{n,t} - \left|S_{n,t}^{\text{conf}} \cap S_{n,t}^{\text{VRCD}}\right|, \tag{15}$$

$$\bar{D} = \frac{\sum_{(n,t)\in\mathcal{T}} D_{n,t}}{|\mathcal{T}|}, \qquad \eta = \frac{\sum_{(n,t)\in\mathcal{T}} D_{n,t}}{\sum_{(n,t)\in\mathcal{T}} K_{n,t}}. \tag{16}$$

These quantities are computed on M^{3}CoT and MMBench for $L \in \{192, 384\}$ and FR $\in \{0.125, 0.25\}$, as shown in Tab. 12. Here $D_{n,t}$ counts the positions that VRCD commits outside the top-$K_{n,t}$ positions by confidence, $\bar{D}$ averages this count over the analyzed steps, and the position-change rate η is the fraction of committed positions that VRCD changes. The rate measures how strongly VRCD alters the position selection.

Table 12: Position-change statistics for VRCD. $\bar{D}$ is the average position-change count per decoding step, and η is the position-change rate.

Dataset	L	FR	$\bar{D}$	η
M^{3}CoT	192	0.125	1.31	0.164
M^{3}CoT	192	0.25	1.03	0.258
M^{3}CoT	384	0.125	1.41	0.177
M^{3}CoT	384	0.25	0.96	0.240
MMBench	192	0.125	1.09	0.137
MMBench	192	0.25	0.79	0.197
MMBench	384	0.125	1.28	0.160
MMBench	384	0.25	0.89	0.222

A.6 Validation of Attention Overlap as a Visual-Grounding Proxy

VRCD uses the pairwise visual overlap o_{ij} between extracted visual saliency distributions as a proxy for shared visual grounding. This section details the test in Sec. 4.5.

Setup. (i) *Data.* From the 1,000 test images of Flickr30K Entities [Plummer et al., 2015], we select one caption per image with two phrases that pass length, scene, and box-size filters and whose boxes barely overlap (Bhattacharyya coefficient of the box masks at most 0.10); 307 images qualify. (ii) *Masking.* We mask the first and the last content token of each phrase, keep the rest of the caption visible, and prompt the model to complete the description from the image. One forward pass predicts the four masked tokens, and we take their final-layer attention as in Appendix A.1. (iii) *Pairs.* We keep the same-phrase and cross-phrase pairs whose two tokens are both predicted correctly, so that the boxes of each phrase annotate the tokens the model predicts; this leaves 510 pairs from 216 of the 307 images. (iv) *Measures.* For each pair, we compute the attention overlap o_{ij} and the annotated-region overlap, i.e., the same Bhattacharyya coefficient applied to the boxes of the two phrases mapped onto the image tokens (1 for same-phrase pairs). We report the gap in mean o_{ij} between same-phrase and cross-phrase pairs and the Spearman correlation ρ between the two overlaps. (v) *Control.* We replace each image with another of the 307 images, resized to the original size so that the image-token layout is unchanged, and recompute o_{ij} for the same pairs.

Table 13: Attention overlap on Flickr30K Entities (510 token pairs, final decoder layer). Brackets give 95% bootstrap intervals over images (10,000 resamples).

Image given to the model	Same-phrase minus cross-phrase gap	Spearman ρ
Original image	0.148 [0.127, 0.169]	0.484 [0.405, 0.559]
Unrelated image	0.073 [0.056, 0.091]	0.343 [0.251, 0.434]
Original minus unrelated	0.075 [0.057, 0.092]	0.141 [0.064, 0.220]

Results. In Tab. 13, the 95% intervals of the same-phrase minus cross-phrase gap, of ρ, and of their reductions under an unrelated image all lie above zero. The correlation ρ also partly reflects how far

apart the two tokens are in the caption and how confident their predictions are. The test evaluates pairwise overlap, the quantity that VRCD uses.

A.7 Token Sampling, Open-Ended Generation, and Decoder Layer

The main experiments rest on three defaults: every method commits the most likely token at each selected position, the benchmarks score multiple-choice or short answers, and VRCD takes visual saliency from the final decoder layer. This section relaxes each default in turn. VRCD keeps its gains over confidence-based unmasking when the committed tokens are sampled (Appendix A.7.1) and when answers are free-form (Appendix A.7.2), and the final decoder layer gives more useful visual saliency than the two alternatives we compare (Appendix A.7.3). Unless stated otherwise, all runs use `lavida-llada-reason`, $L = 192$, $\lambda = 2.0$, $\alpha = 1.5$, and the decoding schedules of Sec. 4; sampled runs draw each committed token by Top-k ($k = 50$) or Top-p ($p = 0.9$) sampling at temperature 1.0.

A.7.1 Stochastic Token Sampling

VRCD decides which positions are committed together, while the token at each committed position comes from the model's prediction. To check that the gains do not depend on committing the most likely token, we sample the committed tokens and compare VRCD with confidence-based unmasking on M^{3}CoT. We sample one output per question, score it three times with the answer-extraction pipeline of the main experiments, and report the mean. VRCD scores higher than confidence-based unmasking at every FR under both samplers (Tab. 14). Choosing which positions to commit together thus remains effective when the committed tokens are sampled.

Table 14: M^{3}CoT accuracy with stochastic token sampling ($L = 192$, $\alpha = 1.5$). Each entry is the mean of three scoring runs on one sampled output per question. Bold: better result in each column for each sampler.

Sampler	Method / FR	0.125	0.25	0.5
Top-k ($k = 50$)	Confidence	30.57	35.13	33.07
	VRCD	**33.72**	**38.48**	**36.86**
Top-p ($p = 0.9$)	Confidence	33.10	35.29	34.01
	VRCD	**34.77**	**37.86**	**37.83**

A.7.2 Open-Ended Generation

To see whether the gains extend to free-form answers, we evaluate MM-Vet [Yu et al., 2024] (218 questions) and LLaVA-Bench-in-the-Wild [Liu et al., 2023] (LLaVA-Wild, 60 questions), judged by GPT-5.6 with the official prompts and rubrics. MM-Vet reports an overall score on a 0–100 scale, and LLaVA-Wild reports a relative score, the mean rating of the generated answers as a percentage of that of the reference answers. We compare VRCD with confidence-based unmasking and Margin under deterministic decoding and under the two samplers of Appendix A.7.1. Deterministic entries average at least three judge runs of one output, and sampled entries average three seeds with three judge runs each. Besides the default $\alpha = 1.5$, we report $\alpha = 2.0$, the best value on MMBench in Tab. 5.

Results. VRCD improves over confidence-based unmasking in open-ended generation under both deterministic decoding and sampling (Tab. 15): with $\alpha = 2.0$ it scores higher in every setting, and with $\alpha = 1.5$ in all but two. VRCD also outperforms Margin in most settings, although Margin scores highest in the sampled settings at $\mathrm{FR} = 0.5$. Because VRCD changes only which positions are committed together, these results show that reranking by visual overlap also benefits free-form generation.

A.7.3 Decoder Layer for Visual Saliency

VRCD takes visual saliency from the token-to-image attention of the final decoder layer (Sec. 3.3), the same layer whose output gives the confidences that VRCD reweights. To check whether another layer gives more useful visual saliency for the redundancy score, we replace the final layer with one

Table 15: Open-ended generation on MM-Vet and LLaVA-Wild ($L = 192$). Entries are GPT-5.6 judge scores (higher is better): the overall score on MM-Vet and the relative score on LLaVA-Wild. Det.: deterministic decoding. Bold: best score in each column, per benchmark.

	Det.			Top-k ($k = 50$)			Top-p ($p = 0.9$)		
Method / FR	0.125	0.25	0.5	0.125	0.25	0.5	0.125	0.25	0.5
MM-Vet									
Confidence	24.62	23.20	24.79	22.02	19.91	21.64	20.76	20.22	22.37
Margin	24.67	24.13	22.64	20.65	22.53	**23.51**	22.61	22.58	**23.88**
VRCD ($\alpha = 1.5$)	26.93	25.46	24.58	21.51	**23.04**	22.32	22.78	23.10	23.03
VRCD ($\alpha = 2.0$)	**28.67**	**26.30**	**25.02**	**22.48**	22.23	22.18	**22.98**	**23.14**	23.33
LLaVA-Wild									
Confidence	16.39	18.84	15.97	15.91	16.79	16.91	16.68	16.23	17.72
Margin	**19.67**	17.39	17.68	14.68	14.75	**19.42**	15.84	16.42	**19.95**
VRCD ($\alpha = 1.5$)	19.20	19.11	**18.79**	**17.93**	17.74	18.64	16.76	**18.79**	18.50
VRCD ($\alpha = 2.0$)	18.36	**19.50**	18.49	17.57	**19.00**	19.20	**17.60**	18.10	18.93

of two alternatives: a single layer at about two-thirds of the depth, or the attention averaged uniformly over the last third of the layers. The final layer gives the highest M^{3}CoT accuracy at all three FR values (Tab. 16); the last-third average comes close, and the two-thirds-depth layer is the lowest at every FR. Attention closer to the output thus gives a more useful estimate of visual overlap, and we keep the final layer as the default.

Table 16: VRCD accuracy on M^{3}CoT ($L = 192$, $\alpha = 1.5$) with visual saliency from three choices of decoder layers. Bold: best accuracy at each FR. The final-layer column repeats the VRCD results on M^{3}CoT in Tab. 1.

FR	Final layer	Two-thirds depth	Last-third average
0.125	**37.94**	32.96	37.76
0.25	**40.38**	35.36	39.72
0.5	**39.60**	34.60	38.75

A.8 When Is Visual Overlap Redundant?

Visual overlap plays two roles in parallel decoding. When tokens committed together repeat the same visual fact, the overlap is redundant: it enlarges the bracketed term in Eq. (10) and, under the same per-step commit size K_t, displaces candidates that would add new evidence. When tokens grounded in the same region state different facts, such as an identity and an attribute, or an action and a relation, the overlap reflects shared grounding that the answer needs (Sec. 3.5). The redundancy score sees only the overlap, which is why VRCD reweights confidence softly and leaves the strength to α. Below, we show where each role appears, first in the α ablation and then in individual questions.

The α ablation traces how the balance shifts as the control grows stronger (Tab. 5 and Fig. 4). On both benchmarks, a larger α lowers VRI. On MMBench, accuracy rises with α over the whole range, so removing repeated visual evidence keeps paying off. On M^{3}CoT, accuracy peaks at $\alpha = 1.5$ and dips at $\alpha = 2.0$ while VRI keeps falling, which indicates that the strongest setting begins to suppress shared grounding that the answers rely on. VRCD therefore keeps confidence as the base score and uses a moderate default: $\alpha = 1.5$ is the best setting on M^{3}CoT and still yields a clear gain on MMBench.

The case studies in Appendix A.9 show the same balance within single questions. In Fig. 6, the answer depends on two separate labels in the image, and lowering overlap lets VRCD state the evidence from both labels before it commits the answer. In Fig. 7, the answer is a single table cell whose tokens share the same grounding, and lowering their overlap splits one reading across steps. Together, the ablation and the cases show where lower overlap stops helping, and α controls how close VRCD comes to that point.

A.9 Case Studies

We present four cases: two MMBench examples that show where the outputs of the two methods begin to diverge (Fig. 5), a case where lower overlap helps (Fig. 6), and a failure case (Fig. 7). In all case figures, green marks the correct option and the output that selects it, and red marks the incorrect option and the output that selects it. For the two M^{3}CoT cases, we also compare, at each step, the VRI of the tokens that VRCD commits with that of the top-K_t positions by confidence in the same decoding state.

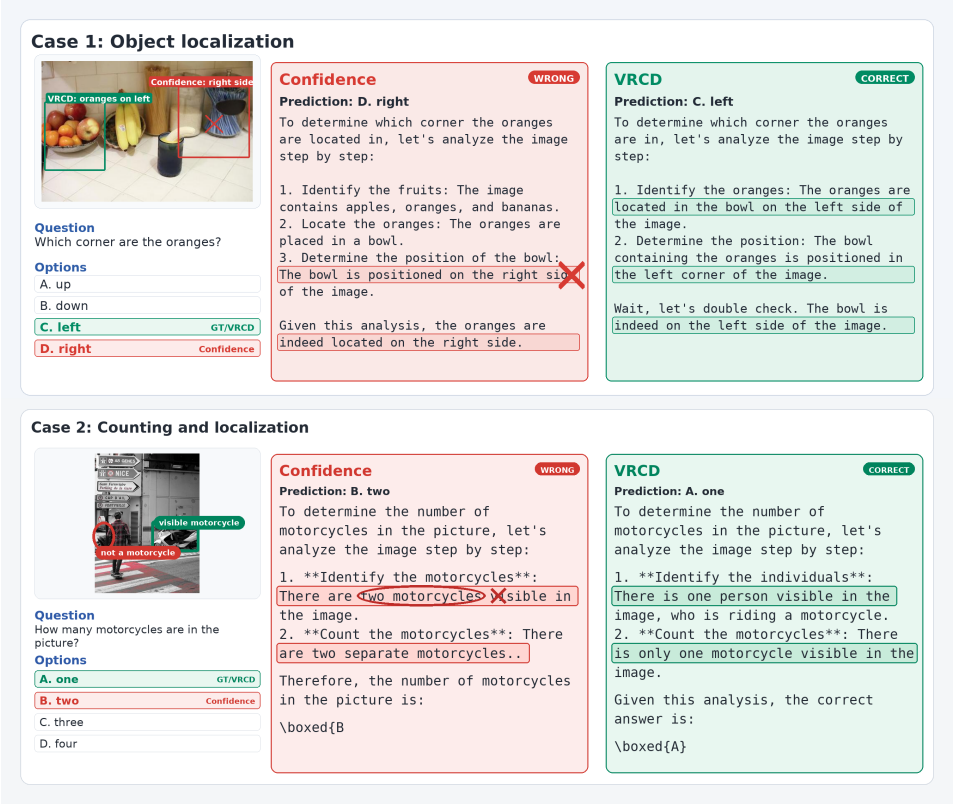


Figure 5: Qualitative MMBench examples. The left column shows the image, question, and choices; the red and green panels show output excerpts. In the object-localization example, confidence-based unmasking assigns the fruit bowl to the right side, while VRCD keeps the oranges on the left. In the counting example, confidence-based unmasking treats the partial car as another motorcycle, while VRCD keeps the count at the single visible motorcycle.

A case where lower overlap helps. Fig. 6 shows an M^{3}CoT question that compares two gas samples. The particles in both samples have the same average speed, and those in Sample A are heavier, so Sample A has the higher temperature (option A). The example comes from the M^{3}CoT runs used for Tab. 4 ($L = 384$, FR $= 0.5$, 192 decoding steps). Confidence-based unmasking commits the answer B at step 123, before its text states either mass. Its later text concludes that Sample A has the higher temperature, yet the committed B remains the final answer. VRCD first commits the digits of both masses (steps 124–129) and the statement that the two speeds are equal (steps 134–139), and it commits A at step 190. Across the decoding steps, the tokens that VRCD commits have a mean VRI of 0.590, against 0.698 for the top-K_t positions by confidence, and the VRCD tokens have the lower VRI in every step that commits a mass digit (for example, 0.384 against 0.584 at step 125).

A failure case. Fig. 7 shows an M^{3}CoT question that asks when the milking demonstration begins in a schedule table. The evidence lies in one row, and the answer is its Begin cell, 9:25 A.M. (option A). The setting is $L = 384$ and FR $= 0.125$, as in Tab. 2. Confidence-based unmasking commits

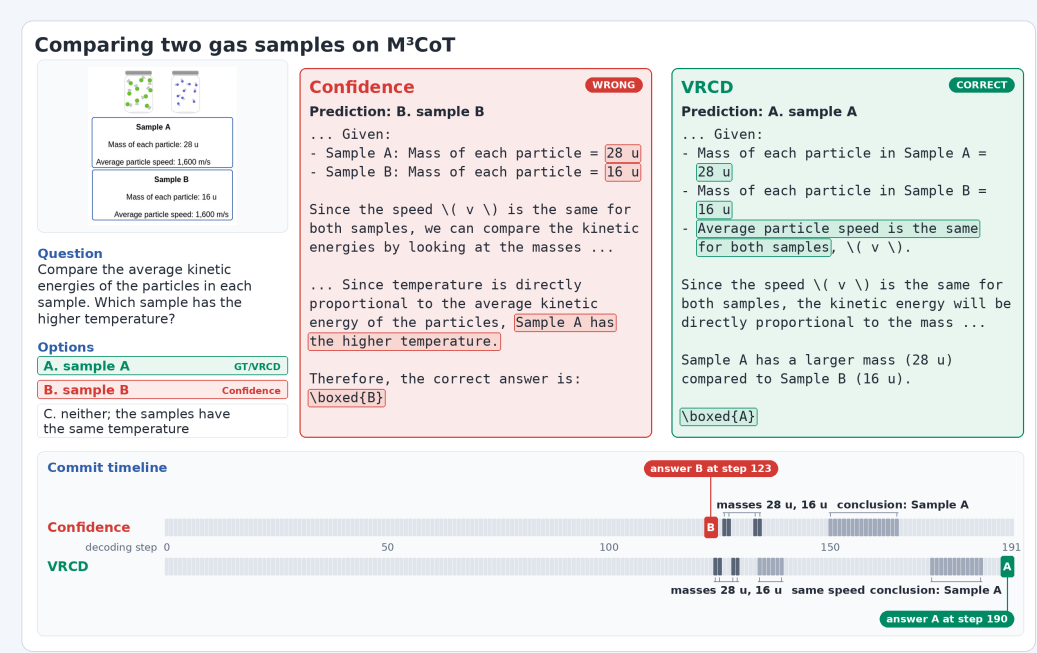


Figure 6: Gas-sample question from M^{3}CoT ($L = 384$, FR $= 0.5$). Left: image with the two sample labels enlarged, question, and options. Right: output excerpts with key phrases boxed. Bottom: the steps at which each method commits its answer, the two masses, and its conclusion; the VRCD timeline also marks the statement that the two speeds are equal. Confidence-based unmasking commits B at step 123, before either mass; VRCD commits A at step 190, after both masses and the equal speeds.

"9", "2", "5", and "AM" together at step 45 and states 9:25 AM, which corresponds to A. VRCD commits the parts of the time over steps 43–45. In each of these steps, the tokens that VRCD commits have lower VRI than the top-K_t positions by confidence (0.786 against 0.819, 0.698 against 0.733, and 0.783 against 0.790). Its text then reads "9:2:50 PM", which combines the Begin cell with the End cell 1:50 P.M., and VRCD selects B (1:50 PM). The tokens of one time value are grounded in the same cell and state different parts of one reading, so their overlap reflects necessary shared grounding (Appendix A.8). Lowering this overlap splits the reading across steps, and the image offers no complementary evidence to take its place.

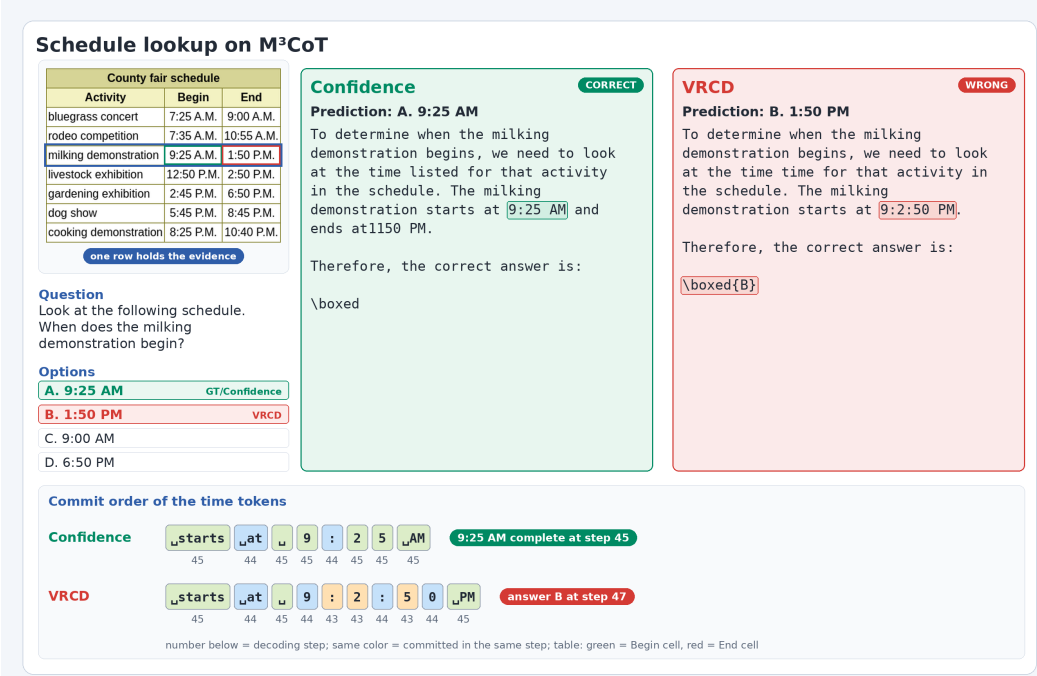


Figure 7: Schedule question from M^{3}CoT ($L = 384$, FR $= 0.125$). The boxes mark the evidence row and its Begin and End cells. Bottom: the decoding step at which each token of the stated time is committed. Confidence-based unmasking commits 9, 2, 5, and AM at step 45; VRCD spreads the time over steps 43–45 and writes 9:2:50 PM.

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 Introduction state the scope of VRCD as a training-free reranking method for dMLLM parallel decoding, and the Method and Experiments sections define and evaluate the corresponding claims.

   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: Appendix A.3 (Limitations, Societal Impact, and Assets) discusses that VRCD reranks candidates within a fixed decoding schedule rather than adapting the schedule or per-step commit size, and notes that target-domain evaluation is needed before use in high-stakes settings.

   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 present theoretical results or formal guarantees; it introduces a decoding method and evaluates it empirically.

   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: The method definition, pseudocode, decoding settings, benchmark list, metrics, answer extraction details, and hardware are described in Sections 3–4 and Appendix A.1–A.2.

   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: [No]

   Justification: The VRCD implementation is publicly available at `https://github.com/infiniteYuanyl/VRCD` under the Apache-2.0 license, with environment setup and inference instructions that use the default VRCD hyperparameters. The current release does not include the benchmark evaluation scripts (prompt construction, answer extraction, and scoring), so it does not reproduce the reported tables end to end; the benchmarks are public, and Section 4 and Appendix A.2 describe the evaluation configuration.

   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 backbone, benchmarks, metrics, decoding schedules, generation lengths, FR settings, and VRCD hyperparameters; Appendix A.2 gives additional implementation details, and Appendices A.4–A.7 describe the settings of the additional experiments.

   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: The main benchmark tables report single runs without error bars; the attention-overlap proxy test (Sec. 4.5 and Tab. 13) reports 95% bootstrap intervals.

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: [No]

   Justification: Appendix A.1 reports the GPU types used for the main comparison and auxiliary experiments, and Tab. 3 reports inference throughput for the measured setting, but the manuscript does not provide per-experiment wall-clock time or complete resource accounting for every run.

   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 work uses existing public benchmarks for evaluation and does not involve new human-subject data collection or high-risk data release; broader ethical considerations are discussed in Appendix A.3.

   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: Appendix A.3 discusses positive impacts from improving inference-time multimodal decoding and possible risks when stronger visual or document understanding is used in sensitive settings.

    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: The paper does not release a new model, dataset, or scraped data asset. It studies a decoding method evaluated on existing benchmarks.

    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: We credit the models, benchmarks, and evaluators we use. Tab. 9 in Appendix A.3 lists the public license or terms information found for each of them, including the LaViDa and MMaDA backbones, the benchmarks of the main and additional experiments (among them Flickr30K Entities, MM-Vet, and LLaVA-Bench-in-the-Wild), the `qwen3:8b` answer extractor, and the GPT-5.6 judge. Datasets are used only for evaluation.

    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: We release the VRCD code under the Apache-2.0 license at `https://github.com/infiniteYuanyl/VRCD`. The repository README documents environment setup, the inference command, and the decoding parameters, and it credits the LaViDa codebase on which the code is built. No new dataset or model is released.

    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: This research does not involve crowdsourcing or new human-subject studies.

    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: This research does not involve crowdsourcing or new human-subject studies, so IRB approval or an equivalent review 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: [N/A]

    Justification: The core method does not use an LLM as an important, original, or non-standard component. LLMs are used only for evaluation: `qwen3:8b` maps generated text to a final multiple-choice option on M^3CoT, MMBench, and SQA-IMG (Appendix A.2), and GPT-5.6 serves as the automatic judge for the open-ended MM-Vet and LLaVA-Bench-in-the-Wild evaluations (Appendix A.7.2).

    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.