VISWAVE: REVISABLE WAVEFRONT DECODING FOR DIFFUSION VISION–LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion multimodal large language models (Diffusion MLLMs) leverage bidirectional context modeling and intra-block multi-token denoising to provide an efficient text-generation path distinct from autoregressive decoding. However, long-form generation remains constrained by serial dependencies across blocks, causing inference efficiency to degrade as the output grows. Existing acceleration methods mainly reduce intra-block denoising cost and do not exploit computation that can overlap between adjacent blocks. The key obstacle is that an early-started successor depends on an unstable predecessor context and may become invalid as that context is updated. We find that this effect is usually local: most successor draft tokens remain reusable, while only a small number require re-verification. Based on this observation, we introduce VISWAVE, a training-free decoder for diffusion vision–language models. VISWAVE overlaps the decoding of adjacent blocks through revisable wavefront execution and uses virtual-remask token-to-token verification (VR-T2T) to re-evaluate affected draft tokens under the latest textual and visual context. Supported tokens are retained, while the remaining positions are remasked and returned to the native denoising process. This local selective revision makes inter-block parallelism reliable. Across seven natural workloads, VISWAVE accelerates BARD-VL-4B by $1.57\times$, with only a 0.27 percentage-point decrease in macro accuracy across six accuracy tasks; at a fixed generation length of 192 tokens, it reaches $2.07\times$ speedup. The efficiency gains remain stable from 2B to 8B and transfer to SDAR-VL.

1 INTRODUCTION

Masked diffusion language models refine many masked positions in parallel rather than commit one token at a time (Sahoo et al., 2024; Nie et al., 2025). Diffusion vision–language models (DVLMs) extend this generation paradigm to visual description, multimodal reasoning, and evidence generation (Cheng et al., 2025; Chen et al., 2026). To produce variable-length natural text, block-diffusion models partition the output into contiguous token blocks, denoise positions within each block in parallel, and append blocks sequentially (Arriola et al., 2025). Existing acceleration methods improve this process along several axes: adaptive caching and confidence-aware decoding reduce repeated denoising computation, visual-token pruning and dynamic token merging lower the cost of each multimodal forward, and recent work explores cross-block generation through additional training or token-level editing (Liu et al., 2025; Wu et al., 2026; Xu et al., 2025; Chang et al., 2025; Wang et al., 2025; Tian et al., 2026).

Despite this progress, two linked gaps remain. First, most training-free methods reduce computation within an active block or a single forward without changing the order in which blocks start and commit. DVLM parallelism consequently remains local to each block, while the sequential depth of long-form generation grows with the number of blocks. Second, existing cross-block methods either learn the transition through additional training or target token editing in text-only diffusion models; they do not provide frozen DVLMs with a reliable mechanism for reconciling early drafts under the latest image–text context. Current approaches therefore do not jointly provide training-free overlap across block boundaries and reliable reuse of provisional multimodal computation.

Simply decoding several blocks concurrently does not close these gaps. A successor started before its predecessor finishes can only draft against an evolving prefix, so part of that draft may become

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

stale when the predecessor commits. Accepting the entire draft can propagate stale state, whereas regenerating it wholesale repeats completed computation and forfeits the overlap. Crucially, predecessor commitment usually affects only a subset of successor positions rather than invalidating the whole block. Figure 1 exposes both sides of this opportunity: inter-block cost accumulates with output length, yet most early tokens remain reusable after the preceding block commits. This suggests a different execution principle: cross-block dependence should determine *which provisional positions are revalidated* under the mature context, rather than *when successor computation may begin*. Cross-block parallelism can then be reframed from a global waiting problem into a localized revision problem.

Based on this principle, we introduce VISWAVE, a training-free revisable wavefront decoder for diffusion vision–language models. VISWAVE overlaps successor drafting with predecessor decoding but keeps early outputs provisional; as the exposed predecessor context evolves, it revisits only affected successor positions. Virtual-remask token-to-token verification (VR-T2T) retains the logical draft while temporarily hiding each candidate from the model, forcing the frozen denoiser to reassess it from the latest text and image. Supported tokens are preserved, while unsupported positions are remasked and returned to native denoising. At predecessor commit, a final fused verification and synchronization step resolves remaining candidates before the successor advances. A commit invariant prevents unverified state from entering stable history, while fused preview and verified prefetch reuse predictions produced during validation.

Under a unified long-form multimodal generation protocol, VISWAVE accelerates BARD-VL-4B by $1.57\times$ on average across seven natural workloads while changing six-task macro accuracy by -0.27pp. At a controlled 192-token length, it reaches $2.07\times$ speedup with 54.5% fewer model forward evaluations. The gains persist from 2B to 8B and transfer to SDAR-VL. Token-level analyses further show that VR-T2T reliably distinguishes reusable from stale draft positions, while changing only the image used for verification systematically alters a subset of token decisions. Revision therefore remains conditioned on the latest multimodal context.

Our contributions are:

- We identify the mismatch between block-local parallelism and the serial inter-block path in long-form DVLM generation. Building on the observation that predecessor commitment typically invalidates only part of a successor draft, we formulate *revisable wavefront execution*, which turns cross-block dependence from a global waiting barrier into localized verification and safe commitment.
- We introduce VR-T2T, a dual-state verifier that removes visible-token self-conditioning while preserving reusable drafts, together with a commit invariant, fused preview, and verified prefetch that convert reliable multimodal revision into realized computation reuse.
- Extensive experiments show that VISWAVE consistently improves the quality–efficiency trade-off across multimodal understanding and long-form generation, remains effective across model scales and on SDAR-VL, and provides reliable, visually conditioned token-level revision.

2 BACKGROUND AND MOTIVATION

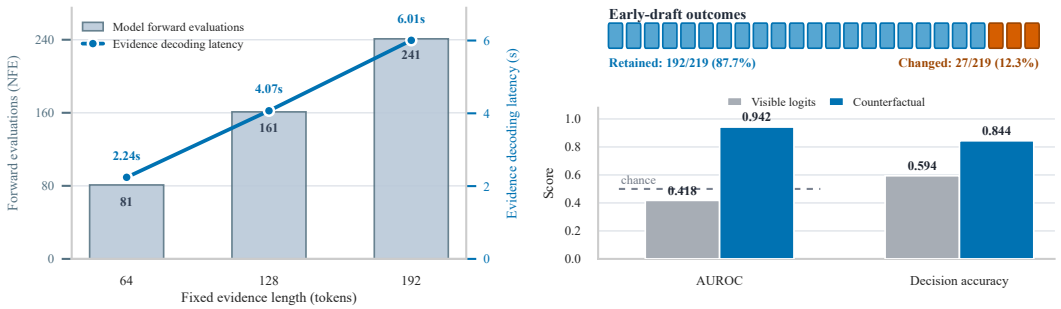
2.1 BLOCK-DIFFUSION GENERATION IN DVLMS

Masked diffusion language models recover text by iteratively refining masked positions rather than extending a sequence token by token (Sahoo et al., 2024; Nie et al., 2025). Diffusion vision–language models condition this refinement on both an image and a text prompt (Cheng et al., 2025; Chen et al., 2026). Given an image I and a question or instruction q, the model first extracts visual representations and uses them together with the textual input as the multimodal condition for denoising.

To produce responses beyond a single diffusion window, the output is divided into contiguous text blocks. When generating block B_m, the model conditions on the image, the prompt, and all previously completed blocks:

$$B_m = \mathrm{Decode}_\theta(I, q, B_{<m}). \tag{1}$$

Each block is initialized as masked positions and refined over multiple denoising steps. A forward pass can update several positions simultaneously, giving diffusion decoding its within-block token



(a) Serial cost scales with length **(b) Local staleness needs verification**

Figure 1: **Empirical motivation.** (a) Serial NFE and synchronized evidence-decoding latency grow with fixed evidence length. (b) Most early drafts survive predecessor commitment, while counterfactual scores reliably identify the changed minority.

parallelism. Once the current block is complete, it is appended to the stable text history and the next block is generated under the updated multimodal context. This process continues until an end-of-sequence token is produced or the generation budget is reached.

2.2 Serial Waiting and Sparse Revision

Our motivation follows two coupled observations. First, strict inter-block dependence forces each successor to wait for its predecessor, creating a serial delay that grows with output length. Second, predecessor commitment usually affects only a small set of positions in a successor draft, making whole-draft regeneration unnecessary.

Serial waiting. Although a DVLM updates multiple tokens in parallel within a block, adjacent blocks execute sequentially. If an output contains M blocks and each block requires K forward evaluations, the serial critical-path depth is approximately

$$D_{\text{serial}} = MK. \tag{2}$$

Within-block parallelism therefore leaves inter-block waiting intact. Figure 1(a) shows that increasing fixed evidence length from 64 to 192 tokens steadily increases both forward evaluations and synchronized decoding latency.

Starting a successor while its predecessor is still decoding can hide part of its computation behind the current block. Accounting for subsequent verification and revision, the resulting critical path is

$$D_{\text{wave}} = D_{\text{serial}} - \sum_{m=1}^{M-1} \Omega_m + D_{\text{verify}}, \tag{3}$$

where Ω_m is the computation hidden by early successor drafting and D_{verify} is the added verification and revision depth. Cross-block overlap shortens generation whenever the hidden computation exceeds this cost.

Sparse revision. Early execution is uncertain because a successor is drafted before its predecessor stabilizes. Let $\hat{b}_i^{\text{early}}$ denote the token drafted at position i and $\hat{b}_i^{\text{final}}$ the token obtained after predecessor commitment. We define the draft revision and reuse rates as

$$\rho_{\text{rev}} = \frac{\left|\left\{ i : \hat{b}_i^{\text{early}} \neq \hat{b}_i^{\text{final}} \right\}\right|}{|B_{m+1}|}, \qquad \rho_{\text{reuse}} = 1 - \rho_{\text{rev}}. \tag{4}$$

In the state-matched audit in Figure 1(b), 192 of 219 early tokens remain unchanged and only 27 are revised, corresponding to a 12.3% revision rate and an 87.7% reuse rate. Context updates therefore induce position-level revision rather than whole-block invalidation.

Visible-token scores are unreliable for identifying these positions because a visible candidate participates in its own conditioning. Temporarily masking the candidate and re-predicting it under the latest

text prefix and original visual input raises AUROC from 0.418 to 0.942 and decision accuracy from 59.4% to 84.4%. Successors can therefore start early: affected positions are revised as the exposed context evolves, followed by final synchronization at predecessor commit.

3 VISWAVE

VISWAVE is a training-free revisable wavefront decoder for diffusion vision–language models. It has two components: **(1) revisable wavefront decoding** launches a successor before its predecessor finishes, overlapping adjacent-block computation while keeping early outputs provisional; **(2) VR-T2T verification** re-evaluates affected draft tokens under the latest multimodal context, retaining supported tokens and remasking unsupported ones. Together, they allow computation to proceed early while admitting only verified state into committed history.

3.1 REVISABLE WAVEFRONT DECODING

Let $c = (I, q)$ be the image–text condition, $H_m = B_{<m}$ the committed prefix of block m, and $\ell_m^{(k)}$ its logical state after denoising step k. A VISWAVE wave maintains one active predecessor B_m and one provisional successor B_{m+1}. The predecessor follows native denoising until its revealed fraction reaches the launch threshold ρ:

$$k_m^{\text{spawn}} = \min\{k : r(\ell_m^{(k)}) \geq \rho\}, \tag{5}$$

after which the successor is drafted under the provisional prefix $\widetilde{H}_m^{(k)} = B_{<m} \oplus \ell_m^{(k)}$. Its tokens remain in logical state rather than stable history. Limiting the active window to two adjacent blocks bounds the propagation of provisional context.

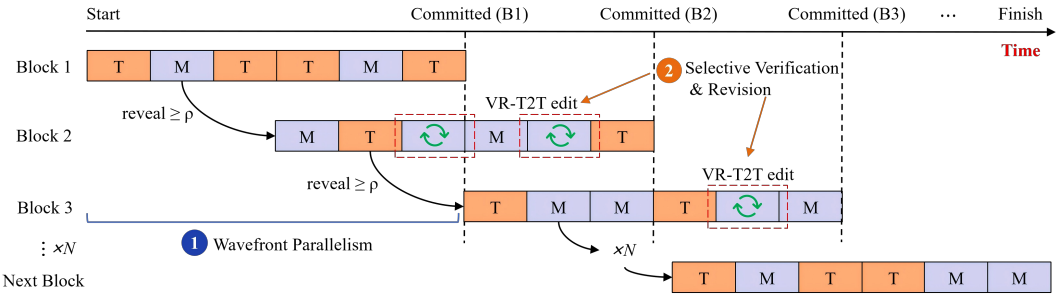


Figure 2: **Draft-and-revise wavefront decoding.** A provisional successor overlaps predecessor decoding and revises affected positions as context evolves. A final boundary synchronization precedes block ownership transfer.

Figure 2 shows the two operations along the execution timeline. Part 1 overlaps successor drafting with predecessor denoising. Part 2 validates stale successor tokens whenever the exposed predecessor context changes; a final synchronization at commit resolves remaining decisions before the successor assumes block ownership. Each drafted token records its source-context version ν_i and support q_i, allowing verification to target predictions measured under an outdated prefix.

Provisional and stable states are separated by the commit invariant. If $\mathcal{A}_m$ is the set of unresolved validation positions, block B_m enters stable history only when

$$H_{m+1} = H_m \oplus \ell_m \quad \Longleftrightarrow \quad r(\ell_m) = 1 \ \wedge \ \mathcal{A}_m = \varnothing. \tag{6}$$

At the boundary, one clean fused forward materializes the predecessor's committed KV while producing the successor's final verification and preview logits. The successor advances only after all keep-or-remask decisions resolve. For M blocks, K native steps, and realized spawn delay $s = k_m^{\text{spawn}}$, the resulting sequential depth is

$$D_{\text{VISWAVE}} = K + (M-1)s + R_{\text{seq}} - G_{\text{reuse}}, \tag{7}$$

where R_{seq} is exposed repair depth and G_{reuse} is depth removed by fused reuse. At the end of evidence generation, VISWAVE stops launching new successors and drains the active wave until the last draft satisfies Eq. 6. Only then does the independent answer tail begin, ensuring that finalization depends exclusively on verified and committed evidence.

3.2 VR-T2T VERIFICATION

Whenever the exposed predecessor context changes, VR-T2T selects revealed successor positions whose source version is stale and whose draft support is below the revision threshold τ_r:

$$\mathcal{A}_{m+1} = \{i \in \mathcal{U}(\ell_{m+1}) : \nu_i < \nu^\star \ \wedge \ q_i < \tau_r\}, \tag{8}$$

where $\mathcal{U}(\ell)$ contains revealed positions and $\nu^\star$ is the latest exposed context version. Let $C_m^{\nu^\star} = H_m \oplus \ell_m^{(k)}$ denote that prefix; at the final boundary it equals the fully committed predecessor prefix. The rule selects stale, insufficiently supported tokens without altering their logical values before verification.

For each candidate, let $d_i = \ell_{m+1,i}$. VR-T2T preserves d_i in logical state but masks all candidates in a temporary model-visible state $\bar{x}$, then re-predicts them together under the latest exposed text and unchanged image:

$$\bar{x}_j = \begin{cases} \texttt{M}, & j \in \mathcal{A}_{m+1}, \\ \ell_{m+1,j}, & j \notin \mathcal{A}_{m+1}, \end{cases} \qquad p_i^{\mathrm{vr}} = \mathrm{softmax}\, F_\theta(c, C_m^{\nu^\star}, \bar{x})_i. \tag{9}$$

The frozen model thereby re-predicts every candidate under the latest exposed predecessor context and the original visual input; at the commit boundary, this context is fully committed. Because d_i is absent from the model-visible state, $p_i^{\mathrm{vr}}(d_i)$ measures renewed multimodal support rather than reinforcement of a visible self-copy. All candidates are verified together in one forward.

The logical state is updated by Top-1 agreement and verification threshold τ_v:

$$\ell^+_{m+1,i} = \begin{cases} d_i, & i \notin \mathcal{A}_{m+1} \text{ or } [\arg\max_v p_i^{\mathrm{vr}}(v) = d_i \ \wedge \ p_i^{\mathrm{vr}}(d_i) \geq \tau_v], \\ \texttt{M}, & \text{otherwise.} \end{cases} \tag{10}$$

Supported drafts receive a *virtual keep* and remain in logical state; rejected positions are truly remasked and returned to native denoising. After verification, every retained token inherits the latest exposed context version $\nu^\star$.

Figure 3 expands this rule for one drafted token. VISWAVE preserves its logical value while temporarily masking the model-visible position, then re-predicts it from the latest text prefix and the same image. A sufficiently supported Top-1 recovery follows the Keep branch; otherwise the position follows the Remask and Re-generate branch.

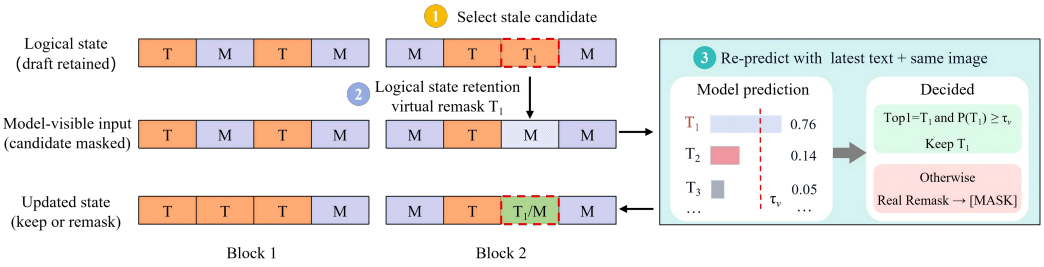


Figure 3: **VR-T2T verification.** Masked draft positions are re-predicted under the latest multimodal context, then kept or remasked. The same operation is used during overlap when the exposed predecessor changes and at the commit boundary for final synchronization.

For verified tokens satisfying the stronger condition $p_i^{\mathrm{vr}}(d_i) \geq \tau_p$, VISWAVE prefetches the prediction into the next wave and assigns it context version $\nu^\star$. This verified prefetch reuses reliable computation without placing unverified state in committed history.

4 VISWAVE ACCELERATES LONG-FORM MULTIMODAL GENERATION

4.1 EXPERIMENTAL SETUP

Models and evaluation. We use frozen BARD-VL-4B (Chen et al., 2026) as the primary model, its 2B and 8B variants for scale generalization, and SDAR-VL (Cheng et al., 2025) for architectural

Table 1: **Natural comparison with Evidence-192 + Answer-8.** Latency is generation wall-clock time after input preparation; parentheses report speedup and score change from BARD-VL-4B.

Dataset	Model / Decoder	Tok./s ↑	Latency (s) ↓	NFE ↓	Tokens	Score ↑
RealWorldQA	BARD-VL-4B	28.89	2.010 (1.000×)	56.1	58.1	72.42 (0.00)
	RedVTP	28.25	2.027 (0.992×)	70.0	57.3	72.03 (−0.39)
	D^3ToM	17.99	3.153 (0.637×)	55.8	56.7	69.41 (−3.01)
	VISWAVE	**38.83**	**1.390 (1.446×)**	**40.0**	54.0	**72.42** (0.00)
MMStar	BARD-VL-4B	34.08	2.300 (1.000×)	76.5	78.4	62.07 (0.00)
	RedVTP	34.26	2.239 (1.027×)	94.3	76.7	**60.80** (−1.27)
	D^3ToM	19.35	4.062 (0.566×)	77.7	78.6	58.07 (−4.00)
	VISWAVE	**53.64**	**1.342 (1.714×)**	**51.9**	72.0	60.20 (−1.87)
MMMU-Val	BARD-VL-4B	34.98	3.355 (1.000×)	115.0	117.4	53.44 (0.00)
	RedVTP	34.10	3.416 (0.982×)	143.7	116.5	53.95 (+0.51)
	D^3ToM	18.23	6.168 (0.544×)	111.3	112.5	53.95 (+0.51)
	VISWAVE	**56.67**	**1.919 (1.748×)**	**76.7**	108.8	**54.35** (+0.91)
MMMU-Pro	BARD-VL-4B	34.76	3.256 (1.000×)	111.0	113.2	32.77 (0.00)
	RedVTP	32.82	3.447 (0.945×)	139.6	113.1	32.54 (−0.23)
	D^3ToM	18.52	5.971 (0.545×)	109.3	110.6	32.25 (−0.52)
	VISWAVE	**51.04**	**2.059 (1.581×)**	**74.1**	105.1	**33.35** (+0.58)
ChartQA	BARD-VL-4B	28.88	0.657 (1.000×)	18.0	19.0	72.88 (0.00)
	RedVTP	26.11	0.732 (0.898×)	24.7	19.1	71.80 (−1.08)
	D^3ToM	18.68	0.975 (0.674×)	19.5	18.2	68.60 (−4.28)
	VISWAVE	**34.85**	**0.536 (1.226×)**	**16.5**	18.7	**72.12** (−0.76)
MME	BARD-VL-4B	34.62	1.795 (1.000×)	60.3	62.1	2288.29 (0.00)
	RedVTP	34.48	1.700 (1.056×)	71.9	58.6	**2286.76** (−1.53)
	D^3ToM	20.37	2.779 (0.646×)	55.9	56.6	2207.15 (−81.14)
	VISWAVE	**52.22**	**1.065 (1.686×)**	**42.7**	55.6	2251.29 (−37.00)
AI2D	BARD-VL-4B	34.14	2.136 (1.000×)	71.0	72.9	82.06 (0.00)
	RedVTP	33.68	2.139 (0.998×)	88.5	72.1	81.31 (−0.74)
	D^3ToM	19.12	3.738 (0.571×)	70.6	71.5	80.57 (−1.49)
	VISWAVE	**49.08**	**1.356 (1.575×)**	**48.4**	66.5	**81.57** (−0.49)

transfer. Evaluation covers RealWorldQA (xAI, 2024), MMStar (Chen et al., 2024), MMMU (Yue et al., 2024a), MMMU-Pro (Yue et al., 2024b), ChartQA (Masry et al., 2022), MME (Fu et al., 2023), AI2D (Kembhavi et al., 2016), and long-form description on DOCCI (Onoe et al., 2024). We port RedVTP (Xu et al., 2025) and D^3ToM (Chang et al., 2025) from their released implementations to the frozen BARD-VL decoder, preserving their original selection and merging rules while adapting model-specific tensor, cache, and generation interfaces. Controlled variants isolate revision, VR-T2T, fused preview, and prefetch.

Protocol and metrics. All methods use the same Evidence-192 + Answer-8 protocol and keep inputs, generation settings, hardware, and scoring fixed; the supplementary material details the prompt stages and output construction. The evidence-first prompt separates grounded evidence collection from answer finalization and elicits task-relevant long-form content without padding: 192 tokens is a maximum budget, and natural-EOS runs allow normal termination. Fixed-length runs instead control output length to compare decoding cost. We report task quality, output length, latency, speedup, and NFE with paired statistical tests.

4.2 QUALITY-PRESERVING ACCELERATION ACROSS TASKS

A single operating point consistently shifts the quality–latency trade-off across diverse workloads. VISWAVE reduces sample-weighted natural latency from 2.042 to 1.282 seconds, yielding a mean

dataset speedup of $1.57\times$, while six-task macro accuracy changes by only -0.27pp. Table 1 summarizes the complete per-dataset comparison; all results use the same VISWAVE configuration without dataset-specific tuning.

This advantage follows from shortening the serial path across blocks. RedVTP and D^{3}ToM reduce computation within individual forwards but leave the inter-block dependency path unchanged, reaching $0.985\times$ and $0.598\times$ speed in the aligned ports. VISWAVE instead overlaps adjacent-block computation and reduces realized latency on every workload.

The gain is not explained by early termination: mean output length decreases by 8.0%, whereas NFE falls by 30.6%. It is smallest on short-output ChartQA, which exposes few block boundaries to overlap, and reaches approximately 1.58–$1.75\times$ on the longer MMMU workloads. This task-level variation suggests that the benefit grows with the amount of successor-block computation available for reuse.

4.3 LONGER GENERATIONS EXPOSE MORE REUSABLE OVERLAP

The cross-task results associate longer outputs with larger speedups. To isolate length from task content, response style, and EOS behavior, we use a fixed-length profile. VISWAVE shortens the critical path of block diffusion by overlapping successor-block computation with active decoding; its wave-launch and synchronization costs should therefore be increasingly amortized as generation spans more blocks.

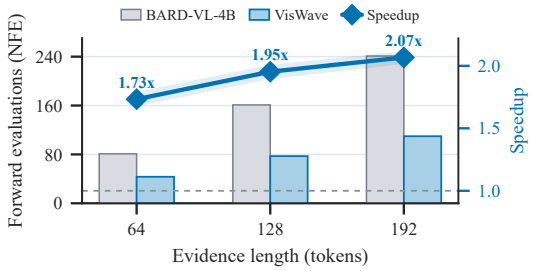


Figure 4: **Fixed-length scaling.** NFE and speedup as output length grows.

Figure 4 confirms this length-dependent behavior. Speedup increases from $1.73\times$ at 64 tokens to $1.95\times$ at 128 tokens and $2.07\times$ at 192 tokens, with every paired 95% bootstrap interval remaining above $1\times$. At 192 tokens, NFE falls by 54.4% and evidence-decoding latency decreases from 6.006 to 2.906 seconds. This monotonic trend matches Eq. 7, showing that the gain comes from reusable cross-block overlap rather than task-specific termination behavior.

4.4 THE GAIN PERSISTS ACROSS MODEL SCALES

Table 2: **VISWAVE scales across BARD-VL checkpoints.** B/V denotes native/VISWAVE; parentheses report speedup and NFE reduction.

Dataset	BARD-VL-2B			BARD-VL-4B			BARD-VL-8B		
	Quality B/V	Latency B/V ↓	NFE B/V ↓	Quality B/V	Latency B/V ↓	NFE B/V ↓	Quality B/V	Latency B/V ↓	NFE B/V ↓
RealWorldQA	60.39/62.35	0.680/0.512 (1.329×)	17.16/12.60 (26.6%)	72.42/72.42	2.010/1.390 (1.446×)	56.12/40.02 (28.7%)	72.68/72.16	1.003/0.882 (1.137×)	15.95/12.70 (20.4%)
MMStar	51.53/51.47	0.794/0.535 (1.483×)	30.53/23.18 (24.1%)	62.07/60.20	2.300/1.342 (1.714×)	76.49/51.88 (32.2%)	67.60/66.80	1.335/0.822 (1.623×)	41.67/27.24 (34.6%)
MMMU-Val	43.52/43.72	1.725/1.147 (1.504×)	74.22/56.63 (23.7%)	53.44/54.35	3.355/1.919 (1.748×)	115.04/76.73 (33.3%)	57.59/58.50	2.824/1.482 (1.906×)	83.87/48.39 (42.3%)
MMMU-Pro	24.39/24.28	1.969/1.331 (1.479×)	85.08/61.41 (27.8%)	32.77/33.35	3.256/2.059 (1.581×)	110.96/74.08 (33.2%)	37.69/37.92	2.446/1.254 (1.951×)	70.49/38.67 (45.2%)
ChartQA	46.04/43.00	0.965/0.595 (1.622×)	40.26/28.26 (29.8%)	72.88/72.12	0.657/0.536 (1.226×)	17.97/16.53 (8.0%)	71.24/70.84	0.877/0.641 (1.368×)	22.86/19.22 (15.9%)
MME	1891.5/1892.0	0.306/0.251 (1.221×)	10.38/10.73 (−3.4%)	2288.3/2251.3	1.795/1.065 (1.686×)	60.32/42.65 (29.3%)	2413.9/2393.3	0.999/0.573 (1.744×)	32.75/20.74 (36.7%)
AI2D	73.15/73.22	0.803/0.514 (1.562×)	30.20/22.76 (24.6%)	82.06/81.57	2.136/1.356 (1.575×)	70.99/48.38 (31.9%)	83.35/84.10	1.030/0.582 (1.769×)	31.22/18.11 (42.0%)
Macro	**49.84/49.67**	**1.46×**	**25.1%**	**62.61/62.34**	**1.57×**	**30.6%**	**65.03/65.05**	**1.64×**	**37.7%**

Table 2 applies the same VISWAVE configuration to BARD-VL-2B, 4B, and 8B. Their six-task macro quality changes remain between -0.27 and $+0.02$ points, while natural-workload speedup reaches

1.46–1.64×; at a fixed 256-token length, all three scales achieve approximately 2.2× speedup. The consistent natural and controlled-length trends show that VISWAVE targets the cross-block execution structure shared across model scales rather than scale-specific output behavior.

4.5 REVISABLE WAVEFRONTS TRANSFER ACROSS DVLMS

Table 3: **VISWAVE transfers to SDAR-VL-4B.** S/V denotes the native decoder/VISWAVE. Parentheses report quality change, speedup, and NFE reduction.

Dataset	Quality S/V	Latency S/V ↓	NFE S/V ↓	Tokens S/V
RealWorldQA	62.75/62.48 (−0.27)	13.393/11.188 (1.197×)	145.1/75.8 (47.8%)	112.1/108.4
MMStar	57.53/57.87 (+0.34)	9.283/6.210 (1.495×)	193.7/99.6 (48.6%)	151.3/146.3
MMMU-Val	51.36/49.82 (−1.54)	12.384/8.820 (1.404×)	239.9/125.4 (47.7%)	188.7/186.7
MMMU-Pro	30.58/29.94 (−0.64)	13.090/9.273 (1.412×)	241.6/126.8 (47.5%)	189.9/188.8
ChartQA	76.76/72.52 (−4.24)	8.903/6.572 (1.355×)	136.1/69.9 (48.6%)	101.9/98.3
MME (points)	2081.8/2020.3 (−61.51)	9.385/6.296 (1.491×)	203.3/110.3 (45.7%)	158.9/155.4
AI2D	78.30/77.95 (−0.35)	11.159/7.416 (1.505×)	223.9/112.4 (49.8%)	175.6/172.7
Macro	**59.55/58.43**	**1.41×**	**47.9%**	**154.1/150.9**

We further integrate VISWAVE with SDAR-VL to test a different DVLM architecture. Across seven datasets, it achieves a 1.41× mean speedup and a 47.9% NFE reduction, while average output length decreases by only 2.0%. Six-task macro quality changes by −1.12 points, with ChartQA accounting for most of the reduction. The consistent efficiency gains on BARD-VL and SDAR-VL show that revisable wavefront execution is not tied to BARD-VL-specific state organization; Table 3 provides the full results.

4.6 ACCELERATION EXTENDS TO OPEN-ENDED LONG-FORM GENERATION

We further evaluate DOCCI, where the final output is an open-ended grounded visual description rather than a selected option. This setting directly tests both long-form generation efficiency and final-content quality.

Table 4: **Long-form generation on DOCCI.** Natural-300 and official `qual_test`-100 results.

(a) Natural-300: length, efficiency, and grounded lexical consistency

Decoder	Tokens	Words	Lat. (s)	Speed	Ref. F1	Number F1	Visual F1
BARD-VL-4B	189.92	161.63	5.818	1.000×	0.3931	0.8608	0.4329
VISWAVE	182.92	155.58	3.840	**1.515**×	0.3950	0.8644	0.4320

(b) Official `qual_test`-100: length, efficiency, and caption metrics

Decoder	Tokens	Words	Lat.	Speed	B-1	B-2	B-3	B-4	R-L	MET	CIDEr
BARD-VL-4B	202.80	172.43	6.065	1.000×	37.52	20.73	10.89	5.80	24.86	17.99	11.24
VISWAVE	183.00	155.67	3.712	**1.634**×	39.61	21.90	11.12	5.82	25.15	17.24	7.67

On Natural-300, latency falls from 5.818 to 3.840 seconds (1.52×), while reference, number, and visual-term F1 remain effectively unchanged. On `qual_test`, speedup reaches 1.63×; BLEU and ROUGE-L are stable or slightly higher, whereas METEOR and CIDEr decrease. Together with the 2.07× speedup in the matched 192-token profile, Table 4 shows that the gain extends from evidence generation to final open-ended multimodal text.

5 HOW REVISION MAKES WAVEFRONT DECODING RELIABLE

5.1 REVISION TURNS OVERLAP INTO RELIABLE ACCELERATION

Wavefront execution shortens the serial path between adjacent text blocks, but the successor is conditioned on context that has not yet stabilized. Table 5 incrementally adds VISWAVE's revision and reuse components. Naive wavefront reaches 2.18× speed, but loses 1.13 accuracy points and shortens mean output from 89.85 to 62.00 tokens, showing that uncorrected early generation changes the decoding trajectory.

Table 5: **Revision-and-reuse ablation.** Rows are cumulative; full VISWAVE further adds verified prefetch.

Variant	Acc.	Speed	NFE	Tokens
BARD-VL-4B	66.60	1.00×	87.83	89.85
Naive wavefront	65.47	2.18×	48.38	62.00
Selective revision	65.27	1.38×	77.05	81.97
+ fused preview	65.60	1.80×	58.10	81.74
+ VR-T2T	65.60	1.83×	57.20	81.26
Full VISWAVE	**65.93**	**1.79×**	**57.98**	**80.80**

Selective revision raises mean output length to 81.97 tokens, while fused preview and VR-T2T further reduce repair overhead. Full VISWAVE achieves 1.79× speed and 65.93 accuracy, the highest quality among accelerated variants, balancing early execution with reliable correction.

5.2 VR-T2T REMOVES VISIBLE-TOKEN SELF-EVIDENCE

A direct FlowBlock-style transfer to a frozen DVLM assumes reliable editing signals from visible drafts. We test this assumption on matched draft states. Visible-token scoring is self-conditioned and reaches only 0.418 AUROC in the 4,744-event audit; VR-T2T instead masks each candidate and re-predicts it under the latest multimodal context, reaching 0.942 AUROC and 0.844 decision accuracy. This removes self-evidence and enables reliable Virtual Keep and remasking.

5.3 VISION PROVIDES LOCAL CAUSAL VERIFICATION

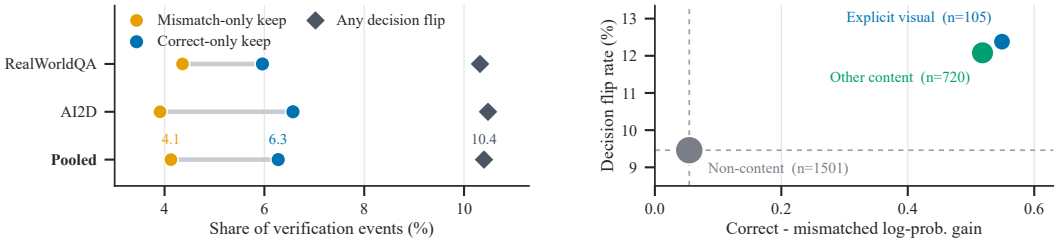


Figure 5: **Visual-causal verification.** Changing only the verification image alters token support and decisions, especially at content-bearing positions.

Figure 5 isolates visual influence by replacing only the verification image while holding the draft, current textual state, candidate positions, and thresholds fixed. Across 2,326 state-matched events, this intervention flips 10.40% of keep/remask decisions. The flip rate is 12.12% at content-bearing positions, compared with 9.46% at non-content positions, showing that visual influence concentrates on factual tokens rather than acting uniformly across the sequence.

These results clarify the role of vision in VISWAVE. Textual context maintains most continuous language structure, while visual input selectively confirms or revises a small set of grounded facts. Vision need not dominate every generated token, yet it retains a local and measurable causal effect on entities, quantities, attributes, and relations. VR-T2T is therefore not a text-only error detector: it uses the latest textual state to identify drafts that require review, then tests whether they remain supported by the current multimodal context. This division lets VISWAVE preserve language-determined drafts while revisiting only visually sensitive positions, protecting factual consistency without paying the cost of revalidating the entire sequence.

6 CONCLUSION

VISWAVE makes cross-block parallelism recoverable by revising provisional drafts and committing only verified state. This training-free design reduces NFE and latency while preserving task quality across workloads, model scales, and diffusion VLM architectures.

Reproducibility Statement

The paper specifies the complete default decoding configuration, prompt budget, answer boundary, dataset sizes, evaluation metrics, and timing protocol. The supplementary material reports the exact evidence and answer prompts, algorithmic pseudocode, fixed-length settings, dataset-level output lengths, and additional statistical results. The implementation records per-example outputs, latency, NFE, parsing status, and configuration metadata.

Ethics Statement

This work changes inference scheduling for existing vision–language models and introduces no additional training data. It inherits the perceptual errors, hallucinations, social biases, and deployment risks of the underlying models. Acceleration can increase the scale at which such models are used, so applications should retain task-appropriate validation and human oversight. Our evaluations use established public research benchmarks and report aggregate results.

AI Use Statement

Generative AI tools were used solely to aid and polish the writing of this manuscript. The authors independently designed and conducted the research, implemented the methods, performed and analyzed the experiments, and verified all quantitative results, citations, and scientific claims. The authors take full responsibility for the final content of the paper.

References

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, 2025.

Shuochen Chang, Xiaofeng Zhang, Qingyang Liu, and Li Niu. D^3ToM: Decider-guided dynamic token merging for accelerating diffusion MLLMs, 2025.

Baoyou Chen, Hanchen Xia, Peng Tu, Haojun Shi, Shan Mu, Weihao Yuan, and Siyu Zhu. BARD: Bridging autoregressive and diffusion vision-language models via highly efficient progressive block merging and stage-wise distillation. *arXiv preprint arXiv:2604.16514*, 2026.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models?, 2024.

Shuang Cheng, Yuhua Jiang, Zineng Zhou, Dawei Liu, Wang Tao, Linfeng Zhang, Biqing Qi, and Bowen Zhou. SDAR-VL: Stable and efficient block-wise diffusion for vision-language understanding, 2025.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. MME: A comprehensive evaluation benchmark for multimodal large language models, 2023.

Ligong Han, Hao Wang, Han Gao, Kai Xu, and Akash Srivastava. S2D2: Fast decoding for diffusion llms via training-free self-speculation, 2026.

Yijie Jin, Jiajun Xu, Yuxuan Liu, Chenkai Xu, Yi Tu, Jiajun Li, Dandan Tu, Xiaohui Yan, Kai Yu, Pengfei Liu, and Zhijie Deng. Multi-block diffusion language models, 2026.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision – ECCV 2016*, pp. 235–251, 2016.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang, and Linfeng Zhang. dLLM-Cache: Accelerating diffusion large language models with adaptive caching, 2025.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, 2022. doi: 10.18653/v1/2022.findings-acl.177.

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, 2025.

Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, Su Wang, and Jason Baldridge. DOCCI: Descriptions of connected and contrasting images. In *Computer Vision – ECCV 2024*, 2024.

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, 2024.

Bing Tian, Haikun Liu, Xiaocheng Zhong, Zhuohui Duan, Zhaokai Luo, Huayi Jin, Zhiyong Wang, and Xiaofei Liao. Flowblock: Wavefront-parallel decoding for self-correcting diffusion language models, 2026.

Xu Wang, Chenkai Xu, Yijie Jin, Jiachun Jin, Hao Zhang, and Zhijie Deng. Diffusion llms can do faster-than-ar inference via discrete diffusion forcing, 2025.

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

xAI. Grok-1.5 Vision Preview. `https://x.ai/news/grok-1.5v`, 2024. Introduces the RealWorldQA benchmark.

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, 2025.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal large diffusion language models, 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, 2025.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024a.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark, 2024b.

Supplementary Material

VisWave: Revisable Wavefront Decoding for Diffusion Vision–Language Models

CONTENTS

A Unified Generation Protocol 13

B Related Work 14

C Algorithm and Default Configuration 15

C.1 Decoding Procedure 15

C.2 Default Configuration 15

D Controlled Fixed-Length Evaluation 15

D.1 Model-Scale Generalization 16

D.2 Aligned External-Method Comparison 16

E Paired Task Statistics 17

F Verification and Visual-Causal Audit 17

G Revision-Threshold Sensitivity 17

A Unified Generation Protocol

All datasets use the same two-stage interface, while the evidence instruction has two format-specific instantiations. Open-ended tasks use:

> Analyze the task using the available visual content and the question. Identify the evidence needed to answer the question. Provide a complete but non-redundant summary of the task-relevant evidence. Include visible objects, attributes, counts, labels, values, spatial relations, textual information, or other explicitly available facts when relevant. Focus on evidence rather than step-by-step reasoning. Do not state the final answer or introduce unsupported assumptions. State uncertainty explicitly when relevant evidence is unclear, ambiguous, or unavailable. Evidence:

For multiple-choice tasks, labeled candidate statements are included in the model input and the opening is made candidate-aware:

> For each candidate statement, separately record the visual or textual evidence that supports it, contradicts it, or leaves it uncertain. Do not rank the candidates, refer to option letters, or state the final answer.

The completeness, non-redundancy, visual grounding, and uncertainty requirements remain unchanged. For tasks requiring an independent answer tail, the decoder appends after evidence synchronization:

> The evidence collection stage is complete. Using the question, the stable evidence above, and the available answer choices if provided, determine the final answer. Apply any necessary reasoning internally. Output only the final answer in the required format. Answer:

The evidence maximum is 192 tokens and the independent answer tail, when used, is 8 tokens.

Figures A1 and A2 illustrate how the same interface covers two distinct workloads. In multiple-choice evaluation, both decoders first produce task-relevant evidence and then emit the answer in the constrained answer tail. For open-ended description, the evidence stage itself is the requested output; colored phrases mark corresponding visual content across the reference and model generations. VisWave changes the execution schedule of the evidence stage, while the prompt, output contract, and finalization rule remain shared with the native decoder.



Question: ***Is this artwork titled madonna degli alberetti? Please answer yes or no.(A. Yes)***

VisWave Output:
Evidence: The artwork in the image is indeed titled " Madonna degli Alberetti" by the Italian Renaissance painter Fra Angelico. The painting depicts the Virgin Mary holding the Christ Child, with a serene and contemplative. The background features a landscape with trees and a clear, blue sky, which is characteristic of Fra Angelico's style. The use of color, the composition of the figures, and the overall atmosphere of the painting are indicative of the artist's work. Therefore, the correct answer to the question is:A. Yes
Answer: A

Bard-VL Output:
Evidence: The artwork in the image is indeed titled "Madonna degli Alberetti." This title is derived from the Italian phrase "Madonna degli Alberetti," which translates to "Madonna of the Trees." The painting depicts the Virgin Mary holding the Christ Child, with a landscape featuring trees in the background, which aligns with the title. Therefore, the correct answer to the question is:A. Yes
Answer: A

Figure A1: An MME multiple-choice example under the unified generation protocol. VisWave and the native Bard-VL decoder receive the same image, question, candidates, evidence instruction, and answer format. Both produce a complete evidence segment followed by the independently finalized answer.

Question: ***Please describe the content of the figure.***



VisWave Output:

"The image captures a nighttime scene on a concrete pathway. A small solar light is planted on the ground, casting a pattern of light rays onto the pavement. The light source is positioned near a circular metal cover, which appears to be a utility access point, possibly for a water or gas meter. The cover is surrounded by a patch of dirt and some vegetation, including a small bush with purple flowers. To the right, there is a stack of large, irregularly shaped stones, possibly part of a garden or decorative feature. The overall atmosphere is serene and quiet, with the light source providing a focal point in the darkness."

Reference

"A high-angle shot of an outdoor path light in a corner of a dried grass yard. The scene is at night, with the patch light illuminating the center of the frame. It has a black rounded cap, and the light creates a round diamond-shaped kaleidoscopic pattern that ripples out with pointed refracted lights towards the outer layer. The upper left and bottom left show driveway concrete appearing mostly dark. On the bottom right is a stack of decorative rocks, mostly out of view. To the right, next to the light, is a brown, round metal sewage cap. The upper right shows a bush appearing black, with a slight view of a similar path light, mostly out of the frame."

Bard-VL Output:

"The image captures a nighttime scene on a concrete pathway. A small solar light is planted on the ground, casting a pattern of light rays onto the pathway. The light is positioned near a circular, metal object that appears to be a utility cover or meter. The cover has some text and symbols on it, but the details are not clearly discernible. To the right of the light, there is a cluster of dark green foliage, possibly a bush or a small tree. The pathway is bordered by a low, white wall on one side and a collection of large stones on the other. The overall atmosphere is serene and quiet, with the light providing a warm glow against the dark surroundings."

Figure A2: A DOCCI open-ended description example under the same evidence-stage protocol. Colored spans identify semantically corresponding visual content in the reference, VISWAVE, and native BARD-VL outputs; colors denote correspondence rather than token alignment.

B RELATED WORK

Diffusion language and vision–language models. Masked diffusion language models replace strictly left-to-right commitment with iterative recovery of masked tokens, enabling bidirectional refinement and parallel token updates (Sahoo et al., 2024; Nie et al., 2025). Block diffusion combines this denoising process with autoregressive dependencies between blocks, recovering arbitrary-length generation and prefix KV reuse while retaining parallelism within each block (Arriola et al., 2025). The same generative paradigm has been extended to visual instruction following and unified multimodal modeling (You et al., 2025; Yang et al., 2025). BARD-VL and SDAR-VL further instantiate block-wise diffusion decoders for vision–language understanding (Chen et al., 2026; Cheng et al., 2025). These models establish efficient parallel denoising inside a block; VISWAVE targets the remaining sequential dependency between adjacent blocks.

Reducing the cost of diffusion decoding. One line of work reduces repeated computation within the existing decoding schedule. dLLM-Cache reuses intermediate features across denoising steps, while Fast-dLLM combines KV reuse with confidence-aware parallel token commitment (Liu et al., 2025; Wu et al., 2026). In diffusion VLMs, RedVTP removes low-importance visual tokens using masked-response attention, and D^3ToM dynamically merges token computation within a denoising step (Xu et al., 2025; Chang et al., 2025). These approaches act on cached states, visual context, or token-level work per forward while leaving the inter-block dependency graph unchanged. VISWAVE acts on a different part of decoding: it overlaps adjacent blocks, revises the successor as its exposed context evolves, and synchronizes the wave at predecessor commit.

Speculation and inter-block parallelism. Speculative decoding separates drafting from target-model verification to accelerate autoregressive generation without changing its output distribution (Leviathan et al., 2023). Recent diffusion methods adapt this idea at different granularities.

D2F and Multi-Block Diffusion Models train the model on noisy multi-block states so that several blocks can be decoded concurrently (Wang et al., 2025; Jin et al., 2026); S2D2 instead invokes the block-size-one mode of the same text model as a local self-verifier (Han et al., 2026). FlowBlock is the closest scheduling counterpart to our work: it admits several text blocks into a gated wavefront and relies on native token-to-token editing to refine drafts produced under stale context (Tian et al., 2026). VISWAVE brings revisable wavefront execution to pretrained diffusion VLMs through a multimodal verification mechanism. It keeps the logical draft intact, temporarily masks only the token being checked in the model-visible state, and re-predicts that position under the latest exposed text and unchanged visual input. This separation converts cross-block context change into selective, visually conditioned revision without requiring additional training. Because BARD-VL is not trained for native visible-token editing, a literal FlowBlock port lacks a reliable revision operator. Our visible-token audit isolates this transfer assumption, while the naive-wavefront ablation measures the corresponding end-to-end schedule without reliable revision.

C ALGORITHM AND DEFAULT CONFIGURATION

Throughout the supplementary material, NFE denotes the number of denoiser forward evaluations used to produce an output.

C.1 DECODING PROCEDURE

1. Decode block B_m with the model's native masked denoising schedule.
2. When its reveal ratio reaches $\rho = 0.5$, initialize and advance draft block D_{m+1} in parallel.
3. During overlap, changes in the exposed predecessor context mark stale, low-support draft positions for provisional VR-T2T in the next joint update; their logical tokens remain stored while the model-visible positions are masked.
4. At predecessor completion, apply the same rule against the final committed context. One clean forward materializes the committed KV for B_m while producing final verification and preview logits for the successor; only the predecessor KV remains in the committed cache.
5. Keep candidates satisfying the VR-T2T keep criterion described in the main paper; genuinely remask all other candidates and return them to native denoising. Prefetch candidates with support at least $\tau_p = 0.9$.
6. After all keep-or-remask decisions are resolved, promote the revised draft to the active left block and continue the next wave.
7. Drain and synchronize the evidence stage, then decode the 8-token answer tail serially.

C.2 DEFAULT CONFIGURATION

Table C1: Core VISWAVE configuration used in the main experiments.

Parameter	Value
Evidence / answer budget	192 / 8 tokens
Block size / denoising steps	4 / 4
Concurrent / preview blocks	2 / 2
Successor spawn ratio ρ	0.5
Revision / keep thresholds τ_r, τ_v	0.4 / 0.4
Verified prefetch threshold τ_p	0.9

D CONTROLLED FIXED-LENGTH EVALUATION

The natural-workload evaluation uses the full test splits: 765 RealWorldQA examples, 1,500 MMStar examples, 988 MMMU-Val examples, 1,730 MMMU-Pro examples, 2,500 ChartQA examples, 2,374 MME sub-questions, and 3,088 AI2D examples. The open-ended evaluation uses 300 sampled DOCCI descriptions and the official 100-example `qual_test` split.

D.1 Model-Scale Generalization

This profile asks whether the reduction in sequential model evaluations persists across checkpoint scales. All comparisons are paired within a model and length; absolute latency is not compared across the scale-specific environments. Token agreement is the mean fraction of fixed output positions whose token IDs match the paired native-decoder output. It is a trajectory diagnostic, not a semantic-quality measure; task accuracy and DOCCI metrics provide the quality evidence.

Table D1: Fixed-length profiling across BARD-VL scales. Speedup compares VISWAVE with the paired native run.

Model	Length	Native (s)	Ours (s)	Speed	Token agreement
2B	128	3.379	1.606	2.104×	0.907
2B	256	6.480	2.929	2.212×	0.941
4B	64	2.281	1.300	1.754×	0.612
4B	128	4.091	2.071	1.975×	0.622
4B	192	6.251	2.986	2.094×	0.631
4B	256	8.604	3.968	2.168×	0.634
8B	128	4.465	2.120	2.106×	0.934
8B	256	8.259	3.709	2.227×	0.962

The table above profiles the three model scales in their native scale-specific environments. The stricter external-method profile in the main paper uses a single BARD-VL-4B environment and a separately paired sample set, which accounts for the small difference between the two reported 192-token speedups (2.094× and 2.067×).

D.2 Aligned External-Method Comparison

All four methods below use the same BARD-VL-4B checkpoint, inputs, output length, RTX 5090, timing boundary, one warm-up, and three measured repetitions on 50 paired examples. Early EOS is disabled for every method. Confidence intervals resample examples after averaging the three repetitions.

Table D2: Synchronized fixed-length evidence-decoding latency for the aligned external-method comparison. Input preparation and the independent answer tail are excluded.

Length	Method	Mean / median / P95 (s)	Tok/s	Speed	95% CI
64	BARD-VL-4B	2.242 / 2.279 / 2.366	28.55	1.000×	–
	RedVTP	2.334 / 2.390 / 2.457	27.42	0.960×	[0.958, 0.963]
	D^3ToM	4.122 / 4.205 / 4.297	15.52	0.544×	[0.542, 0.546]
	VISWAVE	1.294 / 1.294 / 1.525	49.48	**1.733**×	[1.676, 1.790]
128	BARD-VL-4B	4.070 / 4.065 / 4.177	31.45	1.000×	–
	RedVTP	4.155 / 4.150 / 4.252	30.81	0.979×	[0.977, 0.982]
	D^3ToM	7.677 / 7.656 / 7.854	16.67	0.530×	[0.528, 0.532]
	VISWAVE	2.084 / 2.013 / 2.590	61.42	**1.953**×	[1.894, 2.009]
192	BARD-VL-4B	6.006 / 6.007 / 6.116	31.97	1.000×	–
	RedVTP	6.130 / 6.119 / 6.276	31.32	0.980×	[0.977, 0.982]
	D^3ToM	11.312 / 11.306 / 11.504	16.97	0.531×	[0.530, 0.532]
	VISWAVE	2.906 / 2.825 / 3.501	66.07	**2.067**×	[2.014, 2.116]

The external ports reduce different operation counts. At 192 tokens, RedVTP lowers profiled language-side query–key (QK) attention pairs by 7.73% and D^3ToM by 17.90%, but their integration overhead prevents wall-clock gain. VISWAVE instead lowers NFE from 241.00 to 109.78 (54.45%) while its profiled QK-pair count is 2.63% higher because a wavefront forward covers two blocks. This distinction identifies fewer sequential forwards, rather than universal attention sparsity, as the source of its measured speed.

E PAIRED TASK STATISTICS

This section tests whether the natural-workload changes in quality and latency are consistent at the example level rather than artifacts of aggregate averaging. Quality uses paired task outcomes, while latency confidence intervals resample paired examples. McNemar's p tests the paired disagreement counts between the native decoder and VISWAVE.

Table E1: Paired quality and speed statistics for the main BARD-VL-4B evaluation. MME quality changes are normalized to the paired binary sub-question accuracy used by the significance audit; the main text reports official aggregate points.

Dataset	Δ quality [95% CI]	Speed [95% CI]	Native-only / VisWave-only	McNemar p
RealWorldQA	0.00 [−2.22, 2.22]	1.446 [1.405, 1.488]	38 / 38	1.0000
MMStar	−1.87 [−3.67, −0.13]	1.714 [1.673, 1.755]	105 / 77	0.0451
MMMU-Val	+0.91 [−1.11, 3.04]	1.748 [1.707, 1.789]	49 / 58	0.4394
MMMU-Pro	+0.58 [−0.98, 2.08]	1.581 [1.551, 1.611]	87 / 97	0.5071
ChartQA	−0.76 [−1.56, 0.04]	1.226 [1.203, 1.251]	63 / 44	0.0814
MME	−1.32 [−3.74, 0.98]	1.686 [1.653, 1.719]	72 / 85	0.3382
AI2D	−0.49 [−1.42, 0.42]	1.575 [1.549, 1.602]	111 / 96	0.3305

All seven speed intervals remain above one. MMStar is the only 4B quality difference whose paired interval excludes zero at the 5% level; the main table reports this dataset-level change rather than hiding it behind the macro average.

F VERIFICATION AND VISUAL-CAUSAL AUDIT

The verification audit labels a drafted position as *retain* when its token remains unchanged after the corresponding native revision trajectory, and as *change* otherwise. A false keep retains a token that should change; a false remask rejects a token that would remain. Area under the receiver-operating-characteristic curve (AUROC) and expected calibration error (ECE) evaluate the support score against these labels, and decision accuracy evaluates the resulting keep/remask decision. The pooled audit contains 2,438 MMStar and 2,306 AI2D events.

Table F1: Pooled visible-token and virtual-remask verification audit over 4,744 events.

Verifier	AUROC	ECE	False keep	False remask	Decision acc.
Visible token-to-token	0.418	0.397	5	1921	0.594
Masked VR-T2T	0.942	0.133	40	701	0.844

The visual-causal audit is separate from the retention audit above. It holds the drafted token, current textual state, logical state, and threshold fixed, and replaces only the image used by the verification forward. The 2,326 events below come from 200 RealWorldQA and AI2D examples; a flip records a change in the keep/remask decision, and support gain is the log-probability difference for the drafted token under the correct versus mismatched image.

Table F2: Visual counterfactual verification by token group. Gain is the mean log-probability difference for the drafted token between correct and mismatched images.

Token group	Events	Decision flips	Support gain
Explicit visual	105	12.38%	0.549
Other content	720	12.08%	0.518
Non-content	1501	9.46%	0.054
All	2326	10.40%	0.220

G REVISION-THRESHOLD SENSITIVITY

The revision-selection threshold τ_r controls how much provisional work is rechecked as the exposed predecessor context evolves, including final synchronization at commit. Table G1 reports the complete

revision-axis slice of the 150-example MMStar and AI2D scan; all other parameters remain at their defaults. These are natural-EOS system measurements, so a threshold may change output length as well as NFE. The fixed-length profiles in Section D isolate decoding cost. Lower thresholds expose a faster but less conservative operating point, while thresholds above 0.4 spend progressively more evaluations on reconciliation.

Table G1: Revision-threshold sensitivity on 150 paired examples per dataset. The 0.40 row is the default VISWAVE operating point.

Dataset	τ_r	Accuracy	Speedup	NFE	Output tokens
MMStar	0.20	64.00	2.247×	41.45	65.54
	0.30	68.00	1.994×	46.83	68.96
	0.40	64.67	1.691×	56.15	77.87
	0.50	65.33	1.509×	63.28	84.62
	0.60	67.33	1.447×	66.77	85.78
AI2D	0.20	83.33	2.433×	32.64	50.87
	0.30	86.67	1.881×	42.91	62.49
	0.40	83.33	1.672×	48.91	67.85
	0.50	83.33	1.543×	52.93	68.66
	0.60	84.67	1.480×	56.05	69.56

Because 0.30 appeared favorable in the small scan, we re-evaluated it on the full 765-example RealWorldQA and 988-example MMMU-Val sets. Relative to 0.40, it reduced latency from 1.390 to 1.190 seconds on RealWorldQA but changed accuracy by -0.26pp; on MMMU-Val latency changed only from 1.919 to 1.884 seconds while accuracy changed by -1.62pp. The apparent small-sample quality gain therefore did not persist. We retain 0.40 as the single default used in the main evaluation and interpret 0.30 only as a more aggressive Pareto point.