OVERTURE: DECODE BEFORE THE ECHO FOR DMLLM

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing diffusion-based multimodal language models (dMLLMs) typically order decoding by prediction confidence. This strategy prioritizes high-confidence tokens, providing more reliable context for later predictions, but overlooks how decoding timing affects multimodal decision reliability. Analyzing complete denoising trajectories, we find that confidence-based decoding finalizes simple, linguistically predictable tokens first, while deferring more difficult tokens whose interpretation requires precise cross-modal alignment and richer semantic descriptions than those conveyed by formulaic, template-like language. As finalized text accumulates, linguistic context increasingly shapes later predictions, creating a growing linguistic echo, whereas visual evidence exerts less influence on candidate competition. For these visually dependent and more difficult tokens, reliable decisions require stronger visual evidence. Yet confidence-based decoding postpones their commitment until visual evidence exerts weaker influence, creating a difficulty–timing mismatch between evidence requirements and decision timing. This mismatch makes visually dependent decisions more vulnerable to erroneous final predictions. Tracking each token's attention context across denoising steps reveals a stable local Undershoot phenomenon. Pronounced Undershoot is concentrated among tokens associated with subsequent erroneous decisions, providing an observable signal for locating vulnerable generation decisions. We propose **OVERTURE**, a training-free, trajectory-guided priority decoding method that identifies vulnerable content from Undershoot and prioritizes the corresponding positions during regeneration while re-predicting tokens from the current generation state. Experiments across multiple benchmarks and model backbones demonstrate the broad effectiveness of OVERTURE.

1 INTRODUCTION

Diffusion-based multimodal large language models (dMLLMs) reconstruct responses through iterative denoising (Austin et al., 2021; Nie et al., 2025; Li et al., 2025d; 2022; Gong et al., 2022; He et al., 2022; Sahoo et al., 2024). Unlike autoregressive models, dMLLMs predict multiple masked positions per step, enabling parallel prediction and flexible decoding ordered by confidence (Arriola et al., 2025; Kim et al., 2025). High-confidence tokens are finalized first and condition predictions at unresolved positions (Kim et al., 2025; Shu et al., 2026).

Confidence-based decoding thus determines both commitment order and the context available at finalization. It tends to defer visually dependent content while finalizing simple, linguistically predictable tokens. Later-decoded positions receive more candidate updates but also accumulate self-generated linguistic context. This context may increasingly shape predictions while visual evidence may exert weaker influence, leaving visually dependent decisions under weaker visual constraints. We refer to this growing influence of previously committed text on later decisions as a *linguistic echo*. We describe the resulting misalignment between decoding difficulty and decision timing as the difficulty–timing mismatch.

Prior work examines internal and temporal generation dynamics to assess reliability (Wang et al., 2025; Chang et al., 2025; Hemmat et al., 2026). We track changes in each token's attention context across successive denoising steps in step–token space. The resulting field of context-change magnitude forms the *Context-Change Energy* map. At each step, the position with the largest change

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

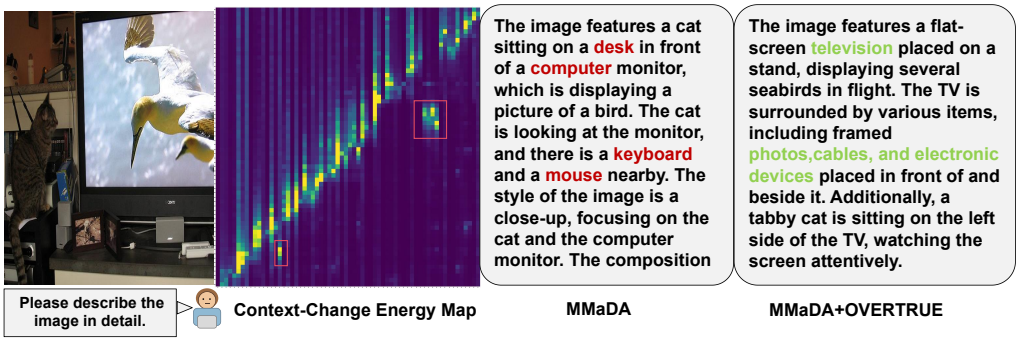


Figure 1: **A motivating example of trajectory-level anomalies during dMLLM generation**. The left panel shows the input and response, with visually unsupported content highlighted in red. The right panel shows the corresponding step–token Context-Change Magnitude map, where local deviations depart from the dominant advancing trajectory.

defines the dominant change peak. These peaks generally form a continuous advancing trajectory across output positions. We define local downward deviations from the stage-specific reference trajectory as trajectory undershoot (Undershoot). Aligning these steps with actual token commitments and their sentences ranks potentially vulnerable content (Figure 1).

Pronounced Undershoot is associated with positions where erroneous decisions ultimately occur. Text-conditioning experiments show stronger linguistic support through multiple finalized positions as context accumulates. Relevant visual information remains in internal representations but may be insufficient to alter candidate competition. Together, these findings provide a process-level account of the difficulty–timing mismatch.

We propose OVERTURE, a trajectory-guided priority decoding method requiring no additional training. OVERTURE first follows the original confidence-based schedule to record attention-context trajectories and select the Top-1 sentence using Undershoot. It then restarts from an all-MASK state and raises the commitment priority of the selected positions while retaining real-time confidence for candidate selection. Only position coordinates are transferred, and tokens are re-predicted from the current generation state.

We evaluate OVERTURE on MMaDA-8B-Base and LaViDa across MathVista, LLaVA-Bench-COCO, ScienceQA-IMG, CHAIR, AMBER, and DetailCaps. The results show consistent improvements over the native scheduler on both backbones, with the strongest overall gains on MMaDA-8B-Base. Our contributions are as follows:

- **We reveal a difficulty–timing mismatch in dMLLM decoding.** Our analysis shows that confidence-based decoding can defer visually dependent decisions to stages where self-generated linguistic context has stronger influence.
- **We introduce Undershoot as a trajectory-level diagnostic signal.** Undershoot captures local deviations in the denoising trajectory and helps identify vulnerable generation decisions.
- **We propose OVERTURE, a training-free trajectory-guided decoding method.** OVERTURE uses Undershoot to prioritize vulnerable content by adjusting commitment timing without modifying model parameters.

Extended discussion of related work is provided in Appendix A.

2 Analysis and Motivation

Confidence-based decoding determines both commitment order and the context available when each decision is finalized. Later-decoded positions undergo more candidate updates while conditioning on an increasing amount of finalized text. This accumulation can change the relative influence of

linguistic and visual information. We examine the resulting difficulty–timing mismatch through denoising trajectories and their implications for decoding order.

2.1 Preliminaries: Diffusion Decoding and Generation Dynamics

Diffusion Decoding Mechanism Given an image I, a text instruction q, and a maximum response length, diffusion-based language models iteratively predict unresolved positions from fully or partially masked responses (Austin et al., 2021; Nie et al., 2025). At step t, $P_{t,i}(v)$ denotes the probability of vocabulary token v at position i, conditioned on the image, instruction, and current response state. Unlike autoregressive models, diffusion models can predict candidates for multiple masked positions within one step (Arriola et al., 2025; Kim et al., 2025). The final response therefore depends on both candidate prediction and commitment order.

Candidate Prediction and Position Commitment At position i, the candidate $\hat{x}_i^{(t)}$ is the most probable token under $P_{t,i}$, and its probability defines the confidence $c_i^{(t)}$. At each step, confidence-based decoding commits up to k_t eligible positions with the highest candidate confidence (Kim et al., 2025; Shu et al., 2026). We denote this commitment set by $\mathcal{U}_t$ and the step at which position i is committed by τ_i. Without remasking, committed positions remain fixed; uncommitted positions may undergo multiple candidate updates before finalization.

Self-Generated Context and Decision Timing Committed tokens enter the response state and condition predictions at the remaining masked positions. Different commitment orders therefore produce different contextual histories, even when model parameters and prediction rules remain unchanged. For position i, τ_i determines the number of candidate updates and how much self-generated text is available at finalization. For difficult positions deferred to later steps, additional context does not necessarily yield more reliable multimodal decisions.

2.2 Step–token Trajectories of Context Change

We track changes in attention context at each output position throughout denoising. Let $H_t(p)$ denote the context-change magnitude at position p and step t, computed as specified in Eq. equation 4. The valid positions used for trajectory analysis form $\mathcal{J}_t$. At each step, the dominant change peak r_t is the position with the largest context-change magnitude. These peaks typically follow a relatively continuous advancing trajectory, with occasional local downward deviations.

Figure 2 shows the attention evolution, the step–token Context-Change Magnitude map, and the corresponding token commitments.

Trajectory Undershoot Let $g(t)$ denote the generation stage at step t and $f_{g(t)}$ its reference trajectory. We write the reference position as $\hat{r}_t = f_{g(t)}(t)$ and define

$$r_t = \operatorname*{argmax}_{p \in \mathcal{J}_t} H_t(p), \qquad d_t = \hat{r}_t - r_t. \tag{1}$$

Undershoot occurs when $d_t \geq \delta$, with $\delta = 4$ in the current setting. The actual peak then lies at least four token positions earlier than its stage-specific reference. We use z_t to indicate whether this condition holds.

2.3 Undershoot Reveals Generation Positions at Which the Model Is Less Certain

Tracing the positions committed at Undershoot steps reveals content with more unstable candidate evolution and a greater tendency toward erroneous final decisions.

From Steps to Generated Content For a diagnostic response, let $\mathcal{U}_t^{\mathrm{diag}}$ contain positions actually committed at step t, and $\mathcal{I}(s)$ contain the token positions of sentence s. The set $\mathcal{S}_D$ contains sentences receiving at least one token committed during an Undershoot step. Each such sentence is

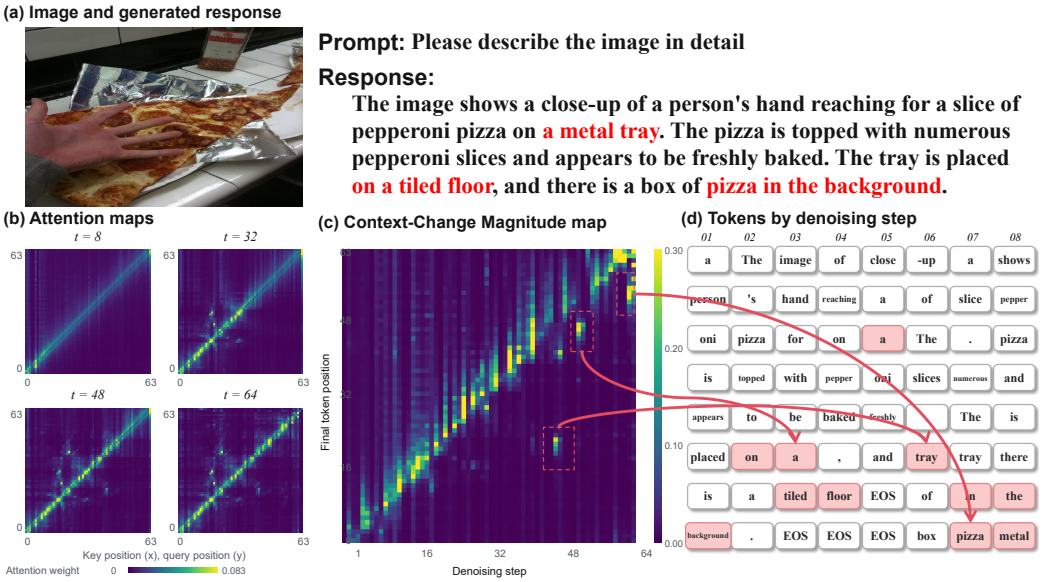


Figure 2: **From attention map dynamics to the step–token Context-Change Magnitude map**. (a) Input image and generated response. (b) Attention maps at representative denoising steps. (c) The resulting step–token Context-Change Magnitude map across the denoising process. (d) Tokens committed at successive denoising steps, enabling trajectory deviations to be aligned with concrete generated content.

scored by its largest associated undershoot depth:

$$q(s) = \max_{\substack{t:\, z_t = 1 \\ \mathcal{U}_t^{\mathrm{diag}} \cap \mathcal{I}(s) \neq \emptyset}} d_t, \qquad s \in \mathcal{S}_D. \tag{2}$$

This score ranks potentially vulnerable content for subsequent intervention. Peak positions determine trajectory deviations; the actual commitments at the same steps determine which content is selected.

Candidate Stability We trace candidate histories and measure the candidate-switching ratio R_i before commitment. This is the fraction of recorded comparisons between adjacent denoising steps in which the candidate token changes. Comparisons require records at both steps; if no valid pair exists, the ratio is undefined. A larger R_i indicates more frequent switching, and positions identified by Undershoot exhibit greater candidate instability in our experiments.

Relationship with Final Errors After controlling for position, generation stage, and confidence, content identified by Undershoot still exhibits a higher proportion of visual inconsistency errors. This association links trajectory deviations to both unstable intermediate predictions and erroneous final decisions.

Core Observation **Undershoot is more likely to occur in content about which the model is less certain, and this content is also more prone to erroneous final decisions.**

This observation clarifies why the trajectory signal is useful beyond instantaneous confidence. Confidence describes the model's preference among candidates at a particular generation state, whereas Undershoot reflects how the context associated with a position evolves over multiple denoising steps. A position may therefore eventually obtain a high-confidence candidate even after undergoing unstable intermediate updates. By tracing this evolution before commitment, Undershoot provides a complementary view of local generation difficulty. Mapping the detected trajectory events back to the positions actually committed at the same steps further connects the internal generation dynamics to concrete response content, allowing the trajectory analysis to operate at the sentence level rather than remaining an abstract hidden-state diagnostic.

2.4 How Later Decoding Changes Decision Conditions

Confidence-based decoding tends to defer difficult positions to later steps. Although additional context might appear to resolve uncertainty, our analysis suggests that this does not always hold for content requiring visual evidence. Self-generated text and visual information exhibit different trends in their influence on candidate decisions.

Finding 1: Candidates Become Increasingly Influenced by Self-Generated Text As finalized tokens accumulate, candidates at uncommitted positions increasingly align with linguistic patterns in the existing text. For example, preceding expressions such as "photo" or "vintage" favor candidates such as "black and white", "classic", or "timeless". These descriptions are not necessarily supported by image evidence.

To examine this dependence, we mask or replace relevant generated text while holding the image and target generation state fixed. We use a common sensitivity measure for the text and visual interventions:

$$\Delta_m(t,i) = D_{\mathrm{JS}}\big(P_{t,i} \| P_{t,i}^{-m}\big), \qquad m \in \{\mathrm{text}, \mathrm{vis}\}. \tag{3}$$

Here, $P_{t,i}^{-m}$ is the candidate distribution after the corresponding intervention, and D_{JS} denotes Jensen–Shannon divergence. The influence of existing text increases during generation, indicating that self-generated context can progressively alter candidate competition.

Finding 2: Visual Information Remains Present, but Its Constraint on the Final Choice Weakens We separately examine whether visual attributes remain recoverable from internal states and whether perturbing the relevant visual input changes candidate predictions. The visual intervention defines $P_{t,i}^{-\mathrm{vis}}$ in Eq. equation 3. Visual information remains recoverable and can affect predictions, but its influence on candidate competition weakens as generation proceeds. It may therefore be insufficient to change choices already supported by linguistic context.

Difficulty–timing mismatch Together, these findings suggest that initially uncertain candidates acquire stronger linguistic support without a corresponding increase in visual constraints. Deferring difficult positions may consequently place visually dependent decisions under stronger linguistic influence and relatively weaker visual constraints. This constitutes the difficulty–timing mismatch examined in this work.

The key consequence is that additional denoising steps do not necessarily provide the same type of information that visually dependent decisions need. Later positions receive more opportunities for candidate revision, but these revisions occur under a response state containing progressively more self-generated text. As a result, delaying commitment changes both the amount and the composition of the context available to the model. For visually dependent content, this distinction is important: a candidate can become more consistent with the emerging linguistic context without becoming better grounded in the image. The difficulty–timing mismatch therefore concerns not only how long a position remains unresolved, but also the contextual conditions under which its final decision is made.

This perspective also distinguishes commitment timing from conventional confidence refinement. Repeated denoising can sharpen the local candidate distribution, yet a sharper distribution does not by itself reveal whether the supporting evidence is primarily visual or linguistic. A late high-confidence decision may therefore be stable with respect to the current response state while still reflecting an unfavorable evidence balance. This motivates treating commitment order as an explicit control variable rather than as a passive consequence of confidence ranking.

2.5 From Risk Localization to Priority Decoding

Undershoot identifies potentially vulnerable content whose predictions may be further shaped by accumulating linguistic context while awaiting commitment. This motivates increasing its decoding priority before the linguistic echo becomes further reinforced. OVERTURE uses Undershoot observed during diagnostic generation to select candidate regions for the priority-decoding intervention.

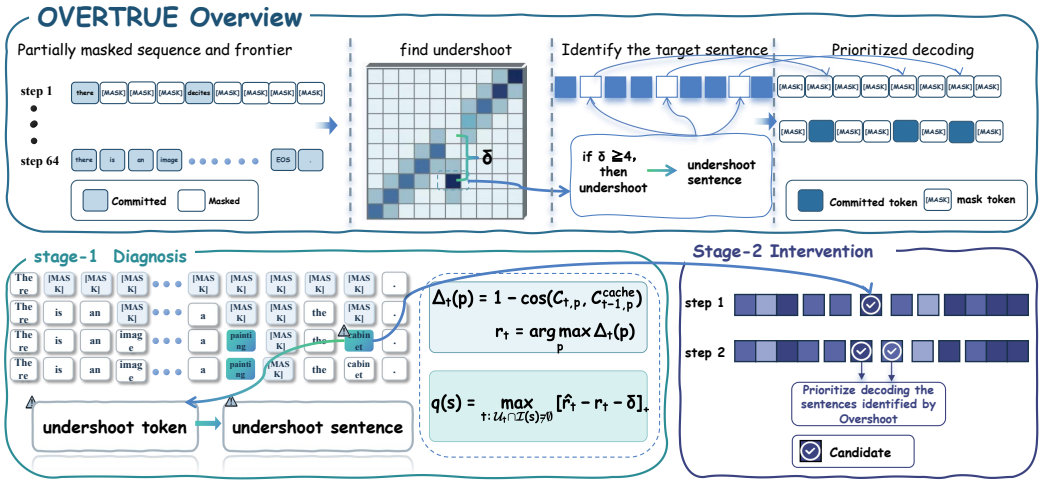


Figure 3: **Overview of OVERTURE**. Stage 1 records the natural denoising trajectory, detects Undershoot events, and maps them to a target sentence. Stage 2 restarts generation from an all-MASK state and prioritizes the identified positions while retaining confidence-based candidate prediction.

3 METHODS

OVERTURE is a training-free inference-time intervention. A diagnostic run follows the original confidence-based schedule and uses Undershoot to select response positions. Priority decoding then restarts from an all-MASK state and adjusts commitment order using these positions together with the EOS/EOT rule in Section 3.2. Model parameters and candidate prediction rules remain unchanged. The overall procedure is shown in Figure 3.

3.1 UNDERSHOOT CANDIDATE LOCALIZATION

We compute cache-relative context-change magnitude from the natural-decoding records. Let $C_{t,l,p}$ denote the concatenated attention context vector at step t, layer l, and output position p. Its cached counterpart is $C^{\text{cache}}_{t-1,l,p}$. We clip each layer's cosine similarity to $[0,1]$ and average its complement across $L = 32$ layers:

$$H_t(p) = \frac{1}{L}\sum_{l=0}^{L-1}\left[1 - \text{clip}\left(\cos\left(C_{t,l,p}, C^{\text{cache}}_{t-1,l,p}\right), 0, 1\right)\right]. \tag{4}$$

Undefined positions remain NaN and are excluded rather than filled with zeros for fitting. At each valid step, we select the dominant change peak using Eq. equation 1. Exact ties are resolved by proximity to the previous peak, followed by the smaller position index.

We fit a rising reference trend to the complete peak sequence. We fit the post-EOS segment separately if three consecutive peaks fall within the EOS region. This also requires at least six tail columns and sufficient valid points in the preceding segment. Reference positions are clipped to the valid output range before computing d_t and identifying Undershoot events. This procedure uses the complete diagnostic trajectory and its EOS position, making the current analysis offline.

Events are mapped through actual commitments to the sentence set $\mathcal{S}_D$. We rank these sentences by $q(s)$ in Eq. equation 2 and select at most one. Score ties are resolved by the earlier triggering step, then the earlier sentence start. The selected sentence defines the original-position set S^*. If no sentence is selected, S^* is empty and the intervention uses non-prioritized scheduling with EOS/EOT deferral.

Algorithm 1 summarizes localization. The dominant change peak measures trajectory deviation; sentence selection uses positions actually committed at the same step.

Algorithm 1: Undershoot Localization

Require: Diagnostic records $\{C_{t,l,p}\}$, cache $\{C^{\text{cache}}_{t-1,l,p}\}$, commitments $\{\mathcal{U}_t^{\text{diag}}\}$, threshold δ
Ensure: Target position set S^*

Compute $H_t(p)$ using Eq. equation 4
for each valid step t **do**
  $r_t \leftarrow \arg\max_{p \in \mathcal{J}_t} H_t(p)$
end for
Fit the reference trajectory $\hat{r}_t = f_{g(t)}(t)$
$d_t \leftarrow \hat{r}_t - r_t$
$\mathcal{D} \leftarrow \{t : d_t \geq \delta, \mathcal{U}_t^{\text{diag}} \neq \emptyset\}$
Map $\{\mathcal{U}_t^{\text{diag}} : t \in \mathcal{D}\}$ to hit sentences $\mathcal{S}_D$
if $\mathcal{S}_D = \emptyset$ **then**
  return $S^* \leftarrow \emptyset$
end if
Compute $q(s)$ for $s \in \mathcal{S}_D$ using Eq. equation 2
$s^* \leftarrow \arg\max_{s \in \mathcal{S}_D} q(s)$
$S^* \leftarrow \mathcal{I}(s^*)$
return S^*

Algorithm 2: OVERTURE Decoding

Require: Image I, instruction q, scheduler Sched, steps T, budget k_t
Ensure: Intervened response y^{OV}

Run natural decoding with Sched
Record trajectory and $\{\mathcal{U}_t^{\text{diag}}\}$
$S^* \leftarrow$ Algorithm 1
Reset all response positions to MASK
Initialize M_1 to all response positions
for $t = 1, \ldots, T$ **do**
  Predict candidates $\hat{y}_i^{(t)}$ and confidence $c_i^{(t)}$
  $B_t \leftarrow M_t \cap \mathcal{E}_t^{\text{int}}$
  $O_t \leftarrow \{i \in B_t : \hat{y}_i^{(t)} \notin \mathcal{Z}\}$
  $P_t \leftarrow S^* \cap O_t$
  Select A_t using Eq. equation 5
  $\mathcal{U}_t^{\text{OV}} \leftarrow \text{TopK}_{i \in A_t}(c_i^{(t)}, k_t)$
  Commit candidates at $\mathcal{U}_t^{\text{OV}}$
  $M_{t+1} \leftarrow M_t \setminus \mathcal{U}_t^{\text{OV}}$
end for
return y^{OV}

3.2 Full-MASK Top-1-Sentence-Priority Intervention

The intervention resets all response positions to MASK. Before step t, M_t contains the uncommitted positions, and $\mathcal{E}_t^{\text{int}}$ specifies those eligible under the current block and budget constraints. The model predicts a candidate $\hat{y}_i^{(t)}$ and confidence $c_i^{(t)}$ for each eligible position.

EOS/EOT commitment deferral. EOS/EOT may be committed before sufficient semantic content is generated, leading to premature truncation. We therefore defer end tokens whenever enough non-ending candidates are available. Let $B_t = M_t \cap \mathcal{E}_t^{\text{int}}$ be the eligible positions, $O_t = \{i \in B_t : \hat{y}_i^{(t)} \notin \mathcal{Z}\}$ the non-ending candidates, and $P_t = S^* \cap O_t$ the prioritized non-ending candidates, where $\mathcal{Z} = \{\text{EOS}, \text{EOT}\}$. We use

$$A_t = \begin{cases} P_t, & |P_t| \geq k_t, \\ O_t, & |P_t| < k_t \text{ and } |O_t| \geq k_t, \\ B_t, & |O_t| < k_t, \end{cases} \qquad \mathcal{U}_t^{\text{OV}} = \text{TopK}_{i \in A_t}\left(c_i^{(t)}, k_t\right). \tag{5}$$

When fewer than k_t non-ending candidates remain, EOS/EOT are released to compete with their original confidence. This rule changes only commitment order and leaves logits and candidate predictions unchanged.

3.3 Diagnostic and Intervention Protocol

The diagnostic and intervention runs use the same image, instruction, model checkpoint, and decoding configuration. The diagnostic run supplies only the priority coordinates S^*, while the intervention recomputes candidate tokens from its current generation state. Because commitment order changes the subsequent context, the resulting tokens, sentence boundaries, and EOS position may differ from those observed during diagnosis.

This separation is important because the intervention does not copy the diagnostic answer or force previously observed tokens. The selected positions only determine which parts of the response receive higher commitment priority. Once the generation state changes, all candidate tokens at those positions are recomputed by the original model. OVERTURE therefore intervenes on decision timing while preserving the model's native candidate prediction mechanism.

Table 1: Main results on multimodal understanding, reasoning, hallucination, and detailed captioning. Bold denotes the best available result and underline denotes the second-best result within each backbone. "–" indicates that the corresponding baseline result is unavailable.

	Understanding & Reasoning			Hallucination			AMBER			DetailCaps
Method	**MathVista↑**	**LLaVA-Bench-COCO↑**	**ScienceQA↑**	**CHAIR$_S$ ↓**	**CHAIR$_I$ ↓**	**Recall↑**	**Cover↑**	**Hal↓**	**Cog↓**	**CAPTURE↑**
MMaDA-8B-Base										
Original	23.50	31.60	50.24	24.58	11.27	39.46	51.93	23.50	1.69	35.51
AdaBlock	24.80	46.80	51.61	22.84	11.27	**41.83**	–	–	–	–
Wavefront	23.00	43.00	50.77	**18.62**	9.43	38.91	–	–	–	–
CACD	28.00	50.40	51.71	19.76	10.28	40.63	–	–	–	–
OVERTURE	**28.70**	**54.10**	**57.61**	22.00	**9.34**	41.51	**54.74**	**17.25**	**0.97**	**36.21**
LaViDa (SigLIP + LLaDA-8B)										
Original	41.50	51.10	71.24	6.93	10.20	34.68	35.20	27.00	0.70	25.62
AdaBlock	45.50	78.60	72.93	6.62	5.02	31.62	–	–	–	–
Wavefront	45.30	82.10	72.09	5.87	**4.31**	31.72	–	–	–	–
CACD	**45.70**	**88.70**	**73.03**	**5.08**	7.58	35.31	–	–	–	–
OVERTURE	41.80	70.60	72.34	5.56	6.87	**37.43**	**38.80**	**18.00**	**0.60**	**27.13**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and baselines. We evaluate OVERTURE on two diffusion-based multimodal language models, MMaDA-8B-Base (Yang et al., 2025) and LaViDa (SigLIP + LLaDA-8B) (Li et al., 2025c). Model parameters remain frozen during trajectory diagnosis and decoding intervention. For each backbone, *Original* denotes its native decoding scheduler. We additionally compare with AdaBlock, Wavefront, and CACD wherever corresponding benchmark results are available. Appendix B further analyzes how the trajectory structure varies across additional dMLLM architectures, including LLaDA-V.

Benchmarks and metrics. We evaluate multimodal understanding and reasoning on MathVista (Lu et al., 2023), LLaVA-Bench-COCO, and ScienceQA-IMG (Lu et al., 2022). Hallucination performance is evaluated with CHAIR$_S$, CHAIR$_I$, and Recall. We further evaluate open-ended hallucination on AMBER using Cover, Hal, and Cog, and detailed image description on DetailCaps using CAPTURE. Higher values are better for MathVista, LLaVA-Bench, ScienceQA, Recall, Cover, and CAPTURE, while lower values are better for CHAIR$_S$, CHAIR$_I$, Hal, and Cog.

Evaluation protocol. OVERTURE and the corresponding Original scheduler are evaluated with the same model checkpoint, prompt, response-length budget, denoising configuration, and sampling setup. The current runs use a fixed 1,000-example subset for each benchmark. The default Undershoot threshold is $\delta = 4$, and at most one target sentence is prioritized during intervention. OVERTURE uses the EOS/EOT commitment-deferral rule in Section 3.2. For external decoding baselines, we report the available benchmark results under their corresponding evaluation settings; unavailable AMBER and DetailCaps entries are marked with "–".

4.2 MAIN RESULTS

Table 1 summarizes the main results. On MMaDA-8B-Base, OVERTURE improves MathVista from 23.50 to 28.70 (+5.20), LLaVA-Bench-COCO from 31.60 to 54.10 (+22.50), and ScienceQA from 50.24 to 57.61 (+7.37). It also reduces CHAIR$_I$ from 11.27 to 9.34 while increasing Recall from 39.46 to 41.51. OVERTURE achieves the best MathVista, LLaVA-Bench-COCO, ScienceQA, and CHAIR$_I$ results among the compared methods on MMaDA-8B-Base.

On LaViDa, OVERTURE also improves all six reported main metrics over the Original scheduler, including an increase in Recall from 34.68 to 37.43. This consistent improvement across a second backbone supports the transferability of trajectory-guided commitment reordering.

The AMBER and DetailCaps results show the same trend. Relative to the Original scheduler, OVERTURE improves AMBER Cover from 51.93 to 54.74 on MMaDA and from 35.20 to 38.80 on LaViDa, while reducing both Hal and Cog. CAPTURE also increases from 35.51 to 36.21 on MMaDA and from 25.62 to 27.13 on LaViDa.

Table 2: LLaVA-Bench-COCO performance on MMaDA-8B-Base across different maximum generation lengths. Bold denotes the best result at each length.

Length	Original	AdaBlock	Wavefront	CACD	OVERTURE
16	45.9	**52.6**	41.9	52.2	46.4
32	40.8	**61.7**	45.5	59.4	54.3
64	33.5	55.9	46.4	**58.8**	56.8
128	31.6	46.8	43.0	50.4	**54.1**
256	32.3	46.0	44.6	47.3	**52.1**
512	35.3	47.4	42.5	49.0	**49.8**
1024	33.0	45.5	43.2	**47.8**	46.5

4.3 Performance across Generation Lengths

The default LLaVA-Bench-COCO setting in Table 1 uses a maximum generation length of 128 tokens. We further vary the generation budget from 16 to 1024 tokens to examine the effect of response length, with the results reported in Table 2.

As shown in Table 2, OVERTURE achieves the best score at 128, 256, and 512 tokens, and remains competitive at 64 and 1024 tokens. At shorter generation lengths, AdaBlock performs best at 16 and 32 tokens. The relative advantage of OVERTURE therefore becomes more apparent once the generation budget moves beyond very short responses. This is consistent with our motivation that commitment timing matters more when unresolved positions remain exposed to an evolving self-generated context over multiple denoising steps. In this regime, changing the order in which vulnerable content is finalized has greater opportunity to alter the subsequent generation path. Overall, the length analysis suggests that OVERTURE is especially effective in the medium-to-long generation regime rather than being tied to a single response budget.

4.4 Selection Range Analysis

Top-1 vs. Top-2 localization. Expanding the selected region from Top-1 to Top-2 covers a broader set of vulnerable content, but the broader priority set produces weaker downstream intervention gains in our experiments. This indicates that increasing localization coverage does not directly translate into better commitment-order intervention. A broader priority set changes more positions simultaneously and may dilute the effect of focusing on the region associated with the strongest Undershoot signal. By contrast, Top-1 selection keeps the intervention more localized and preserves more of the original decoding trajectory outside the identified vulnerable region. We therefore use Top-1 sentence selection throughout the main experiments. Detailed localization statistics and intervention failure cases are provided in Appendix D and Appendix F.

5 Conclusion and Limitations

Our analyses suggest a difficulty–timing mismatch in dMLLMs: delayed commitment can strengthen reliance on self-generated text while weakening visual influence. OVERTURE uses Undershoot to localize potentially vulnerable content and selectively adjust its commitment timing without additional training. On MMaDA-8B-Base and LaViDa, it improves all reported metrics over the corresponding native schedulers on the evaluated subsets (Table 1), supporting trajectory-guided commitment-order intervention.

Limitations Our current evaluation focuses on representative 8B-scale dMLLM backbones and fixed decoding configurations. Different architectures can exhibit different trajectory geometries, motivating broader study of reference estimation and adaptive trajectory modeling across model families. Extending the current trajectory analysis to online settings, where vulnerable regions are estimated from partial denoising histories, is another direction for future work.

Ethics Statement

This work studies inference-time decoding for existing diffusion-based multimodal language models using established benchmarks and pretrained model checkpoints. No new human-subject data or personally identifying information are collected in this study.

Reproducibility Statement

The formulation of OVERTURE, including Context-Change Magnitude, Undershoot localization, priority decoding, and EOS/EOT handling, is specified in Section 3 and Algorithms 1–2. Experimental settings, benchmarks, metrics, and hyperparameters are provided in Section 4.1, with additional analyses and qualitative results included in the appendix.

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. URL https://arxiv.org/abs/2503.09573.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2021. URL https://arxiv.org/abs/2107.03006.

Shenxu Chang, Junchi Yu, Weixing Wang, Yongqiang Chen, Jialin Yu, Philip Torr, and Jindong Gu. Tracedet: Hallucination detection from the decoding trace of diffusion large language models, 2025. URL https://arxiv.org/abs/2510.01274.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models, 2022. URL https://arxiv.org/abs/2210.08933.

Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models, 2022. URL https://arxiv.org/abs/2211.15029.

Zhihao He, Tieyuan Chen, Kangyu Wang, Ziran Qin, Yang Shao, Chaofan Gan, Shijie Li, Zuxuan Wu, and Weiyao Lin. Vidlada: Bidirectional diffusion large language models for efficient video understanding, 2026. URL https://arxiv.org/abs/2601.17868.

Arshia Hemmat, Philip Torr, Yongqiang Chen, and Junchi Yu. Tdgnet: Hallucination detection in diffusion language models via temporal dynamic graphs, 2026. URL https://arxiv.org/abs/2602.08048.

Kai Syun Hou and James Kwok. Rethinking the generation order of block diffusion language models, 2026. URL https://arxiv.org/abs/2607.24306.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation, 2023. URL https://arxiv.org/abs/2311.17911.

Yatai Ji, Teng Wang, Yuying Ge, Zhiheng Liu, Sidi Yang, Ying Shan, and Ping Luo. From denoising to refining: A corrective framework for vision-language diffusion model, 2025. URL https://arxiv.org/abs/2510.19871.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions, 2025. URL https://arxiv.org/abs/2502.06768.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding, 2023. URL https://arxiv.org/abs/2311.16922.

Shufan Li, Jiuxiang Gu, Kangning Liu, Zhe Lin, Zijun Wei, Aditya Grover, and Jason Kuen. Lavida-o: Elastic large masked diffusion models for unified multimodal understanding and generation, 2025a. URL https://arxiv.org/abs/2509.19244.

Shufan Li, Jiuxiang Gu, Kangning Liu, Zhe Lin, Zijun Wei, Aditya Grover, and Jason Kuen. Sparse-lavida: Sparse multimodal discrete diffusion language models, 2025b. URL https://arxiv.org/abs/2512.14008.

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, 2025c. URL https://arxiv.org/abs/2505.16839.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models, 2025d. URL https://arxiv.org/abs/2508.10875.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation, 2022. URL https://arxiv.org/abs/2205.14217.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023. URL https://arxiv.org/abs/2305.10355.

Pan Lu, Swaroop Mishra, Tony 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, 2022. URL https://arxiv.org/abs/2209.09513.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts, 2023. URL https://arxiv.org/abs/2310.02255.

Nikita Mounier and Parsa Idehpour. Review, remask, refine (r3): Process-guided block diffusion for text generation, 2025. URL https://arxiv.org/abs/2507.08018.

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. URL https://arxiv.org/abs/2502.09992.

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. URL https://arxiv.org/abs/2406.07524.

Yingte Shu, Yuchuan Tian, Chao Xu, Yunhe Wang, and Hanting Chen. Deferred commitment decoding for diffusion language models, 2026. URL https://arxiv.org/abs/2601.02076.

Yueyi Sun, Yuhao Wang, Jason Li, Ye Tian, Tao Zhang, Jacky Mai, Yihan Wang, Haochen Wang, Jinbin Bai, Ling Yang, and Yunhai Tong. Perceptiondlm: Parallel region perception with multimodal diffusion language models, 2026. URL https://arxiv.org/abs/2606.19534.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2024. URL https://arxiv.org/abs/2405.09818.

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. URL https://arxiv.org/abs/2607.17652.

Ye Tian, Ling Yang, Jiongfan Yang, Anran Wang, Yu Tian, Jiani Zheng, Haochen Wang, Zhiyang Teng, Zhuochen Wang, Yinjie Wang, Yunhai Tong, Mengdi Wang, and Xiangtai Li. Mmada-parallel: Multimodal large diffusion language models for thinking-aware editing and generation, 2025. URL https://arxiv.org/abs/2511.09611.

Chengcheng Wang, Tingzhang Luo, Wenhao Li, Jianyuan Guo, and Chang Xu. Tacg: Trajectory-aware commit gating for diffusion language model decoding, 2026. URL https://arxiv.org/abs/2607.03236.

Wen Wang, Bozhen Fang, Chenchen Jing, Yongliang Shen, Yangyi Shen, Qiuyu Wang, Hao Ouyang, Hao Chen, and Chunhua Shen. Time is a feature: Exploiting temporal dynamics in diffusion language models, 2025. URL https://arxiv.org/abs/2508.09138.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need, 2024. URL https://arxiv.org/abs/2409.18869.

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, 2025. URL https://arxiv.org/abs/2505.22618.

Yue Wu and Jian Huang. Stability-weighted decoding for diffusion language models, 2026. URL https://arxiv.org/abs/2604.17068.

Liang Yan, Xu Jiang, Jian Ma, Yuhang Liu, Tian Bian, Qichao Wang, Abhishek Basu, Yu Rong, Tingyang Xu, Pengcheng Wu, Le Song, Imran Razzak, Junchi Yan, Zengfeng Huang, and Yutong Xie. A comprehensive survey of multimodal llms for scientific discovery. Preprint, 2026. URL https://www.preprints.org/manuscript/202602.1847. Also presented as a NeurIPS 2025 workshop poster; not peer-reviewed in the supplied preprint version.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models, 2025. URL https://arxiv.org/abs/2505.15809.

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. URL https://arxiv.org/abs/2505.16933.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding, 2025. URL https://arxiv.org/abs/2505.16990.

Lunbin Zeng, Jingfeng Yao, Bencheng Liao, Hongyuan Tao, Wenyu Liu, and Xinggang Wang. Diffusionvl: Translating any autoregressive models into diffusion vision language models, 2025. URL https://arxiv.org/abs/2512.15713.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model, 2024. URL https://arxiv.org/abs/2408.11039.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Llada 1.5: Variance-reduced preference optimization for large language diffusion models, 2025. URL https://arxiv.org/abs/2505.19223.

6 Appendix

A Related Work

A.1 Diffusion-Based Multimodal Large Language Models and Confidence-Based Scheduling

Diffusion-based language models recover discrete responses through iterative denoising, allowing multiple unresolved positions to produce candidates within the same denoising step (Austin et al.,

2021; Nie et al., 2025; Li et al., 2022; Gong et al., 2022; He et al., 2022; Sahoo et al., 2024). Diffusion-based multimodal large language models further combine visual encoding, text instructions, and masked responses, allowing generation to process image information using bidirectional context (Li et al., 2025c; Yang et al., 2025; You et al., 2025; Yu et al., 2025; Yan et al., 2026; Team, 2024; Zhou et al., 2024; Wang et al., 2024). Unlike autoregressive models with their fixed left-to-right order, dMLLMs must decide at each step which positions enter the committed state (Arriola et al., 2025; Kim et al., 2025). Their operation therefore involves two components: candidate prediction and position scheduling (He et al., 2026; Zeng et al., 2025; Li et al., 2025b; Tian et al., 2025; Ji et al., 2025; Li et al., 2025a; Sun et al., 2026; Zhu et al., 2025).

Existing studies have primarily improved the efficiency or overall quality of diffusion inference through parallel decoding, blockwise generation, caching-based acceleration, and confidence-based reordering (Arriola et al., 2025; Wu et al., 2025; Kim et al., 2025; Shu et al., 2026; Wang et al., 2026; Tian et al., 2026; Hou and Kwok, 2026; Wu and Huang, 2026). These studies establish an important fact: although multiple positions can produce candidates in parallel, actual generation remains jointly influenced by the commitment order, candidate confidence, and finalized context (Kim et al., 2025; Shu et al., 2026). When a position is committed determines which self-generated tokens it can access in subsequent steps. However, existing decoding studies have rarely analyzed commitment timing itself as an issue of multimodal reliability. The difficulty–timing mismatch examined in this work lies precisely within this gap. Linguistically predictable content may be finalized first, whereas content requiring visual evidence is deferred for decisions under different contextual conditions.

A.2 Internal Dynamics and Diagnostics of Generation Reliability

Internal model states provide a window into the generation process beyond the final output. Related studies have examined how internal representations contribute to subsequent predictions through context aggregation, information entropy evolution, attention distributions, and token-level importance measures (Wang et al., 2025; Chang et al., 2025; Hemmat et al., 2026; Wu and Huang, 2026). They have also sought to relate these dynamic quantities to generation stability, context dependence, or erroneous outputs. These studies suggest that generation reliability cannot be assessed solely from the final tokens or answers. Context dependence and changes in internal states may also reflect the stability of local predictions.

However, existing process diagnostics typically focus on a single layer, step, or token, making it difficult to jointly characterize the correspondence among denoising time, output positions, and generated content (Wang et al., 2025; Chang et al., 2025; Hemmat et al., 2026). Analyses of dMLLMs based on static attention or internal states at a single step face two limitations. They have difficulty describing the continuous evolution of the same output position throughout the complete denoising process. They also struggle to determine whether trajectory anomalies arising during this process can locate generated content requiring intervention. We therefore continuously track changes in attention context throughout denoising from the perspective of step–token trajectories and further analyze local trajectory anomalies.

A.3 Multimodal Hallucination Mitigation and Inference-Order Intervention

Multimodal hallucination mitigation methods encompass alignment during training, candidate distribution adjustment, visual conditioning enhancement, rejection or rollback resampling, and post-generation verification and correction (Mounier and Idehpour, 2025; Chang et al., 2025; Hemmat et al., 2026; Li et al., 2023; Leng et al., 2023; Huang et al., 2023). These methods alter model parameters, the relative probabilities of candidate tokens, the weights of visual and linguistic signals, or the content of the final answer. For diffusion-based generation, the order in which positions are committed offers an additional degree of freedom for intervention. A method can change which positions enter the response state first without manually replacing candidate tokens.

These approaches provide a range of tools for reliability at inference time, but leave two issues unresolved. First, without a process signal, deciding which positions to prioritize typically requires global rules, instantaneous confidence, or post hoc error labels (Kim et al., 2025; Shu et al., 2026). Second, changing the commitment order changes the subsequent context, so its effect cannot simply be interpreted as a local correction within a fixed answer. OVERTURE uses Undershoot to locate

positions requiring priority from natural decoding trajectories and adjusts their commitment order during second-stage generation. We then evaluate the effects of this decoding-order adjustment on generation outcomes and computational overhead across complete samples.

B Cross-Model Applicability of Undershoot

The applicability of Undershoot localization depends on a structural property of the denoising process: the dominant context-change positions should form a sufficiently stable advancing frontier across denoising steps and output positions. We examine this property on three diffusion-based multimodal language models, namely LaViDa (Li et al., 2025c), MMaDA (Yang et al., 2025), and LLaDA-V (You et al., 2025).

Consistent with the main text, we use the cache-relative *Context-Change Magnitude*. For denoising step t, layer l, and output position p, it is defined as

$$H_t(p) = \frac{1}{L} \sum_{l=0}^{L-1} \left[1 - \mathrm{clip}\left(\cos\left(C_{t,l,p}, C^{\mathrm{cache}}_{t-1,l,p}\right), 0, 1\right)\right]. \tag{6}$$

The dominant context-change position at step t is

$$r_t = \arg\max_p H_t(p), \tag{7}$$

and the sequence $\{r_t\}_t$ describes how the dominant Context-Change Magnitude moves over output positions during denoising. The current Undershoot detector assumes that this sequence exhibits a sufficiently coherent advancing trend, so that a reference trajectory can be fitted and local backward deviations can be meaningfully identified.

As shown in Figure 4, LaViDa and MMaDA generally exhibit stronger spatial continuity in their Context-Change Magnitude maps. Adjacent output positions tend to evolve in similar directions, and the dominant context-change positions form approximately continuous or step-like advancing frontiers. This behavior makes it possible to fit a reference trajectory and identify meaningful local deviations.

LLaDA-V exhibits a qualitatively different pattern. Its dominant context-change activity is more widely distributed over output positions, and the advancing frontier is substantially less stable. Consequently, the trajectory assumption required by the current Undershoot detector is weaker in this model. We therefore restrict the present OVERTURE evaluation to model configurations in which a stable step–token frontier can be observed.

C From Context-Change Magnitude to Undershoot

Our initial observation of Undershoot emerged from visualizing the complete evolution of attention-context changes rather than examining an isolated denoising step. We first organize the Context-Change Magnitude over three dimensions—denoising step, output-token position, and magnitude—and then project this representation onto the step–token plane.

Figure 5 illustrates this analysis process. The three-dimensional representation exposes the global evolution of context-change activity, while the two-dimensional projection makes the dominant advancing structure easier to trace. Most dominant peaks follow the overall progression of the generation process, whereas a subset of steps exhibits pronounced local backward deviations. These observations motivated the trajectory-relative definition of Undershoot used in the main text.

Importantly, Undershoot is not defined by a large Context-Change Magnitude alone. It is determined by the positional deviation of the dominant context-change peak from the reference trajectory at the corresponding generation stage. A strong local Context-Change Magnitude therefore does not necessarily constitute an Undershoot event.

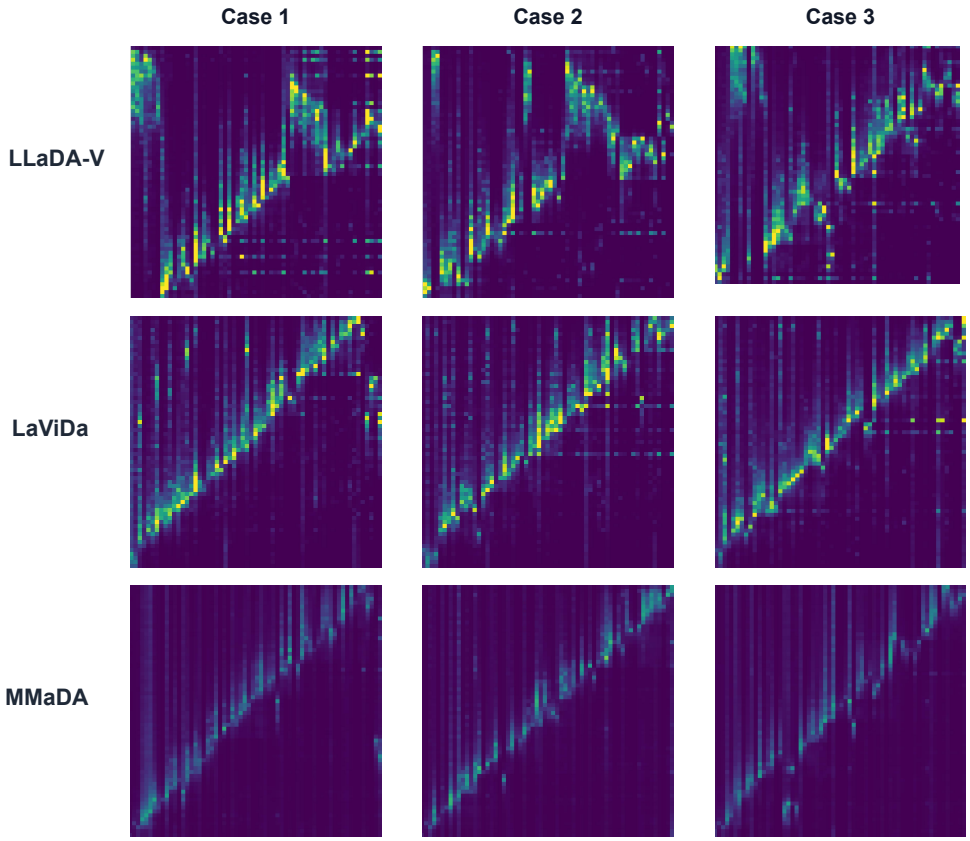


Figure 4: **Cross-model comparison of step–token Context-Change Magnitude maps.** Each row shows representative examples from LLaDA-V, LaViDa, and MMaDA. LaViDa and MMaDA exhibit relatively coherent advancing frontiers across denoising steps and output positions, whereas the dominant context-change activity in LLaDA-V is substantially more dispersed. This structural difference affects the applicability of trajectory-based Undershoot localization.

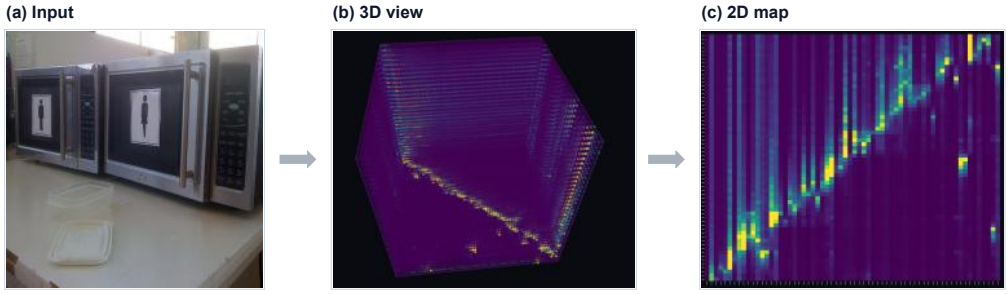


Figure 5: **From three-dimensional Context-Change Magnitude to the step–token trajectory.** From left to right: the input image, a three-dimensional visualization of Context-Change Magnitude over denoising steps and output positions, and its two-dimensional step–token projection. The projection exposes the dominant advancing frontier together with localized deviations that are difficult to identify from individual attention maps alone.

D Statistical Validation of Undershoot Localization

We further examine whether sentences selected by Undershoot are enriched for generation errors. For each diagnostic response, detected Undershoot events are mapped to the token positions committed at the corresponding denoising steps and then to their containing sentences. Candidate sentences are ranked according to their maximum Undershoot depth.

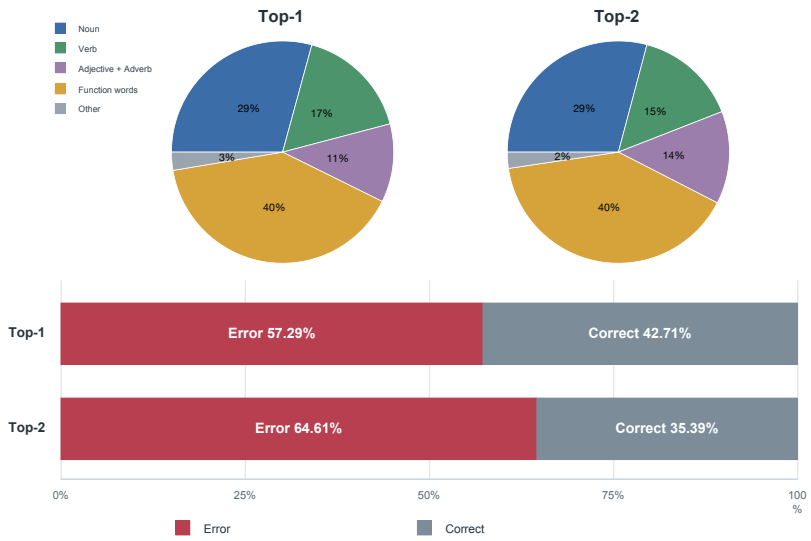


Figure 6: **Error composition of sentences selected by Undershoot.** Under Top-1 selection, 57.29% of selected sentences contain confirmed errors and 42.71% are judged correct. Under Top-2 selection, the confirmed error proportion increases to 64.61%, while 35.39% of selected sentences remain correct. These results characterize Undershoot as a risk-localization signal rather than a deterministic error detector.

As shown in Figure 6, confirmed erroneous content constitutes 57.29% of the Top-1 selected sentences. When the candidate range is extended to Top-2, the confirmed-error proportion increases to 64.61%. The remaining 42.71% and 35.39%, respectively, are sentences for which no confirmed error is identified.

We further analyze the part-of-speech distribution of confirmed reasoning-error tokens. For Top-1, function words account for the largest proportion (40%), followed by nouns (29%), verbs (17%), adjectives and adverbs (11%), and other categories (3%). A similar pattern is observed for Top-2, where function words and nouns remain unchanged at 40% and 29%, respectively, while verbs decrease slightly to 15%, adjectives and adverbs increase to 14%, and other categories decrease to 2%. Overall, the POS distribution remains highly consistent across Top-1 and Top-2, with function words and nouns together constituting the majority of confirmed reasoning-error tokens.

These results indicate that pronounced Undershoot is preferentially associated with vulnerable generation content, but it is not equivalent to an error label. OVERTURE therefore treats Undershoot as a prioritization signal for potentially vulnerable content rather than as a deterministic hallucination classifier.

E Extension to Text-Only Diffusion Language Models

To examine whether the scheduling effect extends beyond multimodal generation, we further apply OVERTURE to two text-only diffusion language models, LLaDA-8B-Instruct and Dream-v0-Instruct-7B. We evaluate generation quality on GSM8K, IFEval, and AlpacaEval. “Original” denotes the trace-recording baseline used in these text-only experiments.

As shown in Table 3, OVERTURE consistently improves all four IFEval metrics on both text-only models, with gains ranging from 8.67 to 17.00 percentage points. The effects on GSM8K and AlpacaEval are more mixed, indicating that the impact of commitment-timing intervention depends on the generation task. These results show that the scheduling behavior exploited by OVERTURE also transfers to text-only diffusion language models.

Table 3: Generation quality on text-only diffusion language models. For LLaDA, GSM8K uses 500 examples and IFEval/AlpacaEval use 100 examples; for Dream, all three benchmarks use 100 examples. AlpacaEval uses Codex-local judging, and the length-controlled win rate is reported as an exploratory metric.

Benchmark	Metric	Original	OVERTURE	Δ (pp)
LLaDA-8B-Instruct				
GSM8K	Accuracy	77.40	**76.20**	−1.20
IFEval	Prompt strict	42.00	**52.00**	+10.00
	Instruction strict	52.67	**61.33**	+8.67
	Prompt loose	46.00	**57.00**	+11.00
	Instruction loose	56.00	**64.67**	+8.67
AlpacaEval	Win rate	7.00	**4.00**	−3.00
	Length-controlled win rate	5.57	**2.83**	−2.75
Dream-v0-Instruct-7B				
GSM8K	Accuracy	77.00	**68.00**	−9.00
IFEval	Prompt strict	33.00	**50.00**	+17.00
	Instruction strict	47.33	**62.67**	+15.33
	Prompt loose	35.00	**52.00**	+17.00
	Instruction loose	48.67	**64.00**	+15.33
AlpacaEval	Win rate	9.50	**11.50**	+2.00
	Length-controlled win rate	20.31	**14.21**	−6.10

F Failure Case: Correct Localization but Ineffective Intervention

Localization and intervention play distinct roles in OVERTURE. Correctly identifying a vulnerable generation region does not guarantee that changing its commitment timing will alter the final prediction.

Figure 7 illustrates this failure mode. The trajectory-based procedure correctly identifies the generation region associated with the vulnerable sentence. Nevertheless, the priority intervention produces essentially the same content at the corresponding region.

In this example, the localized span already appears relatively early in the commitment trajectory. Priority decoding therefore advances the selected content by only a small number of steps, leaving the surrounding generation context close to the original trajectory. The same candidate can consequently remain dominant after intervention, and the final content remains unchanged.

This case illustrates that the effect of timing intervention depends not only on whether the vulnerable region is localized correctly, but also on how much its commitment time can actually be shifted.

G Qualitative Case Studies

We provide additional qualitative cases to illustrate how trajectory localization corresponds to concrete generated content and how changing commitment priority can alter the resulting response. Each case contains the input image, the corresponding Context-Change Magnitude map, the original generation, and the response obtained after the OVERTURE intervention.

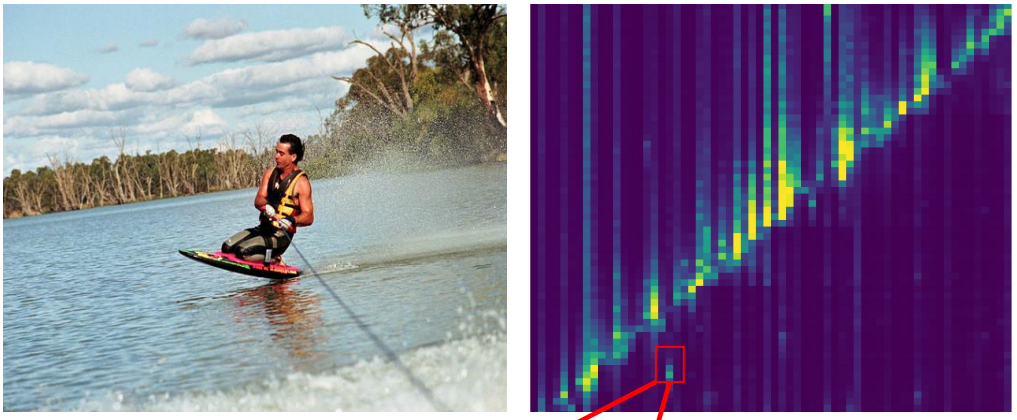


Figure 7: **A failure case with correct localization but ineffective intervention.** The local Undershoot event is correctly mapped to the vulnerable sentence in the generated response. However, after the corresponding positions are prioritized during the intervention, the generated content remains essentially unchanged. This example shows that successful localization does not necessarily lead to successful correction.

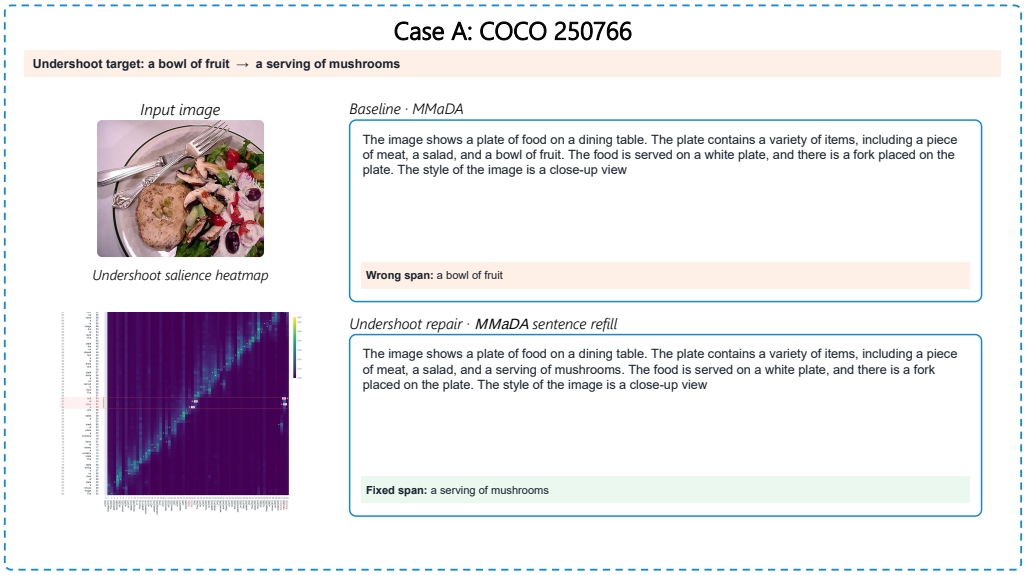


Figure 8: **Qualitative Case 1.** OVERTURE identifies a vulnerable region from the Context-Change Magnitude trajectory and prioritizes the corresponding sentence during the second generation pass. The example compares the original generation with the intervened response and highlights the content modified after priority decoding.

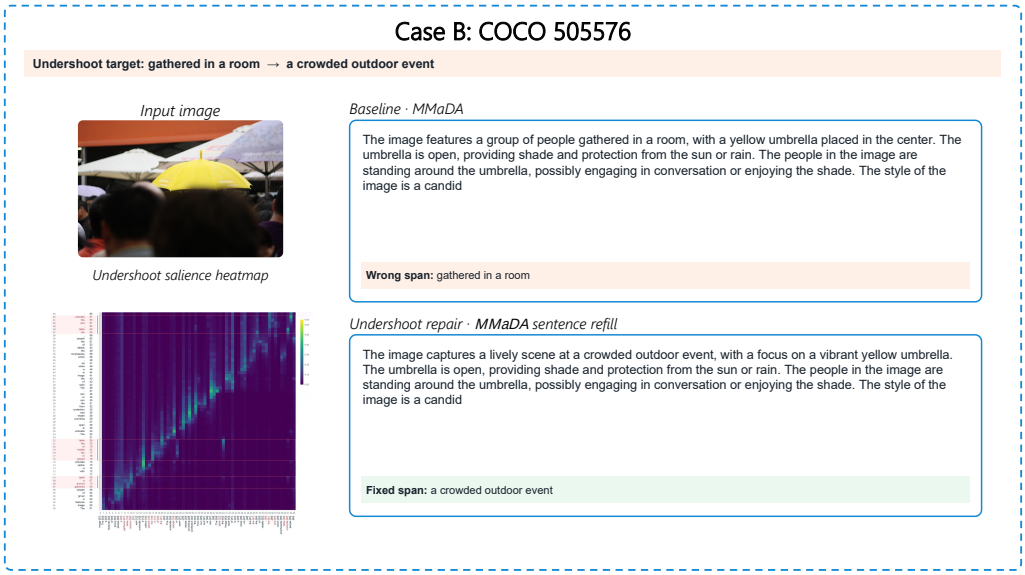


Figure 9: **Qualitative Case 2.** Another example of trajectory-guided intervention. The localized Undershoot region is mapped to sentence-level content, and changing its commitment priority produces a different generation path during intervention.

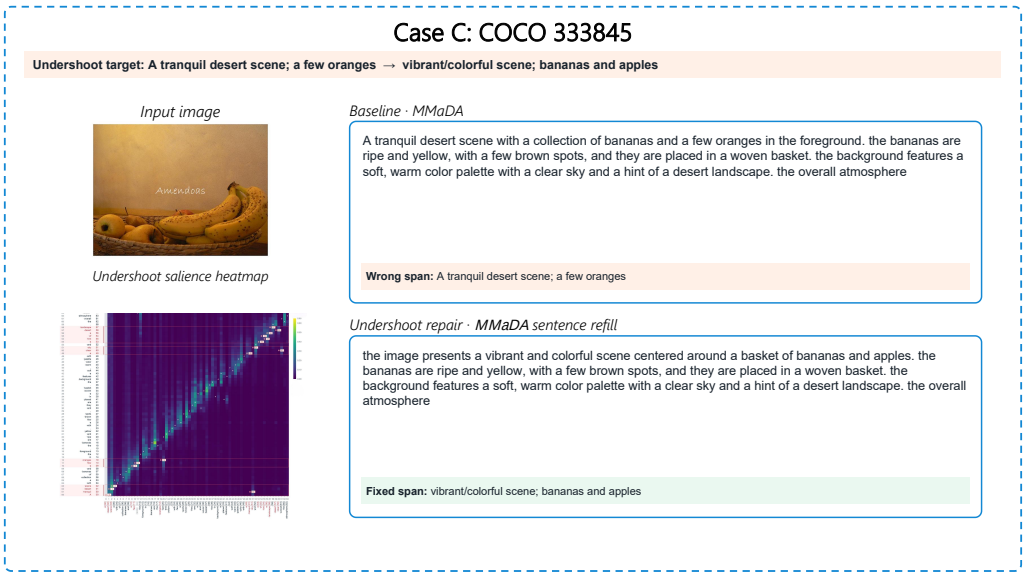


Figure 10: **Qualitative Case 3.** The Context-Change Magnitude trajectory identifies a vulnerable generation region, and OVERTURE prioritizes the corresponding positions during regeneration. The original and intervened responses illustrate the effect of changing decision timing.

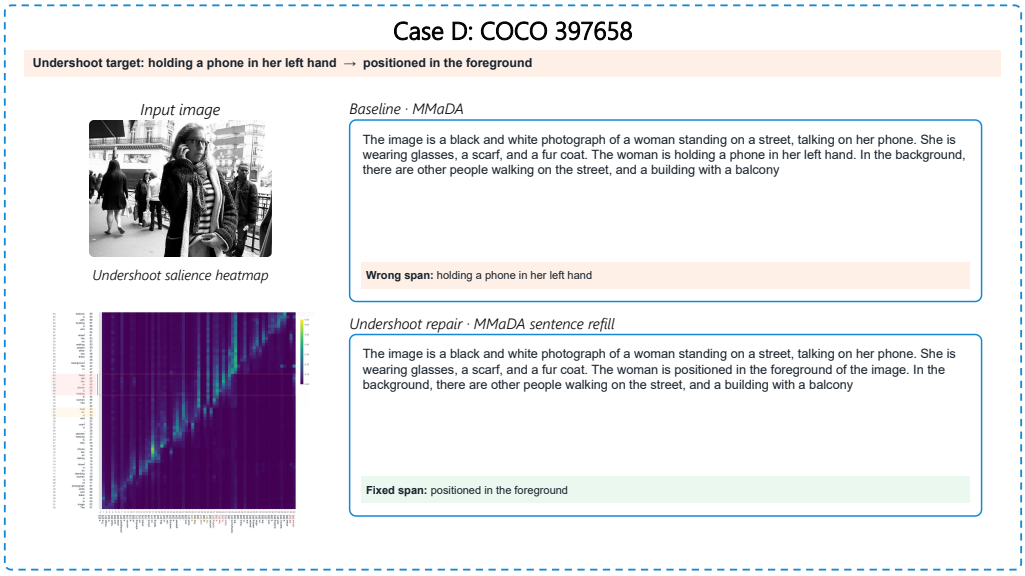


Figure 11: **Qualitative Case 4.** Additional example of trajectory-guided priority decoding. OVERTURE changes the generation path by modifying commitment order while leaving the model parameters, candidate logits, and candidate prediction mechanism unchanged.

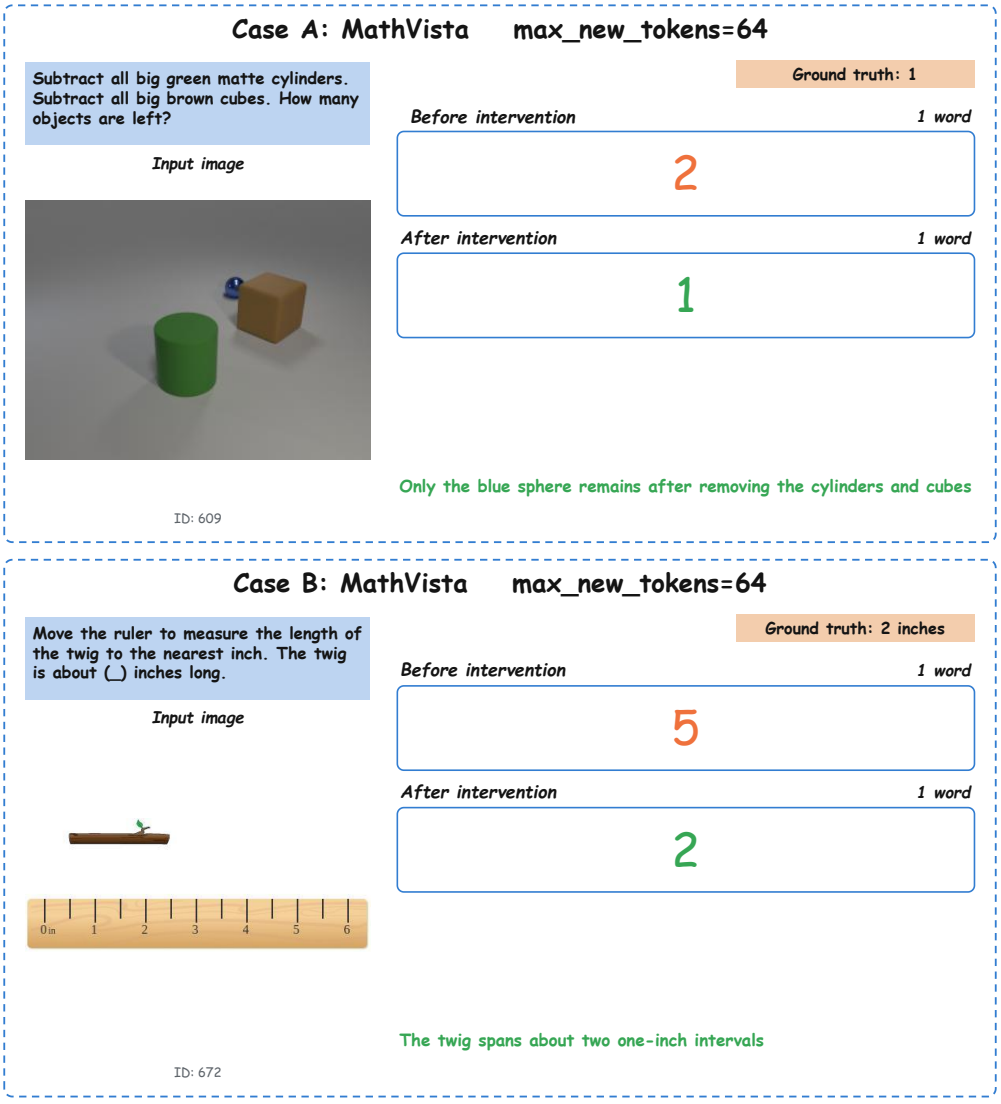


Figure 12: **Additional qualitative examples on MathVista.** Two representative cases in which OVERTURE changes an incorrect short answer to the ground-truth answer. In Case A, the predicted count changes from 2 to 1 after removing the specified objects. In Case B, the estimated twig length changes from 5 to 2 inches. These examples are post-hoc qualitative cases and are not used as evidence of average benchmark gains.

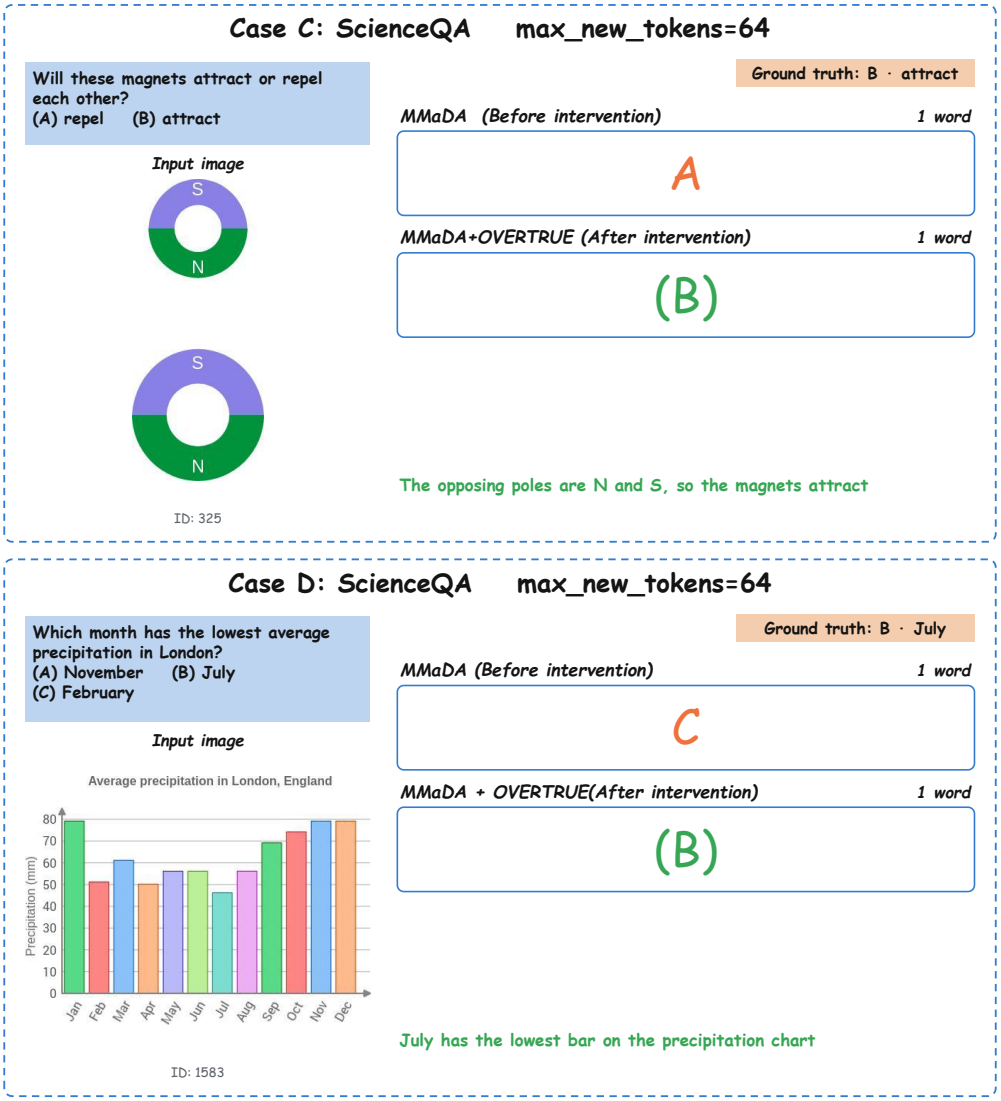


Figure 13: **Additional qualitative examples on ScienceQA.** In Case C, OVERTURE changes the prediction from "repel" to the correct answer "attract" for opposing magnetic poles. In Case D, it changes the precipitation answer from February to the correct month, July. Both cases illustrate corrections of visually grounded multiple-choice reasoning.



Figure 14: **Additional qualitative examples on LLaVA-Bench-COCO.** In Case E, the intervened response incorporates the parking-meter cue and completes the explanation of why the car may be parked at the roadside. In Case F, OVERTURE refines the generic "giraffe" description to identify the head, face, neck, and close-up framing visible in the image.

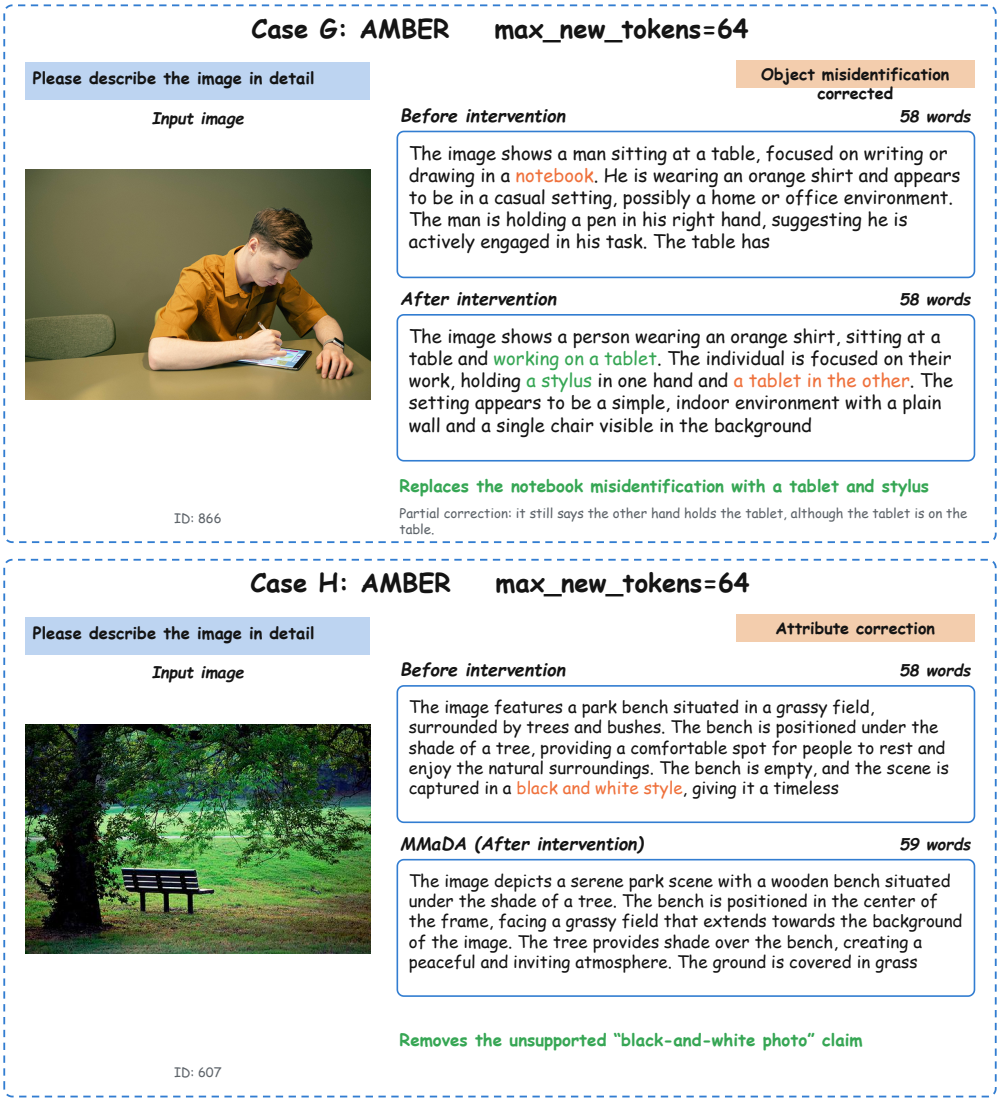


Figure 15: **Additional qualitative examples on AMBER.** Case G corrects the main object description from a notebook and pen to a tablet and stylus, although a residual spatial inconsistency remains in the intervened response. Case H removes the unsupported claim that the scene is a black-and-white photograph. These examples illustrate both successful hallucination reduction and the possibility of partial correction.



Figure 16: **Additional qualitative examples on DetailCaps.** In Case I, OVERTURE corrects the flower category while preserving relevant color and foliage information, increasing CAPTURE from 25.66 to 54.72. In Case J, the intervened response removes unsupported train-count and color claims, increasing CAPTURE from 28.39 to 44.91.