DEP2C: DEPENDENCY-AWARE CREDIT ASSIGNMENT AND COORDINATED DECODING IN MASKED IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Masked generative models (MGMs) enable efficient text-to-image generation through iterative parallel decoding. However, coarse-grained credit assignment may overlook the unequal downstream importance of committed tokens, while confidence-based decoding may jointly commit strongly dependent tokens. We propose **Dep2C**, a dependency-aware framework for fine-grained credit assignment and coordinated decoding. During RL post-training, **Dep2C-Credit** traces dependencies from future, prompt-relevant predictions back to previously committed tokens and accumulates them along the generation trajectory to estimate token importance. These estimates guide hierarchical credit allocation across decoding steps and individual tokens. During inference, **Dep2C-Decoding** combines prediction confidence and dependency degree to prioritize candidates, while using pairwise dependencies to constrain joint commitment. Experiments on Show-o and Lumina-DiMOO demonstrate improvements in both compositional generation and complex prompt following, achieving GenEval scores of 0.86 and 0.92 and DPG-Bench scores of 84.37 and 83.69, respectively.

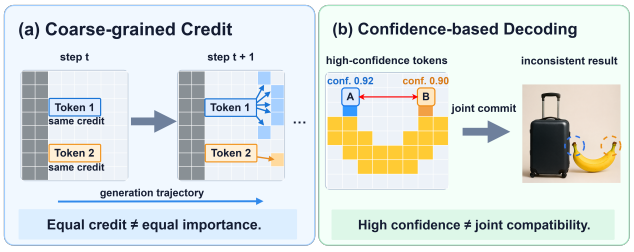

Figure 1: **Motivation.** (a) Unequal token importance. (b) Incompatible joint commitment.

1 INTRODUCTION

Alongside diffusion (Wu et al., 2025; Gao et al., 2025; Labs, 2024; Esser et al., 2024; Chen et al., 2024) and autoregressive (Xin et al., 2025b; Chen et al., 2025; Sun et al., 2024; Wang et al., 2024) models, masked generative models (MGMs) have emerged as a promising paradigm for text-to-image generation (Chang et al., 2022; 2023; Xie et al., 2025; Xin et al., 2025a; Bai et al., 2024; Yang et al., 2025). Operating on discrete visual tokens, MGMs iteratively predict masked positions in parallel and commit a subset at each step to condition subsequent predictions. This iterative yet parallel generation process offers a favorable trade-off between generation quality and efficiency.

Recent studies have explored RL-based method to improve MGMs (Ma et al., 2026; Zhang et al., 2026b;a; Luo et al., 2026). During training, these methods optimize iterative generation using image-level rewards, often allocating credit across generation steps. However, they largely overlook how individual committed tokens differ in their influence on subsequent predictions. During decoding, MGMs typically adopt confidence-based iterative revealing, selecting high-confidence masked tokens for parallel commitment (Chang et al., 2022; 2023). Yet confidence alone does not indicate whether multiple predictions are suitable for joint commitment.

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.

Both issues suggest that a token decision cannot be fully assessed in isolation, but should be considered in relation to other token decisions throughout iterative generation. We characterize such relations through token dependency, which provides a natural signal for guiding token-level decisions. Specifically, we consider two complementary forms: inter-step dependency and intra-step dependency. Across generation steps, later tokens depend on both the prompt and tokens generated at earlier steps. Therefore, the more strongly a token supports semantically related downstream tokens, the greater its influence on subsequent generation, suggesting that it should receive greater emphasis during training, as illustrated in Fig. 1 (a). Within the same step, strong dependency between tokens indicates that their predictions are not conditionally independent. Generating such tokens in parallel may therefore lead to inconsistent or conflicting predictions, as illustrated in Fig. 1(b), suggesting that strongly dependent tokens should be separated during decoding. These two forms of dependency thus play complementary roles: **strong inter-step dependencies should be emphasized during training, whereas strong intra-step dependencies should be avoided during parallel decoding.**

Building on these observations, we propose **Dep2C**, a dependency-aware framework for fine-grained credit assignment and coordinated parallel decoding in discrete image generation. During RL post-training, **Dep2C-Credit** traces dependencies from future, prompt-relevant predictions back to previously committed tokens and accumulates them along the generation trajectory to estimate token-level importance. The resulting importance scores guide hierarchical credit allocation across decoding steps and individual tokens. During inference, **Dep2C-Decoding** models dependencies among currently masked positions, combining prediction confidence and dependency degree to prioritize candidate tokens while using pairwise dependencies to constrain joint commitment. The two components can be applied independently and reuse model attention without additional forward passes. Dep2C-Decoding also preserves the original reveal schedule.

Experiments on Show-o (Xie et al., 2025) and Lumina-DiMOO (Xin et al., 2025a) demonstrate consistent improvements across different masked image generation backbones. Dep2C achieves GenEval scores of 0.86 and 0.92, improving the corresponding pretrained backbones by 0.19 and 0.07. It also reaches 84.37 and 83.69 on DPG-Bench, yielding gains of 1.84 and 1.82 points, respectively, while outperforming matched RL post-training baselines and maintaining strong perceptual quality. Component analyses show that Dep2C-Credit primarily improves compositional correctness, whereas Dep2C-Decoding provides larger gains in complex prompt following. Token-remasking experiments validate the proposed importance signal, and Dep2C-Decoding remains effective across decoding budgets and guidance strengths without additional model forward passes.

Our contributions are summarized as follows:

- We identify **token dependency** as a unified perspective for masked image generation, where inter-step dependencies indicate which tokens deserve greater training credit and intra-step dependencies indicate which tokens should not be committed together.
- We propose **Dep2C**, comprising **Dep2C-Credit** for dependency-aware token-level credit assignment and **Dep2C-Decoding** for coordinated parallel decoding, both without additional forward passes.
- Experiments on **Show-o** and **Lumina-DiMOO** demonstrate consistent gains on GenEval and DPG-Bench, with further analyses validating the proposed importance signal and decoding strategy.

2 Related Work

RL post-training for masked generative models. Reinforcement learning has been extended to masked and discrete generative models using rewards on completed generations (Luo et al., 2026; Wang et al., 2025b; Zhang et al., 2026b;a). Existing methods improve credit assignment through trajectory modeling, timestep reweighting, or stage-aware optimization. Related work on diffusion and flow models further distributes rewards across spatial or spatiotemporal regions (Shen et al., 2026; Savani et al., 2026). These approaches organize credit by trajectory, stage, or spatial location, without explicitly measuring how each committed token affects later predictions. Dep2C-Credit instead accumulates prompt-relevant dependencies from future predictions to estimate token importance, and uses them to allocate image-level advantages across steps and tokens.

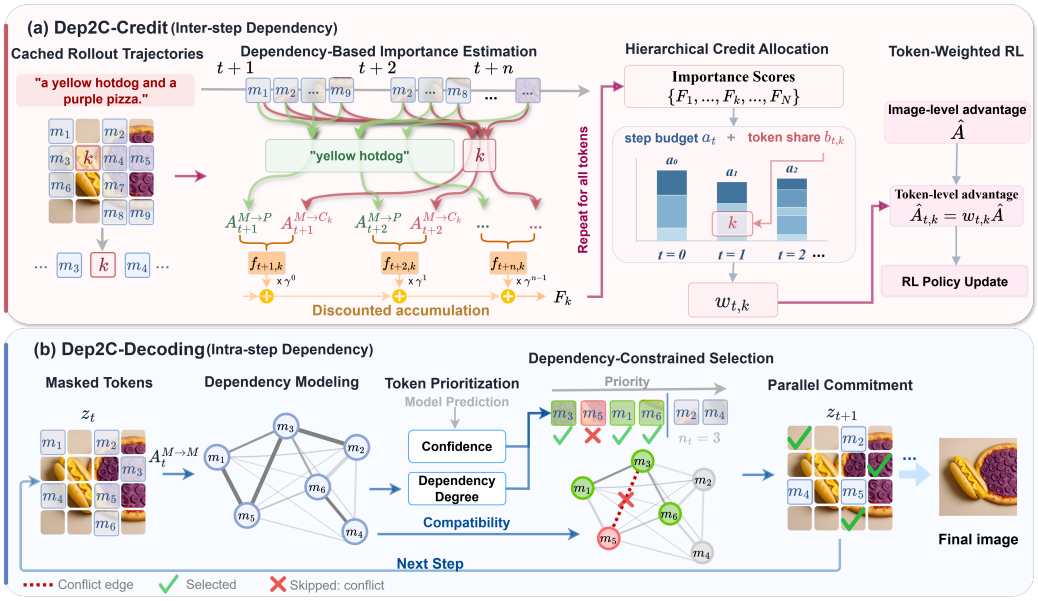

Figure 2: **Overview of Dep2C.** (a) Dep2C-Credit traces downstream dependencies for hierarchical credit allocation. (b) Dep2C-Decoding coordinates parallel commitment through dependency-aware prioritization and compatibility-constrained selection.

Parallel decoding for masked generative models. Masked generative models typically commit high-confidence predictions at each decoding step (Chang et al., 2022; 2023; Xie et al., 2025). Recent studies further improve parallel decoding through adaptive schedules and token selection strategies (Kang et al., 2026; Kim et al., 2026). However, existing decoding strategies largely focus on individual token confidence or selection priority, while interactions among tokens selected for simultaneous commitment remain less explored in masked image generation. Dep2C-Decoding explicitly accounts for these interactions to coordinate parallel commitment under a fixed reveal schedule, complementing Dep2C-Credit's dependency modeling across decoding steps.

3 METHOD

3.1 OVERVIEW

As illustrated in Fig. 2, **Dep2C** consists of two complementary components: **Dep2C-Credit** for RL post-training and **Dep2C-Decoding** for inference-time decoding. Both derive token dependencies from the model's internal attention.

As shown in Fig. 2(a), **Dep2C-Credit** operates on cached rollout trajectories and contains three stages: *dependency-based importance estimation*, *hierarchical credit allocation*, and *token-weighted RL*. After a trajectory is sampled, Dep2C-Credit traces and accumulates future prompt-relevant dependencies along that trajectory to estimate token importance, and then converts these scores into token weights for hierarchical credit allocation and policy optimization.

As shown in Fig. 2(b), **Dep2C-Decoding** consists of *dependency modeling*, *token prioritization*, and *dependency-constrained selection*. It constructs an intra-step dependency graph among masked tokens, combines prediction confidence with dependency degree to prioritize candidates, and constrains joint commitment to avoid selecting strongly dependent tokens together. The selected tokens are committed in parallel under the original reveal schedule, without additional model forward passes.

3.2 ATTENTION-DERIVED DEPENDENCY MODELING

We use the model's internal attention as a proxy for token dependency, avoiding additional prediction modules or forward passes. At decoding step s, let $\mathcal{M}_s$ denote the masked-token indices and $\mathcal{I}_s \subseteq$

$\mathcal{M}_s$ the subset selected for parallel commitment. We average attention over H heads and selected upper Transformer layers $\mathcal{L}$:

$$\overline{A}_s = \frac{1}{|\mathcal{L}|H} \sum_{\ell \in \mathcal{L}} \sum_{h=1}^{H} A_s^{\ell,h},$$

where $\overline{A}_{s,ij}$ denotes attention from query i to key j.

Inter-step dependency. A committed token is considered more important when subsequent prompt-relevant predictions depend on it more strongly. Suppose the prompt contains E entities, with $\mathcal{P}_e$ denoting the prompt-token indices of entity e. Using masked-to-prompt attention $A_s^{M\to P} = \overline{A}_s[\mathcal{M}_s, \mathcal{P}]$, we define the entity-normalized semantic relevance of $m \in \mathcal{M}_s$ as

$$\alpha_{s,m}^{(e)} = \frac{\sum_{p\in\mathcal{P}_e} A_{s,mp}^{M\to P}}{\sum_{m'\in\mathcal{M}_s}\sum_{p\in\mathcal{P}_e} A_{s,m'p}^{M\to P} + \epsilon}. \tag{1}$$

Combining this relevance with masked-to-committed attention $A_s^{M\to C_k} = \overline{A}_s[\mathcal{M}_s, \{k\}]$, we define

$$f_{s,k} = \frac{1}{E}\sum_{e=1}^{E}\sum_{m\in\mathcal{M}_s} \alpha_{s,m}^{(e)} A_{s,m}^{M\to C_k}, \tag{2}$$

where a larger $f_{s,k}$ indicates stronger support from token k to prompt-relevant predictions.

Intra-step dependency. For parallel decoding, strong dependency between two masked tokens indicates that they are less suitable for joint commitment. Using masked-to-masked attention $A_s^{M\to M} = \overline{A}_s[\mathcal{M}_s, \mathcal{M}_s]$, we define the symmetric pairwise dependency

$$D_{s,ij} = \tfrac{1}{2}\big(A_{s,ij}^{M\to M} + A_{s,ji}^{M\to M}\big), \tag{3}$$

for $i, j \in \mathcal{M}_s$. Setting $D_{s,ii} = 0$ yields an undirected weighted dependency graph, with dependency degree $d_{s,i} = \sum_{j\in\mathcal{M}_s} D_{s,ij}$.

3.3 DEP2C-CREDIT: INTER-STEP DEPENDENCY FOR CREDIT ASSIGNMENT

As shown in Fig. 2(a), **Dep2C-Credit** consists of dependency-based importance estimation, hierarchical credit allocation, and token-weighted RL. It traces and accumulates future prompt-relevant dependencies to estimate token importance, redistributes image-level credit across steps and tokens, and uses the resulting weights for fine-grained policy optimization.

Dependency-based importance estimation. For a token $k \in \mathcal{I}_t$ committed at step t, we accumulate its discounted future dependencies as

$$F_k = \sum_{s=t+1}^{T-1} \gamma^{s-t-1} f_{s,k}, \tag{4}$$

where $\gamma \in [0, 1]$ controls temporal discounting. A larger F_k indicates stronger dependency from subsequent prompt-relevant predictions.

To prevent a few abnormally large importance scores from dominating credit allocation, we cap each F_k at the η-quantile of the positive importance scores: $\widetilde{F}_k = \min(F_k, Q_\eta(\{F_j : F_j > 0\}))$.

Hierarchical credit allocation. We hierarchically allocate credit across decoding steps and individual tokens using a step budget a_t and a within-step weight $b_{t,k}$, respectively:

$$a_t = \frac{(q_t + \epsilon_F)^{\beta_t}}{\sum_{s:n_s>0}(q_s + \epsilon_F)^{\beta_t}}, \qquad b_{t,k} = \frac{n_t(\widetilde{F}_k + \epsilon_F)^{\beta_s}}{\sum_{j\in\mathcal{I}_t}(\widetilde{F}_j + \epsilon_F)^{\beta_s}}. \tag{5}$$

Here, $q_t = \frac{1}{n_t}\sum_{k\in\mathcal{I}_t}\widetilde{F}_k$ is the average importance of tokens committed at step t. The step budget a_t and within-step allocation $b_{t,k}$ distribute credit across steps and tokens, with β_t and β_s controlling their respective concentration. The final weight is $w_{t,k} = \frac{N}{n_t} a_t b_{t,k}$, satisfying $\sum_{k\in\mathcal{I}_t} w_{t,k} = N a_t$ and $\frac{1}{N}\sum_t \sum_{k\in\mathcal{I}_t} w_{t,k} = 1$, thus preserving the average credit scale.

Token-weighted RL. For each reward component r, we normalize rewards across G trajectories sampled for the same prompt and combine the resulting advantages as

$$\widehat{A}_g = \sum_r \lambda_r \frac{R_g^{(r)} - \mu^{(r)}}{\max(\sigma^{(r)}, \epsilon_R)}, \tag{6}$$

where $\mu^{(r)}$ and $\sigma^{(r)}$ denote the group mean and standard deviation, respectively, and λ_r controls the contribution of reward r.

We redistribute the image-level advantage to each committed token as $\widehat{A}_{g,t,k} = \widehat{A}_g \, \mathrm{sg}(w_{g,t,k})$, where sg denotes stop-gradient. The resulting token-weighted policy loss is

$$\mathcal{L}_{\text{policy}}^{\text{Dep2C}} = -\frac{1}{G} \sum_{g=1}^{G} \sum_t \sum_{k \in \mathcal{I}_{g,t}} \mathrm{sg}(w_{g,t,k}) \widehat{A}_g \, \ell_{g,t,k}(\theta), \tag{7}$$

where $\ell_{g,t,k}(\theta)$ denotes the per-token policy surrogate, including its baseline normalization factors.

This formulation scales each token's policy-gradient contribution according to its future dependency, strengthening updates for more influential tokens while attenuating those for less influential ones. The original policy objective and any regularization terms are otherwise retained, with no gradients propagating through credit estimation.

3.4 Dep2C-Decoding: Intra-Step Dependency for Coordinated Decoding

As shown in Fig. 2(b), **Dep2C-Decoding** models intra-step dependencies among masked tokens, uses them together with prediction confidence for token prioritization, and constrains joint commitment to separate strongly dependent candidates. This enables coordinated parallel decoding while preserving the original reveal schedule and number of model forward passes.

Dependency-aware prioritization. Highly connected tokens may provide useful context once committed. We therefore combine dependency degree with prediction confidence to prioritize candidate tokens. For each masked position $i \in \mathcal{M}_t$, let $\widetilde{x}_{t,i}$ denote its sampled token and $c_{t,i} = p_\theta(\widetilde{x}_{t,i} \mid \boldsymbol{z}_t, c)$ its prediction confidence. The normalized dependency degree and priority score are

$$\overline{d}_{t,i} = \frac{d_{t,i}}{\frac{1}{|\mathcal{M}_t|} \sum_{j \in \mathcal{M}_t} d_{t,j}}, \qquad s_{t,i} = c_{t,i} \overline{d}_{t,i}^{\,\beta_d}. \tag{8}$$

Here, β_d controls the influence of dependency degree, with $\beta_d = 0$ recovering confidence-based prioritization. When the mean degree is numerically zero, we set $\overline{d}_{t,i} = 1$.

Dependency-constrained selection. High priority does not guarantee compatibility for joint commitment, as strongly dependent candidates cannot condition on each other's newly revealed content when committed simultaneously. We therefore process candidates in descending order of $s_{t,i}$ and greedily construct $\mathcal{I}_t$, accepting a candidate i only if $\max_{j \in \mathcal{I}_t} D_{t,ij} \le \delta_t$, where δ_t limits pairwise dependency and the first candidate is accepted unconditionally. We adapt δ_t through a bounded threshold search, relaxing it when necessary to satisfy $|\mathcal{I}_t| = n_t$. Selected tokens are committed in parallel, while unselected positions remain masked for re-prediction with newly revealed context. Dep2C-Decoding preserves the original number of model forward passes, introducing only auxiliary computation for attention extraction, graph construction, and token selection.

4 Experiments

4.1 Experimental Setup

Training. Our primary experiments use Lumina-DiMOO (Xin et al., 2025a) with LoRA post-training at 512×512 resolution. We use Flow-GRPO prompts (Liu et al., 2025) and AdamW with a learning rate of 3×10^{-6}. Each update samples four prompts and 12 rollouts per prompt. Rollouts use confidence-based decoding with CFG $= 2.0$, 20 steps, a cosine reveal schedule, and temperature 1.0. We independently standardize the GenEval (Ghosh et al., 2023), UnifiedReward (Wang et al.,

Table 1: **GenEval compositional correctness (Ghosh et al., 2023).** Best results are in bold.

Method	Single	Two Obj.	Count	Color	Position	Attr.	Overall
OmniGen (Xiao et al., 2025)	**0.99**	0.86	0.64	0.85	0.31	0.55	0.70
OmniGen2 (Wu et al., 2026)	**0.99**	**0.96**	0.74	**0.98**	0.71	0.75	0.86
BAGEL (Deng et al., 2025)	**0.99**	0.94	0.81	0.88	0.64	0.63	0.82
SANA-1.6B (Xie et al., 2024)	**0.99**	0.77	0.62	0.88	0.21	0.47	0.66
FLUX.1-dev (Labs, 2024)	0.98	0.81	0.74	0.79	0.22	0.45	0.66
Qwen-Image (Wu et al., 2025)	**0.99**	0.92	**0.89**	0.88	**0.76**	**0.77**	**0.87**
Show-o (Xie et al., 2025)	0.97	0.79	0.70	0.81	0.29	0.49	0.67
+ Mask-GRPO (Luo et al., 2026)	**1.00**	0.87	0.74	0.83	0.60	0.63	0.78
+ **Dep2C (Ours)**	0.99	**0.94**	**0.85**	**0.87**	**0.81**	**0.70**	**0.86**
Lumina-DiMOO (Xin et al., 2025a)	**0.99**	0.93	0.79	0.88	0.80	0.73	0.85
+ TraceGRPO (Wang et al., 2025b)	**0.99**	0.94	0.84	**0.90**	0.84	0.74	0.88
+ Mask-GRPO (Luo et al., 2026)	**0.99**	0.94	0.88	0.86	0.89	0.73	0.88
+ **Dep2C (Ours)**	**0.99**	**0.97**	**0.92**	0.89	**0.94**	**0.79**	**0.92**

2025a), HPSv3 (Ma et al., 2025), and CLIP (Radford et al., 2021) rewards within each group and combine their advantages with weights 1.0, 0.2, 0.1, and 0.1, respectively.

Baselines and backbones. We train Mask-GRPO (Luo et al., 2026) under the same setting as Dep2C and adapt TraceRL (Wang et al., 2025b) to Lumina-DiMOO as TraceGRPO. We also compare Mask-GRPO and Dep2C on Show-o (Xie et al., 2025) under matched settings.

Evaluation. We evaluate compositional correctness on GenEval (Ghosh et al., 2023), complex prompt following on DPG-Bench (Hu et al., 2024), and image quality using UniPercept (Cao et al., 2025), ImageReward (Xu et al., 2023), HPSv3 (Ma et al., 2025), and DeQA (You et al., 2025). Unless otherwise specified, Lumina-DiMOO evaluation uses CFG $= 3.5$ and 64 decoding steps. Additional details are provided in Appendix A.1.

4.2 Main Results

We evaluate Dep2C on GenEval, DPG-Bench, and UniPercept across Show-o and Lumina-DiMOO backbones. Overall, Dep2C consistently improves both compositional correctness and complex prompt following across the two backbones, yielding GenEval gains of 0.19 and 0.07 and DPG-Bench gains of 1.84 and 1.82 points on Show-o and Lumina-DiMOO, respectively. These improvements also hold over matched RL post-training baselines, while maintaining strong perceptual quality.

Compositional correctness. Table 1 summarizes the GenEval comparisons, where Dep2C improves the overall score on Show-o from 0.67 to 0.86. It improves all compositional categories over Mask-GRPO except Single Object, with particularly large gains in counting (0.85 versus 0.74) and position (0.81 versus 0.60). On Lumina-DiMOO, Dep2C achieves 0.92, compared with 0.85 for the pretrained model and 0.88 for both TraceGRPO and Mask-GRPO. These consistent gains demonstrate that Dep2C generalizes across both backbones.

Complex prompt following. As summarized in Table 2, Dep2C achieves consistent improvements on DPG-Bench. On Show-o, Dep2C reaches 84.37, improving over the pretrained model and Mask-GRPO by 1.84 and 0.92 points, respectively, and achieving the highest overall score among the compared methods. On Lumina-DiMOO, it obtains 83.69, surpassing the pretrained model, TraceGRPO, and Mask-GRPO by 1.82, 1.52, and 1.18 points, respectively. These results demonstrate consistent improvements in complex prompt following across both backbones.

Image quality and preference. As shown in Table 3, Dep2C preserves strong perceptual quality alongside its semantic improvements. On Show-o, it improves most quality and preference metrics while maintaining a UniPercept mean comparable to Mask-GRPO (59.33 vs. 59.41). On Lumina-DiMOO, it consistently improves DeQA, ImageReward, HPSv3, and the UniPercept mean, reaching 59.38. These results indicate that the gains in compositional generation and prompt following do not come at the cost of overall visual quality.

Table 2: **Complex prompt following on DPG-Bench.** Best results are in bold.

Method	Attribute↑	Other↑	Global↑	Entity↑	Relation↑	Overall↑
DALL-E 3 (Betker et al., 2023)	88.39	89.83	**90.97**	89.61	90.58	83.50
FLUX.1-dev (Labs, 2024)	88.96	88.33	74.35	**90.00**	90.87	83.84
Janus-Pro (Chen et al., 2025)	89.40	89.48	86.90	88.90	89.32	**84.19**
OmniGen2 (Wu et al., 2026)	**90.18**	**90.27**	88.81	88.83	89.37	83.57
Show-o (Xie et al., 2025)	89.42	88.24	83.81	88.18	88.01	82.53
+ Mask-GRPO (Luo et al., 2026)	89.01	88.42	89.24	**90.88**	88.93	83.45
+ **Dep2C (Ours)**	**89.66**	**90.08**	**92.87**	89.31	**89.95**	**84.37**
Lumina-DiMOO (Xin et al., 2025a)	88.73	89.40	88.60	87.13	88.55	81.87
+ TraceGRPO (Wang et al., 2025b)	87.89	88.59	**90.51**	**88.92**	89.91	82.17
+ Mask-GRPO (Luo et al., 2026)	89.10	88.57	84.68	87.58	90.10	82.51
+ **Dep2C (Ours)**	**90.83**	**89.60**	89.15	88.54	**90.96**	**83.69**

Table 3: **Image quality and preference on DPG-Bench.** Best results are in bold.

Method	DeQA↑	ImageReward↑	HPSv3↑	UniPercept↑			
				Aesth.	Quality	Struct.	Mean
Show-o (Xie et al., 2025)	4.16	0.74	9.83	60.31	68.21	46.43	58.32
+ Mask-GRPO (Luo et al., 2026)	4.18	0.81	10.15	61.30	68.83	**48.11**	**59.41**
+ **Dep2C (Ours)**	**4.19**	**0.91**	**10.79**	**61.42**	**69.67**	46.91	59.33
Lumina-DiMOO (Xin et al., 2025a)	4.04	0.85	11.75	61.16	69.04	45.19	58.46
+ TraceGRPO (Wang et al., 2025b)	4.04	0.88	11.85	61.61	68.99	45.47	58.69
+ Mask-GRPO (Luo et al., 2026)	4.07	0.91	12.19	61.61	**69.60**	45.86	59.03
+ **Dep2C (Ours)**	**4.08**	**0.95**	**12.42**	**61.78**	69.58	**46.78**	**59.38**

Qualitative comparisons. As shown in Fig. 3, Dep2C improves spatial relations, attribute binding, and counting on both Show-o and Lumina-DiMOO, better matching the specified object placement, attributes, and quantities than the base model and Mask-GRPO.

4.3 Ablation Studies

We analyze the contributions of Dep2C-Credit and Dep2C-Decoding under the main-experiment setting, followed by an ablation of the coordinated decoding design. For the component analysis, we evaluate the Mask-GRPO and Dep2C-Credit models reported in the main experiments with both confidence-based decoding and Dep2C-Decoding.

Contributions of the two components. We isolate the contributions of Dep2C-Credit and Dep2C-Decoding by evaluating models trained with and without Dep2C-Credit under both confidence-based decoding and Dep2C-Decoding, yielding the four configurations in Table 4. Disabling both components corresponds to Mask-GRPO, while enabling both yields full Dep2C.

Dep2C-Credit improves GenEval from 0.88 to 0.91, with notable gains in position and attribute binding, while Dep2C-Decoding further improves structural consistency and strengthens multi-object generation and spatial relations. Combined, they raise GenEval to 0.92 and the UniPercept mean to 54.13, highlighting the complementary effects of the two components on compositional accuracy and structural quality.

Prioritization and compatibility. We isolate the two mechanisms of Dep2C-Decoding on a frozen pretrained Lumina-DiMOO, as summarized in Table 5. All variants retain confidence-based scoring, while *Degree only*, *Compatibility only*, and *Full* respectively introduce dependency-degree prioritization, pairwise dependency constraints, and their combination.

For all variants, D_t is computed from masked-to-masked attention within the same forward pass; Mean Dep. and Step-max Dep. measure the average and maximum dependency among tokens committed within each step. Degree prioritization modestly improves DPG-Bench from 81.87 to 82.05,



Figure 3: **Qualitative comparisons on GenEval.**

Table 4: **Dep2C component ablation.** UniPercept is evaluated on GenEval-generated images.

Components		GenEval ↑					UniPercept ↑			
Credit	Decoding	Overall	Two Obj.	Color	Position	Attr.	Mean	IAA	IQA	ISTA
–	–	0.88	0.94	0.86	0.89	0.73	53.57	**52.36**	75.16	33.18
✓	–	0.91	0.94	**0.89**	0.93	**0.79**	53.81	50.90	75.20	35.32
–	✓	0.88	0.93	0.88	0.90	0.71	53.66	52.22	75.00	33.77
✓	✓	**0.92**	**0.97**	**0.89**	**0.94**	**0.79**	**54.13**	51.18	**75.29**	**35.91**

while compatibility constraints raise it to 82.35 and substantially reduce within-step dependency. Combining both achieves the highest DPG-Bench score (82.54) and the lowest dependency values, indicating that compatibility constraints provide the main gain, with degree prioritization offering a complementary benefit.

4.4 Analysis of Credit Assignment

We validate F_i by testing whether re-masking a high-score token causes larger changes in subsequent predictions. For each token $i \in \mathcal{I}_t$ committed at step t, we re-mask i in every stored later state while keeping all other tokens fixed. Its discounted effect is measured as

$$C_i = \sum_{s=t+1}^{T-1} \frac{\gamma^{s-t-1}}{|\mathcal{M}_s|} \sum_{j \in \mathcal{M}_s} \mathrm{JS}\big(p_{\theta,j}(\cdot \mid \boldsymbol{z}_s, c) \,\big\|\, p_{\theta,j}(\cdot \mid \boldsymbol{z}_s^{-i}, c)\big)\,, \tag{9}$$

where $\boldsymbol{z}_s^{-i}$ is obtained by re-masking i in $\boldsymbol{z}_s$, and $\mathcal{M}_s$ denotes the originally masked positions. Predictions are recomputed from the stored states without sampling new trajectories. Additional implementation and aggregation details are provided in Appendix A.5.

Figure 4 shows that high-F_i tokens induce larger prediction shifts than random and low-F_i tokens. Binning tokens into within-step quintiles by F_i reveals a monotonic increase in mean downstream influence. Moreover, the original F_i scores exhibit a strong positive rank correlation with the measured effects (Spearman $\rho = 0.719$). Together, these results support F_i as a dependency-based proxy for downstream token influence, providing empirical evidence for dependency-aware credit allocation.

4.5 Analysis of Coordinated Decoding

We isolate the inference-time effect of Dep2C-Decoding on a frozen pretrained Lumina-DiMOO. For each comparison, the two decoding strategies use identical prompts, random seeds, sampling settings, reveal schedules, CFG scales, and decoding budgets, differing only in the token selection strategy. We further vary the CFG scale and decoding budget to evaluate the robustness of Dep2C-Decoding under different inference settings.

Table 5: **Ablation of Dep2C-Decoding on pretrained Lumina-DiMOO.**

Decoder	Mechanism		DPG↑	Dependency ($\times 10^{-3}$)↓	
	Degree	Compat.		Mean	Step-max
Confidence	–	–	81.87	0.902	28.0
Degree only	✓	–	82.05	0.868	27.6
Compatibility only	–	✓	82.35	0.564	13.9
Full Dep2C-Decoding	✓	✓	**82.54**	**0.560**	**13.8**

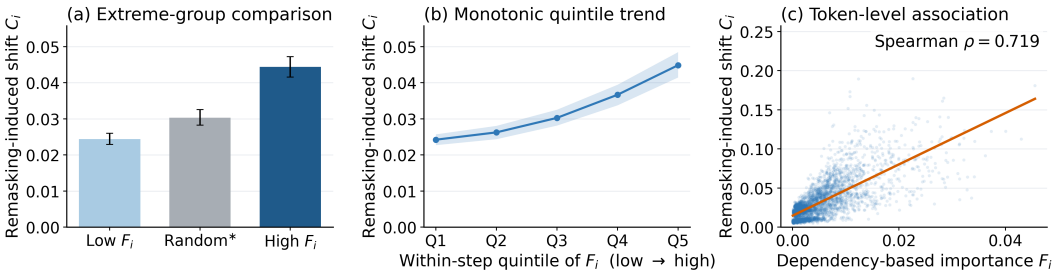

Figure 4: **Token-remasking validation of F_i.** (a) Mean C_i across token groups. (b) Mean C_i across within-step F_i quintiles. (c) Correlation between F_i and C_i. Random* excludes both extreme groups; error bars and shading indicate 95% bootstrap CIs.

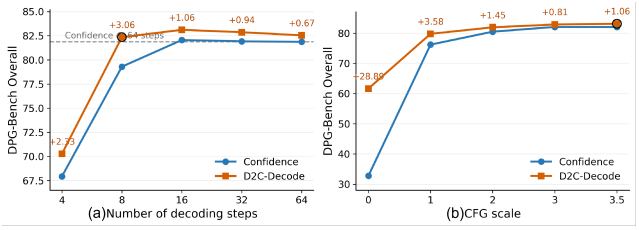

Figure 5: **Dep2C-Decoding across decoding budgets and CFG scales.**

Guidance strength. With the decoding budget fixed at 16 steps, Dep2C-Decoding outperforms confidence-based decoding across all evaluated CFG scales (Fig. 5(b)). The gain is largest under weak guidance, reaching 28.80 points at CFG $= 0$ and 3.58 points at CFG $= 1$, while remaining positive at stronger guidance. This indicates that dependency provides a complementary selection signal beyond token-wise confidence.

Decoding budget. With CFG fixed at 3.5, Dep2C-Decoding improves DPG-Bench across all evaluated budgets (Fig. 5(a)). The gains are largest at small budgets, where more positions are committed simultaneously. With 8 steps, it reaches 82.35, surpassing the 64-step confidence baseline of 81.87, and reaches 82.54 at 64 steps. This highlights its benefit under aggressive parallel decoding.

5 Conclusion

We present **Dep2C**, a dependency-aware framework for fine-grained credit assignment and coordinated decoding in masked image generation. Dep2C-Credit traces dependencies from future prompt-relevant predictions to committed tokens and accumulates their downstream importance along generation trajectories for hierarchical credit allocation. Dep2C-Decoding leverages dependencies among masked positions to prioritize candidates and constrain joint commitment. Experiments on Show-o and Lumina-DiMOO demonstrate consistent improvements in compositional generation and complex prompt following, further supported by component ablations and token-remasking analysis. Both components reuse model attention without additional model forward passes.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing, including language refinement, proofreading, and improving the clarity of technical descriptions. We did not use generative AI tools for developing the proposed method, conducting experiments, analyzing results, or generating scientific claims. All AI-assisted revisions were carefully reviewed and verified by the authors. We take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies masked image generation using publicly available models, datasets, and benchmarks. It does not involve human subjects, private data, or the release of new datasets. We do not identify additional ethical concerns specific to this work beyond the general risks associated with text-to-image generation.

REPRODUCIBILITY STATEMENT

The proposed methods are fully specified in Sec. 3. We provide detailed training and inference configurations, implementation details, ablation settings, hyperparameter analyses, and additional validation results in Sec. 4 and the Appendix. All experiments use publicly available backbones, datasets, reward models, and evaluation benchmarks.

REFERENCES

Jinbin Bai, Tian Ye, Wei Chow, Enxin Song, Qing-Guo Chen, Xiangtai Li, Zhen Dong, Lei Zhu, and Shuicheng Yan. Meissonic: Revitalizing masked generative transformers for efficient high-resolution text-to-image synthesis. In *The Thirteenth International Conference on Learning Representations*, 2024.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

Shuo Cao, Jiayang Li, Xiaohui Li, Yuandong Pu, Kaiwen Zhu, Yuanting Gao, Siqi Luo, Yi Xin, Qi Qin, Yu Zhou, Xiangyu Chen, Wenlong Zhang, Bin Fu, Yu Qiao, and Yihao Liu. Unipercept: Towards unified perceptual-level image understanding across aesthetics, quality, structure, and texture, 2025. URL https://arxiv.org/abs/2512.21675.

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

Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations*, 2024.

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

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

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, et al. Seedream 3.0 technical report. *arXiv preprint arXiv:2504.11346*, 2025.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment, 2023. URL https://arxiv.org/abs/2310.11513.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment, 2024. URL https://arxiv.org/abs/2403.05135.

Wonjun Kang, Byeongkeun Ahn, Minjae Lee, Kevin Galim, Seunghyuk Oh, Hyung Il Koo, and Nam Ik Cho. Uncage: Contrastive attention guidance for masked generative transformers in text-to-image generation. *IEEE Access*, 14:6566–6582, 2026.

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

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

Yifu Luo, Xinhao Hu, Keyu Fan, Haoyuan Sun, Zeyu Chen, Bo Xia, Tiantian Zhang, Yongzhe Chang, and Xueqian Wang. Reinforcement learning meets masked generative models: Mask-grpo for text-to-image generation. *Advances in Neural Information Processing Systems*, 38:108460–108485, 2026.

Tianren Ma, Mu Zhang, Yibing Wang, and Qixiang Ye. Consolidating reinforcement learning for multimodal discrete diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 147950–147971, 2026.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15086–15095, October 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Yash Savani, Branislav Kveton, Yuchen Liu, Yilin Wang, Jing Shi, Subhojyoti Mukherjee, Nikos Vlassis, and Krishna Kumar Singh. Stepwise credit assignment for grpo on flow-matching models, 2026. URL https://arxiv.org/abs/2603.28718.

Jinjie Shen, Wei Deng, Xian Hu, Daiguo Zhou, and Jian Luan. Star: Spatiotemporal adaptive reward allocation for text-to-image rl post-training. *arXiv preprint arXiv:2606.17979*, 2026.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025a.

Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforcement learning framework for diffusion large language models, 2025b. URL https://arxiv.org/abs/2509.06949.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Towards instruction-aligned multimodal generation, 2026. URL https://arxiv.org/abs/2506.18871.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13294–13304, June 2025.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution image synthesis with linear diffusion transformers, 2024. URL https://arxiv.org/abs/2410.10629.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations*, volume 2025, pp. 28240–28264, 2025.

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

Yi Xin, Juncheng Yan, Qi Qin, Zhen Li, Dongyang Liu, Shicheng Li, Victor Shea-Jay Huang, Yupeng Zhou, Renrui Zhang, Le Zhuo, Tiancheng Han, Xiaoqing Sun, Siqi Luo, Mengmeng Wang, Bin Fu, Yuewen Cao, Hongsheng Li, Guangtao Zhai, Xiaohong Liu, Yu Qiao, and Peng Gao. Lumina-mgpt 2.0: Stand-alone autoregressive image modeling, 2025b. URL https://arxiv.org/abs/2507.17801.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023. URL https://arxiv.org/abs/2304.05977.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025.

Zhiyuan You, Xin Cai, Jinjin Gu, Tianfan Xue, and Chao Dong. Teaching large language models to regress accurate image quality scores using score distribution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14483–14494, 2025.

Guohui Zhang, Hu Yu, Xiaoxiao Ma, Yaning Pan, Hang Xu, Jie Huang, and Feng Zhao. Maskfocus: Focusing policy optimization on critical steps for masked image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5956–5966, 2026a.

Mu Zhang, Tianren Ma, Yunfan Liu, Kun Hu, and Qixiang Ye. Rebrl: Reinforcing discrete visual diffusion models with rebalanced timestep credits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43135–43144, 2026b.

Table 6: **Implementation details on Lumina-DiMOO.**

Setting	Value
Model and optimization	
Base model	Lumina-DiMOO
Resolution	512×512
Fine-tuning	LoRA, rank 32, $\alpha = 64$, dropout 0.05
Optimizer	AdamW, lr 3×10^{-6}
Training hardware	$4\times$ NVIDIA H100
Rollout configuration	
Prompts / rollouts per prompt	4 / 12
Decoding steps	20
Reveal schedule	Cosine
CFG / temperature	2.0 / 1.0
Rollout decoder	Confidence-based
Reward configuration	
Reward components	GenEval, UnifiedReward, HPSv3, CLIP
Reward weights	$1.0,\ 0.2,\ 0.1,\ 0.1$
Normalization	Per reward, within each prompt group
Dep2C-Credit	
Attention layers	Final 30%
Head / layer aggregation	Mean / mean
Trajectory discount γ	0.9
Upper clipping quantile η	0.99

A Additional Experimental Details and Analyses

A.1 Implementation Details

We fine-tune Lumina-DiMOO (Xin et al., 2025a) with LoRA using prompts from Flow-GRPO (Liu et al., 2025). Each training update uses four prompts with 12 rollouts per prompt. Each reward component is independently standardized within its prompt group before advantage aggregation. Table 6 summarizes the training setup and Dep2C-Credit configuration.

A.2 Optimization Dynamics

We compare Dep2C-Credit with Mask-GRPO under matched initialization, training data, reward configuration, and rollout settings. Figure 6 reports the online GenEval success rate, defined as the fraction of training rollouts satisfying the strict GenEval criterion.

Both methods improve similarly during the first 3K steps. Thereafter, Mask-GRPO exhibits a marked decrease in online success rate, whereas Dep2C-Credit remains stable through the observed 4K-step training horizon. We therefore conduct an additional component comparison at a matched 3K-step budget in Sec. A.3, before the observed baseline collapse.

These trajectories suggest improved optimization stability with Dep2C-Credit in the evaluated setting. Since the success rate is measured on training rollouts, it serves as an optimization diagnostic rather than an independent benchmark evaluation.

A.3 Matched-Budget Component Ablation

To isolate component contributions under a matched training budget, we evaluate the Mask-GRPO and Dep2C-Credit checkpoints at 3K steps, each with confidence-based decoding and Dep2C-Decoding. The four configurations are compared in Table 7.

Under confidence-based decoding, Dep2C-Credit improves GenEval Overall from 0.88 to 0.90, with gains in counting, position, and attribute binding. Dep2C-Decoding improves UniPercept ISTA on

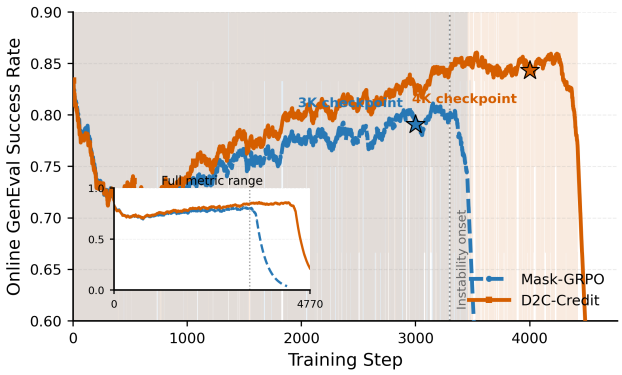

Figure 6: **Online optimization dynamics.** GenEval success rate on training rollouts for Mask-GRPO and Dep2C-Credit.

Table 7: **Component ablation at a matched 3K-step budget.** UniPercept is evaluated on GenEval-generated images.

Components		GenEval ↑							UniPercept ↑			
Credit	**Decoding**	**Overall**	**Single**	**Two Obj.**	**Count**	**Color**	**Position**	**Attr.**	**Mean**	**IAA**	**IQA**	**ISTA**
–	–	0.88	0.99	**0.96**	0.86	0.87	0.87	0.72	53.42	52.07	75.12	33.06
✓	–	0.90	**1.00**	0.95	**0.93**	0.88	**0.92**	0.75	53.42	51.41	75.02	33.85
–	✓	0.88	0.99	0.92	0.89	0.88	0.87	0.74	53.65	**52.20**	75.08	33.68
✓	✓	**0.91**	0.99	**0.96**	**0.93**	**0.89**	0.91	**0.77**	**53.68**	51.48	**75.24**	**34.32**

both checkpoints, from 33.06 to 33.68 for Mask-GRPO and from 33.85 to 34.32 for Dep2C-Credit. Combining both components achieves a GenEval Overall score of 0.91 and a UniPercept Mean of 53.68.

These results support the complementary contributions of credit assignment and coordinated decoding under a matched training budget.

A.4 Hierarchical Credit Allocation

We further examine the two allocation levels in Dep2C-Credit and their sensitivity to the concentration parameters β_t and β_s. All variants are initialized from Lumina-DiMOO and trained for 1K steps with matched data, rewards, and remaining training settings. These short-run experiments are analyzed separately from the 3K-step component comparison in Sec. A.3.

A.4.1 Contributions of the Two Allocation Levels

Hierarchical credit allocation determines both the total credit assigned to each generation step and its distribution among tokens committed within that step. Setting $\beta_t = 0$ assigns equal budgets to nonempty steps, while $\beta_s = 0$ assigns equal token weights within each step.

Notably, $(\beta_t, \beta_s) = (0, 0)$ corresponds to an *equal-step* rather than a token-uniform baseline. With K nonempty steps, the resulting token weight is $w_{t,k} = N/(Kn_t)$.

As shown in Table 8, step-level budgeting and within-step allocation individually improve GenEval from 0.4266 to 0.4604 and 0.4607, respectively. Combining both achieves 0.4738, exceeding either allocation level alone under the matched 1K-step training budget.

The importance scores F_k are held fixed in these comparisons; only the allocation exponents are changed. Within-step allocation here uses the inter-step importance score F_k and is distinct from the intra-step dependencies used by Dep2C-Decoding.

Table 8: **Ablation of hierarchical credit allocation.** All variants are trained under a matched 1K-step budget and evaluated using 16 decoding steps with CFG $= 0$.

Allocation	(β_t, β_s)	**GenEval↑**
Equal-step baseline	$(0, 0)$	0.4266
Step budget only	$(0.25, 0)$	0.4604
Within-step allocation only	$(0, 0.5)$	0.4607
Hierarchical	$(0.25, 0.5)$	**0.4738**

Table 9: **Sensitivity to timestep-budget concentration.** We vary β_t with $\beta_s = 0$ under a 1K-step training budget. Category scores are percentages; Overall is reported on a $[0, 1]$ scale.

β_t	**Single↑**	**Two Obj.↑**	**Counting↑**	**Colors↑**	**Position↑**	**Attr.↑**	**Overall↑**
0	78.44	31.06	50.00	57.45	21.50	17.50	0.4266
0.25	**79.69**	33.33	**55.00**	**62.23**	**23.75**	**22.25**	**0.4604**
0.5	74.06	31.31	49.38	60.90	**23.75**	18.50	0.4298
1.0	76.88	**33.84**	47.50	60.64	21.75	17.50	0.4302

A.4.2 Sensitivity to Allocation Strength

We vary one concentration parameter while fixing the other at zero. These one-dimensional sweeps isolate the effect of each allocation level rather than constituting a joint hyperparameter search.

Timestep budgeting. With $\beta_s = 0$, $\beta_t = 0.25$ achieves the highest overall GenEval score among the tested values (0.4604), compared with 0.4266 under equal-step budgeting (Table 9). Increasing β_t to 0.5 or 1.0 reduces the overall score to 0.4298 and 0.4302, respectively. Thus, stronger step-level credit concentration does not provide further gains in this setting.

Within-step allocation. With $\beta_t = 0$, the overall GenEval score increases from 0.4266 at $\beta_s = 0$ to 0.4491 at $\beta_s = 0.25$ and 0.4607 at $\beta_s = 0.5$, before decreasing to 0.4074 at $\beta_s = 1.0$ (Table 10).

Counting and Position reach their highest scores at $\beta_s = 0.25$, whereas the overall score peaks at $\beta_s = 0.5$. These results suggest that moderate within-step credit concentration is preferable to the strongest tested setting under the short-run protocol.

Together, the one-dimensional sweeps support moderate allocation strength in the evaluated setting. The joint configuration $(\beta_t, \beta_s) = (0.25, 0.5)$ is evaluated separately in Table 8; these sweeps do not establish a globally optimal parameter pair.

Table 10: **Sensitivity to within-step allocation concentration.** We vary β_s with $\beta_t = 0$ under a 1K-step training budget. Category scores are percentages; Overall is reported on a $[0, 1]$ scale.

β_s	**Single↑**	**Two Obj.↑**	**Counting↑**	**Colors↑**	**Position↑**	**Attr.↑**	**Overall↑**
0	78.44	31.06	50.00	57.45	21.50	17.50	0.4266
0.25	79.06	31.31	**54.69**	61.17	**26.00**	17.25	0.4491
0.5	**80.31**	**33.33**	54.37	**64.89**	25.25	**18.25**	**0.4607**
1.0	76.56	26.52	50.31	58.78	18.50	13.75	0.4074

A.5 Implementation Details of the Token-Remasking Analysis

We provide additional details of the token-remasking experiment in Sec. 4.4. This analysis is performed on fixed decoding trajectories and involves no reward computation, policy optimization, or counterfactual trajectory sampling.

Reference trajectories. We generate reference trajectories using confidence-based decoding with 20 decoding steps, CFG $= 2.0$, and temperature 1.0. For each trajectory, we record all intermediate

states $\{\boldsymbol{z}_t\}_{t=0}^{T}$, the step at which each visual token is committed, and its importance score F_i. The counterfactual prediction shift is evaluated using the conditional model distribution corresponding to CFG $= 0$, consistent with the model branch used to compute F_i.

Selection of probe steps and tokens. We select up to four reveal steps from each trajectory. A step is eligible if it commits at least nine tokens and has at least two subsequent stored states. If more than four steps are eligible, approximately evenly spaced steps are selected from the eligible range.

All token groups are constructed separately within each selected reveal step. For a step containing n_s committed tokens, we rank them by F_i and set $k = \min(4, \lfloor n_s/3 \rfloor)$. The top-$k$ and bottom-k tokens form the high-F_i and low-F_i groups, respectively. After excluding these two groups, we uniformly sample k tokens from the remaining positions to form the random group. For the quantile analysis, the committed tokens are additionally divided into five approximately equal groups according to F_i, from which up to two tokens per quintile are sampled. Constructing all groups within the same reveal step controls for differences in the remaining decoding horizon.

Fixed-state intervention. For each sampled token $i \in \mathcal{I}_s$, we revisit every stored state $\boldsymbol{z}_t$ after its commitment, where $t > s$. We construct $\boldsymbol{z}_t^{-i}$ by re-masking only token i, while keeping every other token identical to the original state. The model is then evaluated on both states without continuing either trajectory.

The comparison is restricted to $\mathcal{M}_t$, the positions that remain masked in the original state $\boldsymbol{z}_t$. The newly re-masked position i is therefore excluded from the evaluated queries. At every original masked position, we compute the Jensen–Shannon divergence between the two predictive distributions over the visual codebook. These divergences are averaged uniformly across $\mathcal{M}_t$ and discounted over subsequent steps according to Eq. 9. No entity-level filtering or attention weighting is used when computing C_i.

Aggregation and visualization. We use each trajectory–reveal-step pair as the basic statistical unit. For Fig. 4(a), let $\mathcal{P}_{r,s,g}$ denote the probe tokens assigned to group $g \in \{\mathrm{Low}, \mathrm{Random}, \mathrm{High}\}$ at reveal step s of trajectory r. We first compute the within-unit mean

$$\overline{C}_{r,s,g} = \frac{1}{|\mathcal{P}_{r,s,g}|} \sum_{i \in \mathcal{P}_{r,s,g}} C_i, \tag{10}$$

and then average $\overline{C}_{r,s,g}$ equally across all eligible trajectory–step units. This prevents steps containing more probe tokens from receiving greater weight. Random* is sampled after excluding tokens in the low- and high-F_i groups.

For Fig. 4(b), we apply the same procedure to the five within-step F_i quintiles. Specifically, we first average C_i over the sampled tokens in each trajectory–step–quintile tuple and then average these unit means across trajectories and reveal steps. Because the quintiles are constructed separately at each reveal step, the comparison controls for the number of future states available to the sampled tokens.

Figure 4(c) uses individual sampled probe tokens without the preceding unit-level averaging. We compute Spearman's rank correlation between the raw importance scores $\{F_i\}$ and the corresponding remasking effects $\{C_i\}$. The displayed line is an ordinary least-squares fit included only to visualize the overall trend and is not used for statistical inference.

For panels (a) and (b), 95% confidence intervals are obtained by resampling the trajectory–reveal-step unit means with replacement 2,000 times. The lower and upper bounds correspond to the 2.5th and 97.5th percentiles of the bootstrap means. Accordingly, the error bars in panel (a) and the shaded region in panel (b) reflect variation across trajectory–step units rather than across individual tokens.

A.6 Additional Qualitative Results

We provide additional qualitative results to further examine the generalization and generation quality of Dep2C. We first consider an additional Show-o setting in which CLIP is used as the training

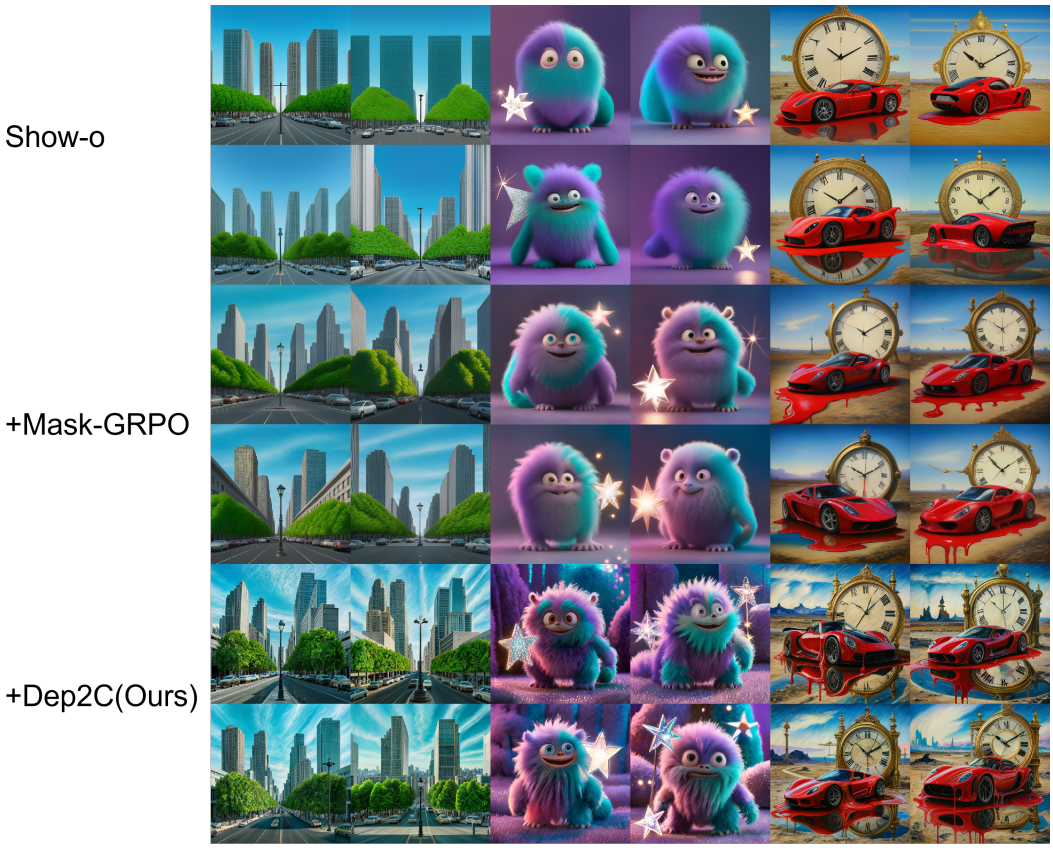

Figure 7: **Qualitative comparison on Show-o with CLIP reward.** Compared with the pretrained Show-o and Mask-GRPO, Dep2C better captures complex prompt semantics while producing more coherent, detailed, and visually appealing generations.

reward. Although this setting is not included in the main experiments, it provides a complementary test of whether dependency-aware credit assignment remains effective under a different reward signal. We then present additional generations from Lumina-DiMOO with Dep2C to illustrate its performance across a broader range of complex prompts.

Show-o with CLIP reward. As an additional study, we apply Dep2C to Show-o using CLIP as the training reward. Figures 7 and 8 compare the pretrained Show-o, Mask-GRPO, and Dep2C under this setting. Across diverse and semantically complex prompts, Dep2C produces generations that more faithfully capture the intended scene composition and semantic concepts, while exhibiting substantially improved visual quality and structural coherence. These results suggest that dependency-aware credit assignment remains effective under a different reward signal, improving both complex prompt following and perceptual quality.

Additional generations on Lumina-DiMOO. Figure 9 presents additional generations from Lumina-DiMOO with Dep2C on diverse DPG-Bench prompts. Rather than providing pairwise comparisons, we show a broader set of outputs to illustrate the generation capability of the full method across complex scenes, multiple objects, attribute binding, and spatial relationships. These results further show that Dep2C maintains strong visual quality while following diverse compositional descriptions.

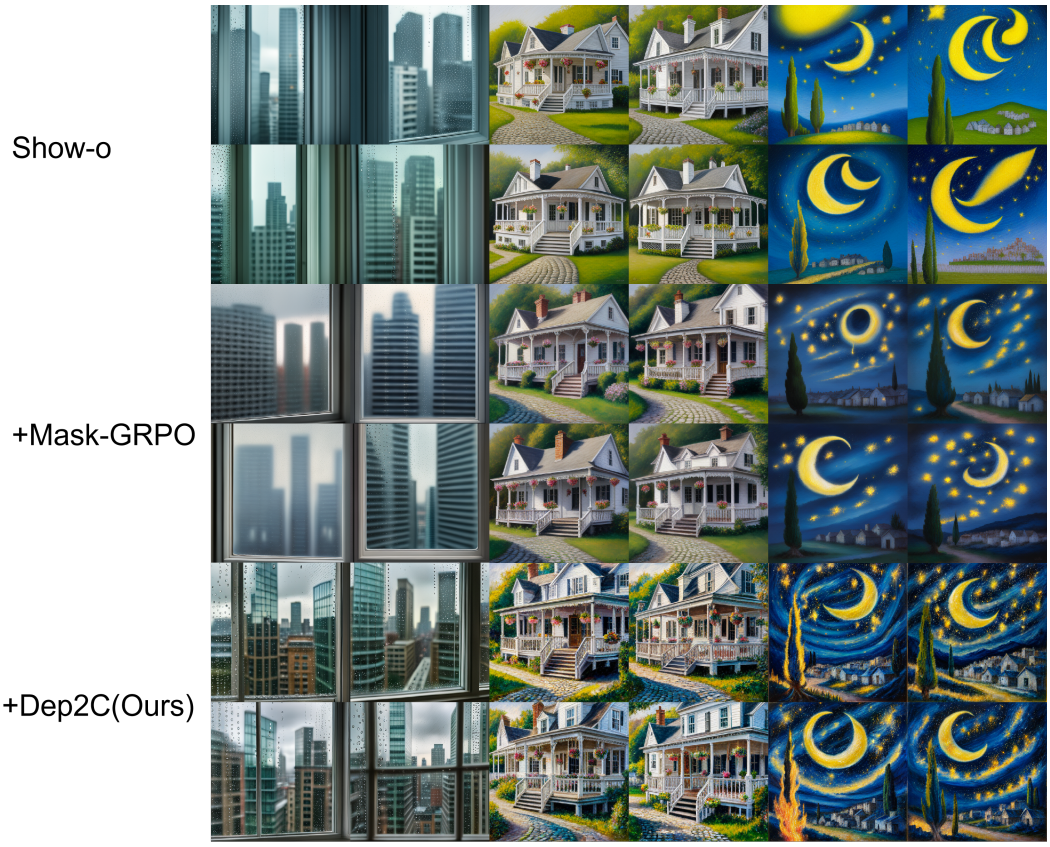

Figure 8: **Qualitative comparison on Show-o with CLIP reward.** Compared with the pretrained Show-o and Mask-GRPO, Dep2C better captures complex prompt semantics while producing more coherent, detailed, and visually appealing generations.

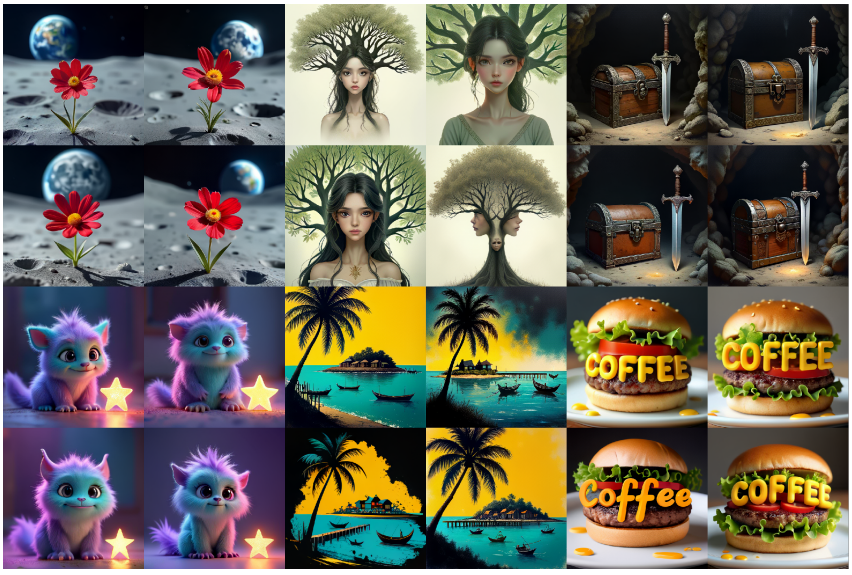

Figure 9: **Additional qualitative results of Dep2C on Lumina-DiMOO.**