IMPROVING FINE-GRAINED CONTROL VIA AGGREGATION OF MULTIPLE DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

While many diffusion models perform well when controlling particular aspects such as style, attributes, and interaction, they struggle with fine-grained control due to dataset limitations and intricate architectural design. This paper introduces a novel training-free algorithm for fine-grained generation, called Aggregation of Multiple Diffusion Models (AMDM). The algorithm aggregates intermediate latent states from multiple compatible diffusion models within the same ecosystem into a target model, providing a training-free mechanism to transfer complementary control signals for fine-grained generation. Experimental results show that AMDM can improve fine-grained control under compatible model settings, e.g., raising instance-level attribute success rate from 34.06% to 54.78% on COCO-MIG and HOI detection score from 16.87 to 26.04 on FGAHOI, while preserving generation quality. AMDM provides a practical way to reuse specialized conditional diffusion models for fine-grained generation, reducing the need for complex multi-condition datasets, architectures, and additional training.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021a;b; Karras et al., 2022) have become a dominant paradigm for generative modeling. Conditional diffusion models (Rombach et al., 2022; Chung et al., 2023; Esser et al., 2024) further enable flexible control in applications such as text-to-image (T2I) (Nichol et al., 2022; Chen et al., 2023b; Lee et al., 2023; Xu et al., 2023) and image-to-image (I2I) generation (Zhang et al., 2023; Mou et al., 2024).

Recent research on conditional diffusion models has focused on achieving fine-grained control, including object attributes (Wu et al., 2023; Wang et al., 2024a), interactions (Hoe et al., 2024; Ge et al., 2024), layouts (Zheng et al., 2023; Chai et al., 2023; Chen et al., 2023a), and style (Wang et al., 2023; Huang et al., 2025; Qi et al., 2024). However, maintaining consistency across diverse, fine-grained controls remains a significant challenge. Generating multiple objects with overlapping bounding boxes can lead to attribute leakage, where the description of one object inappropriately influences others, causing inconsistencies between objects and the background. Fine-grained interaction details may be physically implausible, and style integration may compromise object attributes.

Existing generation approaches only partially address these issues due to the inherent complexity and diversity of fine-grained control, coupled with limitations in datasets and model architectures. Some methods (Li et al., 2023b; Zhou et al., 2024; Wang et al., 2024b) effectively prevent attribute leakage in layout generation but struggle with object interactions; others (Ye et al., 2023; Huang et al., 2025) excel at style transfer yet offer limited layout control.

Many of these methods are built on Stable Diffusion (Rombach et al., 2022), which follows the DDPM theoretical framework (Ho et al., 2020) and commonly uses classifier-free guidance (Ho & Salimans, 2022) for conditional control. In this context, a recent line of research leverages the shared diffusion equation to combine multiple models at inference time, integrating their respective information. The predominant strategy is to apply linear weighting in the score space (Garipov et al., 2023; Bradley et al., 2025; Thornton et al., 2025; Skreta et al., 2025a; He et al., 2026; Skreta et al., 2025b). However, these methods were originally designed to synthesize new data distributions under mutually independent (orthogonal) conditioning variables (Bradley et al., 2025). Although fine-grained generation can be pursued by enforcing identical conditions, this practice violates the

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.

orthogonality assumption underlying these methods, leading to suboptimal generation quality. To mitigate the distributional shift caused by linear weighting, several works have proposed more sophisticated weighting schemes, which, however, introduce additional inference overhead (Skreta et al., 2025b; He et al., 2026). Another line of methods performs linear weighting in the parameter space of the networks (Biggs et al., 2024; Oh et al., 2025; Wang et al., 2025), but this requires the combined models to share exactly the same architecture, thereby substantially limiting their applicability.

This paper proposes a training-free AMDM algorithm that operates directly in the latent data space from a geometric perspective, using only closed-form operations. As a result, it offers an efficient and viable solution for fine-grained control tasks within compatible diffusion ecosystems that are highly coupled and non-orthogonal in nature. Extensive experiments demonstrate clear performance gains under compatible model settings, particularly in regions where previous models exhibited limited controllability.

2 METHOD

2.1 PRELIMINARIES: STABLE DIFFUSION

Stable Diffusion (Rombach et al., 2022) is a class of diffusion models defined in the latent space. The original data $\mathbf{x}_0$ is mapped through the encoder (VAE (Kingma & Welling, 2013)) to obtain the latent variable $\mathbf{z}_0 = E(\mathbf{x}_0)$, which evolves according to the DDPM (Ho et al., 2020) diffusion paradigm:

$$p(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{z}_{t-1}, (1-\alpha_t)\boldsymbol{I}), \tag{1}$$

where $\mathbf{z}_t$ represents the noisy latent data at time step $t \in [0, T]$, α_t is the noise schedule. From (1), the forward marginal distribution is:

$$p(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1-\bar{\alpha}_t)\boldsymbol{I}), \tag{2}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Assuming the denoising neural network is $\boldsymbol{\epsilon}_\theta$ and the condition is y, the reverse sampling process is as follows:

$$\begin{aligned} p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t) =& \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{z}_t, t, y), \sigma_t^2 \boldsymbol{I}) \\ =& \mathcal{N}\left(\sqrt{\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}}\mathbf{z}_t + \left(\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2} - \sqrt{\frac{\bar{\alpha}_{t-1}(1-\bar{\alpha}_t)}{\bar{\alpha}_t}}\right)\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, y), \sigma_t^2 \boldsymbol{I}\right), \end{aligned} \tag{3}$$

where σ_t is a free variable.

2.2 ANALYSIS

Fine-grained conditional generation requires multiple tightly coupled controls, including layout, attributes, interactions, and style, to be satisfied in the same scene. Existing specialized models usually improve only one or a few aspects, with limited capability in other aspects. Existing composition methods offer possible alternatives, but they also have limitations: score-space composition is better suited to relatively independent or orthogonal conditions, while parameter or feature space composition often requires identical or highly compatible model architectures. These assumptions are restrictive for fine-grained generation, where different controls describe the same scene and heterogeneous models may use different control modules. Therefore, we ask whether such aggregation is possible, and proceed to analyze under what conditions intermediate latent variables from different diffusion models can be effectively aggregated.

When are aggregation operations well-defined across different diffusion models? For a latent diffusion model p_θ, the latent data manifold along the diffusion process is $\mathcal{M}_t^\theta(\xi_t) = \{\mathbf{z}_t \in \mathbb{R}^n \mid p_\theta(\mathbf{z}_t) \geq \xi_t\}$, where an appropriate choice of ξ_t ensures that $\mathcal{M}_t^\theta$ is a set of meaningful data. We define the generation domain at time t under condition y as $D_{t,y}^\theta(\tau_{t,y}^\theta) = \{\mathbf{z}_t \in \mathbb{R}^n \mid p_\theta(\mathbf{z}_t \mid y) \geq \tau_{t,y}^\theta\}$. We assume that a threshold $\tau_{t,y}^\theta$ can be chosen such that $D_{t,y}^\theta(\tau_{t,y}^\theta)$ is nonempty and $D_{t,y}^\theta(\tau_{t,y}^\theta) \subset \mathcal{M}_t^\theta(\xi_t)$. Given this choice of $\tau_{t,y}^\theta$, we omit the explicit dependence on the threshold for notational simplicity. Mathematically, the most fundamental requirement for performing aggregation

operations is that all elements must reside in a shared space. However, in the case of aggregation between two models, it is evident that data obtained through conditional sampling from p_{θ_1} should lie within $D^{\theta_1}_{0,y_1}$, whereas data obtained from p_{θ_2} should lie within $D^{\theta_2}_{0,y_2}$. Since $D^{\theta_1}_{0,y_1} \subset \mathcal{M}^{\theta_1}_0$ and $D^{\theta_2}_{0,y_2} \subset \mathcal{M}^{\theta_2}_0$, latent-space aggregation at $t = 0$ requires these manifolds to coincide, i.e., $\mathcal{M}^{\theta_1}_0 = \mathcal{M}^{\theta_2}_0 = \mathcal{M}_0$. This observation directly leads to the **first requirement**: the initial latent manifold must be shared across models, implying that the latent space encoders are consistent, and the aggregation operations at $t = 0$ are well-defined. For $t > 0$, this naturally leads to the **second requirement**: $\mathcal{M}_t$ must coincide across models. In the context of diffusion models, a sufficient condition for this requirement is that the underlying SDEs and noise schedules are identical. Consequently, we obtain $D^{\theta_1}_{t,y_1}, D^{\theta_2}_{t,y_2} \subset \mathcal{M}_t$, which makes aggregation operations well-defined under aligned time steps. This reasoning shows that these requirements are not specific to fine-grained generation, but are shared by general diffusion composition methods.

Which models are suitable for fine-grained aggregation? The ideal goal of aggregation is to obtain a result close to the intersection $D^{\theta_1}_{t,y_1} \cap D^{\theta_2}_{t,y_2}$, since in this region information from both models can be more faithfully incorporated for fine-grained generation. For aggregation to be effective at time t, it is therefore desirable that the domains $D^{\theta_1}_{t,y_1}$ and $D^{\theta_2}_{t,y_2}$ of different models possess a non-empty or non-negligible intersection. We observe that application models developed on top of SD, while satisfying the aforementioned two theoretical requirements, usually fall into the following design patterns in their denoising networks: **1. Additive Architectures:** Typically freeze most or all of the weights of the original base model and then integrate a new, lightweight, and trainable network module in parallel. **2. Modified Architectures:** Replace or alter certain layers within the U-Net while retaining its overall structure. **3. Preserved Architectures:** Fine-tune the model with additional samples from specific domains while keeping the overall architecture intact. Although these approaches adapt the denoising network differently, they preserve much of the base model's generative capability and mainly enhance specific control aspects. We refer to models satisfying these architectural properties together with the two requirements above as belonging to the same **diffusion model ecosystem**. Crucially, the diffusion model ecosystem provides a practical compatibility criterion for latent state aggregation rather than imposing architectural identity: models in the same ecosystem may use different control modules, as long as they share a latent representation and diffusion process that make their intermediate states comparable. For fine-grained generation, the model-specific conditioning signals y_1 and y_2 share the same global prompt, while their additional conditions may take different forms, such as layout, attributes, interactions, or reference images. Since these additional conditions describe semantically aligned aspects of the same target scene and models within the same diffusion ecosystem preserve similar base generative behavior, we assume a *functional proximity*: $\|\boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t, t, y_1) - \boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t, t, y_2)\| \leq L_t$, where L_t captures both architectural differences and the discrepancy between model-specific conditioning signals. This assumption is not expected to hold for models from different ecosystems or for conflicting conditioning signals. Under this assumption, the conditional domains are likely to have non-negligible overlap: $D^{\theta_1}_{0,y_1} \cap D^{\theta_2}_{0,y_2} \neq \varnothing$. Since the models also share the same forward diffusion process, such overlap is expected to be approximately preserved at noised states, making it plausible that $D^{\theta_1}_{t,y_1} \cap D^{\theta_2}_{t,y_2} \neq \varnothing$. Therefore, models within the same diffusion ecosystem are suitable candidates for fine-grained aggregation.

The above analysis suggests that aggregation operations are well-defined for models within the same diffusion model ecosystem, where the models share a common latent representation and diffusion process, and their conditional high-density regions may have non-negligible overlap for aligned fine-grained tasks. This provides the compatibility conditions and geometric intuition underlying the design of AMDM.

2.3 ALGORITHM: AMDM

Spherical Aggregation. To identify a feasible aggregation algorithm, we need to examine the properties of $\mathcal{M}_t$. At the initial stage $t = T$, since $p(\mathbf{z}_T)$ follows a standard Gaussian prior, it follows from the spherical concentration property of high-dimensional Gaussian distributions that $\mathcal{M}_T$ is concentrated on an $(n-1)$-dimensional manifold, namely an n-dimensional hypersphere. The proof is provided in Appendix A. For $t < T$, Chung et al. (2022) demonstrated that $\mathcal{M}_t$ is concentrated on an $(n-1)$-dimensional manifold, which is approximately an n-dimensional hypersphere when t is large. Furthermore, we have the following proposition:

Proposition 2.1. *Let $\mathbf{z}'_t$ denote the aggregated variable at time step t. For the sampling step from t to $t-1$, two diffusion models p_{θ_1} and p_{θ_2} sample $\mathbf{z}^{\theta_1}_{t-1}$ and $\mathbf{z}^{\theta_2}_{t-1}$, respectively, according to* (3)*. Then, with probability at least $1-\gamma$, the absolute difference in norms and the angle is bounded by:*

$$\left| \|\mathbf{z}^{\theta_1}_{t-1}\| - \|\mathbf{z}^{\theta_2}_{t-1}\| \right| \leq L_t + 2\sigma_t \sqrt{2 \ln \frac{8}{\gamma}}, \tag{4}$$

$$\cos\varphi \geq 1 - \frac{\delta^2}{\|\mathbf{z}^{\theta_1}_{t-1}\|^2 + \|\mathbf{z}^{\theta_2}_{t-1}\|^2}, \quad \delta \leq L_t + \sigma_t \sqrt{2\left(n + 2\sqrt{n \ln \frac{2}{\gamma}} + 2 \ln \frac{2}{\gamma}\right)}. \tag{5}$$

where φ is the angle between $\mathbf{z}^{\theta_1}_{t-1}$ and $\mathbf{z}^{\theta_2}_{t-1}$. The angle bound applies only when $\cos\varphi \geq 0$.

The proof is provided in Appendix B. Proposition 2.1 suggests that, under the same diffusion ecosystem and the functional proximity assumption, the sampled variables tend to have comparable norms and, when $\cos\varphi \geq 0$, bounded angular deviations, supporting a local spherical approximation of the latent manifold. Based on this approximation and the global concentration of the Gaussian prior at time T, we adopt spherical interpolation for aggregation, aiming to keep the aggregated state close to the underlying latent manifold:

$$\mathbf{z}'_{t-1} = Slerp(\mathbf{z}^{\theta_1}_{t-1}, \mathbf{z}^{\theta_2}_{t-1}, w) = \frac{\sin((1-w)\varphi)}{\sin(\varphi)} \mathbf{z}^{\theta_1}_{t-1} + \frac{\sin(w\varphi)}{\sin(\varphi)} \mathbf{z}^{\theta_2}_{t-1}, \tag{6}$$

where $w \in [0,1]$ is the weighting factor that balances the contribution of each model. Spherical aggregation is intended to combine the conditional tendencies of p_{θ_1} and p_{θ_2}, while keeping the new variables close to the locally approximated manifold.

Deviation Optimization. Ideally, we would like the aggregated state to remain close to the compatible intersection region $D^{\theta_1}_{t-1,y_1} \bigcap D^{\theta_2}_{t-1,y_2}$, but deviations typically arise in practice. Considering the step from t to $t-1$, since $p_{\theta_1}(\mathbf{z}^{\theta_1}_{t-1} \mid \mathbf{z}^{\theta_1}_t, y_1)$ is high-dimensional Gaussian, its samples concentrate in a thin spherical shell around the mean. Based on this observation, we perform a radial adjustment of $\mathbf{z}'_{t-1}$ toward that shell, encouraging the corrected state $\tilde{\mathbf{z}}_{t-1}$ to move closer to $D^{\theta_1}_{t-1,y_1}$. Although this adjustment alone does not guarantee $\tilde{\mathbf{z}}_{t-1} \in D^{\theta_2}_{t-1,y_2}$, we provide a local probabilistic analysis suggesting that, when the deviation induced by aggregation is small, $\tilde{\mathbf{z}}_{t-1}$ can remain close to the compatible high-density region, leading to the following proposition:

Proposition 2.2. *For the diffusion model p_{θ_1} defined by (3) with $\sigma_t > 0$, and any new intermediate variable $\mathbf{z}'_{t-1}$ from (6), let $r = \left\| \mathbf{z}'_{t-1} - \boldsymbol{\mu}_{\theta_1}(\mathbf{z}^{\theta_1}_t, t, y_1) \right\| > 0$ and define:*

$$\tilde{\mathbf{z}}_{t-1} = \mathbf{z}'_{t-1} - \eta^{\theta_1}_{t-1} \frac{r - \sqrt{n}\sigma_t}{|r - \sqrt{n}\sigma_t|} \frac{\mathbf{z}'_{t-1} - \boldsymbol{\mu}_{\theta_1}(\mathbf{z}^{\theta_1}_t, t, y_1)}{r}, \tag{7}$$

where $0 \leq \eta^{\theta_1}_{t-1} \leq |r - \sqrt{n}\sigma_t|$ is a small correction step size. When $r = \sqrt{n}\sigma_t$, no correction is needed and we set $\tilde{\mathbf{z}}_{t-1} = \mathbf{z}'_{t-1}$. Under the local shell-concentration approximation, there exists $\eta^{\theta_1}_{t-1}$ such that the update in (7) *moves $\mathbf{z}'_{t-1}$ toward $D^{\theta_1}_{t-1,y_1}$. Moreover, an approximate probabilistic interpretation for its compatibility with $D^{\theta_2}_{t-1,y_2}$ is given by:*

$$P\left(\tilde{\mathbf{z}}_{t-1} \in D^{\theta_2}_{t-1,y_2}\right) \geq 1 - 2\exp\left(- \frac{n\left(\epsilon^{\theta_2}_{t-1} - \frac{d}{\sigma_t\sqrt{n}}\right)^2}{4\left(1 + \epsilon^{\theta_2}_{t-1} - \frac{d}{\sigma_t\sqrt{n}}\right)} \right), \tag{8}$$

where d is a deterministic upper bound on $\phi_w(\varphi)\delta + \eta^{\theta_1}_{t-1}$ with $d < \epsilon^{\theta_2}_{t-1}\sigma_t\sqrt{n}$, and $\phi_w(\varphi) = \sin\big((1-w)\varphi/2\big)/\sin(\varphi/2)$.

The proof is given in Appendix C; a detailed analysis of d is presented in Appendix E.3; and the geometry of deviation optimization is illustrated in Figure 1 (Right). Proposition 2.2 provides a local geometric interpretation of deviation optimization. Rather than guaranteeing membership in the conditional domains, it shows that radial correction moves the aggregated state toward the typical reverse-kernel shell of the target model. When the deviation between compatible models is small, this correction is unlikely to disrupt auxiliary control signals.

Remark 2.3. In the deterministic limit $\sigma_t \to 0$, the deviation optimization (7) admits an interpretation as a trajectory-alignment step that pulls the aggregated variable toward the deterministic denoising path. As $\sigma_t \to 0$, the stochastic reverse transition degenerates into a deterministic update centered at the ODE/DDIM trajectory. Therefore, the radial correction in (7) reduces to correcting the deviation from this deterministic denoising path.

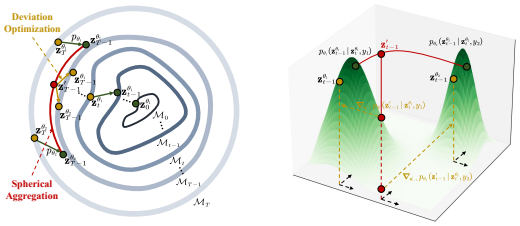


Figure 1: Geometry of AMDM (left) and deviation optimization (right). AMDM incorporates conditional information in early steps before direct target-model sampling.

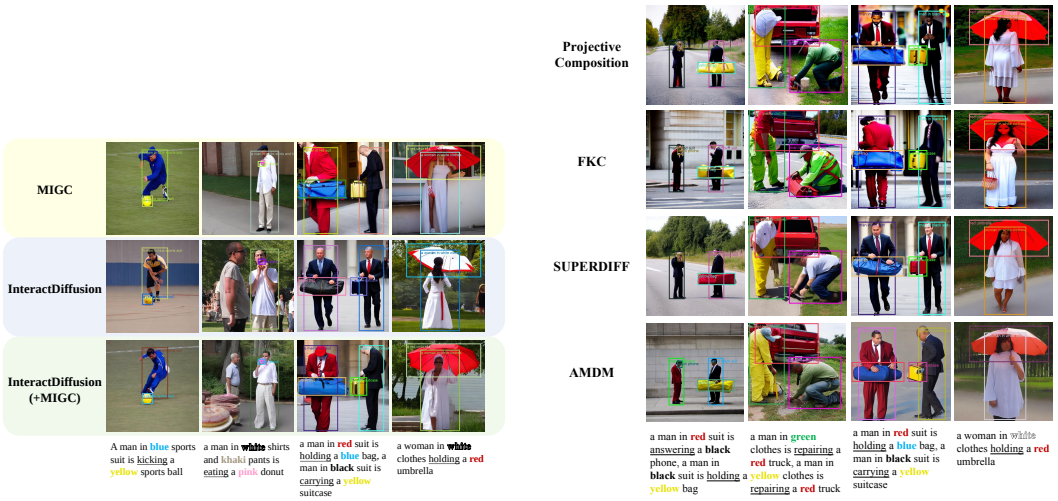
Motivated by the local geometric insights from Propositions 2.1 and 2.2, we combine spherical aggregation in (6) and deviation optimization in (7) to construct AMDM. The algorithm consists of two key components: spherical aggregation aims to keep the new intermediate variables close to the locally approximated manifold, while deviation optimization encourages the aggregated state to stay close to the corresponding high-density region. Following prior observations on denoising stages (Balaji et al., 2022; Hertz et al., 2023; Choi et al., 2022), AMDM performs these two operations only during the first s sampling steps, and then continues direct sampling from the target model p_{θ_1}. This procedure can be regarded as aggregating the features of other models into p_{θ_1}, balancing inference cost against the strength of conditional integration. The algorithm can be readily extended to multiple models. For $N > 2$, we adopt an anchor-centered pairwise strategy: the target model p_{θ_1} acts as the anchor; each auxiliary latent state is first aggregated with the target latent state, and the resulting anchor-aligned states are then recursively merged. This design helps preserve the target model's generative prior and reduces order sensitivity among auxiliary models. The full pseudocode is provided in Algorithm 1 in Appendix F, and the geometry is illustrated in Figure 1 (Left). Moreover, since $\mu_{\theta_i}(\mathbf{z}_t^{\theta_i}, t, y_i)$ can be computed from the already obtained $\epsilon_{\theta_i}(\mathbf{z}_t^{\theta_i}, t, y_i)$, deviation optimization requires only closed-form vector operations and introduces no additional network evaluation.

As for the selection of p_{θ_1}, we naturally select the model with stronger generative capability as p_{θ_1} to ensure high-quality generation in later stages. This choice can be evaluated using a variety of criteria, such as quantitative metrics, model size, general applicability, or direct experimental validation.

Proposition 2.1 supports the local spherical approximation underlying spherical aggregation under the same diffusion ecosystem and the functional proximity assumption. We further show in Proposition D.1 and Appendix D that this geometric property is approximately preserved after deviation optimization under the same local assumptions. These results should be interpreted as local geometric explanations rather than global guarantees. Empirical evidence for the functional proximity assumption and the corresponding geometric trends is provided in Appendices E.2 and E.3.

3 Experiments

In this section, we evaluate AMDM across complementary conditional diffusion models within compatible diffusion ecosystems. Our experiments are designed to assess whether AMDM can aggregate auxiliary control capabilities into a target model, thereby improving its weak control aspects while preserving its original strengths and generation quality. We first study bidirectional aggregation between InteractDiffusion and MIGC to evaluate interaction control and instance-level attribute control, and then extend the evaluation to style transfer with IP-Adapter and multi-model aggregation. We further conduct ablations on spherical aggregation, deviation optimization, aggregation stages, and aggregation weights. To connect the empirical results with our geometric analysis, we provide compatibility measurements based on the norm difference, angular deviation φ, δ, and the maximum-distance term d. These measurements empirically support the functional proximity assumption within the same diffusion ecosystem and are consistent with the local geometric trends underlying AMDM. Appendix G provides baseline adaptation details, fairness controls, runtime comparisons, and supplementary analyses of compositional methods. Appendix H reports evaluation metrics and settings. Additional experiments examine few-step sampling, additional baselines, robustness across seeds, and personalized model composition (Appendices H.7 to H.10). Appendices E.2 and E.3 provide detailed geometric compatibility analyses of the norm difference, angle φ, δ, and d. The



(a) MIGC into InteractDiffusion.

(b) Visual results of related composition baselines.

Figure 2: Visual comparisons of MIGC, InteractDiffusion and related composition baselines.

experiments in this section use a single RTX 3090 GPU; additional hardware settings are reported in Appendix H.

3.1 Aggregation Experiments

InteractDiffusion and MIGC. InteractDiffusion (Hoe et al., 2024) is a T2I model with a locally controlled interaction mechanism, enabling fine-grained human-object interaction control. MIGC (Zhou et al., 2024) adopts a divide-and-conquer strategy for multi-instance generation, showing strong attribute representation and instance isolation.

Table 1: Quantitative results on the HOI Detection Score, FID and CLIP Score across different models.

Method	Default ↑		Known Object ↑		FID ↓	CLIP Score ↑
	Full	Rare	Full	Rare		
MIGC	16.87	18.05	17.84	19.02	30.53	27.36
Projective Composition	22.23	20.14	23.46	21.38	24.68	27.42
Feynman–Kac Correctors	21.74	19.62	22.91	20.83	25.31	27.39
SUPERDIFF	23.15	20.58	24.36	21.71	24.12	27.46
AMDM: MIGC(+InteractDiffusion)	**26.04**	**21.73**	**27.02**	**22.89**	**22.32**	**27.55**
InteractDiffusion	29.53	23.02	30.99	24.93	18.69	26.91
AMDM: InteractDiffusion(+MIGC)	31.40	24.52	32.76	26.32	18.35	27.18

We evaluate AMDM in both directions: aggregating MIGC p_{θ_2} into InteractDiffusion p_{θ_1}, denoted as InteractDiffusion(+MIGC), and aggregating InteractDiffusion p_{θ_1} into MIGC p_{θ_2}, denoted as MIGC(+InteractDiffusion). We further compare AMDM with representative compositional diffusion methods, including projective composition (Bradley et al., 2025), Feynman–Kac Correctors (Skreta et al., 2025a), and SUPERDIFF (Skreta et al., 2025b). These baselines mainly combine model information in the score or density-estimation space, whereas AMDM operates directly in the latent data space. All methods are evaluated under the same MIGC–InteractDiffusion setting. For quantitative evaluation, we use COCO-MIG (Zhou et al., 2024) to measure instance-level attribute control and FGAHOI (Ma et al., 2024) to measure human-object interaction control. Details on baseline adaptation, hyperparameter selection, seed control, and runtime comparison are provided in Appendix G.

The quantitative results of InteractDiffusion(+MIGC) on COCO-MIG are shown in Table 2. InteractDiffusion(+MIGC) improves all metrics related to attribute control compared with InteractDiffusion, and also outperforms the representative compositional diffusion methods. Notably, it even surpasses the original InteractDiffusion in interaction control, as shown in Table 1. Similarly, Table 1 shows that MIGC(+InteractDiffusion) achieves improvements in interaction capability compared with MIGC, while Table 2 shows that it incurs moderate trade-offs in attribute control. These results support the effectiveness of AMDM in aggregating complementary control capabilities under compatible model settings.

Figure 2 qualitatively compares AMDM with source models and composition baselines, showing better attribute preservation while maintaining human–object interactions. Additional qualitative examples and implementation details are provided in Appendix G.



(a) IP-Adapter into InteractDiffusion.



(b) Visual results of aggregating MIGC and IP-Adapter into InteractDiffusion.

Figure 3: Visual results of applying AMDM.

Table 2: Quantitative results on the COCO-MIG benchmark, CLIP Score, and inference time across different models.

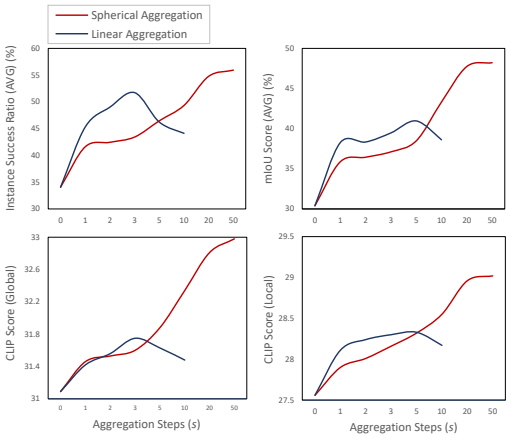
Method	Instance Success Rate (%) ↑						mIoU Score (%) ↑						CLIP Score ↑		Time (s) ↓
	L_2	L_3	L_4	L_5	L_6	Avg	L_2	L_3	L_4	L_5	L_6	Avg	Global	Local	
InteractDiffusion	37.50	35.62	35.31	30.62	34.16	34.06	32.98	31.63	30.82	28.29	30.40	30.40	31.09	27.56	11.35 ± 0.13
Projective Composition	54.38	49.72	48.31	45.69	44.21	47.82	47.36	43.58	42.64	41.02	39.76	42.71	31.76	28.18	17.33 ± 0.21
Feynman–Kac Correctors	52.87	48.21	46.63	43.92	42.76	46.35	45.92	42.38	41.17	39.43	38.21	41.25	31.57	28.03	65.96 ± 0.78
SUPERDIFF	53.16	48.64	46.92	44.15	42.94	46.56	46.28	42.81	41.73	39.88	38.64	41.81	31.65	28.09	33.03 ± 0.60
AMDM: InteractDiffusion(+MIGC)	**62.29**	**54.33**	**56.31**	**53.87**	**52.89**	**54.78**	**54.30**	**46.37**	**48.44**	**47.65**	**46.64**	**47.74**	**32.81**	**28.96**	13.52 ± 0.06
MIGC	64.06	56.04	58.43	56.00	49.89	55.46	54.43	49.33	50.48	48.67	44.74	48.53	32.78	28.61	6.27 ± 0.04
AMDM: MIGC(+InteractDiffusion)	60.00	50.20	50.46	48.25	46.97	49.78	52.58	44.27	43.39	42.67	41.50	43.69	32.41	28.55	10.83 ± 0.04

InteractDiffusion and IP-Adapter. IP-Adapter (Ye et al., 2023) is a lightweight I2I model that employs a decoupled cross-attention mechanism to separately process text and image features, enabling multimodal image generation. Due to its strong ability to preserve the style of the reference image, we integrate the style information from IP-Adapter p_{θ_3} into InteractDiffusion p_{θ_1}, denoted as InteractDiffusion(+IP-Adapter). The qualitative results are shown in Figure 3a. Quantitatively, the style CLIP score increases from 0.533 to 0.581, indicating that AMDM effectively transfers style controllability from IP-Adapter to InteractDiffusion. Appendix H.3 provides additional generalization studies, including GLIGEN, SDXL-based models, and cross-ecosystem aggregation, further supporting the applicability of AMDM within compatible ecosystems and the necessity of the shared-ecosystem assumption.

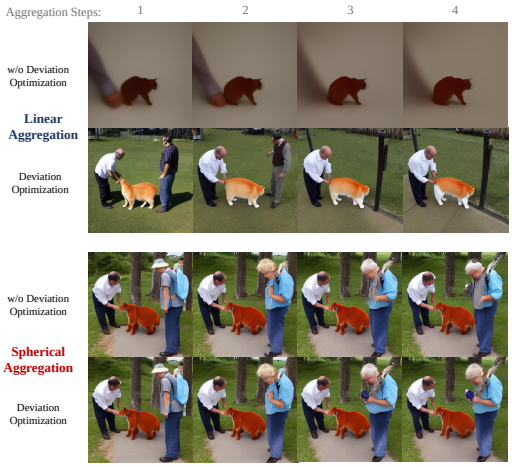
Table 3: Style CLIP for InteractDiffusion(+IP-Adapter).

Method	Style CLIP ↑
InteractDiffusion	0.533
AMDM: InteractDiffusion(+IP-Adapter)	0.581
IP-Adapter	0.588

InteractDiffusion, MIGC and IP-Adapter. Furthermore, we attempt to aggregate the attribute features from MIGC p_{θ_2} and the style features from IP-Adapter p_{θ_3} into InteractDiffusion p_{θ_1} to evaluate the effectiveness of AMDM. The experimental results are presented in Figure 3b. Since no single benchmark jointly evaluates instance-level attribute binding, human-object interaction, and style-related alignment, we provide separate evaluations for the multi-model setting in Appendix H.5. We report MIG-Bench for attribute grounding, HOI metrics for interaction preservation, and Style CLIP for style-related alignment. The additional results show that InteractDiffusion(+MIGC,+IP-Adapter) improves attribute grounding and style-related alignment over InteractDiffusion while largely preserving its interaction ability. The anchor-choice analysis further shows predictable



(a) Metrics of spherical aggregation and linear aggregation under different aggregation steps.



(b) Visual results of deviation optimization for spherical aggregation and linear aggregation under different aggregation steps.

Figure 4: Ablation study of aggregation strategy.

trade-offs: InteractDiffusion and MIGC are stronger anchors for interaction and attribute control, respectively, while IP-Adapter is less suitable as an anchor under the no-reference protocol. Since IP-Adapter behaves closer to a weakly conditioned SD-like branch in this setting, its effective use as an auxiliary branch also provides an additional same-ecosystem generalization case.

We further evaluate AMDM with control methods built on FLUX, extending the study to flow matching with a DiT architecture (Appendix H.6). Aggregating CreatiLayout (Zhang et al., 2025) into OneHOI (Hoe et al., 2026) raises Instance Success Rate from 35.37% to 59.76% on COCO-MIG, while aggregating OneHOI into Regional Prompting (Chen et al., 2024a) raises the Default HOI Detection Score from 4.57% to 22.91% on HOI Bench.

3.2 Ablation Study

We ablate AMDM from three aspects: the aggregation operator, the deviation optimization and weighting hyperparameters, and the sampling stages where aggregation is applied.

We first compare spherical aggregation with linear aggregation in InteractDiffusion(+MIGC). As shown in Figure 4a, linear aggregation is competitive when only a few aggregation steps are used ($s < 5$), but its performance degrades as the number of aggregation steps increases. In contrast, spherical aggregation remains more stable and achieves better results with more aggregation steps. This trend is also reflected in Figure 4b, where linear aggregation causes stronger visual degradation, while spherical aggregation better preserves image quality. These results support the use of spherical aggregation under the local manifold approximation. The corresponding norm, angle, δ, and d trends are analyzed in Appendices E.2 and E.3, and additional comparisons with compositional baselines are discussed in Appendix G.

We next ablate deviation optimization and aggregation weights. Table 4 shows that a moderate absolute deviation optimization step, $\eta = 0.3$, achieves the best overall performance, while overly small or large steps are less effective. We also analyze the aggregation weight w in Appendix H.4, where AMDM remains stable around the balanced region and w serves as a controllable trade-off between different control capabilities.

Finally, we study when aggregation should be applied. As shown in Figure 5a and Figure 5b, the early window achieves the best control metrics and FID, whereas applying the same window in the middle or later stages noticeably degrades both control accuracy and image quality. This further verifies in the model aggregation setting that early stages are more suitable for incorporating coarse control signals such as position, attributes, and style, while later stages predominantly govern quality and consistency. Additional stage-wise visual results are provided in Appendix I.

Table 4: Ablation results of different absolute deviation optimization steps on COCO-MIG benchmark in InteractDiffusion(+MIGC).

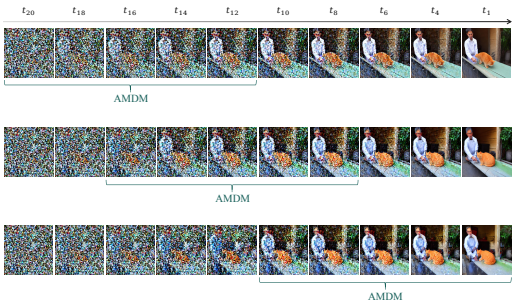
Abs. Deviation Optimization Step	Instance Success Rate (%) ↑						mIoU Score (%) ↑						CLIP Score ↑	
	L_2	L_3	L_4	L_5	L_6	Avg	L_2	L_3	L_4	L_5	L_6	Avg	Global	Local
0	59.24	54.12	54.01	51.68	52.16	53.36	52.96	44.32	45.41	44.12	44.39	46.24	31.36	27.82
0.1	57.60	53.26	53.89	52.40	52.08	53.26	53.27	45.53	46.86	46.31	45.80	46.83	31.43	27.87
0.2	59.62	54.27	54.06	51.25	52.29	53.40	52.71	46.32	47.17	44.68	45.74	46.59	31.92	28.45
0.3	**62.29**	**54.33**	**56.31**	**53.87**	**52.89**	**54.78**	**54.30**	**46.37**	**48.44**	**47.65**	**46.64**	**47.74**	**32.81**	**28.96**
0.4	61.34	53.62	55.32	52.22	52.50	53.96	53.46	46.21	46.82	45.26	46.23	46.78	32.23	28.55
0.5	58.30	53.43	55.14	51.76	51.76	53.27	52.88	45.03	46.52	44.95	45.51	46.14	31.15	27.78

4 Related Work

Fine-Grained Generation. Beyond text-driven models (Nichol et al., 2022; Ramesh et al., 2022; Li et al., 2024; Podell et al., 2024), research is increasingly moving toward finer-grained conditional control. A classical route is personalization-controlled generation, which covers style (Sohn et al., 2023; Hertz et al., 2024; Chen et al., 2024b), subject (Li et al., 2023a; Shi et al., 2024; Jiang et al., 2024; Ye et al., 2023), person (Xiao et al., 2025; Lisanti & Giambi, 2024; Peng et al., 2024), and interactive generation (Huang et al., 2024; Guo et al., 2024; Hoe et al., 2024). Beyond personalization, another line of work focuses on spatial control (Li et al., 2023b; Cheng et al., 2023; Nie et al., 2024; Zhou et al., 2024), which uses bounding boxes or regions as additional conditions to enforce spatial constraints. More recently, several studies have pursued direct fine-grained multimodal control (Huang et al., 2023; Smith et al., 2024; Gu et al., 2023; Kumari et al., 2023), but these approaches typically require extensive multi-condition datasets and increasingly complex model architectures.

(a) InteractDiffusion(+MIGC) with total 20 steps sampling and s=10.

Aggregation Stage	ISR (avg) ↑	Default (Full) ↑	FID ↓
$t_{20} \rightarrow t_{10}$	**52.27**	**30.05**	**19.86**
$t_{15} \rightarrow t_5$	50.14	27.53	22.96
$t_{10} \rightarrow t_0$	47.82	24.17	41.13



(b) Visual Results of Figure 5a.

Figure 5: Ablation on aggregation stages.

Compositional Generation. Some recent works (Du et al., 2020; Liu et al., 2022; Du et al., 2023; Du & Kaelbling, 2024; Garipov et al., 2023; Bradley et al., 2025; Thornton et al., 2025; Skreta et al., 2025a; He et al., 2026; Skreta et al., 2025b) aim to create new distributions by combining different distributions, commonly relying on linear weighting in the score space, optionally combined with simulated annealing or other optimization techniques. Other works (Biggs et al., 2024; Oh et al., 2025; Wang et al., 2025) attempt to combine models by applying weighted interpolation to their parameters. However, this approach requires that the models share an identical architecture, which significantly restricts its applicability. Implementation details and supplementary analysis of these compositional baselines are provided in Appendix G.

5 Conclusion

This paper proposes a novel AMDM algorithm designed for fine-grained generation, which consists of two main components: spherical aggregation and deviation optimization. Experimental results demonstrate the effectiveness of AMDM. AMDM provides a practical way to reuse specialized conditional diffusion models within compatible diffusion ecosystems, requiring comparable latent states rather than identical denoising architectures, and reducing the need for complex multi-condition datasets, architectures, and additional training.

AI USE STATEMENT

We used generative AI tools to assist with drafting and editing parts of the manuscript, reviewing the presentation of theoretical arguments, organizing and describing additional experiments, interpreting reported results, and formatting citations, tables, and figures. The authors reviewed the AI-assisted material and take responsibility for the final content, including its claims and numerical results.

REPRODUCIBILITY STATEMENT

We provide the source code in the supplementary materials. Appendix F describes the AMDM algorithm, and Appendix H provides the evaluation metrics and parameter settings used in the experiments.

REFERENCES

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Benjamin Biggs, Arjun Seshadri, Yang Zou, Achin Jain, Aditya Golatkar, Yusheng Xie, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Diffusion soup: Model merging for text-to-image diffusion models. In *European Conference on Computer Vision*, pp. 257–274. Springer, 2024.

Arwen Bradley, Preetum Nakkiran, David Berthelot, James Thornton, and Joshua M. Susskind. Mechanisms of projective composition of diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 5319–5354. PMLR, 2025. URL https://proceedings.mlr.press/v267/bradley25a.html.

Shang Chai, Liansheng Zhuang, and Fengying Yan. LayoutDM: Transformer-based diffusion model for layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18349–18358, 2023.

Anthony Chen, Jianjin Xu, Wenzhao Zheng, Gaole Dai, Yida Wang, Renrui Zhang, Haofan Wang, and Shanghang Zhang. Training-free regional prompting for diffusion transformers. *arXiv preprint arXiv:2411.02395*, 2024a.

Dar-Yen Chen, Hamish Tennent, and Ching-Wen Hsu. ArtAdapter: Text-to-image style transfer using multi-level style encoder and explicit adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8619–8628, 2024b.

Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. TextDiffuser: Diffusion models as text painters. In *Advances in Neural Information Processing Systems*, volume 36, pp. 9353–9387, 2023a. doi: 10.52202/075280-0410.

Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-Imagen: Retrieval-augmented text-to-image generator. In *The Eleventh International Conference on Learning Representations*, 2023b.

Jiaxin Cheng, Xiao Liang, Xingjian Shi, Tong He, Tianjun Xiao, and Mu Li. LayoutDiffuse: Adapting foundational diffusion models for layout-to-image generation. *arXiv preprint arXiv:2302.08908*, 2023.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11472–11481, 2022.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In *Advances in Neural Information Processing Systems*, volume 35, pp. 25683–25696, 2022. doi: 10.52202/068431-1862.

Hyungjin Chung, Jeongsol Kim, Michael Thompson McCann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023.

Yilun Du and Leslie Pack Kaelbling. Position: Compositional generative modeling: A single model is not all you need. In *Forty-first International Conference on Machine Learning*, 2024.

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In *International conference on machine learning*, pp. 8489–8510. PMLR, 2023.

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.

Timur Garipov, Sebastiaan De Peuter, Ge Yang, Vikas Garg, Samuel Kaski, and Tommi Jaakkola. Compositional sculpting of iterative generative processes. *Advances in neural information processing systems*, 36:12665–12702, 2023.

Mengmeng Ge, Xu Jia, Takashi Isobe, Xiaomin Li, Qinghe Wang, Jing Mu, Dong Zhou, Li Wang, Huchuan Lu, Lu Tian, Ashish Sirasao, and Emad Barsoum. Customizing text-to-image generation with inverted interaction. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 10901–10909, 2024. doi: 10.1145/3664647.3680621.

Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, Yixiao Ge, Ying Shan, and Mike Zheng Shou. Mix-of-Show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15890–15902, 2023. doi: 10.52202/075280-0699.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

Jiajun He, José Miguel Hernández Lobato, Yuanqi Du, and Francisco Vargas. RNE: plug-and-play diffusion inference-time control and energy-based training. In *International Conference on Learning Representations*, pp. 75732–75786, 2026.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *International Conference on Learning Representations*, 2023.

Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4775–4785, 2024.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jiun Tian Hoe, Xudong Jiang, Chee Seng Chan, Yap-Peng Tan, and Weipeng Hu. InteractDiffusion: Interaction control in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6180–6189, 2024.

Jiun Tian Hoe, Weipeng Hu, Xudong Jiang, Yap-Peng Tan, and Chee Seng Chan. OneHOI: Unifying human-object interaction generation and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7664–7673, June 2026.

Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: creative and controllable image synthesis with composable conditions. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 13753–13773, 2023.

Nisha Huang, Yuxin Zhang, Fan Tang, Chongyang Ma, Haibin Huang, Weiming Dong, and Changsheng Xu. DiffStyler: Controllable dual diffusion for text-driven image stylization. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):3370–3383, 2025. doi: 10.1109/TNNLS.2023.3342645.

Ziqi Huang, Tianxing Wu, Yuming Jiang, Kelvin C.K. Chan, and Ziwei Liu. ReVersion: Diffusion-based relation inversion from images. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024. doi: 10.1145/3680528.3687658.

Yuming Jiang, Tianxing Wu, Shuai Yang, Chenyang Si, Dahua Lin, Yu Qiao, Chen Change Loy, and Ziwei Liu. VideoBooth: Diffusion-based video generation with image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6689–6700, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1931–1941, 2023.

Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000. doi: 10.1214/aos/1015957395.

Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Teufel, Marco Bellagente, Minguk Kang, Taesung Park, Jure Leskovec, Jun-Yan Zhu, Fei-Fei Li, Jiajun Wu, Stefano Ermon, and Percy S Liang. Holistic evaluation of text-to-image models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 69981–70011, 2023. doi: 10.52202/075280-3067.

Dongxu Li, Junnan Li, and Steven Hoi. BLIP-Diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In *Advances in Neural Information Processing Systems*, volume 36, pp. 30146–30166, 2023a. doi: 10.52202/075280-1312.

Haoran Li, Haolin Shi, Wenli Zhang, Wenjun Wu, Yong Liao, Lin Wang, Lik-hang Lee, and Peng Yuan Zhou. DreamScene: 3D Gaussian-based Text-to-3D scene generation via formation pattern sampling. In *European Conference on Computer Vision*, pp. 214–230, 2024.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22511–22521, 2023b.

Giuseppe Lisanti and Nico Giambi. Conditioning diffusion models via attributes and semantic masks for face generation. *Computer Vision and Image Understanding*, 244:104026, 2024. doi: 10.1016/j.cviu.2024.104026.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.

Shuailei Ma, Yuefeng Wang, Shanze Wang, and Ying Wei. FGAHOI: Fine-grained anchors for human-object interaction detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46 (4):2415–2429, 2024. doi: 10.1109/TPAMI.2023.3331738.

Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. DiffEditor: Boosting accuracy and flexibility on diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8488–8497, 2024.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pp. 16784–16804. PMLR, 2022.

Weili Nie, Sifei Liu, Morteza Mardani, Chao Liu, Benjamin Eckart, and Arash Vahdat. Compositional text-to-image generation with dense blob representations. In *Forty-first International Conference on Machine Learning*, 2024.

Changdae Oh, Yixuan Li, Kyungwoo Song, Sangdoo Yun, and Dongyoon Han. DaWin: Training-free dynamic weight interpolation for robust adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xu Peng, Junwei Zhu, Boyuan Jiang, Ying Tai, Donghao Luo, Jiangning Zhang, Wei Lin, Taisong Jin, Chengjie Wang, and Rongrong Ji. PortraitBooth: A versatile portrait model for fast identity-preserved personalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27080–27090, 2024.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

Tianhao Qi, Shancheng Fang, Yanze Wu, Hongtao Xie, Jiawei Liu, Lang Chen, Qian He, and Yongdong Zhang. DEADiff: An efficient stylization diffusion model with disentangled representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8693–8702, 2024.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.

Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. InstantBooth: Personalized text-to-image generation without test-time finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8543–8552, 2024.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alan Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-Kac correctors in diffusion: Annealing, guidance, and product of experts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55906–55949. PMLR, 2025a. URL https://proceedings.mlr.press/v267/skreta25a.html.

Marta Skreta, Lazar Atanackovic, Joey Bose, Alexander Tong, and Kirill Neklyudov. The superposition of diffusion models using the Itô density estimator. In *The Thirteenth International Conference on Learning Representations*, 2025b.

James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. Continual diffusion: Continual customization of text-to-image diffusion with C-LoRA. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=TZdEgwZ6f3.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Kihyuk Sohn, Lu Jiang, Jarred Barber, Kimin Lee, Nataniel Ruiz, Dilip Krishnan, Huiwen Chang, Yuanzhen Li, Irfan Essa, Michael Rubinstein, Yuan Hao, Glenn Entis, Irina Blok, and Daniel Castro Chin. StyleDrop: Text-to-image synthesis of any style. In *Advances in Neural Information Processing Systems*, volume 36, pp. 66860–66889, 2023. doi: 10.52202/075280-2920.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

James Thornton, Louis Béthune, Ruixiang Zhang, Arwen Bradley, Preetum Nakkiran, and Shuangfei Zhai. Composition and control with distilled energy diffusion models and sequential Monte Carlo. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 3259–3267. PMLR, 2025. URL https://proceedings.mlr.press/v258/thornton25a.html.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019. doi: 10.1017/9781108627771.

Cong Wang, Kuan Tian, Yonghang Guan, Fei Shen, Zhiwei Jiang, Qing Gu, and Jun Zhang. Ensembling diffusion models via adaptive feature aggregation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ruichen Wang, Zekang Chen, Chen Chen, Jian Ma, Haonan Lu, and Xiaodong Lin. Compositional text-to-image synthesis with attention map control of diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5544–5552, 2024a.

Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. InstanceDiffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6232–6242, 2024b.

Zhizhong Wang, Lei Zhao, and Wei Xing. StyleDiffusion: Controllable disentangled style transfer via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7677–7689, 2023.

Qiucheng Wu, Yujian Liu, Handong Zhao, Ajinkya Kale, Trung Bui, Tong Yu, Zhe Lin, Yang Zhang, and Shiyu Chang. Uncovering the disentanglement capability in text-to-image diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1900–1910, 2023.

Guangxuan Xiao, Tianwei Yin, William T Freeman, Frédo Durand, and Song Han. FastComposer: Tuning-free multi-subject image generation with localized attention. *International Journal of Computer Vision*, 133(3):1175–1194, 2025. doi: 10.1007/s11263-024-02227-z.

Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. BoxDiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7452–7461, 2023.

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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15903–15935, 2023. doi: 10.52202/075280-0700.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Hui Zhang, Dexiang Hong, Yitong Wang, Jie Shao, Xinglong Wu, Zuxuan Wu, and Yu-Gang Jiang. CreatiLayout: Siamese multimodal diffusion transformer for creative layout-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18487–18497, October 2025.

Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10146–10156, 2023.

Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. LayoutDiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22490–22499, 2023.

Dewei Zhou, You Li, Fan Ma, Xiaoting Zhang, and Yi Yang. MIGC: Multi-instance generation controller for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6818–6828, 2024.

TABLE OF CONTENTS FOR APPENDIX

A Concentration of Measure Theorem for High-Dimensional Independent Normal Distribution **17**

B Proof of Proposition 2.1 **17**

C Proof of Proposition 2.2 **20**

D Geometric Properties of AMDM **21**

E Statistical analysis **23**

E.1 Gaussianity in latent space 23

E.2 Numerical Analysis of the Experiments for Proposition 2.1 23

E.3 Maximum distance term 25

F AMDM Algorithm **27**

G Comparison with Compositional Methods **27**

H Experimental Details **29**

H.1 Evaluation Metrics 29

H.2 Parameter Settings 30

H.3 Additional Generalization Studies 30

H.4 Sensitivity to Aggregation Weights 31

H.5 Additional Evaluation of Multi-Model Aggregation 32

H.6 Generalization to FM/RF Models with DiT Architectures 34

H.7 Sampling with Four Steps and One Step 36

H.8 Additional Baselines and Efficiency 38

H.9 Robustness Across Random Seeds 39

H.10 Personalized Model Composition 41

I Visual Results of Figure 5a **42**

J Limitations and Future Work **42**

K Broad Impacts **42**

L Additional Visual Results **42**

A Concentration of Measure Theorem for High-Dimensional Independent Normal Distribution

Lemma A.1. *Let $X = (X_1, X_2, \ldots, X_n)$ be an n-dimensional random vector where each component X_i is independently and identically distributed as $\mathcal{N}(0, \sigma^2)$. Then, as the dimension n increases, the norm $\|X\|$ of X is concentrated around $\sqrt{n}\sigma$. Specifically, for any $0 \leq \epsilon < 1$ and sufficiently large n,*

$$P\left(\left|\|X\| - \sqrt{n}\sigma\right| \leq \epsilon\sqrt{n}\sigma\right) \geq 1 - 2e^{-\frac{n\epsilon^2}{4(1+\epsilon)}}. \tag{9}$$

This means X will concentrate near the hypersphere of radius $\sqrt{n}\sigma$ with high probability.

Proof. First, we consider the squared norm of a random vector X : $\|X\|^2 = X_1^2 + X_2^2 + \ldots + X_n^2$. Since $X_i \sim \mathcal{N}(0, \sigma^2)$, it follows that $Y = \|X\|^2/\sigma^2 \sim \chi^2(n)$.

Using the Chernoff bound of the chi-square distribution:

$$P(Y \geq (1+\delta)n) \leq e^{-\frac{n}{2}[\delta - \ln(1+\delta)]}, \tag{10}$$

$$P(Y \leq (1-\delta)n) \leq e^{-\frac{n}{2}[-\delta - \ln(1-\delta)]}, \tag{11}$$

where $0 \leq \delta < 1$. According to $\ln(1+\delta) \leq \frac{(2+\delta)\delta}{2(1+\delta)}$ and $\ln(1-\delta) \leq -\delta - \frac{\delta^2}{2}$, we can deduce that:

$$\begin{aligned} P(\|X\|^2 \geq (1+\delta)n\sigma^2) &= P(Y \geq (1+\delta)n) \\ &\leq e^{-\frac{n\delta^2}{4(1+\delta)}}, \end{aligned} \tag{12}$$

$$\begin{aligned} P(\|X\|^2 \leq (1-\delta)n\sigma^2) &= P(Y \leq (1-\delta)n) \\ &\leq e^{-\frac{n\delta^2}{4}} \\ &\leq e^{-\frac{n\delta^2}{4(1+\delta)}}. \end{aligned} \tag{13}$$

Given the $\sqrt{1+\delta} \leq 1 + \frac{\delta}{2}$, we have:

$$\begin{aligned} P(\|X\| \geq (1+\frac{\delta}{2})\sqrt{n}\sigma) &\leq P(\|X\| \geq \sqrt{(1+\delta)}\sqrt{n}\sigma) \\ &\leq e^{-\frac{n\delta^2}{4(1+\delta)}}, \end{aligned} \tag{14}$$

which indicates that:

$$\begin{aligned} P(\|X\| \geq (1+\delta)\sqrt{n}\sigma) &\leq e^{-\frac{4n\delta^2}{4(1+2\delta)}} \\ &\leq e^{-\frac{n\delta^2}{4(1+\delta)}}, \end{aligned} \tag{15}$$

Also, given the $\sqrt{1-\delta} \geq 1 - \delta$, we have:

$$\begin{aligned} P(\|X\| \leq (1-\delta)\sqrt{n}\sigma) &\leq P(\|X\| \leq \sqrt{(1-\delta)}\sqrt{n}\sigma) \\ &\leq e^{-\frac{n\delta^2}{4(1+\delta)}}. \end{aligned} \tag{16}$$

Let $\epsilon = \delta$, and then organize equations (15) and (16):

$$P\left(\left|\|X\| - \sqrt{n}\sigma\right| \leq \epsilon\sqrt{n}\sigma\right) \geq 1 - 2e^{-\frac{n\epsilon^2}{4(1+\epsilon)}}, \tag{17}$$

which concludes the proof.

B Proof of Proposition 2.1

Proposition 2.1. Let $\mathbf{z}'_t$ denote the aggregated variable at time step t. For the sampling step from t to $t-1$, two diffusion models p_{θ_1} and p_{θ_2} sample $\mathbf{z}_{t-1}^{\theta_1}$ and $\mathbf{z}_{t-1}^{\theta_2}$, respectively, according to (3). Then, with probability at least $1-\gamma$, the absolute difference in norms and the angle is bounded by:

$$\left|\|\mathbf{z}_{t-1}^{\theta_1}\| - \|\mathbf{z}_{t-1}^{\theta_2}\|\right| \leq L_t + 2\sigma_t\sqrt{2\ln\frac{8}{\gamma}},$$

$$\cos\varphi \geq 1 - \frac{\delta^2}{\|\mathbf{z}_{t-1}^{\theta_1}\|^2 + \|\mathbf{z}_{t-1}^{\theta_2}\|^2},$$
$$\delta \leq L_t + \sigma_t \sqrt{2\left(n + 2\sqrt{n \ln \frac{2}{\gamma}} + 2\ln\frac{2}{\gamma}\right)}.$$

where φ is the angle between $\mathbf{z}_{t-1}^{\theta_1}$ and $\mathbf{z}_{t-1}^{\theta_2}$. The angle bound applies only when $\cos\varphi \geq 0$.

Proof.

According to (3), we have:

$$\begin{aligned} \mathbf{z}_{t-1}^{\theta_1} &= \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1) + \sigma_t \boldsymbol{\epsilon}_1, \\ \mathbf{z}_{t-1}^{\theta_2} &= \boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) + \sigma_t \boldsymbol{\epsilon}_2. \end{aligned} \tag{18}$$

Then, the difference between the norms is:

$$\begin{aligned} \left| \|\mathbf{z}_{t-1}^{\theta_1}\| - \|\mathbf{z}_{t-1}^{\theta_2}\| \right| &= \left| \|\boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1) + \sigma_t \boldsymbol{\epsilon}_1\| - \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) + \sigma_t \boldsymbol{\epsilon}_2\| \right| \\ &= \Big| \|\boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1) + \sigma_t \boldsymbol{\epsilon}_1\| - \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) + \sigma_t \boldsymbol{\epsilon}_1\| \\ &\quad + \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) + \sigma_t \boldsymbol{\epsilon}_1\| - \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) + \sigma_t \boldsymbol{\epsilon}_2\| \Big| \\ &\leq \left| \|\boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1) + \sigma_t \boldsymbol{\epsilon}_1\| - \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) + \sigma_t \boldsymbol{\epsilon}_1\| \right| \\ &\quad + \left| \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) + \sigma_t \boldsymbol{\epsilon}_1\| - \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) + \sigma_t \boldsymbol{\epsilon}_2\| \right| \\ &\leq \|\boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1) - \boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2)\| \\ &\quad + \left| \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) + \sigma_t \boldsymbol{\epsilon}_1\| - \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) + \sigma_t \boldsymbol{\epsilon}_2\| \right|. \end{aligned} \tag{19}$$

For the first term of (19), by the functional proximity assumption, we have

$$\|\boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1) - \boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2)\| \leq L_t. \tag{20}$$

For the second term in (19), it intuitively measures the distance between two samples generated by p_{θ_2}, which admits a controllable upper bound. More formally, define the norm function $f : \mathbb{R}^n \to \mathbb{R}$ with respect to noise $\boldsymbol{\epsilon}$:

$$f(\boldsymbol{\epsilon}) = \|\boldsymbol{\mu}_{\theta_2} + \sigma_t \boldsymbol{\epsilon}\|, \tag{21}$$

the Lipschitz constant derivation:

$$\begin{aligned} |f(\mathbf{x}) - f(\mathbf{y})| &= |\|\boldsymbol{\mu} + \sigma_t \mathbf{x}\| - \|\boldsymbol{\mu} + \sigma_t \mathbf{y}\|| \\ &\leq \|(\boldsymbol{\mu} + \sigma_t \mathbf{x}) - (\boldsymbol{\mu} + \sigma_t \mathbf{y})\| = \sigma_t \|\mathbf{x} - \mathbf{y}\|. \end{aligned} \tag{22}$$

Thus, $f(\cdot)$ is σ_t-Lipschitz. For $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, applying the Gaussian Concentration Inequality for Lipschitz functions (Theorem 2.26 in Wainwright (2019)):

$$P\left(|f(\boldsymbol{\epsilon}) - \mathbb{E}[f]| \geq u\right) \leq 2\exp\left(-\frac{u^2}{2\sigma_t^2}\right) \tag{23}$$

Let $\Delta_f = |f(\boldsymbol{\epsilon}_1) - f(\boldsymbol{\epsilon}_2)|$, then:

$$\Delta_f \leq |f(\boldsymbol{\epsilon}_1) - \mathbb{E}[f]| + |f(\boldsymbol{\epsilon}_2) - \mathbb{E}[f]|. \tag{24}$$

To bound $\Delta_f < 2u$, we bound the failure probability $P(\Delta_f \geq 2u)$:

$$\begin{aligned} P(\Delta_f \geq 2u) &\leq P(|f(\boldsymbol{\epsilon}_1) - \mathbb{E}[f]| \geq u) + P(|f(\boldsymbol{\epsilon}_2) - \mathbb{E}[f]| \geq u) \\ &\leq 2\exp\left(-\frac{u^2}{2\sigma_t^2}\right) + 2\exp\left(-\frac{u^2}{2\sigma_t^2}\right) \\ &= 4\exp\left(-\frac{u^2}{2\sigma_t^2}\right). \end{aligned} \tag{25}$$

Set the failure probability to $\gamma/2$ and solve for the threshold $2u$:

$$\frac{\gamma}{2} = 4\exp\left(-\frac{u^2}{2\sigma_t^2}\right) \implies \ln\frac{\gamma}{8} = -\frac{u^2}{2\sigma_t^2} \implies u = \sigma_t\sqrt{2\ln\frac{8}{\gamma}}. \tag{26}$$

Substituting u back into the bound $2u$:

$$\begin{aligned}&\Big|\|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}'_t, t, y_2) + \sigma_t\boldsymbol{\epsilon}_1\| - \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}'_t, t, y_2) + \sigma_t\boldsymbol{\epsilon}_2\|\Big| \\ &\qquad \le 2\sigma_t\sqrt{2\ln\frac{8}{\gamma}} \quad (\text{w.p. } \ge 1-\gamma/2).\end{aligned} \tag{27}$$

Therefore,

$$\Big|\|\mathbf{z}_{t-1}^{\theta_1}\| - \|\mathbf{z}_{t-1}^{\theta_2}\|\Big| \le L_t + 2\sigma_t\sqrt{2\ln\frac{8}{\gamma}} \quad (\text{w.p. } \ge 1-\gamma/2). \tag{28}$$

Denote φ as the angle between $\mathbf{z}_{t-1}^{\theta_1}$ and $\mathbf{z}_{t-1}^{\theta_2}$, and let $\delta = \|\mathbf{z}_{t-1}^{\theta_1} - \mathbf{z}_{t-1}^{\theta_2}\|$. Then

$$\begin{aligned}\cos\varphi &= \frac{\|\mathbf{z}_{t-1}^{\theta_1}\|^2 + \|\mathbf{z}_{t-1}^{\theta_2}\|^2 - \delta^2}{2\|\mathbf{z}_{t-1}^{\theta_1}\|\|\mathbf{z}_{t-1}^{\theta_2}\|} \\ &= 1 - \frac{\delta^2 - \left(\|\mathbf{z}_{t-1}^{\theta_1}\| - \|\mathbf{z}_{t-1}^{\theta_2}\|\right)^2}{2\|\mathbf{z}_{t-1}^{\theta_1}\|\|\mathbf{z}_{t-1}^{\theta_2}\|} \\ &\ge 1 - \frac{\delta^2}{\|\mathbf{z}_{t-1}^{\theta_1}\|^2 + \|\mathbf{z}_{t-1}^{\theta_2}\|^2},\end{aligned} \tag{29}$$

where the last inequality uses the local assumption $\cos\varphi \ge 0$, equivalently $\delta^2 \le \|\mathbf{z}_{t-1}^{\theta_1}\|^2 + \|\mathbf{z}_{t-1}^{\theta_2}\|^2$. Moreover:

$$\begin{aligned}\delta &= \|\boldsymbol{\mu}_{\theta_1}(\mathbf{z}'_t, t, y_1) - \boldsymbol{\mu}_{\theta_2}(\mathbf{z}'_t, t, y_2) + \sigma_t(\boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_2)\| \\ &\le L_t + \|\sigma_t(\boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_2)\|.\end{aligned} \tag{30}$$

To analyze the term $\|\sigma_t(\boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_2)\|$, we first define $Z = \left\|\frac{\Delta\boldsymbol{\epsilon}}{\sqrt{2}}\right\|^2 = \frac{1}{2}\|\boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_2\|^2$, which follows a Chi-squared distribution $\chi^2(n)$. Then, using the Laurent-Massart Bound for $\chi^2(n)$ (Laurent & Massart, 2000), for any $u > 0$:

$$P\left(Z \ge n + 2\sqrt{nu} + 2u\right) \le \exp(-u). \tag{31}$$

Setting $u = \ln(2/\gamma)$, we have with probability at least $1 - \gamma/2$:

$$\frac{1}{2\sigma_t^2}\|\sigma_t(\boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_2)\|^2 = Z \le n + 2\sqrt{n\ln\frac{2}{\gamma}} + 2\ln\frac{2}{\gamma}. \tag{32}$$

Thus:

$$\|\sigma_t(\boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_2)\| \le \sigma_t\sqrt{2\left(n + 2\sqrt{n\ln\frac{2}{\gamma}} + 2\ln\frac{2}{\gamma}\right)}. \tag{33}$$

Therefore:

$$\cos\varphi \ge 1 - \frac{\delta^2}{\|\mathbf{z}_{t-1}^{\theta_1}\|^2 + \|\mathbf{z}_{t-1}^{\theta_2}\|^2}, \tag{34}$$

where $\delta \le L_t + \sigma_t\sqrt{2\left(n + 2\sqrt{n\ln\frac{2}{\gamma}} + 2\ln\frac{2}{\gamma}\right)}$ (w.p. $\ge 1-\gamma/2$).

Combining (28) and the above bound by a union bound gives the norm-difference and δ bounds simultaneously with probability at least $1-\gamma$; the angle bound additionally requires $\cos\varphi \ge 0$. This concludes the proof.

C PROOF OF PROPOSITION 2.2

Proposition 2.2. For the diffusion model p_{θ_1} defined by (3) with $\sigma_t > 0$, and any new intermediate variable $\mathbf{z}'_{t-1}$ from (6), let $r = \left\|\mathbf{z}'_{t-1} - \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t^{\theta_1}, t, y_1)\right\| > 0$ and define:

$$\tilde{\mathbf{z}}_{t-1} = \mathbf{z}'_{t-1} - \eta_{t-1}^{\theta_1} \frac{r - \sqrt{n}\sigma_t}{|r - \sqrt{n}\sigma_t|} \frac{\mathbf{z}'_{t-1} - \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t^{\theta_1}, t, y_1)}{r},$$

where $0 \le \eta_{t-1}^{\theta_1} \le |r - \sqrt{n}\sigma_t|$ is a small correction step size. When $r = \sqrt{n}\sigma_t$, no correction is needed and we set $\tilde{\mathbf{z}}_{t-1} = \mathbf{z}'_{t-1}$. Under the local shell-concentration approximation, there exists $\eta_{t-1}^{\theta_1}$ such that the update in (7) moves $\mathbf{z}'_{t-1}$ toward $D_{t-1,y_1}^{\theta_1}$. Moreover, an approximate probabilistic interpretation for its compatibility with $D_{t-1,y_2}^{\theta_2}$ is given by:

$$P\left(\tilde{\mathbf{z}}_{t-1} \in D_{t-1,y_2}^{\theta_2}\right) \ge 1 - 2\exp\left(-\frac{n\left(\epsilon_{t-1}^{\theta_2} - \frac{d}{\sigma_t\sqrt{n}}\right)^2}{4\left(1 + \epsilon_{t-1}^{\theta_2} - \frac{d}{\sigma_t\sqrt{n}}\right)}\right),$$

where d is a deterministic local upper bound of $\phi_w(\varphi)\delta + \eta_{t-1}^{\theta_1}$ with $d < \epsilon_{t-1}^{\theta_2}\sigma_t\sqrt{n}$, and $\phi_w(\varphi) = \sin\big((1-w)\varphi/2\big)/\sin(\varphi/2)$.

Proof.

(1) There exists $\eta_{t-1}^{\theta_1}$ such that the update in (7) moves $\mathbf{z}'_{t-1}$ toward $D_{t-1,y_1}^{\theta_1}$:

We begin by defining the spherical shell as the high-probability set of $p_{\theta_i}(\mathbf{z}_{t-1}^{\theta_i} \mid \mathbf{z}_t^{\theta_i}, y_i)$:

$$\mathcal{A}_{t-1,y_i}^{\theta_i}(\epsilon_{t-1}^{\theta_i}) = \left\{\mathbf{z} : \left|\|\mathbf{z} - \boldsymbol{\mu}_{\theta_i}(\mathbf{z}_t^{\theta_i}, t, y_i)\| - \sqrt{n}\sigma_t\right| \le \epsilon_{t-1}^{\theta_i}\sqrt{n}\sigma_t\right\}, \tag{35}$$

where $\epsilon_{t-1}^{\theta_i}$ is chosen such that $\mathcal{A}_{t-1,y_i}^{\theta_i}(\epsilon_{t-1}^{\theta_i}) \subseteq D_{t-1,y_i}^{\theta_i}(\tau_{t-1}^{\theta_i})$; equivalently, it can be taken as any value below the corresponding supremum.

Let

$$r = \left\|\mathbf{z}'_{t-1} - \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t^{\theta_1}, t, y_1)\right\|, \quad \mathbf{u} = \frac{\mathbf{z}'_{t-1} - \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t^{\theta_1}, t, y_1)}{r}.$$

Then $\mathbf{z}'_{t-1} = \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t^{\theta_1}, t, y_1) + r\mathbf{u}$. If $r = \sqrt{n}\sigma_t$, choosing $\eta_{t-1}^{\theta_1} = 0$ makes the claim immediate. Otherwise, from (7), we have

$$\tilde{\mathbf{z}}_{t-1} = \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t^{\theta_1}, t, y_1) + \left(r - \eta_{t-1}^{\theta_1}\frac{r - \sqrt{n}\sigma_t}{|r - \sqrt{n}\sigma_t|}\right)\mathbf{u}. \tag{36}$$

Since $0 \le \eta_{t-1}^{\theta_1} \le |r - \sqrt{n}\sigma_t|$, this radial correction directly gives:

$$\left|\left\|\tilde{\mathbf{z}}_{t-1} - \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t^{\theta_1}, t, y_1)\right\| - \sqrt{n}\sigma_t\right| = \left|r - \sqrt{n}\sigma_t\right| - \eta_{t-1}^{\theta_1}. \tag{37}$$

Thus, choosing $0 \le \eta_{t-1}^{\theta_1} \le |r - \sqrt{n}\sigma_t|$ such that

$$\left|r - \sqrt{n}\sigma_t\right| - \eta_{t-1}^{\theta_1} \le \epsilon_{t-1}^{\theta_1}\sqrt{n}\sigma_t \tag{38}$$

ensures that $\tilde{\mathbf{z}}_{t-1} \in \mathcal{A}_{t-1,y_1}^{\theta_1} \subseteq D_{t-1,y_1}^{\theta_1}$. Such a choice exists, for example $\eta_{t-1}^{\theta_1} = \max\{|r - \sqrt{n}\sigma_t| - \epsilon_{t-1}^{\theta_1}\sqrt{n}\sigma_t, 0\}$, which completes the proof of existence.

(2) The approximate lower bound on the probability that $\tilde{\mathbf{z}}_{t-1} \in D_{t-1,y_2}^{\theta_2}$:

Let $\mathbf{z}_{t-1}^{\theta_2}$ be the corresponding one-step sample from p_{θ_2} used in (6). By the triangle inequality, we have:

$$\left\|\tilde{\mathbf{z}}_{t-1} - \mathbf{z}_{t-1}^{\theta_2}\right\| \le \left\|\mathbf{z}'_{t-1} - \mathbf{z}_{t-1}^{\theta_2}\right\| + \eta_{t-1}^{\theta_1}. \tag{39}$$

Here, the correction vector has norm $\eta_{t-1}^{\theta_1}$ by the normalized radial direction in (7).

From proposition 2.1, we use the local equal-norm spherical approximation $\left\|\mathbf{z}_{t-1}^{\theta_1}\right\| \approx \left\|\mathbf{z}_{t-1}^{\theta_2}\right\| \approx r_{t-1}$. Under this approximation, the chord lengths satisfy:

$$\left\|\mathbf{z}'_{t-1} - \mathbf{z}_{t-1}^{\theta_2}\right\| = 2r_{t-1}\sin\Big(\tfrac{(1-w)\varphi}{2}\Big), \qquad \delta = \left\|\mathbf{z}_{t-1}^{\theta_1} - \mathbf{z}_{t-1}^{\theta_2}\right\| = 2r_{t-1}\sin\Big(\tfrac{\varphi}{2}\Big). \tag{40}$$

Dividing the two identities yields:

$$\left\|\mathbf{z}'_{t-1} - \mathbf{z}_{t-1}^{\theta_2}\right\| = \phi_w(\varphi)\,\delta, \quad \phi_w(\varphi) = \frac{\sin\big(\frac{(1-w)\varphi}{2}\big)}{\sin\big(\frac{\varphi}{2}\big)}. \tag{41}$$

Combining (39) and (41), the realized distance is bounded by $\phi_w(\varphi)\,\delta + \eta_{t-1}^{\theta_1}$. Under the local approximation, let d be a deterministic upper bound of this quantity. We can obtain the bound:

$$\left\|\tilde{\mathbf{z}}_{t-1} - \mathbf{z}_{t-1}^{\theta_2}\right\| \le d. \tag{42}$$

Thus, d can be viewed as a deterministic upper bound on the correction-induced distance to the corresponding sample from p_{θ_2}. Writing $\boldsymbol{\mu}_{\theta_2}$ for $\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t^{\theta_2}, t, y_2)$ and using $d < \epsilon_{t-1}^{\theta_2}\sigma_t\sqrt{n}$, this implies that, if:

$$\big|\,\|\mathbf{z}_{t-1}^{\theta_2} - \boldsymbol{\mu}_{\theta_2}\| - \sigma_t\sqrt{n}\,\big| \le \big(\epsilon_{t-1}^{\theta_2} - \tfrac{d}{\sigma_t\sqrt{n}}\big)\sigma_t\sqrt{n}, \tag{43}$$

then every point within distance d from $\mathbf{z}_{t-1}^{\theta_2}$ belongs to $\mathcal{A}_{t-1,y_2}^{\theta_2}$. Since $\left\|\tilde{\mathbf{z}}_{t-1} - \mathbf{z}_{t-1}^{\theta_2}\right\| \le d$, the above event implies $\tilde{\mathbf{z}}_{t-1} \in \mathcal{A}_{t-1,y_2}^{\theta_2}$. Therefore:

$$\begin{aligned} P\left(\tilde{\mathbf{z}}_{t-1} \in D_{t-1,y_2}^{\theta_2}\right) &\ge P\left(\tilde{\mathbf{z}}_{t-1} \in \mathcal{A}_{t-1,y_2}^{\theta_2}\right) \\ &\ge P\left(\big|\,\|\mathbf{z}_{t-1}^{\theta_2} - \boldsymbol{\mu}_{\theta_2}\| - \sigma_t\sqrt{n}\,\big| \le \big(\epsilon_{t-1}^{\theta_2} - \tfrac{d}{\sigma_t\sqrt{n}}\big)\sigma_t\sqrt{n}\right) \\ &\ge 1 - 2\exp\left(-\frac{n\left(\epsilon_{t-1}^{\theta_2} - \frac{d}{\sigma_t\sqrt{n}}\right)^2}{4\left(1 + \epsilon_{t-1}^{\theta_2} - \frac{d}{\sigma_t\sqrt{n}}\right)}\right). \end{aligned} \tag{44}$$

The first line follows from $\mathcal{A}_{t-1,y_2}^{\theta_2} \subseteq D_{t-1,y_2}^{\theta_2}$, and the last line follows from Lemma A.1 with $\epsilon = \epsilon_{t-1}^{\theta_2} - d/(\sigma_t\sqrt{n})$, which concludes the proof.

D Geometric Properties of AMDM

The following proposition shows that the points obtained by applying the deviation optimization in (7) within the AMDM algorithm also lie on a local spherical manifold; therefore, spherical interpolation remains applicable.

Proposition D.1. *Let $\mathbf{z}'_t$ denote the aggregated variable at time step t, and let $r_i = \|\mathbf{z}'_t - \boldsymbol{\mu}_{\theta_i}(\mathbf{z}_{t+1}^{\theta_i}, t+1, y_i)\|$. Consider the non-degenerate case where $r_i > 0$ and $r_i \ne \sqrt{n}\sigma_t$ for $i = 1, 2$. For the AMDM step from t to $t-1$, the deviation optimization is first performed via (7) using diffusion models p_{θ_1} and p_{θ_2}, followed by sampling according to (3) to obtain $\mathbf{z}_{t-1}^{\theta_1}$ and $\mathbf{z}_{t-1}^{\theta_2}$, respectively. Then, with probability at least $1-\gamma$, the absolute difference in norms and the angles are bounded by:*

$$\Big|\,\|\mathbf{z}_{t-1}^{\theta_1}\| - \|\mathbf{z}_{t-1}^{\theta_2}\|\,\Big| \le L_t + L_{\mathbf{z}}^{\theta_1}\eta_t^{\theta_1} + L_{\mathbf{z}}^{\theta_2}\eta_t^{\theta_2} + 2\sigma_t\sqrt{2\ln\frac{8}{\gamma}}. \tag{45}$$

$$\begin{aligned} \cos\varphi &\ge 1 - \frac{\delta^2}{\|\mathbf{z}_{t-1}^{\theta_1}\|^2 + \|\mathbf{z}_{t-1}^{\theta_2}\|^2}, \\ \delta &\le L_t + L_{\mathbf{z}}^{\theta_1}\eta_t^{\theta_1} + L_{\mathbf{z}}^{\theta_2}\eta_t^{\theta_2} + \sigma_t\sqrt{2\left(n + 2\sqrt{n\ln\frac{2}{\gamma}} + 2\ln\frac{2}{\gamma}\right)}. \end{aligned} \tag{46}$$

The angle bound applies only when $\cos\varphi \ge 0$.

Proof.

According to the definition of AMDM, deviation optimization is introduced in the step from t to $t-1$. Based on (3) and (7), the sampled variable $\mathbf{z}_{t-1}^{\theta_i}$ after deviation optimization can be expressed as:

$$\begin{aligned}\mathbf{z}_{t-1}^{\theta_i} &= \boldsymbol{\mu}_{\theta_i}(\tilde{\mathbf{z}}_t^{\theta_i}, t, y_i) + \sigma_t \boldsymbol{\epsilon}_i, \\ \tilde{\mathbf{z}}_t^{\theta_i} &= \mathbf{z}_t' + \mathbf{d}_i, \\ \mathbf{d}_i &= -\eta_t^{\theta_i} \frac{r_i - \sqrt{n}\sigma_t}{|r_i - \sqrt{n}\sigma_t|} \frac{\mathbf{z}_t' - \boldsymbol{\mu}_{\theta_i}(\mathbf{z}_{t+1}^{\theta_i}, t+1, y_i)}{r_i}.\end{aligned} \tag{47}$$

Then, the absolute difference in norms is:

$$\begin{aligned}\left| \|\mathbf{z}_{t-1}^{\theta_1}\| - \|\mathbf{z}_{t-1}^{\theta_2}\| \right| &= \left| \|\boldsymbol{\mu}_{\theta_1}(\tilde{\mathbf{z}}_t^{\theta_1}, t, y_1) + \sigma_t \boldsymbol{\epsilon}_1\| - \|\boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2) + \sigma_t \boldsymbol{\epsilon}_2\| \right| \\ &\leq \|\boldsymbol{\mu}_{\theta_1}(\tilde{\mathbf{z}}_t^{\theta_1}, t, y_1) - \boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2)\| \\ &\quad + \left| \|\boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2) + \sigma_t \boldsymbol{\epsilon}_1\| - \|\boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2) + \sigma_t \boldsymbol{\epsilon}_2\| \right|.\end{aligned} \tag{48}$$

For the first term of (48):

$$\begin{aligned}&\|\boldsymbol{\mu}_{\theta_1}(\tilde{\mathbf{z}}_t^{\theta_1}, t, y_1) - \boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2)\| \\ =&\|\boldsymbol{\mu}_{\theta_1}(\tilde{\mathbf{z}}_t^{\theta_1}, t, y_1) - \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1) + \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1) - \boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) \\ &\quad + \boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) - \boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2)\| \\ \leq&\|\boldsymbol{\mu}_{\theta_1}(\tilde{\mathbf{z}}_t^{\theta_1}, t, y_1) - \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1)\| \\ &\quad + \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) - \boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2)\| \\ &\quad + \|\boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1) - \boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2)\| \\ \leq&\|\boldsymbol{\mu}_{\theta_1}(\tilde{\mathbf{z}}_t^{\theta_1}, t, y_1) - \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1)\| \\ &\quad + \|\boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2) - \boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2)\| + L_t.\end{aligned} \tag{49}$$

Practically, the denoising network $\mu_{\theta_i}(\cdot)$ is composed of standard neural network operators (e.g., Linear layers, LayerNorm, SiLU) with finite weights, and is differentiable almost everywhere. Therefore, within a small neighborhood induced by the optimization step size $\eta_t^{\theta_i}$, we assume that the input Jacobian remains bounded. Hence, there exists a local constant $L_{\mathbf{z}}^{\theta_i}$ such that $\left\|\boldsymbol{\mu}_{\theta_i}(\tilde{\mathbf{z}}) - \boldsymbol{\mu}_{\theta_i}(\mathbf{z})\right\| \leq L_{\mathbf{z}}^{\theta_i} \left\|\tilde{\mathbf{z}} - \mathbf{z}\right\|$. Then, the first term of (49) is:

$$\begin{aligned}\|\boldsymbol{\mu}_{\theta_1}(\tilde{\mathbf{z}}_t^{\theta_1}, t, y_1) - \boldsymbol{\mu}_{\theta_1}(\mathbf{z}_t', t, y_1)\| &\leq L_{\mathbf{z}}^{\theta_1} \left\|\tilde{\mathbf{z}}_t^{\theta_1} - \mathbf{z}_t'\right\| \\ &= L_{\mathbf{z}}^{\theta_1} \|\mathbf{d}_1\| \\ &= L_{\mathbf{z}}^{\theta_1} \eta_t^{\theta_1}.\end{aligned} \tag{50}$$

Similarly, $\|\boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2) - \boldsymbol{\mu}_{\theta_2}(\mathbf{z}_t', t, y_2)\| \leq L_{\mathbf{z}}^{\theta_2} \eta_t^{\theta_2}$. Hence, (49) can be simplified:

$$\|\boldsymbol{\mu}_{\theta_1}(\tilde{\mathbf{z}}_t^{\theta_1}, t, y_1) - \boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2)\| \leq L_{\mathbf{z}}^{\theta_1} \eta_t^{\theta_1} + L_{\mathbf{z}}^{\theta_2} \eta_t^{\theta_2} + L_t. \tag{51}$$

For the second term of (48), by applying the result in 28, we obtain that:

$$\begin{aligned}&\left| \|\boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2) + \sigma_t \boldsymbol{\epsilon}_1\| - \|\boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2) + \sigma_t \boldsymbol{\epsilon}_2\| \right| \\ &\qquad \leq 2\sigma_t \sqrt{2 \ln \frac{8}{\gamma}} \quad (\text{w.p. } \geq 1 - \gamma/2).\end{aligned} \tag{52}$$

For δ:

$$\begin{aligned}\delta &= \|\mathbf{z}_{t-1}^{\theta_1} - \mathbf{z}_{t-1}^{\theta_2}\| \\ &= \|\boldsymbol{\mu}_{\theta_1}(\tilde{\mathbf{z}}_t^{\theta_1}, t, y_1) - \boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2) + \sigma_t(\boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_2)\| \\ &\leq \|\boldsymbol{\mu}_{\theta_1}(\tilde{\mathbf{z}}_t^{\theta_1}, t, y_1) - \boldsymbol{\mu}_{\theta_2}(\tilde{\mathbf{z}}_t^{\theta_2}, t, y_2)\| + \|\sigma_t(\boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_2)\|.\end{aligned} \tag{53}$$

According to (51) and (33), we obtain:

$$\delta \le L_{\mathbf{z}}^{\theta_1}\eta_t^{\theta_1} + L_{\mathbf{z}}^{\theta_2}\eta_t^{\theta_2} + L_t + \sigma_t\sqrt{2\left(n + 2\sqrt{n\ln\frac{2}{\gamma}} + 2\ln\frac{2}{\gamma}\right)} \quad (\text{w.p. } \ge 1-\gamma/2). \tag{54}$$

By the same law-of-cosines argument as in Proposition 2.1, under the local assumption $\cos\varphi \ge 0$, $\cos\varphi \ge 1 - \delta^2/(\|\mathbf{z}_{t-1}^{\theta_1}\|^2 + \|\mathbf{z}_{t-1}^{\theta_2}\|^2)$. Combining the norm-difference event and the δ event by a union bound gives the norm-difference and δ bounds simultaneously with probability at least $1-\gamma$; the angle bound additionally requires $\cos\varphi \ge 0$. This concludes the proof.

Proposition D.1 shows that the AMDM algorithm shares similar geometric properties with those described in Proposition 2.1, in the sense that the generated samples also lie on the same local spherical manifold. Moreover, the introduction of the deviation optimization strategy leads to a more pronounced improvement in sampling quality. Relevant numerical results are provided in Appendix E.

E Statistical analysis

E.1 Gaussianity in latent space

Chung et al. (2022) have shown that $\mathcal{M}_t$ is concentrated on an $(n-1)$-dimensional manifold, which approximates an n-dimensional hypersphere as t becomes large. Motivated by this property and Proposition 2.1, the AMDM algorithm performs spherical aggregation in order to reduce manifold deviation. Next, we focus on the statistical characteristics of the distribution $p(\mathbf{z}_t)$, in particular its first and second-order moments, to illustrate its behavior in the large t regime.

The continuous form of (1) can be rewritten as:

$$\mathrm{d}\mathbf{z}_t = -\frac{1}{2}g_t^2\mathbf{z}_t\mathrm{d}t + g_t\mathrm{d}\mathbf{w}_t, \tag{55}$$

We can use equations (5.50) and (5.51) (Särkkä & Solin, 2019) to derive the relationship between the mean and variance of (55) as they evolve over time:

$$\frac{\mathrm{d}\mathbf{m}}{\mathrm{d}t} = \mathbb{E}\left[-\frac{1}{2}g_t^2\mathbf{z}_t\right] = -\frac{1}{2}g_t^2\mathbf{m}, \tag{56}$$

$$\begin{aligned}\frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} &= \mathbb{E}\left[-\frac{1}{2}g_t^2\mathbf{z}_t(\mathbf{z}_t-\mathbf{m})^T\right] + \mathbb{E}\left[-\frac{1}{2}g_t^2(\mathbf{z}_t-\mathbf{m})\mathbf{z}_t^T\right] + g_t^2\boldsymbol{I} \\ &= -g_t^2\mathbf{P} + g_t^2\boldsymbol{I}.\end{aligned} \tag{57}$$

The solutions are:

$$\mathbf{m}(t) = \mathbf{m}(0)e^{-\frac{1}{2}\int_0^t g_s^2\mathrm{d}s} \tag{58}$$

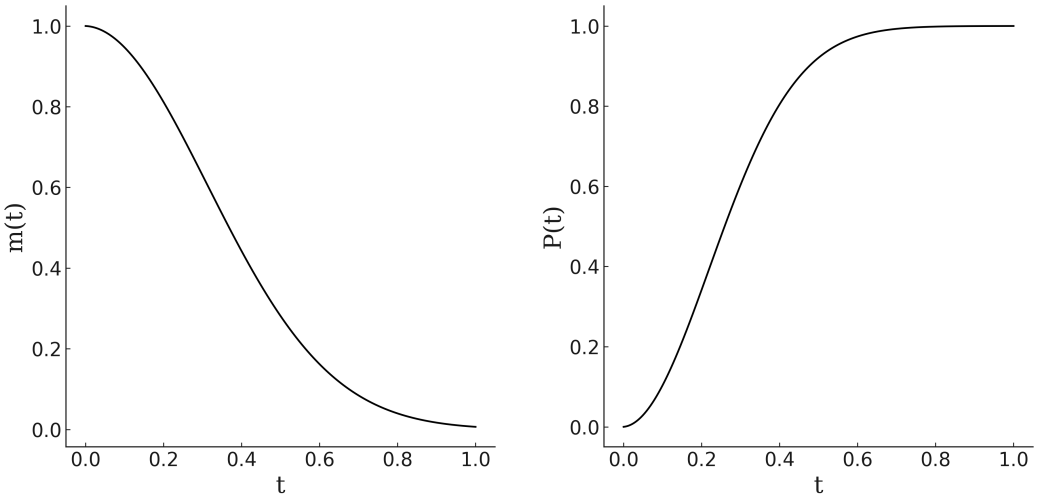
$$\mathbf{P}(t) = \boldsymbol{I} + (\mathbf{P}(0) - \boldsymbol{I})\, e^{-\int_0^t g_s^2\mathrm{d}s}, \tag{59}$$

where $g_t = \sqrt{\beta_t}$, and we set β_t to follow the linear schedule in SD (Rombach et al., 2022). For illustration, we consider $\mathbf{m}(0) = \mathbf{1}$ and $\mathbf{P}(0) = \mathbf{0}$. The resulting mean and variance of $p(\mathbf{z}_t)$ as functions of time are shown in Figure 6.

For this illustrative setting, Figure 6 shows the mean approaching 0 and the variance approaching 1. Gaussian convergence follows from (2): $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is independent of $\mathbf{z}_0$. For $\mathbb{E}\|\mathbf{z}_0\|^2 < \infty$, as $\bar{\alpha}_t \to 0$, the initial-data term vanishes in mean square and $\mathbf{z}_t$ converges in distribution to a standard Gaussian.

E.2 Numerical Analysis of the Experiments for Proposition 2.1

To assess the validity of Proposition 2.1, we log intermediate variables from InteractDiffusion(+MIGC) and report them in Table 5 and Figure 7. During the first $s = 20$ aggregation



(a) Mean over time.

(b) Variance over time.

Figure 6: Mean and variance over time for the illustrative initialization $\mathbf{m}(0) = \mathbf{1}, \mathbf{P}(0) = \mathbf{0}$.

steps ($t = 50$ to $t = 30$), $\left| \|\mathbf{z}_t^{\theta_1}\| - \|\mathbf{z}_t^{\theta_2}\| \right| \approx 0$ and φ is small after the initial two steps, which directly supports the analysis of Proposition 2.1. This behavior is expected for the following reasons. At initialization ($t = 50$), both trajectories are sampled from the standard Gaussian and thus have numerically almost identical magnitudes; the subsequent single-step denoising from $t = 50 \to 49$, although performed by different models, introduces only a small variation, so the magnitudes remain nearly equal. Moreover, the small variations in the angle φ and in the difference norm $\delta = \|\mathbf{z}_t^{\theta_1} - \mathbf{z}_t^{\theta_2}\|$ further indicate that the per-step generative error is well controlled. Once aggregation takes effect, $\mathbf{z}_{t-1}^{\theta_1}$ and $\mathbf{z}_{t-1}^{\theta_2}$ start from the common aggregation point $\mathbf{z}'_t$, undergo small deviation optimization, and are then sampled separately; because the aggregation point is shared, the deviations are limited, and each sampling step introduces only a small error (functional proximity), the magnitudes remain almost equal. Taken together, these observations justify the equal-norm assumption throughout the aggregation process.

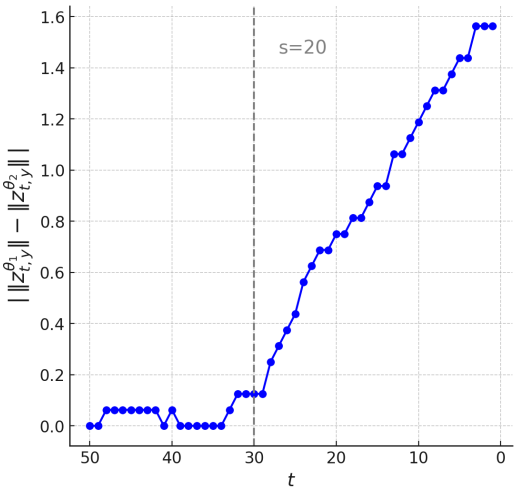
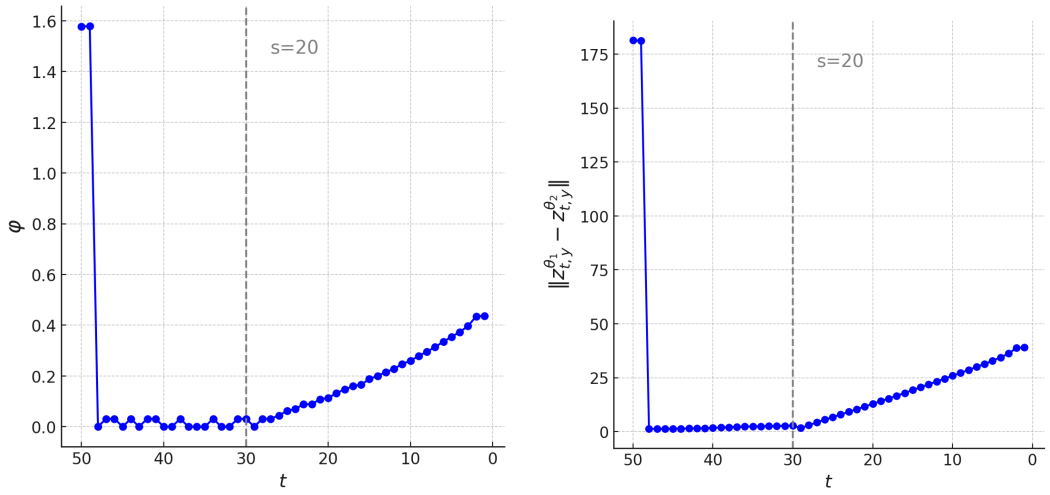


Figure 7: Difference of norm varying with time.



(a) Angle φ varying with time.

(b) Difference norm varying with time.

Figure 8: Variation of angle and the difference norm.

E.3 Maximum distance term

In Proposition 2.2, we derived the approximate lower bound of the probability:

$$P\left(\tilde{\mathbf{z}}_{t-1} \in D_{t-1,y_2}^{\theta_2}\right) \geq 1 - 2\exp\left(-\frac{n\left(\epsilon_{t-1}^{\theta_2} - \frac{d}{\sigma_t\sqrt{n}}\right)^2}{4\left(1 + \epsilon_{t-1}^{\theta_2} - \frac{d}{\sigma_t\sqrt{n}}\right)}\right),$$

where the key quantity is the maximum distance to the model, with d an upper bound on $\phi_w(\varphi)\,\delta + \eta_{t-1}^{\theta_1}$ and satisfying $d < \epsilon_{t-1}^{\theta_2}\sigma_t\sqrt{n}$. The numerical value of d directly determines whether the optimized variable $\tilde{\mathbf{z}}_{t-1}$ is likely to lie on the data manifold of model θ_2, thereby helping preserve auxiliary control signals. Theoretically, the smaller the value of d, the higher the likelihood. To this end, we separately analyze $\phi_w(\varphi) = \frac{\sin\left((1-w)\varphi/2\right)}{\sin(\varphi/2)}$ and $\delta = \left\|\mathbf{z}_{t-1}^{\theta_1} - \mathbf{z}_{t-1}^{\theta_2}\right\|$.

For $\phi_w(\varphi)$, we first examine the variation of the angle φ over time. The detailed experimental results can be referred to in Table 5 and Fig 8a. At the initial step $t = 50$, since random points in high-dimensional Gaussian space are almost orthogonal, the angle satisfies $\varphi \approx \frac{\pi}{2}$. At $t = 49$, a single sampling step of the model introduces only a negligible perturbation, so φ remains essentially unchanged. Subsequently, following an argument similar to the analysis of difference of norm in Appendix E.2: since $\mathbf{z}_{t-1}^{\theta_1}$ and $\mathbf{z}_{t-1}^{\theta_2}$ are both derived from the common aggregation point $\mathbf{z}_t'$, and then undergo deviation optimization and independent sampling, these operations only introduce small perturbations, implying that φ becomes and remains small after aggregation. Expanding $\phi_w(\varphi)$ in a Taylor series gives $\phi_w(\varphi) = (1-w) + O(\varphi^2)$. Since φ is very small, the higher-order terms can be neglected, yielding $\phi_w(\varphi) \approx (1-w)$. In summary, at the initial step, $\phi_w(\varphi) \approx \phi_w\left(\frac{\pi}{2}\right) = \sqrt{2}\,\sin\left(\frac{(1-w)\pi}{4}\right) \leq 1$, whereas after aggregation, $\phi_w(\varphi) \approx (1-w) \leq 1$. Thus, we conclude that $\phi_w(\varphi) \leq 1$.

For δ, the specific experimental values are provided in Table 5 and Figure 8b. Based on a similar analysis as above, it can be inferred that at the initial steps $t = 50$ and $t = 49$, the vectors are generated by random sampling, leading to a large magnitude of their difference. In subsequent steps, however, since both vectors originate from the same aggregation point of the previous step and the following operations introduce only small perturbations, the magnitude of their difference becomes significantly smaller.

In summary, only after the first aggregation optimization may the variable $\tilde{\mathbf{z}}_{T-1}$ deviate from $D_{T-1,y_2}^{\theta_2}$. In subsequent aggregation steps, however, the aggregated variable empirically remains closer to

Table 5: Values of different intermediate variables over time in InteractDiffusion (+MIGC)

t	φ	$\left\|\mathbf{z}_t^{\theta_1} - \mathbf{z}_t^{\theta_2}\right\|$	$\left\|\left\|\mathbf{z}_t^{\theta_1}\right\| - \left\|\mathbf{z}_t^{\theta_2}\right\|\right\|$
50	1.5781	181.3750	0.0000
49	1.5791	181.2500	0.0000
48	0.0000	1.3086	0.0625
47	0.0312	1.3125	0.0625
46	0.0312	1.3281	0.0625
45	0.0000	1.3652	0.0625
44	0.0312	1.4219	0.0625
43	0.0000	1.5244	0.0625
42	0.0312	1.5625	0.0625
41	0.0312	1.6426	0.0000
40	0.0000	1.7812	0.0625
39	0.0000	1.8926	0.0000
38	0.0312	2.0586	0.0000
37	0.0000	2.2402	0.0000
36	0.0000	2.3730	0.0000
35	0.0000	2.4258	0.0000
34	0.0312	2.4707	0.0000
33	0.0000	2.5508	0.0625
32	0.0000	2.6309	0.1250
31	0.0312	2.7324	0.1250
30	0.0312	2.8184	0.1250
29	0.0000	1.6904	0.1250
28	0.0312	2.9727	0.2500
27	0.0312	4.2734	0.3125
26	0.0442	5.5273	0.3750
25	0.0625	6.7500	0.4375
24	0.0699	7.9648	0.5625
23	0.0884	9.1797	0.6250
22	0.0884	10.4141	0.6875
21	0.1083	11.6406	0.6875
20	0.1127	12.8750	0.7500
19	0.1327	14.1250	0.7500
18	0.1467	15.3750	0.8125
17	0.1595	16.6406	0.8125
16	0.1655	17.9219	0.8750
15	0.1877	19.2500	0.9375
14	0.2004	20.5938	0.9375
13	0.2146	21.9219	1.0625
12	0.2280	23.2500	1.0625
11	0.2467	24.5781	1.1250
10	0.2603	25.9219	1.1875
9	0.2786	27.2656	1.2500
8	0.2959	28.6094	1.3125
7	0.3137	29.9844	1.3125
6	0.3352	31.3906	1.3750
5	0.3540	32.8438	1.4375
4	0.3733	34.4062	1.4375
3	0.3967	36.1875	1.5625
2	0.4353	38.8125	1.5625
1	0.4365	38.9688	1.5625

the compatible high-density region of p_{θ_2}, suggesting that auxiliary model characteristics can be partially preserved during aggregation. This observation is also intuitive: since the initial sampling is performed from two random points that are relatively far apart, it is difficult for the first aggregation to lie simultaneously on the manifolds of both models at step $T-1$.

In addition, we conducted 10 additional tests with different random seeds. The results show that the angle φ and the difference of norm $\left|\|\mathbf{z}_t^{\theta_1}\| - \|\mathbf{z}_t^{\theta_2}\|\right|$ remain nearly identical across experiments,

while the difference norm $\left\|\mathbf{z}_{t-1}^{\theta_1} - \mathbf{z}_{t-1}^{\theta_2}\right\|$ fluctuates only within ± 1. Hence, the observed trends appear stable across these random seeds.

F AMDM ALGORITHM

Algorithm 1 gives the full inference procedure of AMDM for N compatible diffusion models. The target model is denoted as p_{θ_1}, which is used for the final decoding and late-stage sampling. All models are initialized from Gaussian latent variables and are sampled in parallel during the first s aggregation steps. At each aggregation step, we first obtain the one-step reverse samples from all models, then apply spherical aggregation to obtain a shared aggregated latent state. Deviation optimization is subsequently applied to each model-specific trajectory to reduce the deviation from its corresponding reverse-kernel shell. After the first s steps, auxiliary models are no longer evaluated, and sampling continues only with the target model p_{θ_1}. For $N > 2$, AMDM uses an anchor-centered pairwise aggregation strategy. The target model is used as the anchor; each auxiliary latent is first aggregated with the anchor latent to obtain anchor-aligned states, and these states are then recursively merged.

Algorithm 1 AMDM

Input: models from the same diffusion ecosystem $p_{\theta_1}, p_{\theta_2}, ..., p_{\theta_N}$, conditions $y_1, y_2, ..., y_N$, aggregation step s, weighting factors $w_1, w_2, ..., w_{N-1}$ and optimization steps $\eta_t^{\theta_1}, \eta_t^{\theta_2}, ..., \eta_t^{\theta_N}$
$\mathbf{z}_T^{\theta_1}, \mathbf{z}_T^{\theta_2}, ..., \mathbf{z}_T^{\theta_N} \sim N(\mathbf{0}, \boldsymbol{I})$
for $t = T, ..., 1$ **do**
  $\mathbf{z}_{t-1}^{\theta_1} \sim p_{\theta_1}(\mathbf{z}_{t-1}^{\theta_1} | \mathbf{z}_t^{\theta_1}, y_1)$
  if $t > T - s$ **then**
    $\mathbf{z}_{t-1}^{\theta_i} \sim p_{\theta_i}(\mathbf{z}_{t-1}^{\theta_i} | \mathbf{z}_t^{\theta_i}, y_i), i \in [2, N]$
    $\mathbf{z}'_{t-1} = Slerp(\mathbf{z}_{t-1}^{\theta_1}, ..., \mathbf{z}_{t-1}^{\theta_N}, w_1, ..., w_{N-1})$
    $r_i = \|\mathbf{z}'_{t-1} - \mu_{\theta_i}(\mathbf{z}_t^{\theta_i}, t, y_i)\|, i \in [1, N]$
    $\mathbf{z}_{t-1}^{\theta_i} = \mathbf{z}'_{t-1} - \eta_{t-1}^{\theta_i} \frac{r_i - \sqrt{n}\sigma_t}{|r_i - \sqrt{n}\sigma_t|} \frac{\mathbf{z}'_{t-1} - \mu_{\theta_i}(\mathbf{z}_t^{\theta_i}, t, y_i)}{r_i}, i \in [1, N]$
  end if
end for
$\mathbf{x}_0^{\theta_1} = \text{Decoder}(\mathbf{z}_0^{\theta_1})$
Output: $\mathbf{x}_0^{\theta_1}$

G COMPARISON WITH COMPOSITIONAL METHODS

Compositional diffusion methods aim to synthesize samples satisfying multiple concepts by combining the distributions or score functions of different diffusion models. In the main paper, we compare AMDM with representative inference-time compositional methods, including projective composition (Bradley et al., 2025), Feynman–Kac Correctors (Skreta et al., 2025a), and SUPERDIFF (Skreta et al., 2025b). These methods mainly operate in the score or density-estimation space, while AMDM performs aggregation directly in the latent data space of models from the same diffusion ecosystem.

Implementation details. For a fair comparison, all compositional baselines are adapted to the same MIGC–InteractDiffusion setting and evaluated with the same prompts, layouts, sampling steps, and random seeds. For score-composition baselines, we compose the model predictions under the same scheduler. For SUPERDIFF, we use its AND-style composition to encourage agreement between the two model distributions. The composition weights and guidance scales are selected on a validation subset and then fixed for all reported experiments. More specifically, projective composition uses the standard shared-background linear composition rule, Du et al. (2023) uses equal-weight score composition followed by Langevin MCMC correction, Feynman–Kac Correctors uses a 4-particle approximation with $\beta = 1.0$ and rejuvenation noise 0.02, and SUPERDIFF uses its original AND-style score fusion under the same scheduler. For the runtime comparison in Table 6, all baselines are measured with 50 denoising steps; for Du et al., we use 3 Langevin corrector steps with step size 10^{-4}.

Runtime comparison. Table 6 reports the average scores and inference time of different compositional methods. Although these baselines improve over the weaker single model, they remain inferior to AMDM in both attribute control and interaction control. Each runtime is tested over five runs and reported as mean $\pm$ standard deviation. Among the non-AMDM baselines, projective composition is the lightest because it only performs linear score fusion at each denoising step, but it still runs both branches throughout sampling. Du et al. (2023) is the slowest because each denoising step is followed by additional Langevin correction steps, while Feynman–Kac Correctors is also expensive due to particle propagation, reweighting, resampling, and rejuvenation. SUPERDIFF is intermediate because its AND-style fusion requires extra score/divergence computations at every step. In contrast, AMDM uses closed-form latent aggregation and deviation correction without MCMC, particle propagation, or extra score/divergence estimation. It only runs both branches during early aggregation and then continues with the target branch, making AMDM the fastest compositional method in the comparison.

Table 6: Supplementary comparison of MIGC and InteractDiffusion composition on COCO-MIG and HOI detection.

Method	Instance Success Rate Avg ↑	mIoU Score Avg ↑	Default (Full) ↑	Time (s) ↓
Du et al. (2023)	45.14	40.03	19.23	74.48 ± 0.69
Projective Composition (Bradley et al., 2025)	47.82	42.71	22.23	17.33 ± 0.21
Feynman–Kac Correctors (Skreta et al., 2025a)	46.35	41.25	21.74	65.96 ± 0.78
SUPERDIFF (Skreta et al., 2025b)	46.56	41.81	23.15	33.03 ± 0.60
InteractDiffusion(+MIGC)	**54.78**	**47.74**	**31.40**	$\underline{13.52 \pm 0.06}$
MIGC(+InteractDiffusion)	<u>49.78</u>	<u>43.69</u>	<u>26.04</u>	$\mathbf{10.83 \pm 0.04}$

Discussion. Under the same scheduler and aligned sampling states, score-space composition can be viewed as related to linearly combining model predictions in the reverse update. However, such linear composition does not explicitly preserve the geometric structure of the latent data manifold. This explains why the linear aggregation baseline in Figure 4b becomes unstable when more aggregation steps are used. In contrast, AMDM uses spherical aggregation and deviation optimization to keep the aggregated variables close to the latent data manifold.

The difference is also reflected in the assumptions of existing compositional methods. Projective composition (Bradley et al., 2025) is most effective when the composed conditions lie in factorized or approximately orthogonal feature spaces. Fine-grained generation, however, usually involves highly coupled conditions, such as object attributes, spatial layouts, interactions, and styles describing the same scene. Feynman–Kac Correctors (Skreta et al., 2025a) and SUPERDIFF (Skreta et al., 2025b) provide more sophisticated correction or weighting mechanisms, but they also require additional computation during inference. AMDM is instead designed for compatible models within the same diffusion ecosystem, where different models preserve a shared latent space while specializing in different control capabilities.

Limitations on general compositional distributions. AMDM is not intended to replace general-purpose compositional generation methods. Its design relies on the shared latent space and the approximately spherical concentration of high-dimensional diffusion latents within the same ecosystem. When these structural assumptions are violated, AMDM fails, whereas dedicated compositional samplers that operate directly on the combined score function remain effective.

To illustrate this boundary, we construct a 2D Gaussian mixture composition task. We define two mixture distributions: $p_1(x) = \frac{1}{2}\mathcal{N}(\mu_1^{(1)}, \sigma^2 I) + \frac{1}{2}\mathcal{N}(\mu_2^{(1)}, \sigma^2 I)$ with modes at $(\pm 3, 0)$, and $p_2(x) = \frac{1}{2}\mathcal{N}(\mu_1^{(2)}, \sigma^2 I) + \frac{1}{2}\mathcal{N}(\mu_2^{(2)}, \sigma^2 I)$ with modes at $(0, \pm 3)$, where $\sigma = 1$. The target is the product distribution $p_1(x) \cdot p_2(x)$, which has four modes near $(\pm 3, \pm 3)$. We use the analytic score functions of each mixture and simulate a DDPM-style diffusion process in 2D. AMDM applies Slerp and deviation optimization on the intermediate samples at inference time. For comparison, we use the annealed Langevin composition method of Du et al. (2023), which combines the score functions $\nabla_x \log p_1 + \nabla_x \log p_2$ via MCMC sampling.

AMDM fails in this setting for two reasons: (1) In 2D, the concentration-of-measure phenomenon that underlies the spherical approximation does not hold—samples do not concentrate on a thin shell,

so Slerp has no geometric justification. (2) The target distribution is multi-modal; samples drawn from different modes of p_1 and p_2 are interpolated by Slerp toward the interior, landing in low-density regions rather than at the product modes. In contrast, Du et al. directly combines the score functions without relying on geometric structure, and correctly recovers the four-mode product distribution.

As shown in Table 7 and Figure 9, AMDM is substantially inferior to Du et al. on all metrics. Notably, in the fine-grained aggregation setting (Table 6), the relationship is reversed: AMDM significantly outperforms Du et al. in both attribute control and interaction control. This complementary failure pattern confirms that AMDM and general compositional samplers address fundamentally different problems: AMDM is designed for aggregating compatible models within a shared high-dimensional diffusion ecosystem, not for composing arbitrary low-dimensional or multi-modal distributions.

Table 7: Comparison of AMDM and Du et al. (2023) under a 2D non-Gaussian mixture composition task.

Method	**ln(MMD)** ↓	**LL** ↑	**Var**
AMDM	-0.77	-19.53	19.16
Du et al. (2023)	**-2.27**	**-6.77**	**4.85**
Ground Truth	–	–	9.56

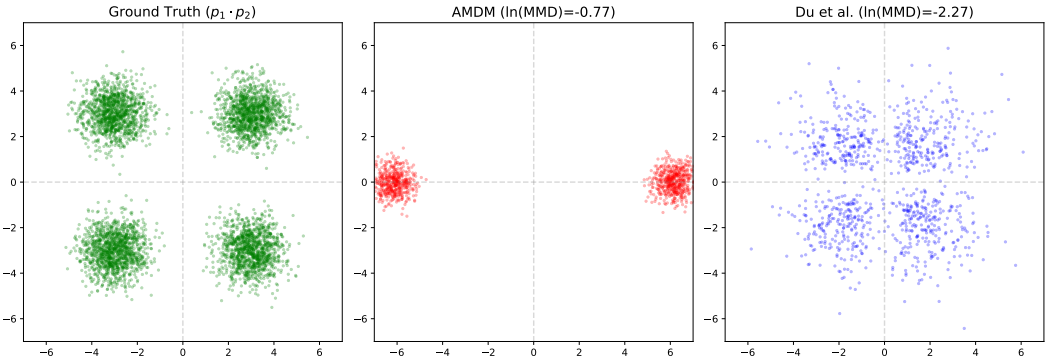


Figure 9: Visualization of the 2D mixture composition experiment. The target product distribution $p_1 \cdot p_2$ has four modes at $(\pm 3, \pm 3)$. AMDM collapses samples toward the center due to the failure of spherical interpolation in low dimensions with multi-modal distributions, while Du et al. correctly recovers the four-mode structure via score composition.

H Experimental Details

H.1 Evaluation Metrics

COCO-MIG Benchmark (Zhou et al., 2024) assesses the enhancement in attribute metrics. The COCO-MIG Benchmark is based on the COCO-position layout, where each instance is assigned a color attribute, requiring the generated instances to satisfy both position and color constraints. The process includes sampling layouts from COCO and categorizing layouts into five levels (L2-L6) based on the number of instances. Then, colors are assigned to each instance, global prompts are constructed, and a test file containing 800 entries is generated. The COCO-MIG metrics primarily include Instance Success Rate and mIoU Score. The Instance Success Rate measures the probability of each instance being generated correctly, while mIoU Score calculates the average of the maximum IoU for all instances; if the color attribute is incorrect, the IoU value is set to 0. Since the MIG-Benchmark does not contain interactive prompts, we set the "action" input in the InteractDiffusion model to "and".

FGAHOI (Ma et al., 2024) can measure interaction controllability. We use the FGAHOI model based on the Swin-Tiny architecture, as an HOI detector to evaluate the model's ability to control

interactions. HOI Detection Scores are categorized into two types: Default and Known Object. The Default setting is more challenging, as it requires distinguishing between irrelevant images.

CLIP (Hessel et al., 2021) is the CLIPScore of the generated images with captions of the image prompts.

H.2 Parameter Settings

InteractDiffusion and MIGC. The total sampling steps T are set to 50, the aggregation step s is set to 20, the weighting factor w is set to 0.5 and the optimization steps $\eta_t^{\theta_1}$ and $\eta_t^{\theta_2}$ are both set to 0.3. We use InteractDiffusion v1.0, and MIGC is modified to use the DDIM sampling method to align with the same diffusion process.

InteractDiffusion and IP-Adapter. The total sampling steps T are set to 10, the aggregation step s is set to 5, the weighting factor w is set to 0.5, ip scale is set to 0.8 and the optimization steps $\eta_t^{\theta_1}$ and $\eta_t^{\theta_3}$ are both set to 0.3. We utilized IP-Adapter based on SD1.5 while keeping InteractDiffusion unchanged.

InteractDiffusion, MIGC and IP-Adapter. The pretrained models for the three architectures remain consistent with those mentioned above. The total sampling step T is set to 10, an aggregation step s set to 5, and weight factors w_1 and w_2 set to 0.5 and 0.65. The optimization steps $\eta_t^{\theta_1}$, $\eta_t^{\theta_2}$, and $\eta_t^{\theta_3}$ are all simply set to 0.3, respectively.

H.3 Additional Generalization Studies

GLIGEN. To further examine the generalization of AMDM beyond the MIGC–InteractDiffusion pair, we conduct an additional experiment with GLIGEN and InteractDiffusion. Both models are developed within the Stable Diffusion ecosystem but provide different control capabilities: GLIGEN focuses on grounded layout control, while InteractDiffusion provides stronger human-object interaction control. As shown in Table 8, aggregating InteractDiffusion into GLIGEN consistently improves HOI generation across all Default and Known Object settings, while also reducing FID from 24.60 to 20.80. Conversely, aggregating GLIGEN into InteractDiffusion mainly improves Rare interaction cases under both Default and Known Object settings, while preserving overall image quality. These results provide additional evidence that AMDM can transfer complementary control signals across different compatible models within the same diffusion ecosystem.

Table 8: Quantitative results of AMDM on GLIGEN and InteractDiffusion within the same Stable Diffusion ecosystem.

Method	**Default ↑**		**Known Object ↑**		**FID ↓**	**CLIP Score ↑**
	Full	Rare	Full	Rare		
GLIGEN	23.50	21.80	25.00	27.90	24.60	27.18
GLIGEN(+InteractDiffusion)	27.30	25.40	28.90	32.00	20.80	27.08
InteractDiffusion	29.53	23.02	30.99	24.93	18.69	26.91
InteractDiffusion(+GLIGEN)	29.68	27.90	31.25	32.80	18.52	27.12

SDXL. We further evaluate AMDM within the SDXL ecosystem using SDXL-based InteractDiffusion and IP-Adapter. This setting tests whether AMDM remains applicable when moving from the SD1.x ecosystem to another compatible diffusion ecosystem. As shown in Table 9, AMDM improves the style CLIP score from 0.545 to 0.597, approaching the performance of IP-Adapter. This suggests that AMDM is not restricted to SD1.x models, as long as the aggregated models share a compatible latent space and diffusion process.

It is worth noting that AMDM operates on the intermediate latent variables $\mathbf{z}_t$ of the sampling process rather than on the parameterization of the denoising network. Therefore, the method is largely decoupled from whether the underlying model uses ϵ-prediction or v-prediction. As long as the latent state $\mathbf{z}_t$ at the same time step is accessible and the aggregated models follow a compatible diffusion process, AMDM can be applied without modifying model weights or training objectives. This property supports the portability of AMDM across compatible diffusion ecosystems and samplers.

Table 9: Style CLIP scores for InteractDiffusion(+IP-Adapter) based on SDXL.

Method	Style CLIP ↑
InteractDiffusion	0.545
InteractDiffusion(+IP-Adapter)	0.597
IP-Adapter	0.613

Cross-Ecosystem. Finally, we evaluate AMDM under a cross-ecosystem setting by combining InteractDiffusion based on SD1.4 with MIGC using EDM sampling. Compared with DDPM/DDIM sampling, EDM adopts a different noise schedule and therefore induces a different diffusion trajectory. In this case, the two models no longer share the same diffusion process, and the intermediate latent variables are not expected to lie on the same time-dependent data manifold $\mathcal{M}_t$. As a result, the latent aggregation operation is no longer well aligned with the compatibility analysis in Section 2.2.

As shown in Table 10, AMDM fails in this setting, producing substantially worse attribute control and image-text alignment. This failure case supports the central assumption of AMDM: the method is not a general-purpose distribution composition algorithm, but a latent aggregation method for compatible diffusion models within the same ecosystem. In other words, the shared latent space and shared diffusion process are not merely implementation conveniences, but necessary conditions for meaningful aggregation.

This requirement is different from, and in some sense weaker than, the constraints imposed by parameter-space model merging methods, which often require identical denoising network architectures. AMDM does not require the denoising networks to be identical, since aggregation is performed on intermediate latent variables rather than model parameters. Thus, many models that are developed from the same pretrained diffusion backbone but use different control modules can still be compatible with AMDM. This makes the same-ecosystem assumption relatively easy to satisfy in practice, while the cross-ecosystem experiment clarifies the boundary where AMDM should not be applied.

Table 10: Quantitative results in different diffusion model ecosystems on the COCO-MIG benchmark and CLIP Score in InteractDiffusion(+MIGC).

Method	Instance Success Rate (%) ↑						mIoU Score (%) ↑						CLIP Score ↑	
	L_2	L_3	L_4	L_5	L_6	Avg	L_2	L_3	L_4	L_5	L_6	Avg	Global	Local
InteractDiffusion	37.50	35.62	35.31	30.62	34.16	34.06	32.98	31.63	30.82	28.29	30.40	30.40	31.09	27.56
InteractDiffusion(+MIGC)	20.03	18.67	18.45	15.62	16.29	18.13	17.64	16.38	15.60	13.25	14.98	15.03	14.77	13.91
MIGC	67.70	59.61	58.09	56.16	56.88	58.43	59.39	52.73	51.45	49.52	49.89	51.48	33.01	28.95

H.4 Sensitivity to Aggregation Weights

We analyze the sensitivity of AMDM to the aggregation weight w in the InteractDiffusion(+MIGC) setting. The weight w is user-controlled and reflects the relative emphasis assigned to different models. As shown in Table 11, increasing w generally improves the attribute-related metrics, including the average instance success rate and mIoU score, since the generation is increasingly biased toward MIGC. Meanwhile, the interaction-related metrics remain stable when w is in a moderate range, but deteriorate when w approaches the endpoint corresponding to MIGC. This indicates that AMDM is not highly sensitive to w around the balanced region, while extreme weights naturally reduce the contribution of one model and weaken its corresponding control capability.

The results suggest that w provides an intuitive trade-off between the two models. When w increases, AMDM incorporates more attribute-control capability from MIGC, leading to higher ISR and mIoU scores. However, when w becomes too large, the interaction-control capability inherited from InteractDiffusion is weakened, as reflected by the drop in HOI detection scores. In practice, we find that balanced values such as $w = 0.4$ or $w = 0.5$ provide a favorable trade-off between attribute control and interaction control.

Table 11: Sensitivity analysis of the aggregation weight w in InteractDiffusion(+MIGC). The results show that AMDM remains stable around the balanced region, while extreme weights bias the generation toward one model and weaken the other control capability.

w	ISR (avg) ↑	mIoU Score (avg) ↑	Default (Full) ↑	Known Object (Full) ↑
0.0 (InteractDiffusion)	34.06	30.40	29.53	30.99
0.2	41.67	38.42	31.18	32.54
0.4	54.22	47.36	31.47	32.81
0.5	54.78	47.74	31.40	32.76
0.7	55.03	47.92	29.44	30.12
0.8	55.19	48.11	22.39	24.73
1.0 (MIGC)	55.46	48.53	16.87	17.84

This behavior can be understood from the spherical interpolation coefficients. For two variables $z_{t-1}^{\theta_1}$ and $z_{t-1}^{\theta_2}$ with angle φ, AMDM computes

$$z'_{t-1} = \frac{\sin((1-w)\varphi)}{\sin(\varphi)} z_{t-1}^{\theta_1} + \frac{\sin(w\varphi)}{\sin(\varphi)} z_{t-1}^{\theta_2}. \tag{60}$$

Therefore, the effective coefficients of the two models are

$$a_1(w,\varphi) = \frac{\sin((1-w)\varphi)}{\sin(\varphi)}, \quad a_2(w,\varphi) = \frac{\sin(w\varphi)}{\sin(\varphi)}. \tag{61}$$

When $w = 0$, we have $a_1(w,\varphi) = 1$ and $a_2(w,\varphi) = 0$, so the aggregation reduces to the first model, i.e., InteractDiffusion in this setting. When $w = 1$, we have $a_1(w,\varphi) = 0$ and $a_2(w,\varphi) = 1$, so the aggregation reduces to the second model, i.e., MIGC. More generally, when w approaches 0, $a_2(w,\varphi)$ approaches 0, and the aggregated variable is dominated by InteractDiffusion. When w approaches 1, $a_1(w,\varphi)$ approaches 0, and the aggregated variable is dominated by MIGC. Therefore, extreme values of w naturally suppress one model and weaken its corresponding control capability.

This coefficient behavior is consistent with the quantitative results in Table 11. When w increases from 0 to 0.5, the contribution of MIGC increases, and the attribute-related metrics improve substantially. At the same time, the contribution of InteractDiffusion remains sufficiently large, so the interaction-related metrics remain stable. However, when w further increases to 0.8 or 1.0, the contribution of InteractDiffusion becomes too small, and the interaction-control metrics drop sharply. This shows that AMDM is robust around the balanced region, but naturally degenerates toward a single-model sampler near the endpoints.

It is also worth noting that the Slerp coefficients are not identical to linear interpolation coefficients. For small angular deviations, using $\sin x \approx x$, we have

$$a_1(w,\varphi) \approx 1 - w, \quad a_2(w,\varphi) \approx w, \tag{62}$$

so Slerp locally behaves similarly to linear interpolation. However, when the angular deviation is non-negligible, the sine-based interpolation follows the spherical geometry rather than moving directly through the interior of the latent sphere. This helps the aggregated variable remain close to the local spherical manifold while incorporating complementary control information from different models, which is consistent with the geometric motivation of AMDM.

H.5 Additional Evaluation of Multi-Model Aggregation

A comprehensive benchmark for jointly evaluating instance-level attribute binding, human-object interaction, and style-related alignment is currently unavailable. Therefore, we provide separate evaluations for the three-model setting using existing benchmarks. Specifically, we report attribute-related results on MIG-Bench, interaction-related results using HOI metrics, and style-related alignment using Style CLIP. Style CLIP measures the CLIP similarity between the generated image and the style-related conditioning, and is used here as a style-related alignment metric rather than a full measure of reference-conditioned fidelity. MIG-Bench mainly measures localization and color grounding, and therefore only partially reflects the contribution of IP-Adapter, whose intended role is reference-conditioned style or appearance control.

Because MIG-Bench does not provide reference images, IP-Adapter cannot be evaluated under its intended reference-conditioned protocol. In this no-reference setting, IP-Adapter behaves more like a generic same-ecosystem SD branch and provides limited task-relevant information for localization, attribute grounding, and interaction control. Therefore, the MIG-Bench and HOI results should not be interpreted as a complete evaluation of the reference-conditioned capability of IP-Adapter or the full three-model capability of AMDM. Instead, they evaluate how AMDM behaves when an auxiliary model from the same diffusion ecosystem provides weakly task-relevant information under the available benchmark protocol.

Table 12 reports the results of different generation modes. Compared with InteractDiffusion, InteractDiffusion(+MIGC,+IP-Adapter) improves the instance success rate from 31.56% to 50.78% and the mIoU from 28.69% to 43.24%, while increasing the Style CLIP score from 0.533 to 0.574. At the same time, it preserves strong interaction performance, achieving 30.81 on HOI Default Full and 31.63 on HOI Known Object Full, comparable to or slightly better than the original InteractDiffusion scores of 29.10 and 30.46. These results show that the three-model aggregation can improve attribute grounding and style-related alignment while maintaining the interaction ability of the target model.

The three-model result is slightly lower than InteractDiffusion(+MIGC) on MIG-Bench, but the gap is moderate. This is expected because MIGC directly provides localization and instance-level attribute control, whereas IP-Adapter is not activated by reference images in this benchmark and contributes little to the measured MIG-Bench objective. Nevertheless, adding IP-Adapter substantially improves Style CLIP, from 0.536 for InteractDiffusion(+MIGC) to 0.574 for InteractDiffusion(+MIGC,+IP-Adapter), while preserving comparable MIG-Bench and HOI performance. This indicates that the IP-Adapter branch still contributes style-related information without severely disrupting the effective aggregation between InteractDiffusion and MIGC. These results also suggest that multi-model aggregation is not necessarily monotonic with the number of auxiliary models; its benefit depends on whether the added branch provides task-relevant complementary information under the evaluation protocol.

We further evaluate different anchor choices for multi-model aggregation. Using either InteractDiffusion or MIGC as the anchor yields strong overall performance, indicating that AMDM is robust when aggregating compatible models from the same diffusion ecosystem. The InteractDiffusion-anchor setting achieves slightly stronger interaction preservation, with 30.81 on HOI Default Full and 31.63 on HOI Known Object Full. The MIGC-anchor setting achieves slightly stronger attribute grounding, improving ISR from 50.78% to 51.86% and mIoU from 43.24% to 44.92%, while still maintaining competitive HOI performance. This behavior is consistent with the role of the anchor: the anchor tends to preserve the primary capability of the selected target model, while auxiliary models provide complementary control signals.

By contrast, using IP-Adapter as the anchor leads to lower MIG-Bench and HOI scores. This does not indicate that IP-Adapter is incompatible with AMDM. Rather, under the no-reference protocol, IP-Adapter loses its intended reference-conditioned input and behaves closer to a generic SD-like branch with weak task-relevant control for the evaluated attribute and interaction objectives. Consequently, it is less suitable as the anchor when the evaluation emphasizes localization, attribute binding, and human-object interaction. However, the fact that both the InteractDiffusion-anchor and MIGC-anchor settings remain strong after incorporating this weakly conditioned SD-like branch further suggests that AMDM can tolerate and aggregate such generic same-ecosystem branches without disrupting the effective aggregation between InteractDiffusion and MIGC.

Overall, the results suggest that AMDM remains effective in the three-model setting. The mixed model benefits from the MIGC branch for attribute grounding and from the IP-Adapter branch for style-related alignment, while preserving the interaction capability provided by InteractDiffusion. The anchor study further shows that different compatible anchors lead to predictable trade-offs rather than unstable degradation: a MIGC anchor slightly favors attribute grounding, an InteractDiffusion anchor slightly favors interaction preservation, and an IP-Adapter anchor is less suitable under the no-reference evaluation protocol. Importantly, since IP-Adapter behaves as a weakly conditioned SD-like branch in this setting, the strong performance of the InteractDiffusion-anchor and MIGC-anchor variants also provides additional evidence that AMDM can aggregate with generic same-ecosystem SD-based branches when the anchor and auxiliary controls are compatible. The observed differences mainly reflect the roles of the aggregated models and the limitations of current separate benchmarks, rather than an intrinsic failure of multi-model aggregation. A fairer evaluation of this setting

would require a benchmark that simultaneously measures instance-level attribute binding, interaction correctness, and reference-conditioned style preservation.

Table 12: Additional evaluation of multi-model aggregation using MIG-Bench, HOI, and Style CLIP metrics.

Method	MIG-Bench ↑		HOI Default ↑		HOI Known Object ↑		Style CLIP ↑
	ISR Avg.	mIoU Avg.	Full	Rare	Full	Rare	
IP-Adapter (no ref.)	2.28	4.20	0.68	0.12	0.64	0.10	0.588
MIGC	53.41	46.33	16.11	17.63	17.15	17.97	0.527
InteractDiffusion	31.56	28.69	29.10	22.63	30.46	24.63	0.533
InteractDiffusion(+MIGC)	52.19	45.66	31.20	24.19	32.16	25.98	0.536
InteractDiffusion(+MIGC,+IP-Adapter)	50.78	43.24	30.81	23.97	31.63	24.87	0.574
InteractDiffusion(+MIGC,+IP-Adapter, MIGC anchor)	51.86	44.92	29.73	23.86	31.18	24.31	0.562
InteractDiffusion(+MIGC,+IP-Adapter, IP-Adapter anchor)	38.64	32.37	23.82	19.41	23.76	21.28	0.553

H.6 Generalization to FM/RF Models with DiT Architectures

We further evaluate AMDM with three control methods built on FLUX. Regional Prompting (Chen et al., 2024a) controls regional layouts, CreatiLayout (Zhang et al., 2025) controls instance layouts and attributes, and OneHOI (Hoe et al., 2026) controls human-object interactions. We aggregate CreatiLayout into OneHOI to improve attribute control, and OneHOI into Regional Prompting to improve interaction control.

Settings. Within each benchmark, methods use the same prompts and seeds, with $T = 28$ sampling steps and $s = 10$ aggregation steps. We evaluate attributes and localization on COCO-MIG (also called MIG-Bench) using Instance Success Rate and mIoU Score, and interaction control on 1,000 HOI Bench samples using HOI Detection Scores from FGAHOI under the Default and Known Object settings. The target model completes the remaining sampling steps after aggregation.

Table 13: Attribute and interaction control with methods built on FLUX. All scores are reported as percentages; higher is better.

COCO-MIG Method	Instance Success Rate	mIoU Score
OneHOI	35.37	38.71
OneHOI(+CreatiLayout)	**59.76**	**54.46**
HOI Bench **Method**	**Default**	**Known Object**
Regional Prompting	4.57	5.32
Regional Prompting(+OneHOI)	**22.91**	**25.05**

Results. Table 13 shows that AMDM improves OneHOI's Instance Success Rate by 24.39 percentage points and mIoU Score by 15.75 percentage points. For Regional Prompting, the Default and Known Object scores improve by 18.34 and 19.73 percentage points, respectively. Figures 10 and 11 show the corresponding changes in object colors, instance separation, and interactions.

Runtime and memory. We measure efficiency with a fixed prompt containing three instances, using one warmup run followed by five repeated runs. Table 14 reports the time to generate the target image and the sum of peak memory across participating GPUs. OneHOI(+CreatiLayout) takes 38.69 seconds, close to OneHOI alone at 38.25 seconds, but uses three GPUs and 84.08 GiB in total. Adding OneHOI to Regional Prompting increases both time and total GPU memory.

Table 14: Efficiency of the experiments with FLUX. Time is mean ± standard deviation over five runs. Memory sums the peak usage on each participating GPU.

Method	Time (s)	Memory (GiB)	GPUs
OneHOI	38.25 ± 0.10	33.64	1
OneHOI(+CreatiLayout)	38.69 ± 0.04	84.08	3
Regional Prompting	18.62 ± 0.01	32.52	1
Regional Prompting(+OneHOI)	30.31 ± 0.06	67.97	2

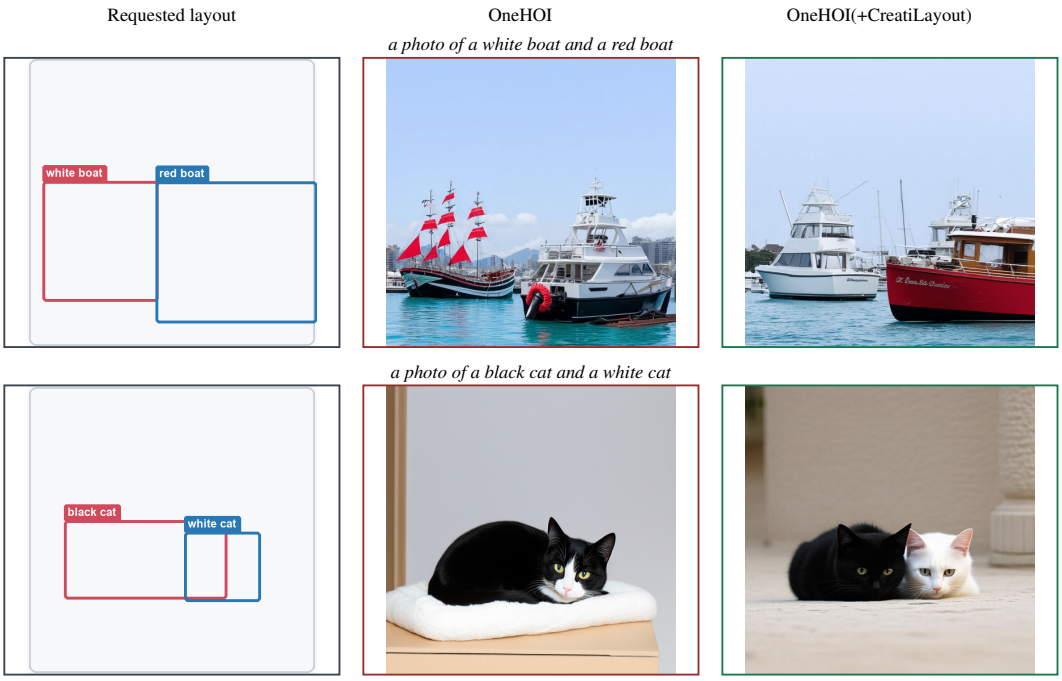


Figure 10: Attribute and layout control on COCO-MIG with methods built on FLUX. AMDM improves the requested color assignment for the boats and produces two distinct cats with the requested colors. Each row uses the same prompt and seed across methods.

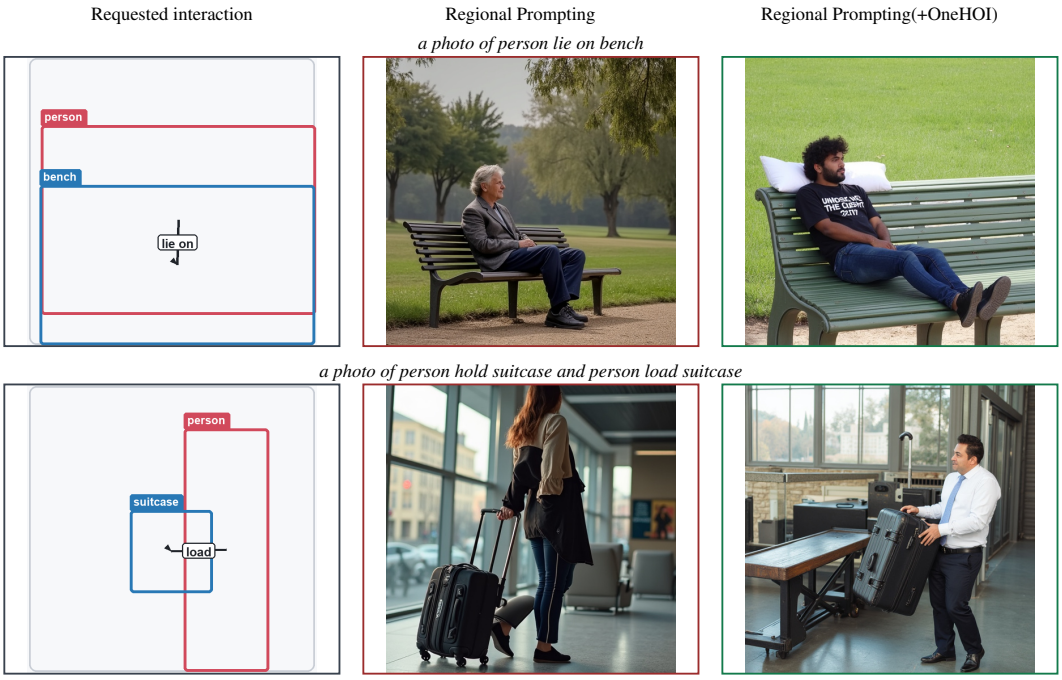


Figure 11: Interaction control with methods built on FLUX. Aggregating OneHOI into Regional Prompting changes an upright sitting pose to a reclined pose and changes pulling a suitcase to lifting it toward a raised surface.

H.7 Sampling with Four Steps and One Step

We next reduce the number of sampling steps for the same two model pairs. The setting with four sampling steps uses two aggregation steps followed by two steps from the target model alone. The setting with one sampling step performs aggregation once, with no subsequent sampling by the target model.

Table 15: AMDM with four sampling steps or one sampling step. Parentheses give changes from the corresponding target model at the same number of sampling steps, in percentage points.

COCO-MIG: OneHOI(+CreatiLayout)			
T	s	Instance Success Rate	mIoU
4	2	43.29 (+6.71)	44.01 (+4.63)
1	1	13.41 (−10.37)	20.78 (−8.75)
HOI Bench: Regional Prompting(+OneHOI)			
T	s	Default	
4	2	10.62 (+10.21)	
1	1	0.00 (−0.40)	

With four sampling steps, AMDM improves both tasks (Table 15), and Figure 12 shows better instance control and a person wielding a baseball bat. With one sampling step, all three reported scores fall below those of the corresponding target model; Figure 13 shows reduced instance separation on COCO-MIG and poor interaction generation from both methods on HOI Bench.

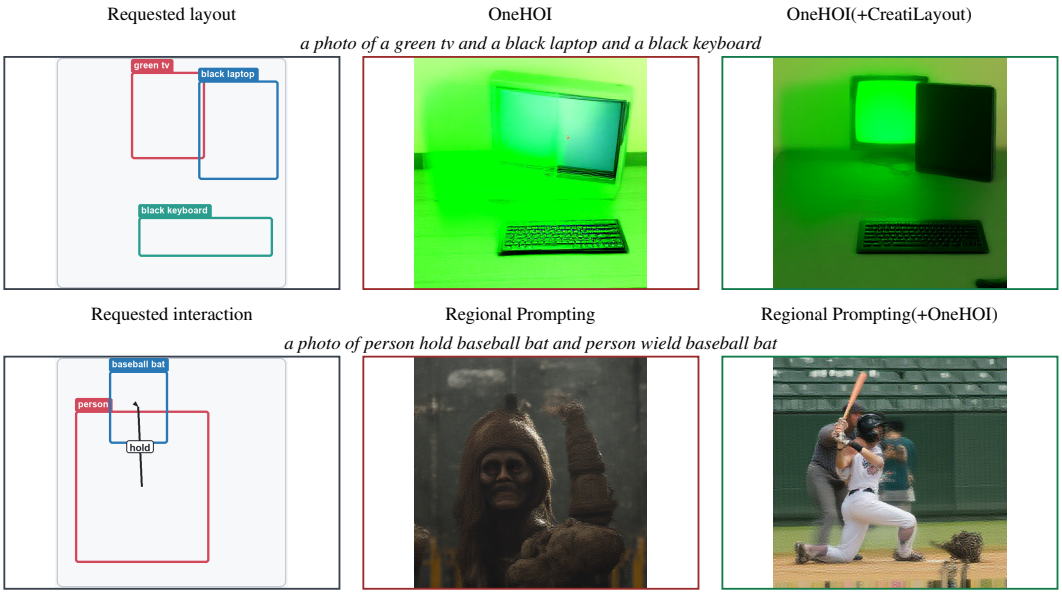


Figure 12: Examples with four sampling steps and two aggregation steps. In the first row, the instance evaluator marks two of the three requested instances as correct in the AMDM output, compared with none for the target model alone. In the second row, the AMDM output shows a person wielding a baseball bat, with visible color and shape artifacts.

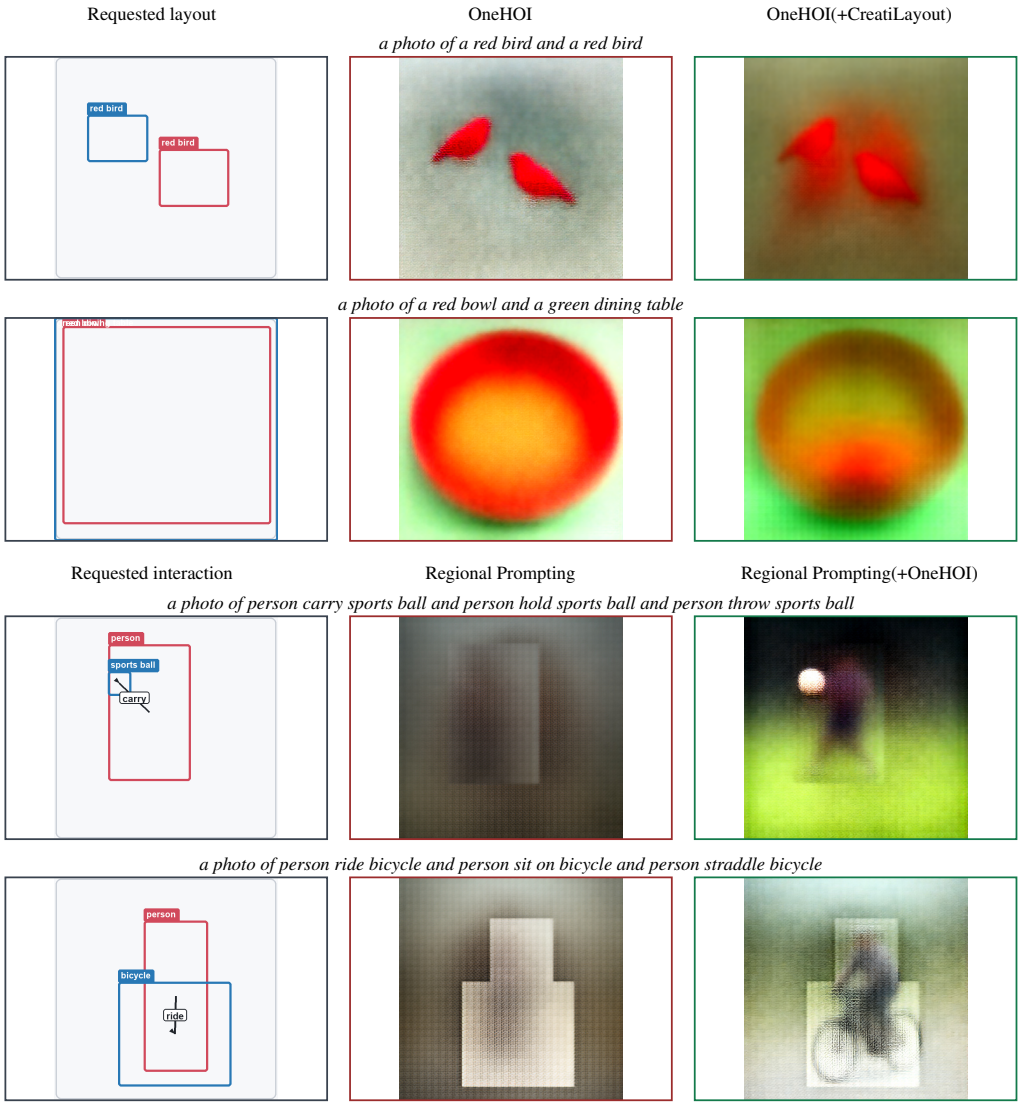


Figure 13: Failure cases with one sampling step. The first two rows show reduced instance separation and less distinct object boundaries after aggregation. The last two rows show that both Regional Prompting and AMDM fail to produce clear interactions.

H.8 Additional Baselines and Efficiency

We compare AMDM with two InteractDiffusion baselines: one uses 100 instead of 50 sampling steps, and the other applies BoxDiff (Xie et al., 2023) guidance for 20 of its 50 steps. All methods use the same prompts and seeds.

Table 16: Additional comparisons on COCO-MIG. The Control column counts BoxDiff guidance steps or AMDM aggregation steps within the sampling trajectory. ISR denotes Instance Success Rate. Scores are percentages.

Method	Sampling steps	Control steps	ISR ↑	mIoU Score ↑	Time (s) ↓
InteractDiffusion	50	0	34.06	30.40	11.35 ± 0.13
InteractDiffusion	100	0	35.98	28.85	14.77 ± 0.12
InteractDiffusion with BoxDiff	50	20	52.00	44.37	16.04 ± 5.35
InteractDiffusion(+MIGC)	50	20	**54.78**	**47.74**	13.52 ± 0.06

Increasing the target model's sampling steps produces a small increase in Instance Success Rate, from 34.06 to 35.98, while mIoU Score decreases. AMDM achieves 54.78 and 47.74, respectively, with lower measured latency than the target model using 100 sampling steps. BoxDiff also improves both control metrics, but AMDM obtains higher scores with lower latency in this comparison.

Table 17: Runtime and peak GPU memory for the SD experiments on a single RTX 3090. Time is mean $\pm$ standard deviation over five runs. Standalone models are included as references.

Method	Time (s) $\downarrow$	Memory (GiB) $\downarrow$
Projective Composition	17.33 ± 0.21	13.71
Feynman–Kac Correctors	65.96 ± 0.78	13.71
SUPERDIFF	33.03 ± 0.60	13.71
InteractDiffusion(+MIGC)	13.52 ± 0.06	13.71
InteractDiffusion	11.35 ± 0.13	7.44
MIGC	6.27 ± 0.04	8.86

Relative to InteractDiffusion alone, AMDM adds 2.17 seconds of latency and 6.27 GiB of peak memory. It is faster than the three compositional baselines in Table 17, with the same reported peak memory. AMDM evaluates the auxiliary branch only during the aggregation window, reuses the model predictions for deviation optimization, and then completes sampling with the target model alone.

H.9 Robustness Across Random Seeds

We test InteractDiffusion, MIGC, and InteractDiffusion(+MIGC) on 10 COCO-MIG prompts containing two to six instances. The three methods use the same five random seeds, giving 50 outputs per method. We use $T = 50$, $s = 20$, $w = 0.5$, and correction step size 0.3 for both branches.

Table 18: Mean and standard deviation across five random seeds on 10 COCO-MIG prompts. All scores are percentages; higher is better. Image Success measures whether all requested instances in an image pass the evaluator; Instance Success Rate measures the fraction of instances that pass.

Method	Image Success	Instance Success Rate	Localization	mIoU
InteractDiffusion	8.00 ± 4.47	33.00 ± 3.71	77.00 ± 3.26	27.59 ± 3.40
MIGC	6.00 ± 5.48	50.50 ± 8.18	78.00 ± 4.47	41.55 ± 5.95
InteractDiffusion(+MIGC)	14.00 ± 5.48	48.00 ± 7.16	82.50 ± 5.00	43.47 ± 6.29

AMDM improves InteractDiffusion's Instance Success Rate for all five seeds, with a paired gain of 15.00 ± 5.30 percentage points. It also achieves a higher mean mIoU Score than MIGC.

We also examine the 47 outputs for which MIGC does not satisfy all requested instance controls. Relative to InteractDiffusion, AMDM increases Instance Success Rate in 23 cases, leaves it unchanged in 22, and decreases it in two. Figures 14 and 15 show improved color and location assignments in some cases and degraded assignments in others.

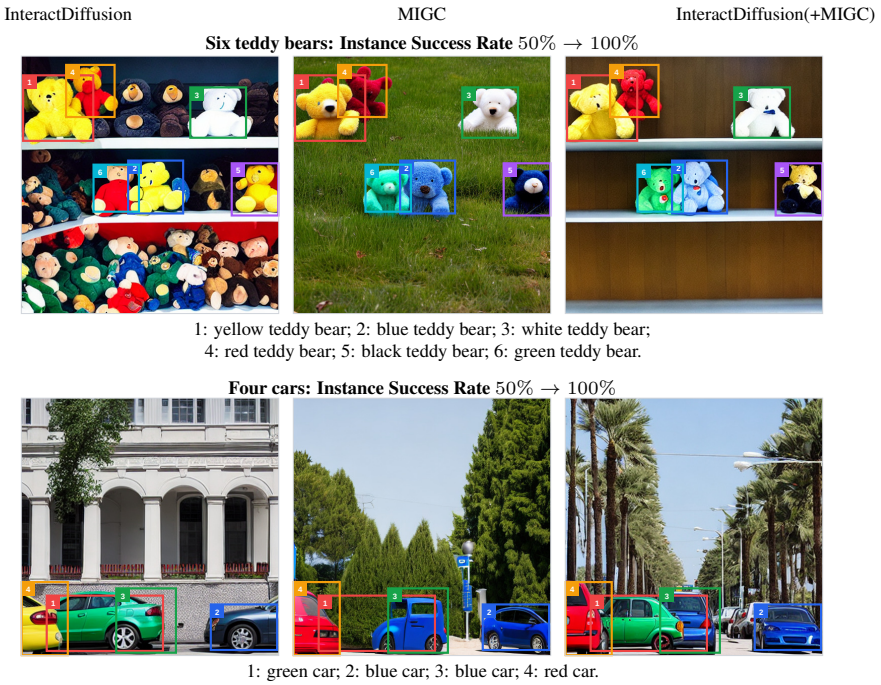


Figure 14: In two cases where MIGC misses requested controls, AMDM raises Instance Success Rate from 50% to 100% relative to InteractDiffusion. All three columns use the same ground truth layout boxes.



Figure 15: In two cases, AMDM lowers Instance Success Rate relative to InteractDiffusion, from 33% to 17% and from 50% to 33%. All columns use the same ground truth boxes, and rates are rounded to whole percentages.

H.10 PERSONALIZED MODEL COMPOSITION

Finally, we test AMDM on two personalized model combinations: Identity+Style and Two Identities. All LoRAs are derived from the same SD1.5 base model. For each setting, we use 10 prompts, two model pairs, and four shared random seeds, yielding 80 images per method with 50 sampling steps. AMDM aggregates the first 30 steps with correction step size 0.3. The aggregation weight is $w = 0.7$ for Identity+Style and $w = 0.5$ for Two Identities.

We compare AMDM with Multi-LoRA loading at weights $1.0 + 1.0$ and LoRA weight interpolation at weights $0.5 + 0.5$. We also report the scores of each individual LoRA. All methods run on a single RTX 4090.

Identity and style. We measure identity with FaceNet (Schroff et al., 2015) similarity and style with a CLIP style score. Table 19 shows that AMDM improves identity similarity over both composition baselines, reaching 3.41 compared with Multi-LoRA's 1.65. Its CLIP style score remains close to Multi-LoRA's, at 22.22 versus 22.72, and both scores exceed those of weight interpolation.

Table 19: Identity+Style composition. Individual LoRAs provide reference scores for each concept separately. Bold marks the best result among the three composition methods.

Method	Identity similarity ↑	CLIP style score ↑
Multi-LoRA ($1.0 + 1.0$)	1.65	**22.72**
LoRA weight interpolation ($0.5 + 0.5$)	0.68	19.35
AMDM	**3.41**	22.22
Identity LoRA (reference)	13.99	13.54
Style LoRA (reference)	1.10	23.91

Two identities. We detect faces in each generated image and match them one to one with the two identity references. Dual-ID similarity uses the lower of the two matched identity similarities and assigns zero when two faces are not detected. We also report the percentage of outputs containing two detected faces.

Table 20 shows that AMDM and weight interpolation both produce two detected faces in 55% of outputs, while AMDM achieves a higher Dual-ID similarity of 6.73 versus 6.55. Multi-LoRA at full strength produces no outputs with two detected faces in this evaluation.

Table 20: Two Identities composition. Dual-ID similarity penalizes missing faces by assigning zero. Two-Face Rate is a percentage. Individual Identity LoRAs are included as references.

Method	Dual-ID similarity ↑	Two-Face Rate ↑
Multi-LoRA ($1.0 + 1.0$)	0.00	0.00
LoRA weight interpolation ($0.5 + 0.5$)	6.55	**55.00**
AMDM	**6.73**	**55.00**
Identity-A LoRA (reference)	7.42	62.50
Identity-B LoRA (reference)	6.95	47.50

Efficiency. Multi-LoRA and weight interpolation require approximately 2.91 to 2.93 seconds per image and 2.63 GiB of peak memory. AMDM requires approximately 3.86 seconds and 4.65 GiB. Maintaining two denoising branches during the aggregation window therefore adds latency and memory in this setting.

I VISUAL RESULTS OF FIGURE 5A

To better illustrate the different roles of sampling stages verified in the model aggregation setting, we further visualize the experiment where $s = 5$.

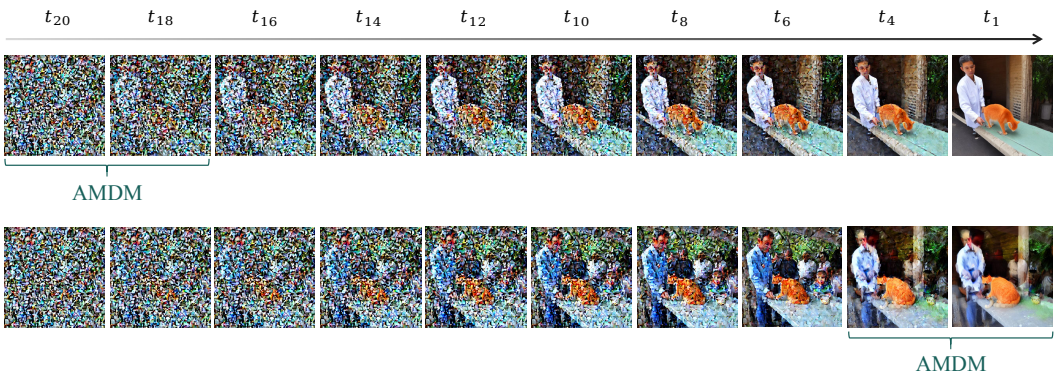


Figure 16: Different stages of the AMDM when aggregation steps are $s = 5$.

From Figure 16, it becomes even clearer that when aggregation occurs in the later stages, especially once the diffusion model begins focusing on detail generation, applying the aggregation algorithm at that point induces manifold deviation, causing a sharp deterioration in generation quality.

This further verifies in the model aggregation setting that early stages are more suitable for incorporating coarse control signals such as position, attributes, and style, while later stages predominantly govern quality and consistency.

J LIMITATIONS AND FUTURE WORK

We propose a novel aggregation algorithm AMDM. This algorithm is specifically designed for fine-grained generation tasks within the same diffusion model ecosystem. The theoretical assumption in this work, namely functional proximity, is primarily tailored to fine-grained tasks within the same diffusion model ecosystem, and this is empirically supported by our observations. These assumptions form the key distinction of AMDM from existing compositional diffusion approaches. AMDM does not provide global guarantees, and we leave the development of more relaxed theoretical foundations to future work. In addition, the selection of the hyperparameter η in the AMDM algorithm is largely heuristic. In future work, we will investigate dynamic adaptive selection strategies that are grounded in data-driven criteria or theoretical analysis to enhance the method's robustness and generalization.

K BROAD IMPACTS

AMDM may reduce the need to train new multi-condition generative models by reusing specialized diffusion models, which can lower data and compute requirements for fine-grained generation research. At the same time, stronger controllable image generation could be misused to create harmful or misleading visual content. We therefore advocate applying AMDM only with responsibly released models and appropriate safety filters, and against injecting harmful features into high-quality models.

L ADDITIONAL VISUAL RESULTS

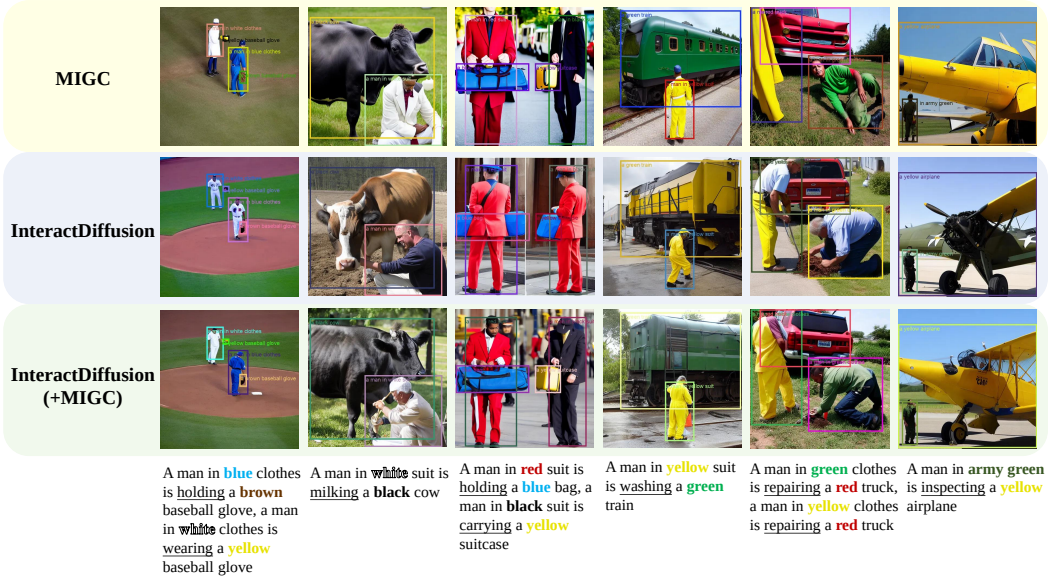


Figure 17: Additional visual results of aggregating MIGC into InteractDiffusion using the AMDM algorithm.



Figure 18: Additional visual results of aggregating IP-Adapter into InteractDiffusion using the AMDM algorithm.

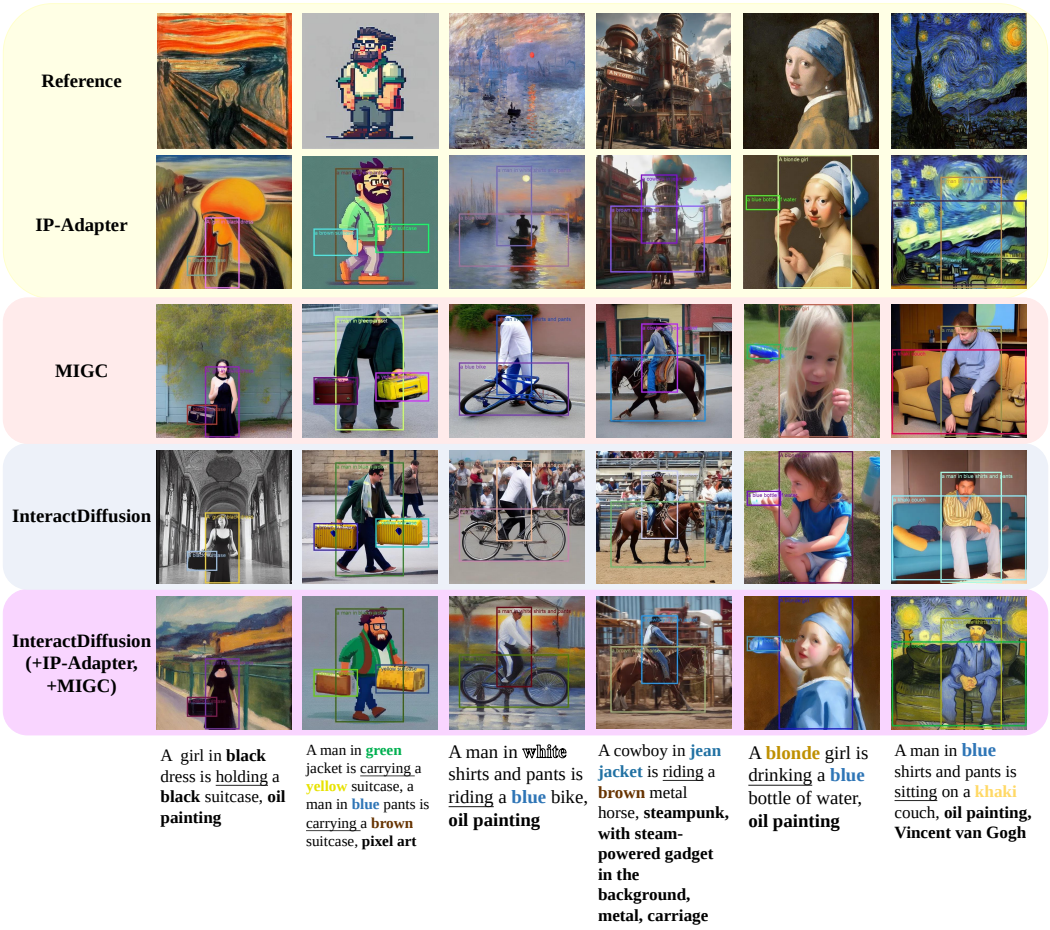


Figure 19: Additional visual results of aggregating MIGC and IP-Adapter into InteractDiffusion using the AMDM algorithm.