# UNDERSTANDING AND IMPROVING DIFFUSION TRAINING VIA CROSS-TIMESTEP TASK AFFINITY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Non-uniform training schemes have been widely used to accelerate diffusion model training; however, most existing approaches rely on static strategies and often exhibit unstable performance. Empirical evidence suggests that prior static methods are built on the assumption that timestep importance remains constant throughout training, an assumption that is inherently limited because it fails to capture the dynamically evolving interdependencies among denoising tasks during optimization. To address this limitation, we introduce task affinity to quantify the optimization influences among denoising tasks at different timesteps. By further relating task dependencies to the training optimization state, an affinity-aware adaptive timestep sampling strategy is derived. This strategy dynamically adjusts timestep preferences and yields robust and consistent acceleration across a wide range of model architectures, model scales, and datasets.

## 1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2020; Nichol & Dhariwal, 2021) have emerged as a foundational framework in generative modeling, but standard training pipelines rely on uniform timestep sampling, implicitly treating all noise levels as equally important. This assumption ignores substantial variations in denoising difficulty and optimization impact across timesteps, leading to inefficient allocation of training resources. To alleviate efficiency bottlenecks in diffusion model training, prior work has proposed a variety of timestep-based non-uniform training schemes (Choi et al., 2022; Hang et al., 2023; Karras et al., 2022; Wang et al., 2025; Zheng et al., 2024; 2025).

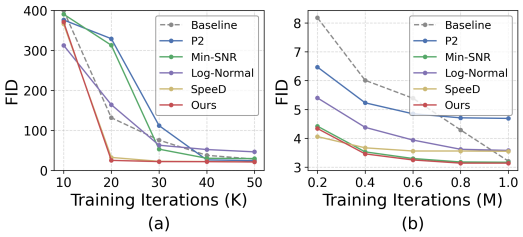



Figure 1: Performance comparison of non-uniform timestep training schemes across datasets and architectures: (a) results on MetFaces with DiT-XL/2, and (b) results on CIFAR-10 with a U-Net. The effectiveness of these methods varies substantially across settings and, in some cases, even falls below the baseline.

Notably, a common pattern emerges across existing methods: their training schemes are kept fixed throughout the entire training process. From an optimization standpoint, such static designs implicitly assume that the relative importance of timesteps remains unchanged during training. Yet, despite this shared assumption, prior methods exhibit markedly different timestep preferences, with no clear consensus on which timesteps are truly critical. This divergence calls into question the validity of assuming fixed timestep importance. To further probe this issue, two experimental settings are examined, as shown in Figure 1, where existing methods consistently display unstable acceleration behavior. Together, these observations suggest that prescribing a fixed set of critical timesteps is unlikely to provide an optimal solution for diffusion training.

A more plausible hypothesis is that timesteps associated with higher optimization gains are not static, but instead evolve dynamically with the experimental configuration, model state, and stage of training. Consequently, the instability observed in the acceleration performance of existing static strategies can be attributed to their inability to track such evolving dynamics. Related work that studies diffusion denoising training from a multi-task learning perspective has reported similar behaviors (Go et al.,



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.

2023; Hang et al., 2023; Kim et al., 2025), revealing important interactions among the optimization of different denoising tasks. However, how these interactions evolve during training and how they affect overall convergence efficiency remain insufficiently studied. Therefore, we believe that investigating and designing acceleration methods from a multi-task learning perspective is a promising direction.

To disentangle the complex effects induced by joint optimization, under a multitask optimization objective, we group the overall tasks into task groups and quantify the optimization gain relationships between pairs of task groups, which we term inter-group task affinity. This metric enables the identification of favorable inter-task optimization dependencies that emerge during training. We further aggregate the inter-group task affinity of each task group to derive its group-to-context task affinity, which characterizes the influence of a given task group on the overall optimization process. Based on the above understanding of the two task affinity measures, a unified perspective is provided for explaining the following two key phenomena: (1) existing acceleration methods achieve speedup by implicitly exploiting such optimization-related dependencies in different ways; and (2) static acceleration strategies often exhibit performance instability, as they fail to adapt to the dynamically evolving structure of these optimization dependencies throughout training.

Building on these insights, we propose an affinity-aware adaptive timestep sampling strategy that dynamically adjusts timestep preferences in accordance with the evolving group-to-context task affinity. This adaptive mechanism enables each task group to contribute effectively throughout training, thereby facilitating the optimization of the overall multitask objective. Our key contributions can be summarized as follows:

1. We arrive at a key conclusion: in diffusion training, the optimization influence of different denoising tasks evolves over the course of training and varies across experimental settings.
2. By introducing task affinity, we reveal the mechanism by which non-uniform timesteps affect training efficiency: acceleration arises from implicit alignment with multi-task optimization gains, while instability stems from static timestep strategies failing to adapt to evolving cross-task optimization relationships.
3. We introduce an affinity-aware adaptive timestep sampling strategy that robustly accelerates diffusion training by tracking evolving inter-task optimization dependencies.

## 2 Related Works

### 2.1 Non-uniform training of diffusion models

Diffusion models (Ho et al., 2020; Song et al., 2020) are a class of generative models that achieve high sample quality via learned denoising and exhibit strong transferability to a wide range of downstream image synthesis tasks (Dhariwal & Nichol, 2021; Podell et al., 2023; Zhang et al., 2025) and large-scale applications (Ho et al., 2022; Blattmann et al., 2023; Chong et al., 2024; Wu et al., 2025). However, diffusion model training faces severe computational resource bottlenecks.

Existing acceleration methods address this challenge through non-uniform training schemes, which can be categorized into three groups:**(1) Avoid over-emphasizing low-noise timesteps.** P2 (Choi et al., 2022) and Min-SNR (Hang et al., 2023) downweight very low-noise timesteps to emphasize perceptually meaningful features. Similar preferences are adopted by curriculum-based methods such as CLTS (Xu et al., 2024) and Diff-CL (Kim et al., 2024). **(2) Emphasize low-noise timesteps.** B-TTDM (Zheng et al., 2024) and BB-TDM (Zheng et al., 2025) observe that the most pronounced distributional changes in the forward diffusion process occur at early timesteps and therefore emphasize high-noise stages to better capture global structural information. **(3) Prioritize intermediate-noise timesteps.** Log-Normal (Karras et al., 2022) samples timesteps from a log-normal distribution, while SpeeD (Wang et al., 2025) prioritizes timesteps with rapidly changing process dynamics to improve training gains per unit computational cost. Both emphasize the importance of intermediate-noise regimes for training efficiency. Although these methods can effectively accelerate diffusion model training, the mechanisms underlying their effectiveness remain unclear. In particular, recent adaptive methods often rely on predefined adaptation rules (Xu et al., 2024; Kim et al., 2024) or additional black-box estimators(Kim et al., 2025) to determine timestep importance, without explicitly analyzing why such adaptations lead to faster optimization. This marks a key distinction from our work, which aims to uncover the underlying optimization mechanism and derive timestep adaptation from it.

### 2.2 MULTI-TASK LEARNING

Multi-task learning (MTL) (Crawshaw, 2020; Javaloy & Valera, 2021; Wang et al., 2020) trains a single model to perform multiple tasks through parameter sharing, which can reduce model complexity but may also induce negative transfer due to task conflicts. Diffusion training can be viewed as multi-task learning across timesteps; however, existing analyses fail to elucidate how negative transfer arising from different training strategies manifests during training and relates to final sample quality.

From a gradient-based perspective (Go et al., 2023), negative transfer has primarily been attributed to two factors: gradient conflict and gradient magnitude imbalance. Gradient conflict arises when gradients from different tasks interfere or cancel each other, reducing the effectiveness of parameter updates. Moreover, imbalanced gradient magnitudes may cause certain tasks to dominate optimization, suppressing others. However, these explanations are insufficient to account for acceleration strategies that are not explicitly designed at the gradient level.

From a loss-based perspective (Hang et al., 2023; Kim et al., 2025), the large number of denoising tasks and imbalanced loss scales are key contributors to instability and inefficiency in multi-task optimization during diffusion training. To mitigate these issues, several reweighting strategies (Choi et al., 2022; Hang et al., 2023; Wang et al., 2025) have been proposed to balance task losses across timesteps; however, the robustness of such heuristic approaches has been shown to be limited. Moreover, the lack of efficient metrics for monitoring negative transfer among tasks during training further limits their use in guiding the design of acceleration strategies.

## 3 METHOD

From a multi-task learning perspective, analyzing the efficiency of diffusion training centers on assessing the optimization interactions among denoising tasks at different noise levels. To this end, we introduce task affinity measures to characterize optimization dependencies under a multitask objective, revealing three key insights regarding the effectiveness and instability of existing acceleration strategies. These insights further guide the design of an affinity-aware adaptive timestep sampling strategy, which dynamically aligns timestep preferences with the evolving structure of task dependencies during training.

### 3.1 PRELIMINARIES

Diffusion models consist of two key processes: a predefined forward noising process that progressively corrupts clean data with Gaussian noise, and a learned reverse denoising process trained to invert this corruption.

**Forward Process.** Starting from data $x_0 \sim q(x_0)$, the forward diffusion process incrementally adds Gaussian noise, producing a sequence of latent variables $\{x_1, \dots, x_T\}$. Specifically, it follows a Markov Gaussian transition given by $q(x_t \mid x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\, x_{t-1},\, \beta_t I)$, where $\{\beta_t\}_{t=1}^T$ denotes a predefined noise schedule. Equivalently, by marginalizing the forward process, latent variables can be sampled in closed form as $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$, $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$.

**Reverse Process.** Diffusion models generate data by learning a reverse denoising process $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta(x_t, t),\, \sigma_t^2 I)$, which progressively removes noise from $x_T$ to recover $x_0$. Following (Ho et al., 2020), the mean is parameterized as $\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\, \epsilon_\theta(x_t, t)\Big)$, where $\epsilon_\theta$ denotes a neural network that predicts the injected noise given $(x_t, t)$.

**Training Objective.** The training of diffusion models can be derived by minimizing the variational lower bound on the data log-likelihood, which can be reformulated as a noise-prediction objective. In practice, the model is trained by minimizing a timestep-weighted mean-squared error loss

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0, \epsilon, t \sim p(t)}\Big[w(t)\, \big\|\epsilon_\theta(\alpha_t x_0 + \sigma_t \epsilon, t) - \epsilon\big\|_2^2\Big] \tag{1}$$

where $w(t)$ denotes the loss weight and $p(t)$ is typically chosen to be uniform.

### 3.2 TASK AFFINITY

**Definition 3.1 (Multitask Optimization Objective)** *Diffusion model training can be formulated as a multi-task learning problem over timesteps, since denoising sub-tasks at different noise levels share the same denoising network $\epsilon_\theta$. Let $\mathcal{T} = \{D_t\}_{t=0}^{T-1}$ denote the set of denoising sub-tasks, where $D_t$ corresponds to the denoising prediction task at timestep $t$ and is associated with a task-specific loss $L_t(\theta)$. The multi-task optimization objective is to jointly minimize a weighted combination of per-timestep losses*

$$\theta^* = \arg\min_\theta \sum_{t=0}^{T-1} L_t(\theta). \tag{2}$$

**Definition 3.2 (Task Grouping)** *To reduce the task scale, temporally adjacent denoising tasks are assigned to the same task group, as denoising tasks at neighboring timesteps are observed to exhibit highly similar gradient directions (Go et al., 2023) and optimization behaviors (Hang et al., 2023). The denoising tasks are evenly partitioned into $k$ task groups, each containing $n$ tasks. The $i$-th timestep task group is defined as:*

$$\mathcal{G}_i = \{\, D_t \mid t \in [(i-1)n,\ in-1)\,\}, \quad i = 1, \ldots, k. \tag{3}$$

Based on the above multi-task learning formulation, diffusion training efficiency is governed by the interaction effects among the optimization of different denoising tasks. To isolate and quantify these interactions from the coupled multi-task objective, a computationally efficient metric termed *task affinity* is introduced, which characterizes how optimization on a given set of timesteps influences the losses at other timesteps throughout training.

**Definition 3.3 (Inter-Group Task Affinity)** *Let $\mathcal{G}_i$ and $\mathcal{G}_j$ denote two denoising task groups. The inter-group task affinity from $\mathcal{G}_i$ to $\mathcal{G}_j$ at training iteration $t$ is defined as:*

$$\mathcal{A}^t_{\mathcal{G}_i \to \mathcal{G}_j} = 1 - \frac{L_{\mathcal{G}_j}\left(\theta^{t+1}_{\{\mathcal{G}_i, \mathcal{G}_j\}}\right)}{L_{\mathcal{G}_j}\left(\theta^{t+1}_{\{\mathcal{G}_j\}}\right)} \tag{4}$$

*where $\theta^{t+1}_{\{\mathcal{G}_i, \mathcal{G}_j\}}$ denotes the parameters after a joint update using samples from both $\mathcal{G}_i$ and $\mathcal{G}_j$, and $\theta^{t+1}_{\{\mathcal{G}_j\}}$ denotes the parameters after an update using only samples from $\mathcal{G}_j$. Here $L_{\mathcal{G}_j}(\theta)$ denotes the inter-group training loss for $\mathcal{G}_j$, evaluated under the two parameter settings. This formulation quantifies how incorporating $\mathcal{G}_i$ into the update affects the objective of $\mathcal{G}_j$. As it is pairwise, this characterization does not directly capture the aggregated interactions that arise when training over many groups ($k > 2$), motivating the group-to-context extension.*

**Definition 3.4 (Group-to-Context Task Affinity)** *Based on the inter-group formulation above, a more general setting encountered in practice is considered. Given a target group $\mathcal{G}_i$, its complement $\mathcal{B}_i = \bigcup_{k \neq i} \mathcal{G}_k$ is regarded as an aggregated task. Assuming that $\mathcal{B}_i$ corresponds to the task set selected by the current sampling strategy, the group-to-context task affinity from $\mathcal{G}_i$ to $\mathcal{B}_i$ measures the overall effect of optimizing $\mathcal{G}_i$ on $\mathcal{B}_i$ at training iteration $t$:*

$$\mathcal{A}^t_{\mathcal{G}_i \to \mathcal{B}_i} = 1 - \frac{L_{\mathcal{B}_i}\left(\theta^{t+1}_{\{\mathcal{G}_i, \mathcal{B}_i\}}\right)}{L_{\mathcal{B}_i}\left(\theta^{t+1}_{\{\mathcal{B}_i\}}\right)}. \tag{5}$$

Although the group-to-context affinity $\mathcal{A}^t_{\mathcal{G}_i \to \mathcal{B}_i}$ is directly computable by definition, it exhibits two practical limitations. First, estimating it during training typically requires maintaining additional parameter update trajectories and relies on large-batch computations to obtain stable estimates, which incurs substantial storage overhead. Second, as an aggregated measure, it only reflects the overall effect on the context and obscures fine-grained optimization dependencies among individual denoising task groups. For these reasons, we approximate it using the average of inter-group task affinities.

**Proposition 3.5** *At iteration $t$, the shared parameters are updated by a single gradient descent step with learning rate $\eta_t > 0$. The following assumptions are imposed:*

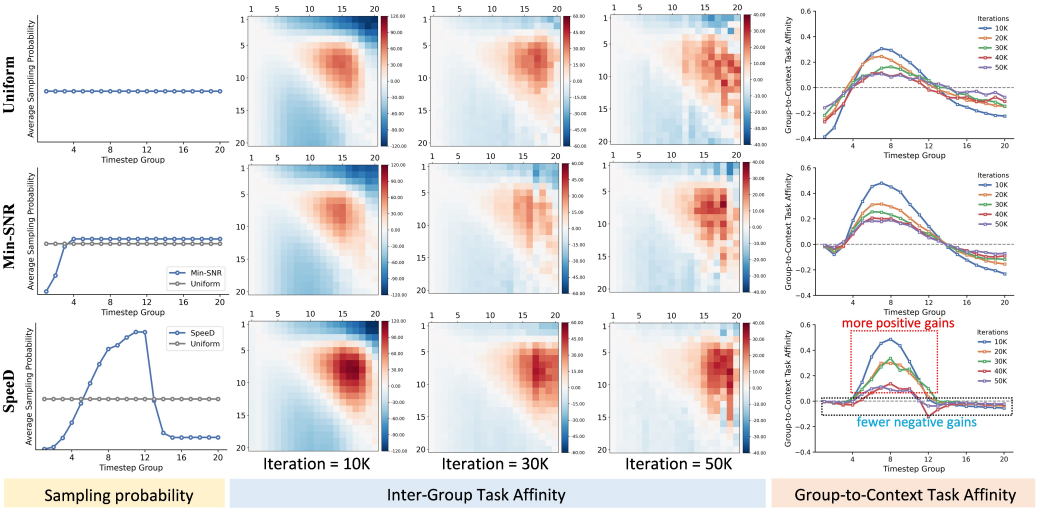



Figure 2: Denoising tasks are partitioned into $k$ noise-level groups ($k = 20$), where non-uniform training schemes such as SpeeD prioritize specific task intervals via higher sampling probabilities. Inter-group task affinity analysis reveals that the affinity structure evolves dynamically across training stages, where each entry $\mathcal{A}^t_{\mathcal{G}_i \to \mathcal{G}_j}$ at the $i$-th row and $j$-th column indicates the influence from group $\mathcal{G}_i$ to $\mathcal{G}_j$, with deeper red colors denoting stronger positive gains. These positive interactions form gain regions of favorable inter-task dependencies, and aligning the sampling distribution with them allows SpeeD to achieve higher group-to-context task affinity and the accelerated training behavior shown in Figure 1(a). For the weighting-based method Min-SNR, the weights assigned to each timestep are converted into sampling probabilities.

1. *For any denoising task group $\mathcal{G}_i \in \mathcal{T}$, the associated loss function $L_i(\theta)$ is differentiable and $l$-Lipschitz continuous with respect to the shared parameters $\theta$, implying that $\|\nabla_\theta L_i(\theta)\| \leq l$ for all $\theta$.*

2. *There exists a constant $C > 0$ such that $L_i(\theta) \geq C$ for all $\mathcal{G}_i \in \mathcal{T}$ and all $\theta$.*

*Then, the absolute difference between the inter-group task affinity and the group-to-context task affinity admits the following upper bound:*

$$\left| \mathcal{A}^t_{\mathcal{G}_i \to \mathcal{B}_i} - \frac{1}{|\mathcal{B}_i|} \sum_{\mathcal{G}_k \in \mathcal{B}_i} \mathcal{A}^t_{\mathcal{G}_i \to \mathcal{G}_k} \right| \leq \frac{\eta_t (1 + |\mathcal{B}_i|) l^2}{C}. \tag{6}$$

*Collectively, adopting a sufficiently small learning rate $\eta_t$, enforcing practical constraints on the group size $|\mathcal{B}_i|$, and employing techniques that guarantee a positive lower bound on the loss jointly ensure that the upper bound $\eta_t(1 + |\mathcal{B}_i|)l^2/C$ remains small. A detailed derivation is provided in Appendix A.*

Based on the above proposition, under a sampling-weighted aggregation of the training objective, the influence of group $\mathcal{G}_i$ on the overall optimization dynamics at iteration $t$ scales with its sampling probability $p^t_{(\mathcal{G}_i)}$

$$\widetilde{\mathcal{A}}^t_{\mathcal{G}_i \to \mathcal{B}_i} = \frac{1}{|\mathcal{B}_i|} p^t_{(\mathcal{G}_i)} \sum_{\mathcal{G}_k \in \mathcal{B}_i} \mathcal{A}^t_{\mathcal{G}_i \to \mathcal{G}_k}. \tag{7}$$

### 3.3 Insights

Building on the proposed evaluation metric, the optimization interactions among denoising tasks learned at different noise levels are quantified, which is central to analyzing diffusion training efficiency from a multi-task learning perspective.

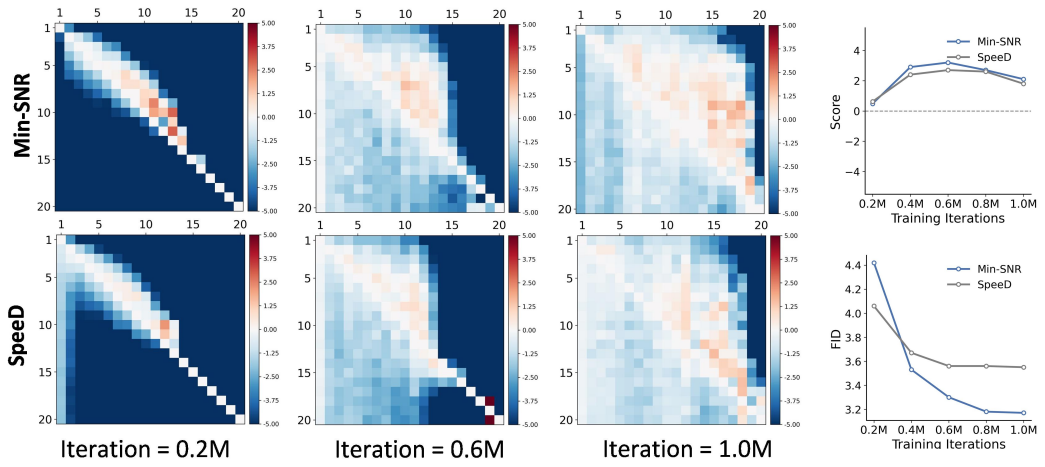



Figure 3: Across different experimental settings, gain regions consistently exhibit dynamic evolution during training, and stronger inter-group to context task affinity gains are positively associated with faster convergence.

**(Insight 1) Gain Regions During Training.** Based on inter-group task affinity, gain regions are defined to indicate the presence of positive interactions among cross-timestep task optimizations, reflecting favorable inter-group dependencies, as highlighted by the red regions in Figure 2. Such gain regions are consistently observed across different acceleration methods. From the perspective of multi-task cooperative optimization efficiency, if training at specific timesteps can continuously yield positive benefits for other timesteps, this property may constitute a key factor underlying the acceleration achieved by non-uniform training methods.

**(Insight 2) A Unified Explanation of Current Acceleration Methods.** Existing acceleration methods exploit gain regions more effectively during training by reshaping the timestep sampling distribution. Specifically, at each training stage, the sampling probabilities are used as weighting factors in the computation of group-to-context task affinity, thereby determining the contribution of each task group to the overall optimization dynamics, as illustrated in the last column of Figure. 2. Taking SpeeD as an example, compared to uniform sampling and Min-SNR, this method more effectively leverages gain regions and consistently accumulates greater positive gains throughout training. In addition, as training progresses, the optimization dynamics among tasks gradually decouple, and inter-task influences weaken, regardless of whether they are positive or negative. This trend becomes particularly evident in the later stages of training (e.g., after approximately 50K iterations), where the corresponding curves exhibit a clear flattening behavior.

**(Insight 3) Instability of Static Acceleration Strategies.** Fixed timestep sampling distributions struggle to track the gain regions that evolve throughout training. As shown in Figure 3, these gain regions are highly sensitive to experimental factors, including model architecture and dataset scale, and exhibit pronounced dynamics over the course of training. Under such conditions, maintaining alignment between static sampling strategies and the evolving gain regions becomes difficult, leading to unstable acceleration performance across different training stages and experimental settings. For example, in the experiments shown in Figure 3, the overall group-to-context task affinity at each training stage is obtained by aggregating affinities across all task groups. Based on this aggregated measure, SpeeD consistently exhibits lower overall group-to-context task affinity than Min-SNR at every training stage. Moreover, gains in the overall group-to-context task affinity show a positive correlation with convergence speed.

### 3.4 Affinity-Aware Adaptive Timestep Sampling

To consistently and stably exploit the latent gains arising from inter-task optimization during training, this paper proposes an affinity-aware adaptive timestep sampling strategy. Rather than relying on fixed, manually designed sampling schemes, the proposed strategy dynamically updates the sampling distribution to maximize the overall training benefit induced by inter-group interactions. Based on the

weighted group-to-context task affinity, the expected global gain under a given sampling distribution is formulated as the following optimization objective:

$$\max_{p} \sum_{i} p^{t}_{(\mathcal{G}_i)} \mathcal{A}^{t}_{\mathcal{G}_i \to \mathcal{B}_i} \tag{8}$$

At each iteration, the affinity score of group $\mathcal{G}_i$ is computed as $\omega_i = \sum_{\mathcal{G}_k \in \mathcal{B}_i} \mathcal{A}^{t}_{\mathcal{G}_i \to \mathcal{G}_k}$. Based on these scores, an exponentially weighted affinity scoring scheme is employed to derive the sampling probabilities, enabling a principled balance between exploitation of high-affinity groups and exploration of potentially beneficial ones

$$p^{t}_{(\mathcal{G}_i)} = \frac{\exp(\omega_i/\tau)}{\sum_{j=1}^{k} \exp(\omega_j/\tau)} \tag{9}$$

where $\tau > 0$ denotes a temperature hyperparameter. Larger values of $\tau$ produce a flatter probability distribution, thereby favoring exploration, whereas smaller values of $\tau$ yield a more concentrated distribution and place greater emphasis on exploitation. After computing the group-level sampling probability by Eq. equation 9, we distribute it uniformly across all timesteps within the group. This group-wise allocation is motivated by the similar optimization behavior of neighboring diffusion timesteps, providing a simple and effective adaptive granularity.

# 4 EXPERIMENT

**Datasets and Network Architectures.** The proposed method is evaluated on a diverse set of benchmarks that cover both unconditional and conditional image generation scenarios. For unconditional image synthesis, experiments are conducted on MetFaces (Karras et al., 2020) and FFHQ (Karras et al., 2019). Conditional image generation is evaluated on CelebAHQ (Karras et al., 2017), CIFAR-10 (Krizhevsky et al., 2009), and ImageNet-1K (Russakovsky et al., 2015). The proposed approach is implemented on two representative diffusion backbones: the convolutional U-Net architecture (Rombach et al., 2021) and the transformer-based DiT model (Peebles & Xie, 2023). We select widely used, open-source methods. Since adaptive methods have only recently been proposed, some related works remain unavailable or are evaluated only in limited settings, and are therefore not included; a more detailed discussion is provided in Appendix E.

**Evaluation protocols.** For each model, 50,000 samples are generated at inference time. Generation quality is assessed using the Fréchet Inception Distance (FID) (Heusel et al., 2017), a standard metric for measuring sample fidelity. To ensure a fair comparison of training efficiency, we normalize the total wall-clock training time of the uniform-sampling baseline to 1.0 and evaluate all methods at predefined fractions of this budget, as shown in Table 1. For instance, a normalized time of 0.2 denotes performance achieved using 20% of the baseline's total training time. For our method, the reported budget includes both standard model optimization and the additional overhead of task-affinity estimation.

## 4.1 COMPARISONS WITH OTHER STRATEGIES

As shown in Table 1, the acceleration performance of the same method varies across different experimental settings. In experiments on the CIFAR dataset, most acceleration strategies improve optimization efficiency during the early stages of training. However, as training approaches convergence, some methods fail to maintain this advantage, and their final performance can even fall below that of the baseline, including the state-of-the-art SpeeD method. A similar pattern is also observed on higher-resolution datasets, where performance gains achieved in the early or intermediate training stages tend to diminish as training progresses.

These observations are consistent with the earlier analysis of performance instability in existing acceleration methods. Static strategies rely on fixed weighting adjustments or resampling schemes, implicitly assuming that timestep importance remains stable throughout the entire training process. However, this assumption is inconsistent with practical optimization behavior: as model optimization progresses, the importance of timesteps changes, and static designs are unable to track these variations, thereby limiting their ability to provide sustained improvements.

Table 1: Comparison of FID↓ with baseline methods on two model architectures (U-Net and DiT) across three datasets. Wall-clock time is used for fair comparison, with the baseline as the reference. Here, 1.0 corresponds to 1.0M iterations on CIFAR 32×32, 50K iterations on MetFaces 256×256, and 100K iterations on FFHQ 256×256. The proposed method achieves the best performance on all three datasets. Bold entries indicate the best results.

| dataset | U-Net CIFAR 32×32 | | | | | DiT-XL/2 MetFaces 256×256 | | | | | DiT-XL/2 FFHQ 256×256 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| wall-clock time | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
| Baseline | 8.18 | 6.01 | 5.39 | 4.29 | 3.21 | 399.74 | 131.66 | 75.73 | 37.68 | 28.34 | 337.51 | 37.37 | 12.58 | 9.86 | 7.91 |
| P2 (Choi et al., 2022) | 6.47 | 5.23 | 4.85 | 4.71 | 4.69 | 376.15 | 329.27 | 112.04 | 26.31 | 24.36 | 360.06 | 35.22 | 14.78 | 11.85 | 8.13 |
| Min-SNR (Hang et al., 2023) | 4.42 | 3.53 | 3.30 | 3.18 | 3.17 | 390.47 | 312.94 | 53.06 | 30.34 | 29.58 | 333.17 | 33.78 | 11.09 | 8.47 | 6.29 |
| Log-Normal (Karras et al., 2022) | 5.40 | 4.38 | 3.94 | 3.62 | 3.58 | **312.83** | 164.18 | 62.95 | 52.14 | 46.37 | **167.03** | 52.24 | 38.36 | 21.54 | 15.94 |
| SpeeD (Wang et al., 2025) | **4.06** | 3.67 | 3.56 | 3.56 | 3.55 | 366.36 | 32.42 | 22.63 | 22.17 | 21.12 | 319.22 | 21.75 | 8.14 | 6.92 | 6.18 |
| **Ours** | 4.34 | **3.46** | **3.25** | **3.14** | **3.14** | 371.28 | **25.06** | **22.02** | **21.83** | **21.11** | 247.13 | **17.34** | **7.82** | **6.34** | **5.97** |

In contrast, the proposed method achieves stable acceleration across diverse experimental settings by employing a dynamic sampling strategy that adapts to the evolving inter-group optimization relationships among denoising tasks, as shown in Figure. 4. Although the proposed method is not the fastest in the early stages in some experiments, it exhibits more stable acceleration behavior overall.

Table 2: Comparison across different architectures on the CelebAHQ dataset evaluated by FID. For a more comprehensive evaluation, results reported in the original papers of methods marked with an asterisk are also included for reference.

| Method | Baseline | P2 | Min-SNR | SpeeD | B-TTDM* | **Ours** |
|---|---|---|---|---|---|---|
| DiT-B/2 | 8.87 | 8.32 | 7.51 | 6.94 | - | **6.85** |
| U-Net | 16.45 | 14.52 | 14.29 | 14.16 | 13.62 | **13.59** |

Table 3: Additional cross-architecture robustness evaluation between our method and baseline approaches, evaluated by FID.

| Method | Metfaces | | FFHQ | | ImageNet | |
|---|---|---|---|---|---|---|
| | DiT | U-Net | DiT | U-Net | DiT | U-Net |
| baseline | 28.34 | 44.12 | 7.91 | 9.36 | 27.31 | 46.54 |
| **Ours** | 21.11 | 23.41 | 5.97 | 8.74 | 21.77 | 37.36 |

## 4.2 Generalization Evaluation

**Cross-architecture robustness evaluation.** To evaluate robustness across different architectures, the proposed method is implemented on the DiT-XL/2 (Peebles & Xie, 2023) and LDM (Rombach et al., 2021) architectures and evaluated on the MetFaces, FFHQ, and ImageNet-1K datasets. In addition, it is implemented on the DiT-B/2 (Peebles & Xie, 2023) and ADM (Dhariwal & Nichol, 2021) architectures and evaluated on the CelebAHQ dataset. Following standard experimental protocols, models are trained for 50K iterations on MetFaces and FFHQ, 400K iterations on ImageNet-1K, and 100K iterations on CelebAHQ. As shown in Tables 2 and 3, the proposed method achieves strong performance across all evaluation settings. These results demonstrate its robustness and generalization across architectural choices.

Table 4: Comparison of different model scales under the same architecture (DiT-S/8, DiT-S/4, DiT-S/2), focusing on early-stage convergence speed measured by FID on ImageNet-1K.

| wall-clock time | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
|---|---|---|---|---|---|
| DiT-S/8 | 398.6 | 382.3 | 310.4 | 230.7 | 213.4 |
| **Ours** | 395.2 | 360.8 | 275.6 | 205.3 | 188.9 |
| DiT-S/4 | 290.7 | 245.1 | 205.8 | 178.3 | 165.0 |
| **Ours** | 285.4 | 228.9 | 185.2 | 158.7 | 142.6 |
| DiT-S/2 | 240.3 | 205.6 | 178.9 | 150.4 | 132.7 |
| **Ours** | 235.8 | 190.2 | 160.5 | 132.8 | 118.3 |

**Robustness evaluation across model scales.** To investigate the robustness of the proposed method across different model scales, experiments are conducted using DiT architectures with varying patch resolutions, including DiT-S/8, DiT-S/4, and DiT-S/2. Model performance is evaluated at multiple training stages, ranging from 10K to 50K iterations, with FID reported as the quantitative metric. The results are summarized in Table 4. Across all model scales and training iterations, the proposed method consistently outperforms the corresponding baselines, with gains observed throughout training, indicating that its effectiveness is not limited to a specific training stage or model capacity.

Table 5: FID comparison under different timestep schedules, evaluating the generalization ability of the proposed method across linear, quadratic, and cosine schedules.

| | CIFAR-10 | | | FFHQ | | |
|---|---|---|---|---|---|---|
| | linear | quadratic | cosine | linear | quadratic | cosine |
| baseline | 3.21 | 3.15 | 3.19 | 7.91 | 7.48 | 7.63 |
| **Ours** | 3.14 | 3.11 | 3.12 | 5.97 | 5.34 | 6.25 |

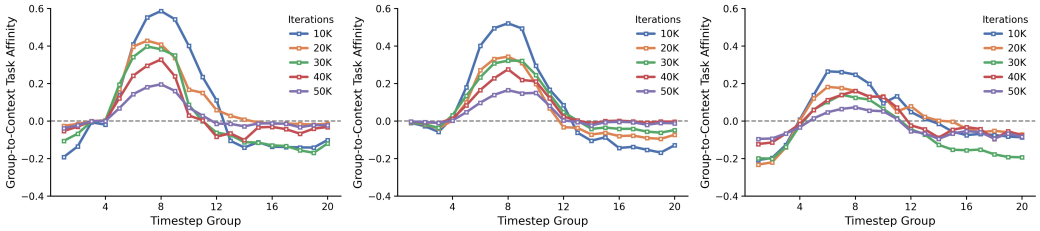



Figure 4: Hyperparameter analysis on the FFHQ dataset with different values of $\tau$. From left to right, $\tau = 2, 4$, and $8$.

Table 6: Analysis of the hyperparameter $\tau$. Unless otherwise specified, we set Groups = 20, Interval = 2000, and use the Original grouping strategy.

| $\tau$ | MetFaces | FFHQ | ImageNet |
|---|---|---|---|
| 2 | 23.78 | 6.32 | 23.18 |
| 4 | 21.11 | 5.97 | 21.77 |
| 8 | 26.53 | 7.15 | 25.09 |

Table 7: Comparison of the number of groups, refresh intervals, and grouping strategies on MetFaces and FFHQ.

| | | MetFaces 256×256 | | FFHQ 256×256 | |
|---|---|---|---|---|---|
| Groups | Interval | Original | SNR | Original | SNR |
| 4 | 2000 | 25.45 | 24.36 | 7.58 | 7.29 |
| 4 | 3000 | 27.82 | 27.44 | 7.72 | 7.65 |
| 10 | 2000 | 23.24 | 22.82 | 6.42 | 6.31 |
| 10 | 3000 | 23.73 | 23.17 | 6.94 | 6.57 |
| 20 | 2000 | <u>21.11</u> | **21.08** | <u>5.97</u> | **5.92** |
| 20 | 3000 | 21.25 | 21.19 | 6.25 | 6.08 |

**Cross-schedule robustness evaluation.** Within the diffusion framework, multiple timestep scheduling schemes are typically employed, such as linear (Ho et al., 2020), quadratic, and cosine (Nichol & Dhariwal, 2021) formulations. The robustness of the proposed approach is examined under these heterogeneous scheduling settings, using the FID as the evaluation criterion. As summarized in Table 5, consistent performance gains are observed across all considered schedules.

### 4.3 Hyperparameter Analysis

Hyperparameter analysis of $\tau$ is provided in Table 6. As the temperature of the sampling distribution, higher $\tau$ yields a flatter distribution and promotes exploration, while lower $\tau$ concentrates sampling and emphasizes exploitation. The best performance is obtained at $\tau = 4$. As shown in Figure 4, on FFHQ with DiT-XL/2, $\tau = 8$ produces a near-uniform distribution and $\tau = 2$ focuses on high-gain regions, whereas $\tau = 4$ balances the two and yields more stable optimization. Previous methods (Go et al., 2023; Esser et al., 2024) suggest that SNR-based grouping better reflects the diffusion process. We therefore adopt an SNR-based clustering strategy (Go et al., 2023) for timestep grouping. As shown in Table 7, SNR-based grouping consistently outperforms uniform grouping, which we attribute to its better alignment with denoising difficulty across noise levels, yielding more meaningful task separation. Performance degrades slightly with fewer groups, as reduced flexibility in inter-group sampling distribution scheduling limits adaptivity. The refresh interval has a minor impact, since training dynamics evolve smoothly under the optimization schedule, though excessively long intervals may fail to capture timely changes and weaken adaptation.

## 5 Conclusion

By formulating diffusion training as a dynamic multi-task optimization problem, we observe that training efficiency is influenced by complex optimization gains arising from interactions among different denoising tasks. By introducing task affinity as an effective measure of cross-task optimization dependencies, we provide a unified explanation for both the gains and the instability observed in prior non-uniform training schemes: acceleration is achieved when the sampling design aligns with favorable dependency structures, whereas performance degrades when static preferences drift out of alignment as training progresses. Building on this perspective, we propose an affinity-aware adaptive timestep sampling strategy that continuously updates timestep preferences to match the evolving model state, thereby enabling more stable and robust acceleration across diverse settings.

### AI USE STATEMENT

In this work, we used generative AI tools to assist with broad literature exploration and the identification of potentially relevant prior work, as well as to cross-check the correctness of selected mathematical derivations in the development of our method. All literature identified with AI assistance was independently verified against the original sources, and all AI-assisted derivations were manually reviewed and validated by the authors.

## REFERENCES


Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 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.

Zheng Chong, Xiao Dong, Haoxiang Li, Shiyue Zhang, Wenqing Zhang, Xujie Zhang, Hanqing Zhao, Dongmei Jiang, and Xiaodan Liang. Catvton: Concatenation is all you need for virtual try-on with diffusion models. *arXiv preprint arXiv:2407.15886*, 2024.

Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

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.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34: 27503–27516, 2021.

Everette S Gardner Jr. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1):1–28, 1985.

Hyojun Go, Yunsung Lee, Seunghyun Lee, Shinhyeok Oh, Hyeongdon Moon, and Seungtaek Choi. Addressing negative transfer in diffusion models. *Advances in Neural Information Processing Systems*, 36:27199–27222, 2023.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7441–7451, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. *arXiv preprint arXiv:2103.02631*, 2021.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.

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.

Jin-Young Kim, Hyojun Go, Soonwoo Kwon, and Hyun-Gyoon Kim. Denoising task difficulty-based curriculum for training diffusion models. *arXiv preprint arXiv:2403.10348*, 2024.

Myunsoo Kim, Donghyeon Ki, Seong-Woong Shim, and Byung-Jun Lee. Adaptive non-uniform timestep sampling for accelerating diffusion model training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2513–2522, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, et al. Pytorch distributed: Experiences on accelerating data parallel training. *arXiv preprint arXiv:2006.15704*, 2020.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

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. *arXiv preprint arXiv:2307.01952*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Kai Wang, Mingjia Shi, Yukun Zhou, Zekai Li, Zhihang Yuan, Yuzhang Shang, Xiaojiang Peng, Hanwang Zhang, and Yang You. A closer look at time steps is worthy of triple speed-up for diffusion model training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12934–12944, 2025.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *arXiv preprint arXiv:2010.05874*, 2020.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26024–26035, 2025.

Tianshuo Xu, Peng Mi, Ruilin Wang, and Yingcong Chen. Towards faster training of diffusion models: An inspiration of a consistency phenomenon. *arXiv preprint arXiv:2404.07946*, 2024.

Jinjin Zhang, Qiuyu Huang, Junjie Liu, Xiefan Guo, and Di Huang. Diffusion-4k: Ultra-high-resolution image synthesis with latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23464–23473, 2025.

Tianyi Zheng, Peng-Tao Jiang, Ben Wan, Hao Zhang, Jinwei Chen, Jia Wang, and Bo Li. Beta-tuned timestep diffusion model. In *European Conference on Computer Vision*, pp. 114–130. Springer, 2024.

Tianyi Zheng, Jiayang Zou, Peng-Tao Jiang, Hao Zhang, Jinwei Chen, Jia Wang, and Bo Li. Bidirectional beta-tuned diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

# A Proof and Analysis of Task Affinity

## A.1 Proof of Proposition 3.5

Each timestep group $\mathcal{G}_k$ has a differentiable loss $L_{\mathcal{G}_k}(\theta)$. For the context, we define the context loss as a sum:

$$L_{\mathcal{B}_i}(\theta) := \sum_{\mathcal{G}_k \in \mathcal{B}_i} L_{\mathcal{G}_k}(\theta). \tag{10}$$

Assume for all $\theta$ and all groups $\mathcal{G}_k$,

$$\|\nabla_\theta L_{\mathcal{G}_k}(\theta)\| \le l, \tag{11}$$

$$L_{\mathcal{G}_k}(\theta) \ge C > 0, \tag{12}$$

and similarly (consistent with equation 10) for the context loss,

$$L_{\mathcal{B}_i}(\theta) \ge C > 0. \tag{13}$$

Moreover, by equation 10 and equation 11,

$$\|\nabla_\theta L_{\mathcal{B}_i}(\theta)\| = \left\| \sum_{\mathcal{G}_k \in \mathcal{B}_i} \nabla_\theta L_{\mathcal{G}_k}(\theta) \right\| \le \sum_{\mathcal{G}_k \in \mathcal{B}_i} \|\nabla_\theta L_{\mathcal{G}_k}(\theta)\| \le |\mathcal{B}_i|\, l. \tag{14}$$

One-step shared-parameter updates. Let $\eta_t > 0$ be the step size at iteration $t$ and define $g^t_{\mathcal{G}_k} := \nabla_\theta L_{\mathcal{G}_k}(\theta^t)$. For any set $\mathcal{S}$ of groups, define the one-step update

$$\theta^{t+1}_{\mathcal{S}} := \theta^t - \eta_t \sum_{\mathcal{G}_k \in \mathcal{S}} g^t_{\mathcal{G}_k}. \tag{15}$$

In particular,

$$\theta^{t+1}_{\mathcal{B}_i} = \theta^t - \eta_t \sum_{\mathcal{G}_k \in \mathcal{B}_i} g^t_{\mathcal{G}_k}, \tag{16}$$

$$\theta^{t+1}_{\mathcal{G}_i \cup \mathcal{B}_i} = \theta^t - \eta_t \left( g^t_{\mathcal{G}_i} + \sum_{\mathcal{G}_k \in \mathcal{B}_i} g^t_{\mathcal{G}_k} \right), \tag{17}$$

thus

$$\theta^{t+1}_{\mathcal{B}_i} - \theta^{t+1}_{\mathcal{G}_i \cup \mathcal{B}_i} = \eta_t\, g^t_{\mathcal{G}_i}. \tag{18}$$

Similarly, for any $\mathcal{G}_k \in \mathcal{B}_i$,

$$\theta^{t+1}_{\mathcal{G}_k} - \theta^{t+1}_{\mathcal{G}_i \cup \mathcal{G}_k} = \eta_t\, g^t_{\mathcal{G}_i}. \tag{19}$$

Affinity definitions. We define group-to-context and group-to-group affinities at step $t$ as

$$\mathcal{A}^t_{\mathcal{G}_i \to \mathcal{B}_i} := 1 - \frac{L_{\mathcal{B}_i}\big(\theta^{t+1}_{\mathcal{G}_i \cup \mathcal{B}_i}\big)}{L_{\mathcal{B}_i}\big(\theta^{t+1}_{\mathcal{B}_i}\big)}, \tag{20}$$

$$\mathcal{A}^t_{\mathcal{G}_i \to \mathcal{G}_k} := 1 - \frac{L_{\mathcal{G}_k}\big(\theta^{t+1}_{\mathcal{G}_i \cup \mathcal{G}_k}\big)}{L_{\mathcal{G}_k}\big(\theta^{t+1}_{\mathcal{G}_k}\big)}. \tag{21}$$

Step 1: rewrite as normalized loss differences.

$$\mathcal{A}^t_{\mathcal{G}_i \to \mathcal{B}_i} = \frac{L_{\mathcal{B}_i}\big(\theta^{t+1}_{\mathcal{B}_i}\big) - L_{\mathcal{B}_i}\big(\theta^{t+1}_{\mathcal{G}_i \cup \mathcal{B}_i}\big)}{L_{\mathcal{B}_i}\big(\theta^{t+1}_{\mathcal{B}_i}\big)}, \tag{22}$$

$$\mathcal{A}^t_{\mathcal{G}_i \to \mathcal{G}_k} = \frac{L_{\mathcal{G}_k}\big(\theta^{t+1}_{\mathcal{G}_k}\big) - L_{\mathcal{G}_k}\big(\theta^{t+1}_{\mathcal{G}_i \cup \mathcal{G}_k}\big)}{L_{\mathcal{G}_k}\big(\theta^{t+1}_{\mathcal{G}_k}\big)}. \tag{23}$$

Step 2: apply the Mean Value Theorem. By the mean value theorem, there exists a point $\xi_{\mathcal{B}}$ on the line segment between $\theta^{t+1}_{\mathcal{B}_i}$ and $\theta^{t+1}_{\mathcal{G}_i \cup \mathcal{B}_i}$ such that

$$L_{\mathcal{B}_i}\big(\theta^{t+1}_{\mathcal{B}_i}\big) - L_{\mathcal{B}_i}\big(\theta^{t+1}_{\mathcal{G}_i \cup \mathcal{B}_i}\big) = \nabla_\theta L_{\mathcal{B}_i}(\xi_{\mathcal{B}})^\top \Big(\theta^{t+1}_{\mathcal{B}_i} - \theta^{t+1}_{\mathcal{G}_i \cup \mathcal{B}_i}\Big). \tag{24}$$

Combining equation 24 with equation 18 yields

$$\mathcal{A}^t_{\mathcal{G}_i \to \mathcal{B}_i} = \frac{\eta_t \, \nabla_\theta L_{\mathcal{B}_i}(\xi_{\mathcal{B}})^\top g^t_{\mathcal{G}_i}}{L_{\mathcal{B}_i}\left(\theta^{t+1}_{\mathcal{B}_i}\right)}. \tag{25}$$

Similarly, for each $\mathcal{G}_k \in \mathcal{B}_i$, there exists $\xi_k$ on the line segment between $\theta^{t+1}_{\mathcal{G}_k}$ and $\theta^{t+1}_{\mathcal{G}_i \cup \mathcal{G}_k}$ such that

$$\mathcal{A}^t_{\mathcal{G}_i \to \mathcal{G}_k} = \frac{\eta_t \, \nabla_\theta L_{\mathcal{G}_k}(\xi_k)^\top g^t_{\mathcal{G}_i}}{L_{\mathcal{G}_k}\left(\theta^{t+1}_{\mathcal{G}_k}\right)}. \tag{26}$$

Step 3: bound the deviation from the average of pairwise affinities. Let

$$\Delta := \mathcal{A}^t_{\mathcal{G}_i \to \mathcal{B}_i} - \frac{1}{|\mathcal{B}_i|} \sum_{\mathcal{G}_k \in \mathcal{B}_i} \mathcal{A}^t_{\mathcal{G}_i \to \mathcal{G}_k}.$$

Substituting equation 25 and equation 26, and using the loss lower bounds equation 13–equation 12, we have

$$\begin{aligned} |\Delta| &= \left| \frac{\eta_t}{L_{\mathcal{B}_i}(\theta^{t+1}_{\mathcal{B}_i})} \nabla_\theta L_{\mathcal{B}_i}(\xi_{\mathcal{B}})^\top g^t_{\mathcal{G}_i} - \frac{1}{|\mathcal{B}_i|} \sum_{\mathcal{G}_k \in \mathcal{B}_i} \frac{\eta_t}{L_{\mathcal{G}_k}(\theta^{t+1}_{\mathcal{G}_k})} \nabla_\theta L_{\mathcal{G}_k}(\xi_k)^\top g^t_{\mathcal{G}_i} \right| \\ &\leq \eta_t \left\| g^t_{\mathcal{G}_i} \right\| \left( \frac{\|\nabla_\theta L_{\mathcal{B}_i}(\xi_{\mathcal{B}})\|}{L_{\mathcal{B}_i}(\theta^{t+1}_{\mathcal{B}_i})} + \frac{1}{|\mathcal{B}_i|} \sum_{\mathcal{G}_k \in \mathcal{B}_i} \frac{\|\nabla_\theta L_{\mathcal{G}_k}(\xi_k)\|}{L_{\mathcal{G}_k}(\theta^{t+1}_{\mathcal{G}_k})} \right) \end{aligned} \tag{27}$$

where we used triangle inequality and Cauchy–Schwarz.

Applying equation 11–equation 14 to equation 27 yields

$$\begin{aligned} |\Delta| &\leq \eta_t \left\| g^t_{\mathcal{G}_i} \right\| \left( \frac{|\mathcal{B}_i| \, l}{C} + \frac{1}{|\mathcal{B}_i|} \sum_{\mathcal{G}_k \in \mathcal{B}_i} \frac{l}{C} \right) \\ &= \eta_t \left\| g^t_{\mathcal{G}_i} \right\| \cdot \frac{(|\mathcal{B}_i| + 1) \, l}{C}. \end{aligned} \tag{28}$$

Finally, since $\|g^t_{\mathcal{G}_i}\| = \|\nabla_\theta L_{\mathcal{G}_i}(\theta^t)\| \leq l$ by equation 11, we obtain

$$\left| \mathcal{A}^t_{\mathcal{G}_i \to \mathcal{B}_i} - \frac{1}{|\mathcal{B}_i|} \sum_{\mathcal{G}_k \in \mathcal{B}_i} \mathcal{A}^t_{\mathcal{G}_i \to \mathcal{G}_k} \right| \;\leq\; \frac{\eta_t (1 + |\mathcal{B}_i|) \, l^2}{C}, \tag{29}$$

which proves the claim.

### A.2 Differences from TAG Task Affinity

Our task affinity formulation differs fundamentally from that defined in TAG (Fifty et al., 2021). TAG defines task affinity as

$$Z^t_{i \to j} = 1 - \frac{L_j(\theta^{t+1}_i)}{L_j(\theta^t_j)},$$

which measures the effect of updating task $\mathcal{G}_i$ on the loss of task $\mathcal{G}_j$. In contrast, our affinity score $\mathcal{A}^t_{i \to j}$ characterizes the effect of jointly training $\mathcal{G}_i$ and $\mathcal{G}_j$ on the optimization of $\mathcal{G}_j$. This distinction is crucial, as it enables us to establish a direct relationship between affinity values and loss reduction without introducing additional assumptions. Specifically, if $\mathcal{A}^t_{i \to j} > \mathcal{A}^t_{k \to j}$, then joint training $\{\mathcal{G}_i, \mathcal{G}_j\}$ induces a larger loss decrease for $\mathcal{G}_j$ than joint training $\{\mathcal{G}_k, \mathcal{G}_j\}$. By contrast, TAG relies on restrictive conditions, such as strong convexity of the loss function, to derive a similar implication.

(1) TAG further definesgroup-to-context as a simple average of pairwise affinities,

$$Z^t_{i \to \{j,k\}} = \frac{1}{2} \left( Z^t_{i \to j} + Z^t_{i \to k} \right),$$

which directly aggregates task-level quantities and lacks theoretical guarantees that the resulting score faithfully reflects joint optimization effects. In contrast, we explicitly establish the connection between



Figure 5: Generated images of FFHQ.

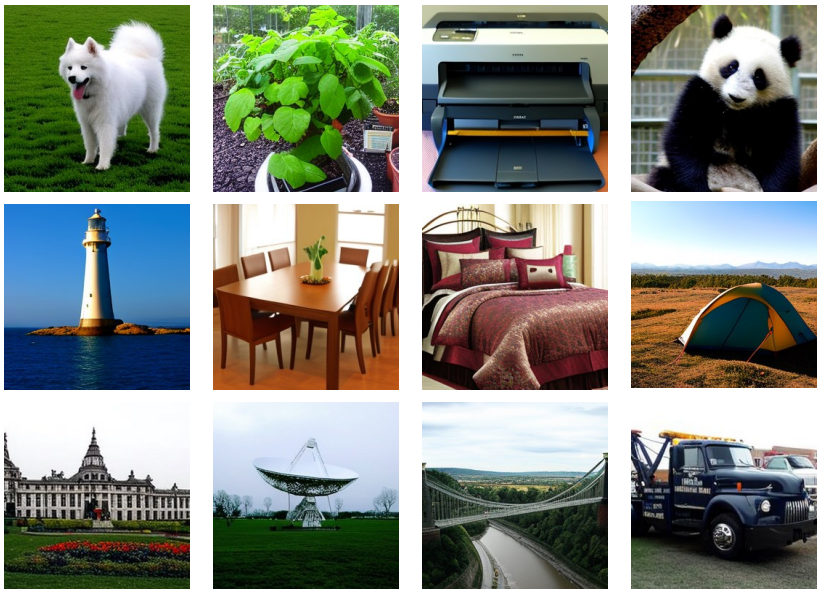


Figure 6: Generated images of ImageNet-1K.

inter-group task affinity and group-to-context task affinity, providing both theoretical justification and practically valid operations at the implementation level.

(2) Diffusion model training simultaneously samples a large number of timesteps, inducing complex higher-order optimization interactions among denoising tasks. In this regime, pairwise quantities such as $Z^t_{i\to j}$ are insufficient to capture the resulting dependency structure. Our formulation naturally extends to the group-to-context setting, enabling an accurate characterization of optimization dependencies under large-scale timestep sampling.

## B Qualitative Results

Figures 5 and 6 present qualitative examples generated by the proposed method on representative datasets.

## C Training Details

All models are optimized using the AdamW optimizer (Kingma & Ba, 2014) with a fixed learning rate of $1 \times 10^{-4}$. The diffusion process is configured with 1000 training timesteps and a linear variance schedule by default, except in experiments that explicitly explore alternative scheduling strategies.

Random horizontal flipping is applied as the data augmentation technique. Following common practice in diffusion-based generative modeling, an exponential moving average (EMA) (Gardner Jr, 1985) of model parameters is maintained with a decay rate of 0.9999, and all reported results are obtained using the EMA weights. We report the practical overhead on DiT-XL/2 for FFHQ 256×256, using 8 NVIDIA A6000 GPUs. Under the setting of batch size = 16, the wall-clock runtime is 22.76 h, the peak GPU memory usage is 42,397 MB, the FLOPs are approximately 119 GFLOPs, and the training throughput is 0.61 steps/s. Since our method does not introduce any new model modules, its additional overhead mainly arises from the training-time strategy rather than from increased model size or architectural complexity.

## D Computational Cost

Computing the inter-group task affinity in Eq. equation 4 requires constructing one-step updated parameters under different group configurations. For the denominator $L_{\mathcal{G}_j}(\theta^{t+1}_{\{\mathcal{G}_j\}})$, the update depends only on the target group $\mathcal{G}_j$ and can therefore be computed once and reused across all affinities associated with the same $j$. As a result, evaluating all denominators requires $k$ computations, each incurring a cost of $(2\,\mathrm{t}_{\mathrm{fwd}}(\theta) + 1\,\mathrm{t}_{\mathrm{bwd}}(\theta))$, where $\mathrm{t}_{\mathrm{fwd}}(\theta)$ denotes the computational cost of a forward pass and $\mathrm{t}_{\mathrm{bwd}}(\theta)$ denotes the cost of a backward pass. In contrast, the numerator $L_{\mathcal{G}_j}(\theta^{t+1}_{\{\mathcal{G}_i,\mathcal{G}_j\}})$ depends on a pairwise combination of distinct task groups $(\mathcal{G}_i, \mathcal{G}_j)$ with $i \neq j$ and must be evaluated separately for each such pair. Computing all numerators therefore requires $k(k-1)/2$ computations, each incurring the same cost of $(2\,\mathrm{t}_{\mathrm{fwd}}(\theta) + 1\,\mathrm{t}_{\mathrm{bwd}}(\theta))$. Overall, the computational cost of constructing the full inter-group task affinity matrix scales as

$$\mathcal{O}\left(\frac{k(k+1)}{2}\left(2\,\mathrm{t}_{\mathrm{fwd}}(\theta) + \mathrm{t}_{\mathrm{bwd}}(\theta)\right)\right) \tag{30}$$

During each iteration of diffusion model training, particularly for sufficiently large models, the dominant computational cost arises from the forward and backward passes of the objective $L_t(\theta)$. In practice, the backward pass is substantially more expensive due to the communication overhead incurred by gradient synchronization and typically takes about three times longer than the forward pass (Li et al., 2020). Thus, a single training iteration approximately incurs a cost of

$$\mathrm{t}_{\mathrm{iter}}(\theta) = \mathrm{t}_{\mathrm{fwd}}(\theta) + \mathrm{t}_{\mathrm{bwd}}(\theta) \approx 4\,\mathrm{t}_{\mathrm{fwd}}(\theta).$$

With $k = 20$ groups and computing the group-to-context task affinity once every 2000 iterations, the amortized overhead per iteration is approximately $13.12\%$ of a standard training step.

## E Discussion on Adaptive Methods

**Discussion on the Difference from ADA(Kim et al., 2025).** Our method differs from ADA primarily in the perspective used to understand timestep sampling and the optimization signal used for adaptation. ADA largely follows the conventional view that timestep sampling should reflect the optimization difficulty or importance of individual timesteps. Since different loss- or gradient-based criteria may yield inconsistent importance estimates, ADA learns a dynamic sampler to implicitly infer the sampling distribution. In contrast, we revisit diffusion training from a multi-task learning perspective, where different timestep groups are treated as correlated tasks sharing the same model parameters. This perspective shifts the focus from the difficulty of an individual timestep to how optimizing one timestep group affects the optimization of others, providing a unified interpretation of timestep sampling through cross-timestep optimization interactions.

More importantly, the two methods rely on fundamentally different adaptation signals. ADA adjusts timestep importance according to the overall loss reduction induced by sampling and optimizing particular timesteps, which is closely related to the loss-based importance criteria explored in prior timestep-sampling methods. Our method instead introduces task affinity, which explicitly measures whether optimizing one timestep group facilitates or interferes with other groups. By modeling both positive and negative transfer across timestep groups, task affinity directly captures their optimization relationships and uses them to globally update the sampling distribution. Therefore, our adaptation mechanism is not based solely on estimating individual timestep importance, but on explicitly modeling cross-timestep interactions.

We do not include ADA in our experiments for two practical reasons. First, its original evaluation is limited to DDPM on CIFAR-10 and LDM on CelebA-HQ at $256 \times 256$ resolution, leaving its applicability to larger-scale datasets and architectures insufficiently validated. Moreover, reusable configurations and independent implementations for these broader settings are currently unavailable. Second, ADA requires training an additional timestep sampler and periodically updating it according to changes in the global variational objective. Extending it to new datasets or architectures therefore introduces several sampler-specific hyperparameters that require careful retuning. Given our limited computational budget, we cannot guarantee sufficiently thorough tuning to ensure a fair and representative comparison.

**Discussion on the Difference from CLTS (Xu et al., 2024).** CLTS adapts timestep sampling through a predefined transformation of the sampling distribution, typically from a uniform distribution toward a Gaussian form. As reported in SpeeD, its performance can be unstable and is generally inferior to SpeeD under the evaluated settings. This suggests that predefined distributional transformations may not consistently reflect the evolving optimization state during training. In contrast, our method determines sampling probabilities from explicit cross-timestep optimization relationships, allowing the sampling strategy to adapt directly to the current training dynamics.

**Discussion on the Difference from Diff-CL (Kim et al., 2024).** Diff-CL constructs a curriculum based on timestep training difficulty and allocates the sampling budget according to a predefined curriculum schedule. The method also notes that dynamically adjusting the curriculum according to the model state could further improve performance. From this perspective, Diff-CL relies primarily on a predefined rule to organize timestep sampling, whereas our method performs dynamic adaptation through task affinity, which explicitly captures the optimization interactions among timestep groups.