Multi-Architecture Multi-Expert Diffusion Models

Yunsung Lee[*] **JinYoung Kim**[*] **Hyojun Go**[*]
Myeongho Jeong **Shinhyeok Oh** **Seungtaek Choi**[†]
Riiid AI Research
{yunsung.lee, jinyoung.kim, hyojun.go,
myeongho.jeong, shinhyeok.oh, seungtaek.choi}@riiid.co

Abstract

Diffusion models have achieved impressive results in generating diverse and realistic data by employing multi-step denoising processes. However, the need for accommodating significant variations in input noise at each time-step has led to diffusion models requiring a large number of parameters for their denoisers. We have observed that diffusion models effectively act as filters for different frequency ranges at each time-step noise. While some previous works have introduced multi-expert strategies, assigning denoisers to different noise intervals, they overlook the importance of specialized operations for high and low frequencies. For instance, self-attention operations are effective at handling low-frequency components (low-pass filters), while convolutions excel at capturing high-frequency features (high-pass filters). In other words, existing diffusion models employ denoisers with the same architecture, without considering the optimal operations for each time-step noise. To address this limitation, we propose a novel approach called Multi-architecturE Multi-Expert (MEME), which consists of multiple experts with specialized architectures tailored to the operations required at each time-step interval. Through extensive experiments, we demonstrate that MEME outperforms large competitors in terms of both generation performance and computational efficiency.

1 Introduction

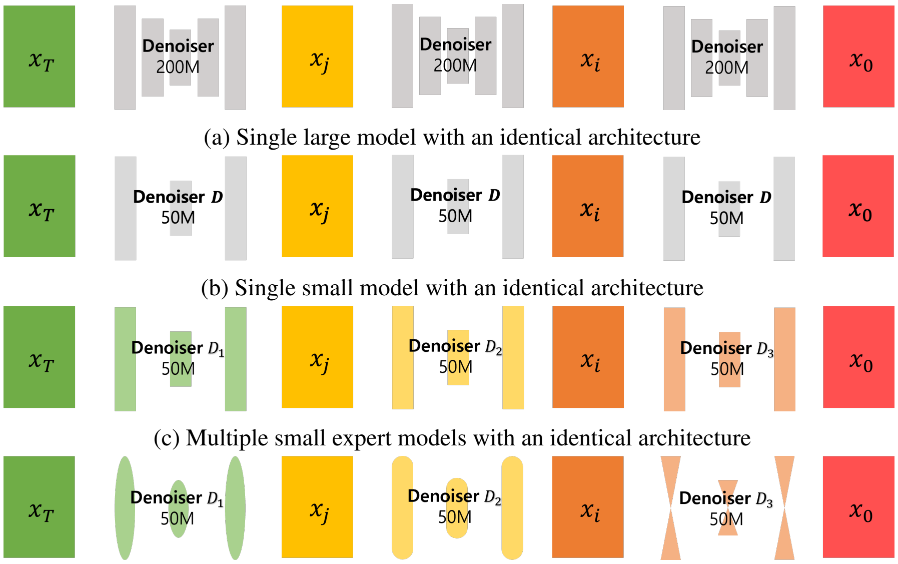
Diffusion models [49, 52, 20] are a promising approach for generative modeling, and they are likely to play an increasingly important role in diverse domains, including image [7, 41, 44, 42, 1], audio [29, 40, 23, 27], video [22, 19, 70], and 3D generation [39, 68, 46]. However, despite their impressive performance, diffusion models suffer from high computation costs, which stem from the following two orthogonal factors: (i) the lengthy iterative denoising process, and (ii) the cumbersome denoiser networks. Though there have been several efforts to overcome such limitations [2, 30, 32, 45, 36, 50, 51, 15, 21, 42, 61, 57], most of these efforts have focused only on resolving the first factor, such that the cumbersome denoisers still limit their applicability to real-world scenarios. A few efforts reduce the size of the denoisers based on post-training low-bit quantization [47] and distillation [67], as we illustrated in Fig. 1b, but they usually achieve such efficiency by compromising on accuracy.

In this paper, we thus aim to build a diffusion model that is compact yet comparable in performance to the large models. For this purpose, we first ask a research question "*why the traditional diffusion models require such massive parameters?*". From a frequency perspective, there was a theoretical explanation that it is because the models should learn too many different features in varying time-steps [67], where diffusion models tend to initially form low-frequency components (e.g., overall

[*] Co-first author

[†] Corresponding author

Preprint. Under review.



(a) Single large model with an identical architecture

(b) Single small model with an identical architecture

(c) Multiple small expert models with an identical architecture

(d) Multiple small expert models with multiple architectures (MEME)

Figure 1: **Comparative illustration of single/multiple-expert models with single/multiple architectures.** Figure 1a depict the standard diffusion models approach, which employs a single large denoiser. To reduce the cost due to the large-scale of the model, a diffusion model with a small denoiser is designed with post-training low-bit or distillation as illustrated in Fig.1b. In Fig. 1c, to alleviate the performance drop, we consider a configuration with multiple small expert models having identical architectures. Finally, our proposed method, the Multi-architecturE Multi-Expert diffusion models (MEME), constructs small expert models with unique optimal architectures for their respective assigned time-step intervals, as visualized in Fig. 1d.

image contour) and subsequently fill in high-frequency components (e.g., detailed textures). However, as they assume the denoiser network to be a linear filter, which is not practical, we aim to investigate empirical evidence to support this claim. Specifically, we analyze the per-layer Fourier spectrum for the input x_t at each diffusion time-step t, finding that there are significant and consistent variations in the relative log amplitudes of the Fourier-transformed feature maps as t progresses. This finding indicates that the costly training process of large models indeed involves learning to adapt to the different frequency characteristics at each time-step t.

One way to leverage this finding is to assign distinct time-step intervals to multiple diffusion models [13, 1], referred to as the *multi-expert* strategy, in order for models to be specialized in the assigned time-step intervals as shown in Fig. 1c. However, since [13] utilized the multi-expert strategy for the conditioned generation with guidance and [1] focused on high performance, the efficiency is not considered while ignoring the fact that different operations may be more suitable at different time-step from a frequency perspective.

To this end, we propose to assign different models with **different architectures** for each different time-step interval, whose base operations vary according to their respective frequency ranges, which we dub **Multi-architecturE Multi-Expert diffusion models (MEME)** (Fig. 1d). Specifically, we leverage the recent insights from [10, 6, 37, 48, 66, 9] that convolutions are advantageous for handling high-frequency components ($t \sim 0$), while multi-head self-attention (MHSA) excels in processing low-frequency components ($t \sim T$). However, naively placing two different architectures in different time-step intervals would be suboptimal because the features are inherently a combination of high- and low-frequency components [10, 48, 37].

In order to better adapt to such a complex distribution of frequency-specific components, we propose a more flexible denoiser architecture called **iU-Net**, which incorporates an iFormer [48] block that allows for adjusting the channel-wise balance ratio between the convolution operations and MHSA operations. We take advantage of the characteristic of diffusion models we discovered that first recover low-frequency components during the denoising process and gradually add high-frequency features. Consequently, we configure each architecture to have a different proportion of MHSA, effectively tailoring each architecture to suit the distinct requirements at different time-step intervals of the diffusion process.

We further explore methods for effectively assigning focus on specific time-step intervals to our flexible iU-Net. Specifically, we identify a soft interval assignment strategy for the multi-expert models that prefers a soft division over a hard segmentation. This strategy allows the experts assigned to intervals closer to T to have more chance to be trained with the entire time-step, which prevents excessive exposure to meaningless noises at the time-step $t \sim T$.

Empirically, our MEME diffusion models effectively perform more specialized processing for each time-step interval, resulting in improved performance compared to the baselines. MEME, with LDM as the baseline, has managed to reduce the computation cost by 3.3 times while training on FFHQ [26] and CelebA-HQ [25] datasets from scratch and has simultaneously improved image generation performance by 0.62 and 0.37 in FID scores, respectively. By comparing the Fourier-transformed feature maps of MEME and multi-expert with identical architecture, we have confirmed that MEME's multi-architecture approach allows for distinct frequency characteristics suitable for each interval. Furthermore, MEME not only improves performance when combined with the LDM baseline but also demonstrates successful performance enhancements when integrated with the other diffusion model, DDPM [20].

In summary, our contributions can be distilled into three main points:

- As far as we know, we are the first to identify and address the issue that existing diffusion models rely solely on identical operations at all time-steps, despite the vastly different functionalities required at each time-step in diffusion processes.
- We propose MEME, a novel diffusion models framework composed of multi-architecture multi-expert denoisers which can balance operations for low-frequency and high-frequency components, performing distinct operations for each time-step interval.
- MEME surpasses its counterparts not only in computational efficiency but also in terms of generation quality. Trained from scratch on the FFHQ and CelebA datasets, MEME operates 3.3 times faster than baselines while improving FID scores by 0.62 and 0.37, respectively.

2 Related Work

2.1 Diffusion Models

Diffusion models [49, 52, 20], a subclass of generative models, generate data through an iterative denoising process. Trained by denoising score-matching objectives [53], these models demonstrate impressive performance and versatility in various domains, including image [7, 41, 44, 42, 1], audio [29, 40, 23, 27], video [22, 19, 70], and 3D [39, 68, 46] generation. However, diffusion models suffer from significant drawbacks, such as high memory and computation time costs [28, 32, 50]. These issues primarily stem from two factors: the lengthy iterative denoising process and the substantial number of parameters in the denoiser model. A majority of studies addressing the computation cost issues of diffusion models have focused on accelerating the iteration process. Among these, [63, 64, 8] have employed more efficient differential equation solvers. Other studies have sought to reduce the lengthy iterations by using truncated diffusion [34, 69] or knowledge distillation [45, 51, 33].

In contrast, [47] and [67] focus on reducing the size of diffusion models. Shang et al. [47] proposes a post-training low-bit quantization specifically tailored for diffusion models. Yang et al. [67] analyze diffusion models based on frequency, enabling small models to effectively handle high-frequency dynamics by applying wavelet gating and spectrum-aware distillation. However, these attempts at lightweight models usually failed to match the performance of large models and rely on resource-intensive training, which assumes the availability of a pretrained diffusion model. However, despite the well-known insight [67, 5, 16] that diffusion models have very different functionality to learn at each time-step, previous diffusion models use the same structured denoiser model with the same operations at every time-step. Our research aims to provide more optimized operations for denoiser models at each time-step of the diffusion process. To achieve this, we embrace the multi-expert strategy, which has been previously utilized in plug-and-play guidance models [13] or for increasing architectural capacity [1].

2.2 Combination of Convolutions and Self-attentions

Since the advent of the Vision Transformer [9], there has been active research into why self-attention works effectively in the image domain and how it differs from convolution operations. [9, 10, 60] suggest that this is because self-attention operations better capture global features and act as low-pass filters [37, 48]. There have been efforts [37, 6, 66, 10, 48] aiming to design optimal architectures that better combine the advantages of self-attention and convolution. [37, 6] suggest structuring networks with convolution-focused front layers, which are advantageous for high-pass filtering, and self-attention-focused rear layers, which are advantageous for low-pass filtering. [10, 48] propose new blocks that perform operations intermediate between convolution and self-attention. Notably, iFormer [48] proposes a block that allows for adjustable ratios between convolution and self-attention operations. Despite numerous efforts to reveal the benefits of different operations depending on the frequency and design better architectures for image recognition, current diffusion models still rely on a single fixed architecture to learn vastly different noise levels. This approach proves to be highly suboptimal and calls for multi-architecture.

3 Background

3.1 Diffusion Models and Spectrum Evolution over Time

Diffusion models [49, 52, 20] work by inverting a stepwise noise process using latent variables. Data points $\mathbf{x}_0$ from the true distribution are perturbed by Gaussian noise with zero mean and β_t variance across T steps, eventually reaching Gaussian white noise. As in [20], efficiently sampling from the noise-altered distribution $q(\mathbf{x}_t)$ is achieved through a closed-form expression to generate arbitrary time-step $\mathbf{x}_t$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \text{where} \quad \epsilon \sim \mathcal{N}(0,\mathbf{I}),\ \alpha_t = 1-\beta_t,\ \bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s \tag{1}$$

The denoiser, a time-conditioned denoising neural network $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)$ with trainable parameters θ, is trained to reverse the diffusion process by minimizing re-weighted evidence lower bound (ELBO) [52, 20], adapting to the noise as follows:

$$\mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[||\nabla\mathbf{x}_t \log p(\mathbf{x}_t|\mathbf{x}_0) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)||_2^2\Big] \tag{2}$$

In essence, the denoiser learns to recover the gradient that optimizes the data log-likelihood. Utilizing the trained denoiser, the previous step data x_{t-1} is generated by inverting the Markov chain:

$$\mathbf{x}_{t-1} \leftarrow \frac{1}{\sqrt{1-\beta_t}}(\mathbf{x}_t + \beta_t\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)) + \sqrt{\beta_t}\boldsymbol{\epsilon}_t \tag{3}$$

In this reverse process, the insight that diffusion models evolve from rough to detailed was gained through several empirical observations [5, 20, 35, 42]. Beyond them, [67] provides a numerical explanation of this insight from a frequency perspective by considering the network as a linear filter. In this case, the optimal filter, known as the Wiener filter [65], can be expressed in terms of its spectrum response at every time-step. Under the widely accepted assumption that the power spectra $\mathbb{E}[|X_0(f)|^2] = A_s(\theta)/f^{\alpha_S(\theta)}$ of natural images x_0 follows a power law [62, 3, 11, 58], the frequency response of the signal reconstruction filter is determined by the amplitude scaling factor $A_s(\theta)$ and the frequency exponent $\alpha_S(\theta)$. As the reverse denoising process progresses from $t = T$ to $t = 0$, and $\bar{\alpha}$ increases from 0 to 1, diffusion models, as analyzed by [67], exhibit spectrum-varying behavior over time. Initially, a narrow-banded filter restores only low-frequency components responsible for rough structures. As t decreases and $\bar{\alpha}$ increases, more high-frequency components, such as human hair, wrinkles, and pores, are gradually restored in the images.

3.2 Inception Transformer

The limitation of transformers in the field of vision is well-known as they tend to capture low-frequency features that convey global information but are less proficient at capturing high-frequency features that correspond to local information, as noted in previous works [9, 48]. To address this shortcoming, [48] introduced the Inception Transformer, which combines a convolution layer with a transformer, utilizing the Inception module [55, 56, 54] To elaborate, the input feature $\mathbf{Z} \in \mathbb{R}^{N \times d}$ is first separated into $\mathbf{Z}_h \in \mathbb{R}^{n \times d_h}$ and $\mathbf{Z}_l \in \mathbb{R}^{n \times d_l}$ along the channel dimension, where $d = d_h + d_l$. The iFormer block then applies a high-frequency mixer to $\mathbf{Z}_h$ and a low-frequency mixer to $\mathbf{Z}_l$. Specifically, $\mathbf{Z}_h$ is further split into $\mathbf{Z}_{h_1}$ and $\mathbf{Z}_{h_2}$ along the channel dimension as follows:

$$\mathbf{Y}_{h_1} = \text{FC}(\text{MP}(\mathbf{Z}_{h_1})), \tag{4}$$

$$\mathbf{Y}_{h_2} = \text{D-Conv}(\text{FC}(\mathbf{Z}_{h_2})), \tag{5}$$

where $\mathbf{Y}$ denotes the outputs of high-frequency mixer, FC is fully-connected layer, MP represents max pooling layer, and D-Conv is depth-wise convolutional layer.

In the low-frequency mixer, MHSA is utilized to acquire a comprehensive and cohesive representation, as shown in Eq. 6. This global representation is then combined with the output from the high-frequency mixer as in Eq. 7. However, due to the potential oversmoothing effect of the upsample operation described in Eq. 6, a fusion module outlined in Eq. 8 is introduced to counteract this issue and produce the final output.

$$\mathbf{Y}_l = \text{Up}(\text{MHSA}(\text{AP}(\mathbf{Z}_{h_2}))), \tag{6}$$

$$\mathbf{Y}_c = \text{Concat}(\mathbf{Y}_{h_1}, \mathbf{Y}_{h_2}, \mathbf{Y}_l), \tag{7}$$

$$\mathbf{Y} = \text{FC}(\mathbf{Y}_c + \text{D-Conv}(\mathbf{Y}_c)), \tag{8}$$

where Up denotes upsampling, AP is average pooling, and Concat represents concatenation.

4 Frequency Analysis for Diffusion Models

As in [10, 60], it is useful to design the architecture with distinct blocks capturing appropriate frequency according to the depth of the block. In this section, we analyze the frequency-based characteristics of latents and extracted features by the model according to time-step.

4.1 Frequency Component from Latents.

From the fact that a Gaussian filter prioritizes the filtering out of high-frequency [14], it is evident that the training data fed into diffusion models progressively lose their high-frequency spectrum as t increases. As shown in Fig. 2, by illustrating the Fourier coefficients of the periodic function against the corresponding frequency, we demonstrate that the training data gradually lose their high-frequency spectrum as t increases. It is thus clear to design the diffusion model which filters different frequency components according to the time-step for dealing with the corresponding features.

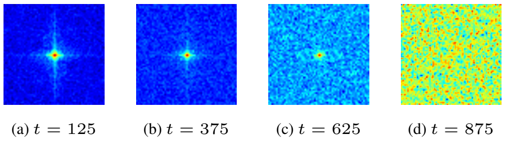
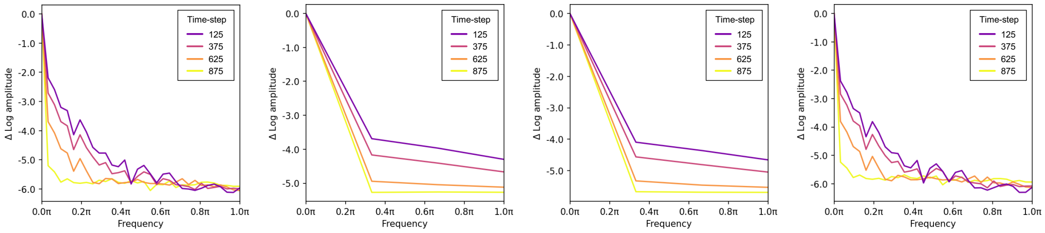

(a) $t = 125$ (b) $t = 375$ (c) $t = 625$ (d) $t = 875$

Figure 2: **Visualization of the Fourier spectrum of the inputs** used in the training of diffusion models. As t increases from 0 to T, we observe the high-frequency feature spectrum, initially concentrated towards the center, gradually disappearing.

4.2 Frequency Component Focused by Model.

Here, we first introduce the analysis on the frequency for each layer with the distinct depth as [9, 10] did. We examine examining the relative log amplitudes of Fourier-transformed feature maps obtained from the pre-trained latent diffusion model (LDM). As shown in Fig. 3, it reveals that image recognition deep neural networks primarily perform high-pass filtering in earlier layers and low-pass filtering in later layers. Additionally, in this paper, we further analyze the frequency components focused by the diffusion model with respect to the diffusion time-step. The captured feature with frequency perspective is illustrated in each subfigure of Fig. 3, indicating that diffusion models tend to attenuate low-frequency signals more prominently as t increases. These findings align with the well-established characteristics of a Gaussian filter, known for its tendency to suppress high-frequency components primarily [14].



(a) The former layer in the encoder (b) The later layer in the encoder (c) The former layer in the decoder (d) The later layer in the decoder

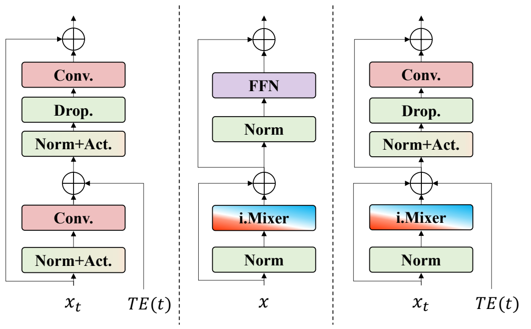
Figure 3: **Visualization of relative log amplitudes of Fourier transformed feature map obtained from the pre-trained large LDM.** The ΔLog amplitude of high-frequency signals is a difference with log amplitudes at the frequency of 0.0π and π. We compute it with 10K image samples at each time-step $t \in \{125, 375, 625, 875\}$. We can confirm that the tendency of ΔLog amplitude is interpolated as t is changed. In particular, as $t \to T$ (i.e., more noised input), the Fourier transformed features from the model are rapidly changed after 0.0π. Following [37], we provide half-diagonal components of two-dimensional Fourier-transformed feature maps for better visualization.

5 Multi-Architecture Multi-Expert Diffusion Models

Based on the above observations in Section 4, we propose the following significant hypothesis: *By structuring the denoiser model with operations that vary according to each time-step interval, it could potentially enhance the efficiency of the diffusion model's learning process.* To validate this hypothesis, two key elements are needed: i) a denoiser architecture with the capacity to adjust the degree of its specialization towards either high or low frequencies, and ii) a strategy for varying the application of this tailored architecture throughout the diffusion process.

5.1 iU-Net Architecture

We propose the iU-Net architecture, a variant of U-Net [43] that allows for adjusting the ratio of operations favorable to high and low frequencies. We utilize a block referred to as the inception transformer (iFormer) [48], which intertwines convolution operations, suitable for high-pass filtering, and Multi-Head Self-Attention (MHSA) operations, suitable for low-pass filtering, with an inception [55] mixer. Figure 4 illustrates the manner in which we have adapted the iFormer block to fit the denoiser architecture of diffusion. This setup allows the iFormer block to regulate the ratio between the convolution-heavy high-frequency mixer and the MHSA-heavy low-frequency mixer in the architecture's composition. Following [37, 6, 66, 10, 48] that tried to combine convolution and MHSA, we set the iU-Net encoder to perform more MHSA operations in the later layers. We discuss it more technically in Section 5.2. Furthermore, as in [4], rather than completely replacing the block architecture from the U-Net block to the iFormer block, asymmetrically merging the two is effective in constructing an architecture for diffusion model that exploits iFormer.



(a) U-Net (b) iFormer (c) Ours (iU-Net)

Figure 4: **Comparative illustration of the block in the diffusion models.** Figure 4a illustrates the U-Net block for the diffusion model as in [20]. Our proposed architecture, iU-Net, exploits the iFormer depicted in Fig. 4b based on conventional U-Net for frequency dynamic feature extraction in the diffusion process as shown in Fig. 4c. The $\oplus$ denotes element-wise addition and TE denotes the time-embedding lookup table.

5.2 Multi-Architecture Multi-Expert Strategy

Architecture Design for Experts To facilitate the construction of structures capable of accommodating diverse architectures, we employ a multi-expert strategy [13, 1], but assign different architectures

Table 1: **Overall Results of Unconditional Generation on FFHQ and CelebA-HQ** We use the Clean-FID implementation to ensure reproducibility. We sample 200 steps using DDIM on the FFHQ, and 50 steps on the CelebA-HQ. Even with N models trained using Multi-Expert and MEME, the total training cost was equivalent to that of a large model. SD is trained through knowledge distillation, which is dependent on having a large pretrained model already, but we can build an efficient model from scratch. The symbols denote †: values reported in the original source; ‡: average value across four architectures; $\ast$: calculated using checkpoints from our training; $\ast\ast$: recalculated using pretrained checkpoints from the official repository.

FFHQ 256×256 (DDIM-200)

Model	#Param↓	MACs↓	FID↓	Prec.↑	Recall ↑
LDM-L$^{*}\ast$ (635K iter) [42]	274.1M	288.2G	9.03	**0.72**	0.49
Lite-LDM† [67]	22.4M	23.6G	17.3	-	-
SD (with Distill.) [67]	21.1M	-	10.5	-	-
LDM-L* (540K iter)	274.1M	288.2G	9.14	**0.72**	0.48
LDM-S*	89.5M(3.1×)	94.2G(3.1×)	11.41(−2.27)	0.66(−0.06)	0.44(−0.04)
iU-LDM-S*	**82.6M**(3.3×)	90.5G(3.2×)	11.64(−2.50)	0.65(−0.07)	0.45(−0.03)
Multi-Expert* (w/o Soft)	89.5M×4(3.1×)	94.2G(3.1×)	10.42(−1.28)	0.69(−0.03)	0.46(−0.02)
Multi-Expert*	89.5M×4(3.1×)	94.2G(3.1×)	9.58(−0.44)	0.70(−0.02)	0.46(−0.02)
MEME*(w/o Soft)	82.9M‡×4(3.3×)	**90.4G‡**(3.3×)	9.20(−0.06)	0.70(−0.02)	0.48(+0.00)
MEME*	82.9M‡×4(3.3×)	**90.4G‡**(3.3×)	**8.52**(+0.62)	**0.72**(+0.00)	**0.50**(+0.02)

CelebA-HQ 256×256 (DDIM-50)

Model	#Param↓	MACs↓	FID↓	Prec.↑	Recall ↑
LDM-L$^{*}\ast$ (410K iter) [42]	274.1M	288.2G	5.92	0.71	**0.49**
Lite-LDM† [67]	22.4M	23.6G	14.3	-	-
SD† (with Distill.) [67]	21.1M	-	9.3	-	-
LDM-S*	89.5M(3.1×)	94.2G(3.1×)	9.11(−3.19)	0.61(−0.10)	0.45(−0.04)
iU-LDM-S*	**82.6M**(3.3×)	90.5G(3.2×)	9.06(−3.14)	0.60(−0.11)	0.47(−0.02)
Multi-Expert*	89.5M×4(3.1×)	94.2G(3.1×)	7.00(−1.08)	0.67(−0.04)	0.48(−0.01)
MEME*	82.9M‡×4(3.3×)	**90.4G‡**(3.2×)	**5.55**(+0.37)	**0.73**(+0.02)	**0.49**(+0.00)

to each expert according to the frequency component. In each architecture, the ratio of dimension sizes for high and low channels is defined by two factors: layer depth and diffusion time-step. The former is well-known to enable the frequency dynamic feature extraction by focusing on lower frequency as a deeper layer [37, 6]. For more technical derivation, let d^k be the channel size in the k-th layer, d^k_h be the dimension size for the high mixer, and d^k_l for the low mixer, satisfying $d^k = d^k_h + d^k_l$. Based on the analysis in Fig. 3, the ratio in each iFormer block is defined for dealing with appropriate frequency components according to the depth; d^k_h/d^k_l decreases as a deeper block. On the other hand, the latter (diffusion time-step) can be associated with the frequency components based on the observation we found in Section 4; as time-step t increases, the lower frequency components are focused. Therefore, we configure the iU-Net architecture such that the ratio of d^k_h/d^k_l decreases faster for the denoiser taking charge of the expert on the larger t.

Soft Expert Strategy. As suggested in [13], one of N experts Θ_n is trained on the uniform and disjoint interval $\mathbb{I}_n = \left\{ t \middle| t \in \left(\frac{(n-1)}{N}T, \frac{n}{N}T \right] \right\}$ for $n = 1, ..., N$. However, for the large n, expert Θ_n takes as noised input images by near Gaussian noise $\epsilon_n \sim \mathcal{N}(\sqrt{\bar{\alpha}_n}x_0, (1-\bar{\alpha}_n)\mathbf{I})$, which makes it challenging for meaningful learning to take place with Θ_n. To address this, we propose a *soft expert strategy*, where each Θ_n learns on the interval $\mathbb{I}_n$ with a probability of p_n denoted as the expertization probability[3]. Otherwise, it learns on the entire interval $\bigcup_{n=1}^{N} \mathbb{I}_n$ with the remaining probability of $(1 - p_n)$.

Since it is evident that Θ_n for large n takes more noised images, larger p_n as $n \to N$ is a more flexible strategy for training multi-expert, yielding $p_1 \geq \cdots \geq p_N$.

6 Experiments

In this section, we demonstrate the capability of MEME to enhance the efficiency of diffusion models. Section 6.1 showcases how our model can achieve superior performance over the baseline models,

[3] Note that When $p_n = 1$ regardless of n, it can be denoted as *hard expert strategy* proposed by [13].

despite being executed with less computation. In Section 6.2, we verify if our MEME model, as hypothesized, indeed incorporates appropriate Fourier features for each time-step interval input.

We evaluated the unconditional generation of models on two datasets, FFHQ [26] and CelebA-HQ [25]. We construct models based on the LDM framework [42]. All pre-trained auto-encoders for LDM were obtained from the official repository[4].

MEME employs a multi-expert structure composed of multiple small models, each of which has its channel dimension reduced from 224 to 128. The use of these smaller models is denoted by appending an 'S' to the model name, such as in LDM-S and iU-LDM-S. We set the number of experts N to 4 for all multi-expert settings, including MEME.

All experiments were conducted on a single NVIDIA A100 GPU. We primarily utilize the AdamW optimizer [31]. The base learning rate is set according to the oigianl LDM [42]. Notably, our smaller models employ a setting that doubles the batch size, which is not feasible with the original LDM on a single A100. Correspondingly, the base learning rate for our smaller models is also doubled compared to the standard settings.

We assess the quality of our generated models using the FID score [18]. As the FID score can be challenging to replicate due to the settings of the reference set, we calculate it using the publicly available Clean-FID [38] implementation[5]. Particularly for the FFHQ dataset, the availability of a fixed reference set allows for a fair comparison of generation quality across all evaluated generative models on Clean-FID. To verify the efficiency of our trained model, we compare its model size and computational cost using the number of parameters and Multiply-Add cumulation (MACs)[6] as metrics. We provide detailed configurations regarding the models in the Appendix.

6.1 Image Generation Results

Performance and Cost The results of our model trained on FFHQ [26] and CelebA-HQ [25] datasets are shown in Table 1. Despite requiring only 3.3 times less computation cost (MACs), our model demonstrates an improvement in performance (FID). Specifically, we observe a gain of 0.62 in FID for FFHQ and 0.4 in FID for CelebA. In the case of MEME and Multi-Expert, they require N models to be loaded into system memory for inference. However, in large-scale sample inference scenarios, it is possible to load only one expert into system memory while storing intermediate outputs on the disk, yielding less cost to the inferring process. Our approach allows for an improvement of 3.3 times in memory cost, which is equivalent to that of a single denoiser. In our experimental setting with $N = 4$, even if all experts are loaded into system memory for inference, it only requires an additional 20.9% of memory compared to the single large model. Further details regarding these two inference scenarios can be found in the Appendix. It is also worth noting that in our experiments, the four experts of Multi-Expert and MEME incurred less than 30% of the computation time compared to LDM-L based on A100 GPU. Therefore, the overall training cost requires less than an additional 20% of resources. The qualitative results illustrated in 5 show that the generated images by our methods are superior to those by baseline.

Module Ablation Table 1 provides ablation study for various methods on FFHQ dataset. Firstly, when training with the baseline LDM-S, which involves standard diffusion training, performance drop (FID -2.27, -2.50 for LDM-S, iU-LDM-S, respectively) occurs. Although the incorporation of Multi-Expert mitigates the performance drop to some extent, there is still a degradation (FID -1.28) compared to the baseline LDM-L. In contrast, MEME not only reduces computational cost through the use of smaller-sized denoisers but also achieves performance improvement (FID +0.62).

Additionally, as mentioned in Section 5.2, we found that the soft-expert strategy is more efficient than the hard-expert strategy, where each expert focuses on its designated region. We empirically discovered that a staretegy for training the multi-expert with the expertization probability, denoted as p_n, is beneficial. We configured the probabilities as follows: $p_1 = 0.8$, $p_2 = 0.4$, $p_3 = 0.2$, and $p_4 = 0.1$. Different configurations for p_n are provided in the Appendix.

[4] https://github.com/CompVis/latent-diffusion

[5] https://github.com/GaParmar/clean-fid

[6] https://github.com/sovrasov/flops-counter.pytorch

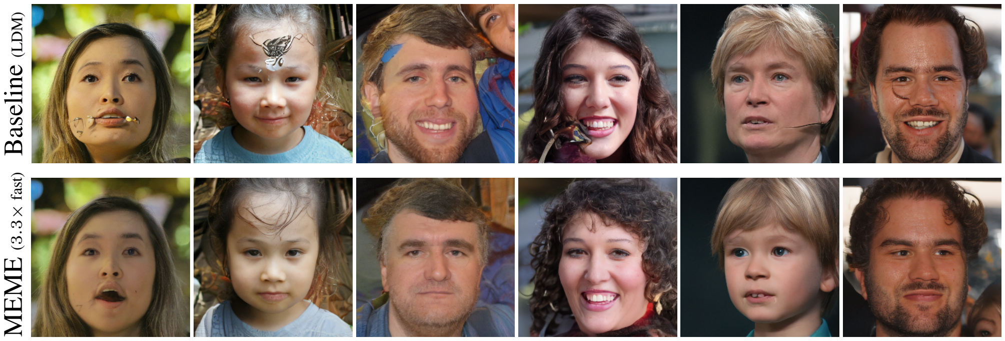
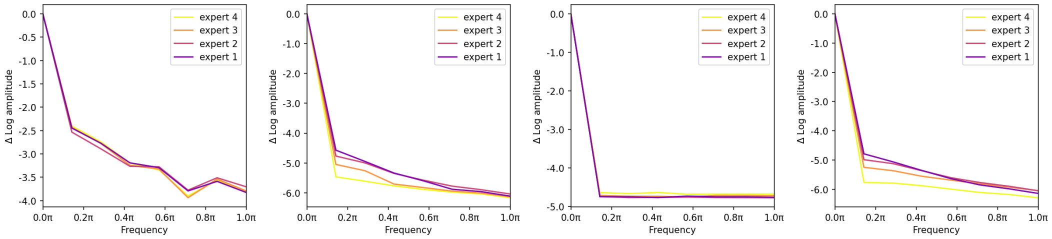


Figure 5: **Samples from baseline LDM-L and MEME trained on FFHQ.** The baseline often generates unnatural aspects in images, whereas our approach MEME shows fewer such cases.



(a) Multi-Expert, $t = 125$ (b) MEME, $t = 125$ (c) Multi-Expert, $t = 875$ (d) MEME, $t = 875$

Figure 6: **Fourier Analysis Comparison between Multi-Expert and MEME** Even with the same input t, we can confirm that MEME exhibits different characteristics for each expert. MEME demonstrates a similar trend as the pre-trained large model shown in Fig. 3, where experts responsible for intervals close to $t = T$ rapidly reduce high frequencies. In contrast, Multi-Expert composed of the same architecture shows that the frequency characteristics of features for the same time-step input are not significantly distinguished from each other.

6.2 Fourier Analysis of MEME

In this section, unlike the analysis shown in Fig. 3, we investigate whether the experts in MEME possess the ability to capture the corresponding frequency characteristics that are advantageous for their respective intervals as illustrated in Fig. 6. MEME, composed of various architectures, exhibits different characteristics for each expert; experts responsible for intervals closer to $t = T$ quickly reduce high frequencies. In contrast, the Multi-Expert, composed of the same architecture, fails to significantly differentiate the frequency characteristics of features when the same time-step input is provided. Particularly for $t = 875$, which requires the ability to capture low-frequency components, it is difficult to distinguish the features of all experts.

6.3 MEME on Top of the Other Diffusion Baseline

In order to explore the generalizability of MEME, we adopted the experimental setup used for architecture validation in [5]. We trained a lightweight version of ADM [7] (referred to as ADM-S) on the CelebA-64 dataset with batch size 8 and 200,000 iterations. The FID measurement was conducted from 10,000 samples from DDIM [50] with 50 steps. The results demonstrate that our MEME exhibits effective performance (FID +6.47) even in the context of ADM. Furthermore, the consistent trend is in line with the results observed in the LDM experiments.

CelebA 64×64		
Model	#Param↓	FID↓
ADM-S	90M	49.56
iU-ADM-S	**82M**(1.1×)	50.08(−0.52)
Multi-Expert	90M ×4	47.29(+2.27)
MEME	**82M** ×4(1.1×)	**43.09**(+6.47)

Table 2: **Results when applied to ADM baseline** MEME outperforms ADM-S baseline [5]. Note that ADM-S already has a much smaller parameter than the original ADM (552.8M).

7 Conclusion

In this paper, we studied the problem of enhancing diffusion models, with the distinction of adopting multiple architectures to suit the specific frequency requirements at different time-step intervals. By incorporating the iU-Net architecture, we provide a more flexible and efficient solution for handling the complex distribution of frequency-specific components across time-steps. Our experiments validated that the proposed method, named **MEME**, outperforms existing baselines in terms of both generation performance and computational efficiency, making it a more practical solution for real-world applications.

References

[1] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[2] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022.

[3] Geoffrey J Burton and Ian R Moorhead. Color and spatial structure in natural scenes. *Applied optics*, 26(1):157–170, 1987.

[4] He Cao, Jianan Wang, Tianhe Ren, Xianbiao Qi, Yihao Chen, Yuan Yao, and Lei Zhang. Exploring vision transformers as diffusion learners. *arXiv preprint arXiv:2212.13771*, 2022.

[5] 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*, pages 11472–11481, 2022.

[6] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[8] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. *arXiv preprint arXiv:2210.05475*, 2022.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[10] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pages 2286–2296. PMLR, 2021.

[11] David J Field. Relations between the statistics of natural images and the response properties of cortical cells. *Josa a*, 4(12):2379–2394, 1987.

[12] Mary Anne Franks and Ari Ezra Waldman. Sex, lies, and videotape: Deep fakes and free speech delusions. *Md. L. Rev.*, 78:892, 2018.

[13] Hyojun Go, Yunsung Lee, Jin-Young Kim, Seunghyun Lee, Myeongho Jeong, Hyun Seung Lee, and Seungtaek Choi. Towards practical plug-and-play diffusion models. *arXiv preprint arXiv:2212.05973*, 2022.

[14] Rafael C Gonzalez and Richard E Woods. Digital image processing. upper saddle river. *J.: Prentice Hall*, 2002.

[15] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022.

[16] 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. *arXiv preprint arXiv:2303.09556*, 2023.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] 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.

[19] 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.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[21] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(47):1–33, 2022.

[22] Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[23] R Huang, MWY Lam, J Wang, D Su, D Yu, Y Ren, and Z Zhao. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 4157–4163. IJCAI: International Joint Conferences on Artificial Intelligence Organization, 2022.

[24] Niharika Jain, Alberto Olmo, Sailik Sengupta, Lydia Manikonda, and Subbarao Kambhampati. Imperfect imaganation: Implications of gans exacerbating biases on facial data augmentation and snapchat face lenses. *Artificial Intelligence*, 304:103652, 2022.

[25] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[26] 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*, pages 4401–4410, 2019.

[27] Heeseung Kim, Sungwon Kim, and Sungroh Yoon. Guided-tts: A diffusion model for text-to-speech via classifier guidance. In *International Conference on Machine Learning*, pages 11119–11133. PMLR, 2022.

[28] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021.

[29] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

[30] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[32] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[33] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

[34] Zhaoyang Lyu, Xudong Xu, Ceyuan Yang, Dahua Lin, and Bo Dai. Accelerating diffusion models via early stop of the diffusion process. *arXiv preprint arXiv:2205.12524*, 2022.

[35] Hengyuan Ma, Li Zhang, Xiatian Zhu, and Jianfeng Feng. Accelerating score-based generative models with preconditioned diffusion sampling. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, pages 1–16. Springer, 2022.

[36] Chenlin Meng, Ruiqi Gao, Diederik P Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. *arXiv preprint arXiv:2210.03142*, 2022.

[37] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2022.

[38] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11410–11420, 2022.

[39] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[40] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, pages 8599–8608. PMLR, 2021.

[41] 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.

[42] 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*, pages 10684–10695, 2022.

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[44] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[45] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

[46] Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937*, 2023.

[47] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. *arXiv preprint arXiv:2211.15736*, 2022.

[48] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng YAN. Inception transformer. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[49] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

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

[51] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[52] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[53] 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.

[54] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[55] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[57] Zhicong Tang, Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Improved vector quantized diffusion models. *arXiv preprint arXiv:2205.16007*, 2022.

[58] David J Tolhurst, Yoav Tadmor, and Tang Chao. Amplitude spectra of natural images. *Ophthalmic and Physiological Optics*, 12(2):229–232, 1992.

[59] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[60] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[61] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

[62] van A Van der Schaaf and JH van van Hateren. Modelling the power spectra of natural images: statistics and information. *Vision research*, 36(17):2759–2770, 1996.

[63] Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. In *International Conference on Learning Representations*, 2022.

[64] Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*, 2021.

[65] Norbert Wiener, Norbert Wiener, Cyberneticist Mathematician, Norbert Wiener, Norbert Wiener, and Cybernéticien Mathématicien. *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*, volume 113. MIT press Cambridge, MA, 1949.

[66] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

[67] Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. *arXiv preprint arXiv:2211.17106*, 2022.

[68] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. LION: Latent point diffusion models for 3d shape generation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[69] Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models. *stat*, 1050:7, 2022.

[70] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

[71] Barret Zoph and Quoc Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017.

Appendix: Multi-Architecture Multi-Expert Diffusion Models

Contents

A Experimental Details

In this section, we provide the details of experiments in Section 6. All experiments are conducted with a single A100 GPU.

A.1 Experimental Details for LDM Baseline

Commonly used hyperparameters of the models involved in image generation experiments on FFHQ [26] and CelebA [25] datasets are presented in Table. 3. Additionally, the hyperparameters specific to our expert models within the Multi-architecturE Multi-Expert (MEME) framework, employing the iU-Net, are outlined in Table. 4. Our implementation for the models in experiments is based on the official Latent Diffusion Models (LDM [42]) repository[7].

In the FFHQ experiments, all models generate 50K samples for evaluation via the Denoising Diffusion Implicit Models (DDIM [50]) with a 200-step sampling process. The Fréchet Inception Distance (FID [18]) is computed utilizing the Clean-FID [38] official code[8], with the entire 70K FFHQ dataset serving as the reference image set. The constancy of this reference set bolsters the reproducibility of the Clean-FID computations. The LDM-L model, trained for 635K iterations, is incorporated with pretrained weights obtained from the LDM [42] official repository. In contrast, another model is independently trained for 540K iterations to provide a comparable measure against Multi-Expert and MEME, specifically in terms of GPU memory and time costs. For both Multi-Expert and MEME, the batch size is set twice that of the large model, equating the GPU memory costs when using a single A100 GPU. This configuration leads to an approximate usage of 50GB VRAM in the given system. The process of sequentially training four small experts, each for 135k iterations on a single A100 GPU, exhibits similar time costs to training a large model for 520K iterations.

For experiments involving the CelebA-HQ [25] dataset, all models are subject to a DDIM 50-step sampling process to generate 50K samples. To align experimental settings with the FFHQ dataset, the Clean-FID score is computed with the entire 30K CelebA-HQ dataset employed as the reference image set.

Table 3: Hyperparameters for the LDMs producing the numbers shown in Table. 1. All models are trained on a single NVIDIA A100. Further details for iU-LDM-S and MEME architectures are shown in 4

	Large		Small	
	CelebA-HQ 256×256	FFHQ 256×256	CelebA-HQ 256×256	FFHQ 256×256
f	4	4	4	4
z-shape	$64 \times 64 \times 3$	$64 \times 64 \times 3$	$64 \times 64 \times 3$	$64 \times 64 \times 3$
$\|\mathcal{Z}\|$	8192	8192	8192	8192
Diffusion steps	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear
N_{params}	274M	274M	89.5M	89.5M
Channels	224	224	128	128
Depth	2	2	2	2
Channel Multiplier	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
Attention resolutions	32, 16, 8	32, 16, 8	32, 16, 8	32, 16, 8
Head Channels	32	32	32	32
Batch Size	48	42	96	84
Iterations*	410k	520k, 635k	85k	135k
Learning Rate	8.4e-5	9.6e-5	1.68e-4	1.92e-4

A.2 Experimental Details for ADM-S Baseline

In an effort to investigate the potential for employing MEME within the scope of pixel-level diffusion models other than LDM, we incorporate the specific experimental configurations previously utilized by Choi *et al.* [5]. These configurations are originally devised for the purpose of validating p2-weighting [5] model architecture. In this particular experimental context, we have implemented the small ADM model[7], termed ADM-S, equipped with a total of 90 million parameters. Fundamentally,

[7] https://github.com/CompVis/latent-diffusion

[8] https://github.com/GaParmar/clean-fid

Table 4: Configurations of the proposed expert models with iU-Net. † denotes the architecture used for iU-LDM-S

Stage	Layer	expert† Θ_1	expert Θ_2	expert Θ_3	expert Θ_4
enc#1	iFormer Block	$\begin{bmatrix} 3\times3, \text{stride } 1, 128 \\ \left\{\begin{matrix} d_h/d=3/4 \\ d_l/d=1/4 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 128 \\ \left\{\begin{matrix} d_h/d=3/4 \\ d_l/d=1/4 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 128 \\ \left\{\begin{matrix} d_h/d=3/4 \\ d_l/d=1/4 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 128 \\ \left\{\begin{matrix} d_h/d=3/4 \\ d_l/d=1/4 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$
	Res Block	$3\times3, \text{stride } 1, 128$	$3\times3, \text{stride } 1, 128$	$3\times3, \text{stride } 1, 128$	$3\times3, \text{stride } 1, 128$
enc#2	iFormer Block	$\begin{bmatrix} 3\times3, \text{stride } 1, 256 \\ \left\{\begin{matrix} d_h/d=5/8 \\ d_l/d=3/8 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 256 \\ \left\{\begin{matrix} d_h/d=1/2 \\ d_l/d=1/2 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 256 \\ \left\{\begin{matrix} d_h/d=3/8 \\ d_l/d=5/8 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 256 \\ \left\{\begin{matrix} d_h/d=1/4 \\ d_l/d=3/4 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$
	Res Block	$3\times3, \text{stride } 1, 256$	$3\times3, \text{stride } 1, 256$	$3\times3, \text{stride } 1, 256$	$3\times3, \text{stride } 1, 256$
enc#3	iFormer Block	$\begin{bmatrix} 3\times3, \text{stride } 1, 384 \\ \left\{\begin{matrix} d_h/d=1/2 \\ d_l/d=1/2 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 384 \\ \left\{\begin{matrix} d_h/d=3/8 \\ d_l/d=5/8 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 384 \\ \left\{\begin{matrix} d_h/d=1/4 \\ d_l/d=3/4 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 384 \\ \left\{\begin{matrix} d_h/d=1/8 \\ d_l/d=7/8 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$
	Res Block	$3\times3, \text{stride } 1, 384$	$3\times3, \text{stride } 1, 384$	$3\times3, \text{stride } 1, 384$	$3\times3, \text{stride } 1, 384$
enc#4	iFormer Block	$\begin{bmatrix} 3\times3, \text{stride } 1, 512 \\ \left\{\begin{matrix} d_h/d=1/4 \\ d_l/d=3/4 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 512 \\ \left\{\begin{matrix} d_h/d=1/8 \\ d_l/d=7/8 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 512 \\ \left\{\begin{matrix} d_h/d=1/16 \\ d_l/d=15/16 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, \text{stride } 1, 512 \\ \left\{\begin{matrix} d_h/d=1/16 \\ d_l/d=15/16 \\ \text{pool stride } 2 \end{matrix}\right\} \end{bmatrix} \times 2$
#Param. (M)		82.56	82.85	83.11	83.32

our experimental model implementation is based on the official repository[9] of Choi *et al.* [5]. The hyperparameters pertaining to this experiment can be found in Table. 5.

Table 5: Hyperparameters for the ADM-S [7] producing the numbers shown in Table. 2. All models are trained on a single NVIDIA A100. ‡ denotes: average value across four architectures.

	ADM-S	iU-ADM-S	Multi-Expert	MEME
T	1000	1000	1000	1000
β_t	linear	linear	linear	linear
Model Size	90	82	90 ×4	$82^{\ddagger}$ ×4
Channels	128	128	128	128
Blocks	1	1	2	2
Self-attn	bottle	bottle	bottle	bottle
Heads Channels	64	32	64	64
BigGAN Block	yes	yes	yes	yes
Dropout	0.1	0.1	0.1	0.1
Learning Rate	$2e^{-5}$	$2e^{-5}$	$2e^{-5}$	$2e^{-5}$
Images (M)	1.6	1.6	1.6	1.6

A.3 Practical Benefits and Limitations

The foremost benefit derived from the multi-expert strategy is the considerable reduction in computational time costs. This mirrors the empirical observations made by Balaji *et al.* [1], who found that within a practical setting, the total inference speed of the model does not vary with the number of experts, N. The inference speed stays constant with increasing N, as it's defined by the average model size of the experts.

However, a potential limitation of this approach lies in the associated memory cost. If all the expert models are loaded into the GPU system memory, the memory cost increases proportionally with the number of experts, N. Despite this, the multi-expert strategy for diffusion models provides an option to prioritize memory efficiency over computational time costs under memory-critical circumstances. This can be achieved by loading only the expert model responsible for inference at each time-step into the GPU system memory, thereby reducing memory cost at the expense of computation time.

[9] https://github.com/jychoi118/P2-weighting

Moreover, in situations where there is a demand to process large sample inferences simultaneously, the multi-expert strategy can offer drastic reductions in both system memory cost and computational time costs. This is possible by loading one expert model into the GPU system memory and processing features up to its limit. Intermediary outputs can then be stored in disk memory before unloading the current expert from the GPU system memory. The same procedure is then sequentially repeated for each subsequent expert, resulting in significant overall savings.

B The Effects of Expertization Probability in Soft-Expert

In establishing the soft-expert strategy, we hypothesize that experts dealing with more highly diffused inputs suffer from learning meaningful semantics. Therefore, we posit that setting the expertization probabilities, p_n, to decrease as n increases ($p_1 \geq \cdots \geq p_N$) would be more effective than maintaining them all constant. We conducted an experimental comparison to test this hypothesis, contrasting hard-expert [13], soft-expert with constant p_n, and soft-expert with decreasing p_n.

FFHQ 256×256		
Expert Strategy	Multi-Expert	MEME
[1.0, 1.0, 1.0, 1.0]	10.42	9.20
[0.6, 0.6, 0.6, 0.6]	10.13	9.08
[0.8, 0.4, 0.2, 0.1]	9.58	8.52

Table 6: FID values on the FFHQ dataset depending on how the expertization probability is assigned. The 'Expert Strategy' column represents $[p_1, p_2, p_3, p_4]$. In this context, [1.0, 1.0, 1.0, 1.0] denotes the hard-expert [13], [0.6, 0.6, 0.6, 0.6] denotes the soft-expert with a constant expertization probability, and [0.8, 0.4, 0.2, 0.1] denotes the soft-expert with decreasing expertization probabilities.

The results of this comparative study are presented in Table. 6. The elements in the Expert Strategy column represent $[p_1, p_2, p_3, p_4]$. Thus, [1.0, 1.0, 1.0, 1.0] denotes the hard-expert [13], [0.6, 0.6, 0.6, 0.6] denotes the soft-expert with a constant expertization probability, and [0.8, 0.4, 0.2, 0.1] denotes the soft-expert with decreasing expertization probability. The training and evaluation process details followed those outlined in Section. 6.1 and Section. A.1.

C The Effects of The Number of Experts

As noted in Section. A.3, the total inference time in the multi-expert strategy does not increase as the number of experts N increases. However, the cost of GPU memory may increase proportionally with N. To understand the performance difference based on the number of experts, we varied the number of experts in a multi-expert setting with hard expertization probability and measured the FID [18] value.

FFHQ 256×256		
	#Expert N	FID
LDM-S Multi-Expert	$N = 2$	10.97
	$N = 4$	10.42
	$N = 6$	10.28

Table 7: Changes in FID according to the number of experts N in LDM-S Multi-Expert training. All results were trained under the hard-expert expertization probability setting.

In Table. 7, we show the FID values obtained from training the multi-expert model with different values of N: 2, 4, and 6. The performance improvement from $N = 2$ to $N = 4$ is substantial, but there is not a large increase from $N = 4$ to $N = 6$. Hence, we have set $N = 4$ as our default value. The details of the training and evaluation process followed those outlined in Section. 6.1 and Section. A.1.

D Experimental Details for Fourier Analysis

Following Park *et al.* [37], we analyze the feature maps in the Fourier space to confirm what frequency the pretrained large diffusion models focus on at each time-step (as shown in Fig. 3), or to verify that our proposed MEME learns the characteristics of the responsible intervals more effectively than a single architecture multi-expert (as shown in Fig. 6).

We perform a Fourier transformation on the feature maps, converting them into a two-dimensional frequency domain. These converted feature maps are then represented in a normalized frequency domain, where the highest frequency components correspond to $f = -\pi, +\pi$, while the lowest frequency components coincide with $f = 0$. To enhance the clarity of our visualizations, we focus on presenting only the half-diagonal components. The features required for conducting Fourier analysis are calculated based on the average of features derived from 10,000 randomly sampled input data from the FFHQ [26] dataset.

The approach of Park *et al*. [37] involves visualizing the Δ log amplitude for all layers within a single model. However, in our analysis, we incorporate multi-expert diffusion models which add two additional dimensions: the time-step and expert models. This necessitates a different visualization approach, where instead of plotting values corresponding to multiple layers within one model, we chart values per time-step (Fig. 3), or alternatively, per expert model (Fig. 6). This enables a more nuanced understanding of how each time-step is handled across the denoising process, or how each expert model performs.

E Societal Impacts

Generative models, including diffusion models, have the potential to significantly impact society, particularly in the context of DeepFake applications [12] and biased data [24, 59]. One of the key concerns lies in the amplification of misinformation and the erosion of trust in visual media. Furthermore, if generative models are trained on biased data or intentionally manipulated content, they can inadvertently perpetuate and exacerbate social biases [24], leading to the dissemination of misleading information and the manipulation of public perception.

F Limitations

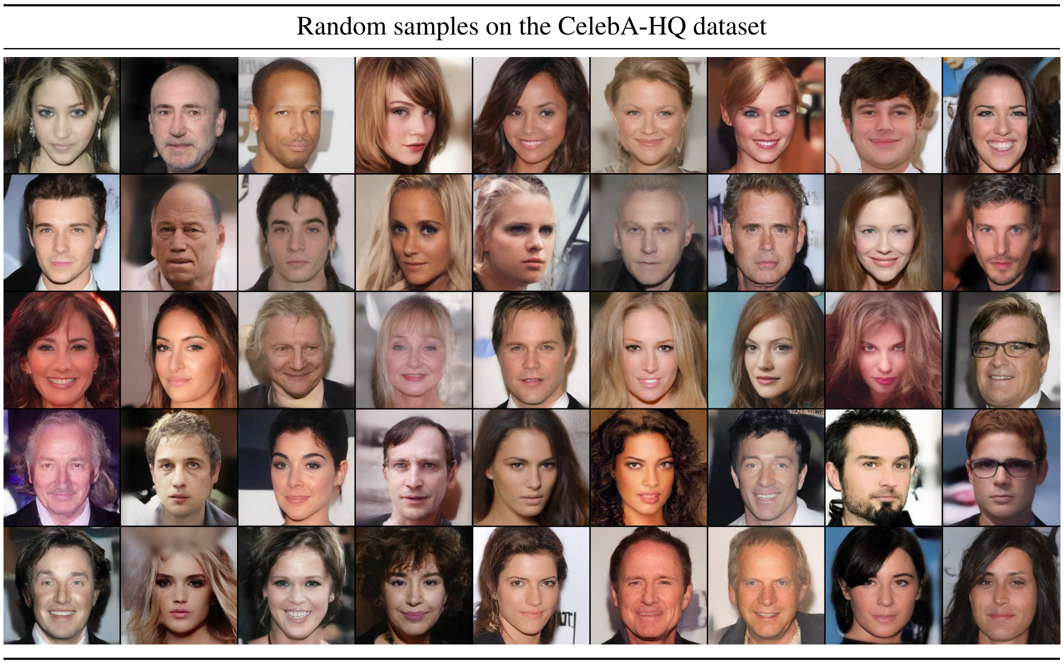
Our research highlights the significance of customizing architectural designs to align with the specific timestep of diffusion models. In order to achieve this, our primary focus lies in tuning the operations within each layer through the modulation of the mixing ratio between convolution and self-attention. However, there are two limitations that can be addressed in future work.

Firstly, our research recognizes the yet unexplored territory of determining the optimal mixing ratio between convolution and self-attention. Although we demonstrate that increased convolution leads to enhanced performance in latent spaces with lower noise, the precise optimization of the mixing ratio remains a task yet to be accomplished, similar to the advancements achieved through neural architecture search [71]. To address this, introducing a neural architecture scheme that adapts to varying timesteps can hold significant potential for advancing diffusion models.

Secondly, our research does not delve into exploring other architectural design factors, such as pooling techniques. Future work can focus on investigating the impact of different pooling techniques on the performance of diffusion models. Additionally, exploring the combination of convolution, self-attention, and other architectural elements, such as residual connections [17] or skip connections [43], could provide further insights into optimizing the overall architecture for diffusion models.

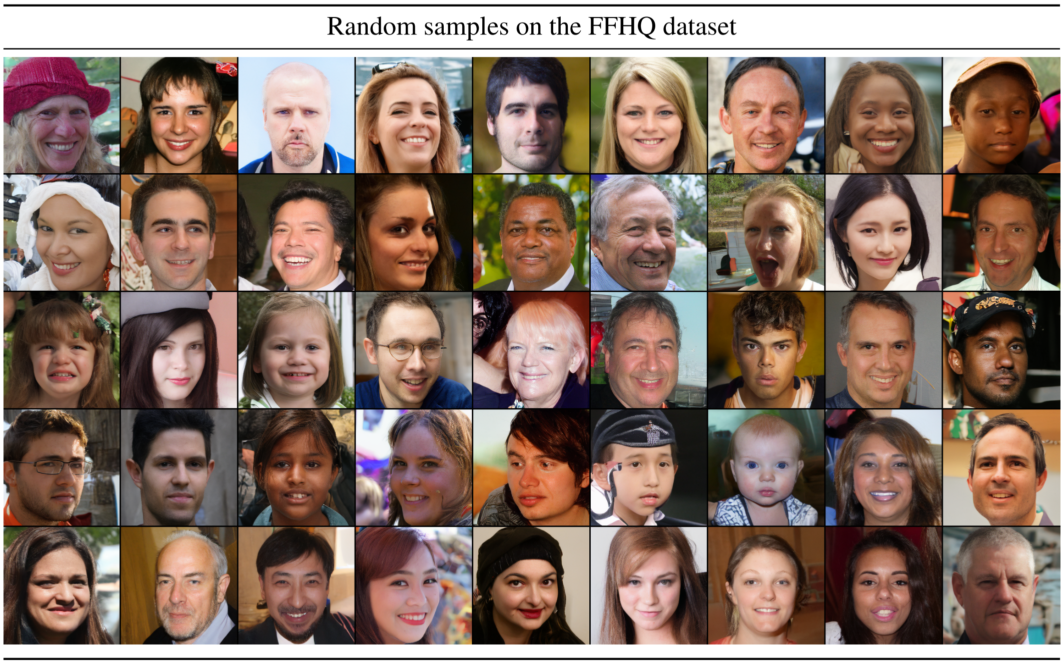
G Qualitative Resutls

We provide additional qualitative results for our MEME models for the CelebA-HQ [25], and FFHQ datasets (Fig. 7 - 8).



Random samples on the CelebA-HQ dataset

Figure 7: Random samples of our MEME on the CelebA-HQ dataset. Sampled with 50 DDIM steps and $\eta = 0$.



Random samples on the FFHQ dataset

Figure 8: Random samples of MEME on the FFHQ dataset. Sampled with 200 DDIM steps and $\eta = 1$.