A Diffusion Model Translator for Efficient Image-to-Image Translation

Mengfei Xia, Yu Zhou, Ran Yi, Yong-Jin Liu, *Senior Member, IEEE*, and Wenping Wang, *Fellow, IEEE*

***Abstract*—Applying diffusion models to image-to-image translation (I2I) has recently received increasing attention due to its practical applications. Previous attempts inject information from the source image into each denoising step for an iterative refinement, thus resulting in a time-consuming implementation. We propose an efficient method that equips a diffusion model with a lightweight translator, dubbed a Diffusion Model Translator (DMT), to accomplish I2I. Specifically, we first offer theoretical justification that in employing the pioneering DDPM work for the I2I task, it is both feasible and sufficient to transfer the distribution from one domain to another only at some intermediate step. We further observe that the translation performance highly depends on the chosen timestep for domain transfer, and therefore propose a practical strategy to automatically select an appropriate timestep for a given task. We evaluate our approach on a range of I2I applications, including image stylization, image colorization, segmentation to image, and sketch to image, to validate its efficacy and general utility. The comparisons show that our DMT surpasses existing methods in both quality and efficiency. Code is available at https://github.com/THU-LYJ-Lab/dmt.**

***Index Terms*—Diffusion models, image translation, deep learning, generative models.**

I. Introduction

A DIFFUSION probabilistic model [1], [2], [3], [4], also known as a diffusion model, is a generative model that consists of 1) a forward diffusion process that gradually adds noise to a data distribution until it becomes a simple latent distribution (e.g., Gaussian), and 2) a reverse process that begins with a random sample in the latent distribution and employs a learned network to revert the diffusion process, thereby generating a data point in the original distribution. Among all the variants of the diffusion model, the denoising diffusion probabilistic model (DDPM) [2] offers the advantage of a simple training procedure by exploring an explicit connection between the diffusion model and denoising score matching. Recent studies have demonstrated the compelling performance of DDPM in high-fidelity image synthesis [2], [5], [6].

Despite its rapid development, there are relatively few studies on applying the diffusion model to conditional generation, which is a key requirement for many real-world applications, such as the well-known image-to-image (I2I) task [7] that translate a source image of one style into another target image of a different style. Unlike unconditional generation, conditional generation necessitates constraining synthesized result with an input sample in the source domain as the content guidance. Therefore, to handle an I2I task using DDPM, existing methods [8], [9], [10], [11], [12] inject the information from an input source sample into every single denoising step in the reverse process (see Fig. 1(a)). In this way, each denoising step explicitly relies on its previous step, making it inefficient to learn the step-wise injection.

In this work, we investigate a more efficient approach to applying DDPM to I2I tasks by endowing a pre-trained DDPM with a translator, which we name *Diffusion Model Translator* (DMT). First, we provide a *theoretical* proof that given two diffusion processes on two different image domains involved in an I2I task, it is feasible to accomplish the I2I task by shifting a distribution from one process to another at a particular timestep with appropriate reparameterization. Based on this theoretical justification, we develop a new efficient DDPM pipeline, as illustrated in Fig. 1(b). Assuming that a DDPM has been prepared for one image domain y_0, we use it to decode the latent that is shifted from another domain x_0. To accomplish the domain shift, we apply the same forward diffusion process onto x_0 and y_0 until a pre-defined timestep t, and then employ a neural network to translate x_t to y_t as a typical I2I problem.

There are two major advantages to our approach. First, the training of DMT is independent of DDPM and can be executed very efficiently. Second, DMT can benefit from using all the previous techniques in the I2I field (e.g., such as Pix2Pix [7], TSIT [13], SPADE [14], and SEAN [15]), for a better performance. Furthermore, regarding the choice of the timestep t to perform domain transfer, we propose a practical strategy to automatically select an appropriate timestep for a given data distribution.

To empirically validate the efficacy of our method, we conducted evaluation on four I2I tasks: image stylization, image colorization, segmentation to image, and sketch to image. Both

Manuscript received 24 March 2024; accepted 23 July 2024. Date of publication 30 July 2024; date of current version 5 November 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U2336214, Grant 62332019, and Grant 62302297, in part by Beijing Natural Science Foundation under Grant L222008, in part by Shanghai Sailing Program under Grant 22YF1420300, and in part by Young Elite Scientists Sponsorship Program by CAST under Grant 2022QNRC001. Recommended for acceptance by G. Hua. *(Corresponding authors: Ran Yi; Yong-Jin Liu.)*

Mengfei Xia, Yu Zhou, and Yong-Jin Liu are with the MOE-Key Laboratory of Pervasive Computing, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: xmf20@mails.tsinghua.edu.cn; yzhou20@mails.tsinghua.edu.cn; liuyongjin@tsinghua.edu.cn).

Ran Yi is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: ranyi@sjtu.edu.cn).

Wenping Wang is with the Department of Computer Science and Computer Engineering, Texas A&M University, College Station, TX 77840 USA (e-mail: wenping@tamu.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3435448, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3435448

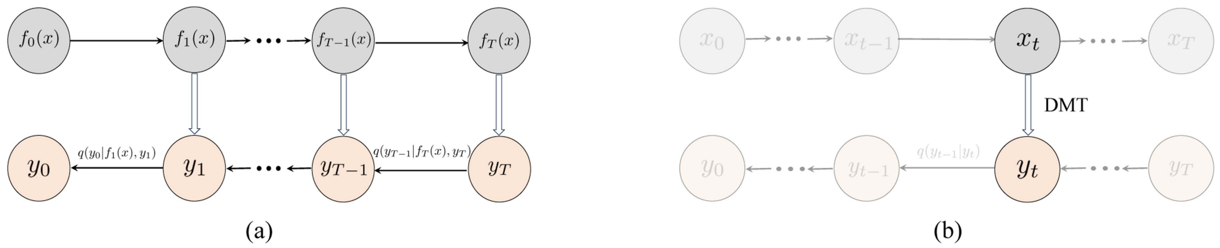


Fig. 1. *Conceptual comparison* between (a) existing methods [9], [10], [11] and (b) our DMT. $\{x_t\}_{t=0}^{T}$ represent different states of the input from the source domain, while $y_T \to y_0$ stands for the denoising process of DDPM. Here, T denotes the total number of noise-adding steps in the diffusion process. Instead of using the information $f_t(x)$ from the source domain (which can be the original or noisy image) for an iterative refinement at *each* denoising step $t, t = 0, 1, \dots, T$, DMT accomplishes the I2I task efficiently by learning an efficient translation module at just one *preset* timestep and fully reusing the pre-trained DDPM. How to select an appropriate translation timestep is discussed in Section III-D.

qualitative and quantitative results demonstrate the superiority of our method over existing diffusion-based alternatives as well as the GAN-based counterparts of DMT.

II. Related Work

In a forward diffusion process, a *Diffusion probabilistic model (DPM)* [1], [2] transforms a given data distribution into a simple latent distribution, such as a Gaussian distribution. Due to its strong capabilities, DPM has achieved great success in various fields, including speech synthesis [16], [17], video synthesis [18], [19], image super-resolution [20], [21], conditional generation [10], [12], and image-to-image translation [8], [9]. Denoising diffusion probabilistic model (DDPM) [2] assumes the Markovian property of the forward diffusion process. For a dataset of images, the forward diffusion process is realized by corrupting each image x_0 through the addition of standard Gaussian noise to reduce it into a completely random noise image. Formally, given the variance schedules $\alpha_t \in [0, 1], t = 1, 2, \dots, T, \beta_t = 1 - \alpha_t$, we can write the Markov chain as:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \tag{1}$$

$$q(x_t|x_{t-1}) \sim \mathcal{N}(x_t; \sqrt{\alpha_t} x_{t-1}, \beta_t I), \tag{2}$$

where $x_T \sim \mathcal{N}(x_T; 0, I)$ and I is the identity matrix.

When reversing this diffusion process, DDPM serves as a generator for data generation in the form $p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}$ starting from x_T:

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \tag{3}$$

$$p_\theta(x_{t-1}|x_t) \sim \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{4}$$

so that any sample x_T in the latent distribution will be mapped back to x_0 in the original data distribution. To achieve its reverse process for image synthesis, DDPM parameterizes the mean $\mu_\theta(x_t, t)$ by a time-dependent model $\epsilon_\theta(x_t, t)$ and optimizes the following simplified objective function:

$$\mathcal{L} = \mathbb{E}_{q(x_0,t,\epsilon)} \left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|^2 \right]. \tag{5}$$

Faster DPM attempts to explore shorter trajectories rather than the complete reverse process, while ensuring that the synthesis performance is comparable to the original DPM. Some existing methods seek the trajectories using the grid search [16]. However, this is only suitable for short reverse processes because its time complexity grows exponentially. Other methods try to find optimal trajectories by solving a least-cost-path problem with a dynamic programming (DP) algorithm [22], [23]. Another representative category of fast sampling methods uses high-order differential equation (DE) solvers [24], [25], [26], [27], [28]. Some GAN-based methods also consider larger sampling step size. For instance, [29] demonstrates learning a multi-modal distribution within a conditional GAN using a larger step size.

Image-to-image translation (I2I) aims to translate an input image from a given source domain to another image in a given target domain, with input-output paired training data [7]. To this end, the conditional generative adversarial network (cGAN) is designed to inject the information of the input image into the generation decoder with the adversarial loss [30], [31]. The cGAN-based algorithms has demonstrated high quality on many I2I tasks [13], [14], [15], [32], [33], [34], [35], [36], [37], [38]. However, due to their training instability and the severe mode collapse issue, it is hard for the cGAN-based methods to generate diverse high-resolution images. Recently, DPM has been applied to the I2I task. Palette [9] introduces the novel DPM framework to the I2I task by injecting the input into each sampling step for refinement. Some methods use pre-trained image synthesis models for the I2I task [12]. Despite the high quality of synthesized images, the generation process of these existing methods is extremely time-consuming. Our work tackles this issue by proposing a new DDPM method for the I2I task that works efficiently, without the time-consuming requirement of having to inject the input source information in every denoising step. Although unpaired data are more accessible for translation tasks, the advantages of paired image-to-image (I2I) tasks, such as reduced data demands and enhanced synthesis quality, have made them a significant research focus.

III. Method

A. Markov Process of Translation Mappings

For an I2I task, traditional DDPM methods directly approximate the real distribution $q(y_0|x_0)$ in which x_0, y_0 are paired data from the source domain $\mathcal{D}_x$ and the target domain $\mathcal{D}_y$, respectively. In contrast, we construct a translation module

$p_\theta(y_t|x_t)$, which bridges the input condition and the pre-trained DDPM. Accordingly, we can approximate the $q(y_0|x_0)$ using the learned intermediate translation module. Specifically, given a noise-adding schedule of the forward variance process $\beta_i \in [0,1], t = 1, 2, \ldots T$, $\alpha_i = 1 - \beta_i$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_t$, we first generalize the forward Markov process to the joint distribution of $(x_{1:t}, y_{1:t})$ as below:

$$q(y_{1:t}, x_{1:t}|y_0, x_0) = \prod_{i=1}^{t} q(x_i|x_{i-1}) \prod_{j=1}^{t} q(y_j|y_{j-1}), \tag{6}$$

$$q(x_i|x_{i-1}) \sim \mathcal{N}(x_i; \sqrt{\alpha_i} x_{i-1}, \beta_i I), \tag{7}$$

$$q(x_t|x_0) \sim \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I), \tag{8}$$

$$q(y_j|y_{j-1}) \sim \mathcal{N}(y_j; \sqrt{\alpha_i} y_{i-1}, \beta_i I), \tag{9}$$

$$q(y_t|y_0) \sim \mathcal{N}(y_t; \sqrt{\bar{\alpha}_t} y_0, (1 - \bar{\alpha}_t) I). \tag{10}$$

The corresponding DDPM trained on the target domain provides a reverse Markov process to approximate $q(y_0)$ from a sample y_T drawn from the standard Gaussian distribution, i.e., $y_T \sim \mathcal{N}(y_T; 0, I)$. Note that during the denoising process, y_i is only determined by y_{i+1} and irrelevant to $x_{0:t}$ for $i \in [0, t-1]$. We choose to construct the translation mapping at some specified step[1] of the diffusion forward process using $p_\theta(y_t|x_t)$, which induces the following Markov process:

$$p_\theta(y_{0:t}, x_{1:t}|x_0) = p_\theta(y_t|x_t) \prod_{i=1}^{t} q(x_i|x_{i-1}) \prod_{j=1}^{t} q(y_{j-1}|y_j), \tag{11}$$

where $q(y_{j-1}|y_j)$ is the denoising process of the pre-trained DDPM.

B. Translation Mappings of DDPM

Let $p_\theta(y_0|x_0) = \int p_\theta(y_{0:t}, x_{1:t}|x_0) dy_{1:t} dx_{1:t}$ represent the sampling distribution of $q(y_0|x_0)$, where $p_\theta(y_t|x_t)$ serves to bridge the two domains. By making use of the variational lower bound to optimize the negative log-likelihood, we have the following lemma:

Lemma 1: The negative log-likelihood of $-\log p_\theta(y_0|x_0)$ has the following upper bound,

$$-\log p_\theta(y_0|x_0) \leqslant \mathbb{E}_q \left[\log \frac{q(y_{1:t}, x_{1:t}|y_0, x_0)}{p_\theta(y_{0:t}, x_{1:t}|x_0)} \right], \tag{12}$$

where $q = q(y_{1:t}, x_{1:t}|y_0, x_0)$.

In other words, the translation mapping can be learned by optimizing the variational lower bound:

$$\mathcal{L}_{CE} = \nmid\mathbb{E}_{q(y_0|x_0)} [\log p_\theta(y_0|x_0)] \tag{13}$$

$$\leqslant \mathbb{E}_{q(y_{0:t}, x_{1:t}|x_0)} \left[\log \frac{q(y_{1:t}, x_{1:t}|y_0, x_0)}{p_\theta(y_{0:t}, x_{1:t}|x_0)} \right] := \mathcal{L}_{VLB}. \tag{14}$$

First, we claim that the optimal $p_\theta(y_t|x_t)$ follows a Gaussian distribution up to a non-negative constant of (13).

Theorem 1 (Closed-Form Expression): The loss function in (13) has a closed-form representation. The training is equivalent to optimizing a KL-divergence up to a non-negative constant, i.e.,

$$\mathcal{L}_{VLB} = \mathbb{E}_{q(y_0, x_t|x_0)} \left[D_{KL}(q(y_t|y_0) \| p_\theta(y_t|x_t)) \right] + C. \tag{15}$$

For the given closed-form expression in (15), the optimal $p_\theta(y_t|x_t)$ follows a Gaussian distribution and its mean μ_θ has an analytic form, as summarized in the Theorem 2 below:

Theorem 2 (Optimal Solution to (15)): The optimal $p_\theta(y_t|x_t)$ follows a Gaussian distribution with its mean being

$$\mu_\theta(x_t) = \sqrt{\bar{\alpha}_t} y_0. \tag{16}$$

Detailed proofs of the above lemma and theorems are provided in Appendix B, available online.

C. Reparameterization of μ_θ

Given the DDPM trained on the target domain, we first apply the same diffusion forward process on both x_0 and y_0 as a shared encoder to represent the mean $\mu_\theta(x_t)$:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} z_t, \quad y_t = \sqrt{\bar{\alpha}_t} y_0 + \sqrt{1 - \bar{\alpha}_t} z_t. \tag{17}$$

Theorem 2 reveals that μ_θ needs to approximate the expression $\sqrt{\bar{\alpha}_t} y_0$ with x_t as the only available input. Then, we apply the following parameterization,

$$\mu_\theta(x_t) = f_\theta(x_t) - \sqrt{1 - \bar{\alpha}_t} z(x_t), \tag{18}$$

where f_θ is a trainable function and $z(x_t) = z_t$, which is set to the shared noise component of x_0 and y_0. The KL-divergence in (15) is optimized by minimizing the difference between the two means together with the variance Σ_θ of $p_\theta(y_t|x_t)$. Noting that $\Sigma_\theta = (1 - \bar{\alpha}_t) I$, the objective function then has the following form,

$$\mathcal{L}_t = \mathbb{E}_q \left[\frac{1}{2(1 - \bar{\alpha}_t)} \| f_\theta(x_t) - y_t \|^2 \right]. \tag{19}$$

Equation (18) implies that inferring $y_t \sim p_\theta(y_t|x_t)$ is to compute $f_\theta(x_t) - \sqrt{1 - \bar{\alpha}_t} z_t + \sqrt{1 - \bar{\alpha}_t} z$, where $z \sim \mathcal{N}(0, I)$.

D. Determining an Appropriate Timestep for Translation

Recall that we encode the same forward diffusion process onto both x_0 and y_0 using a shared encoder (ref. to (17)), where z_t is independent of x_0 and y_0. As t tends to T, x_t and y_t will converge to the same Gaussian noise simultaneously, since $x_t, y_t \rightarrow z_T \sim \mathcal{N}(0, I)$. Hence, as t increases, the distance between (x_t, y_t) will decrease and the distance between (x_0, x_t) will increase. In other words, the training of DMT faces a trade-off between the gap between the two potential domains and the strength of the condition signal. The larger timestep t makes it easier for the DMT to learn the translation mapping, while the strength of inference information will be weakened since the injected noise corrupts the origin signal.

To address this trade-off issue, we provide a theoretical analysis below. Recall that our proposed diffusion-model-based I2I system consists of three sub-systems: 1) the forward diffusion

[1] The selection of this specified step is discussed in Section IV.

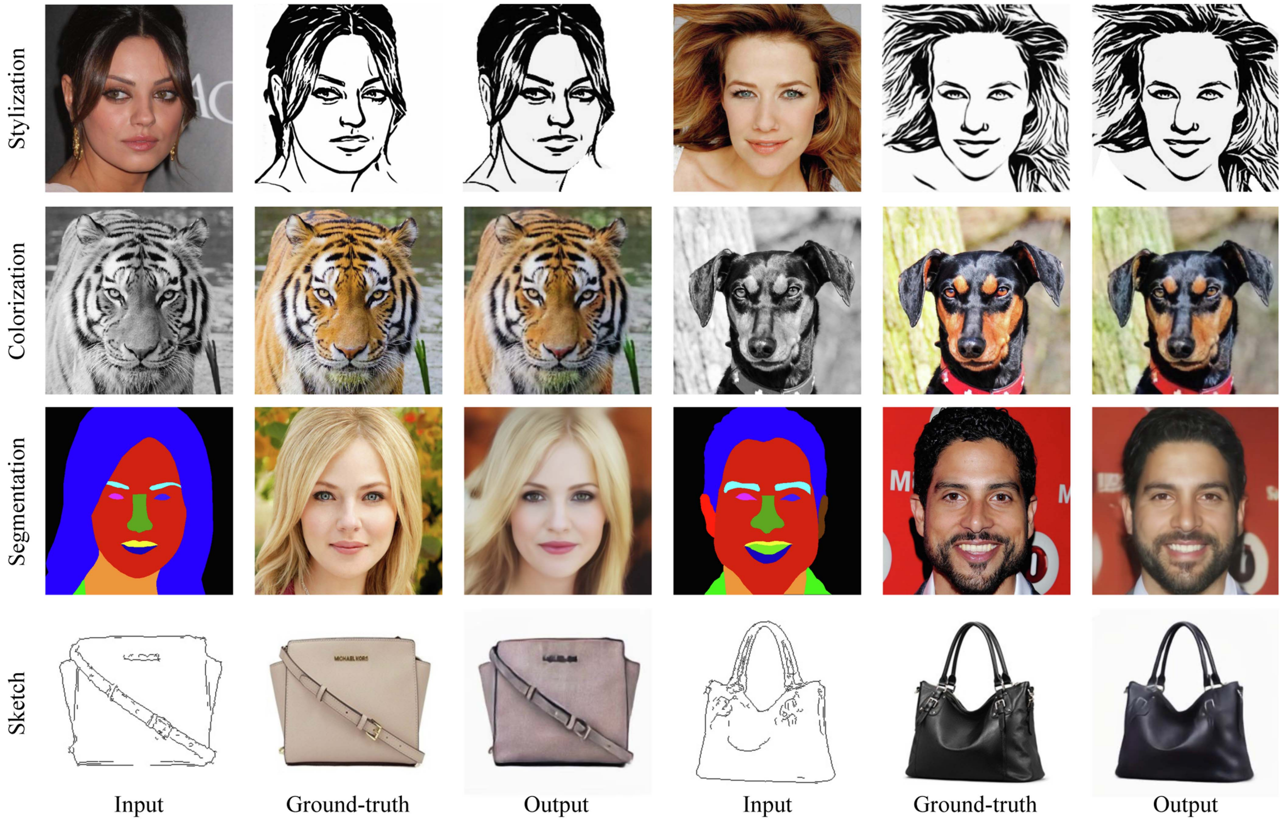


Fig. 2. *Qualitative results* of our proposed DMT on four I2I tasks: image stylization, image colorization, segmentation to image, and sketch to image. Here we equip a pre-trained DDPM with an efficient translation module. Our approach makes adequate use of the content information from the input condition as well as the domain knowledge contained in the learned denoising process.

process from x_0 to x_t, 2) DMT from x_t to y_t, and 3) the denoising process via pre-trained diffusion model from y_t to y_0. Our analysis is based on the following observation: the complexity C of the whole system S is determined by the maximal one among the complexities of three sub-systems (S_1, S_2, S_3), i.e., $C(S) = \max\{C(S_1), C(S_2), C(S_3)\}$. Given a timestep t, let $C(S_1) = f(t)$, $C(S_2) = g(t)$, $C(S_3) = h(t)$, where $f(t)$, $g(t)$ and $h(t)$ are complexity curves of diffusing x_0 to x_t, translating x_t to y_t, and denoising y_t to y_0 w.r.t. the timestep t, respectively. First, we assume[2] $f(t) \approx h(t)$. Then $C(S) = \max\{f(t), g(t)\}$. Second, we assume[3] that $f(t)$ and $g(t)$ are monotone curves. Then we have the conclusion that $C(S)$ *takes the minimum value at the intersection point of two monotone curves* $f(t)$ *and* $g(t)$.

Accordingly, we propose a simple and effective strategy to determine an appropriate timestep t before training. We calculate the L_1, L_2, Peak Signal-to-Noise Ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS) [39], Fréchet Inception Distance (FID) [40], and Structure Similarity Index Measure (SSIM) [41] between (x_t, y_t) and between (x_0, x_t), among which SSIM achieves the timestep with the best performance. The results shown in Fig. 3 are consistent with our aforementioned findings: the distance between (x_t, y_t) drops rapidly, while the distance between (x_0, x_t) grows monotonically as the timestep t grows. Note that the intersection point of the two curves offers a good approximation for the minimum of system complexity. This observation provides us with a pre-selecting strategy that chooses the timestep t of this intersection point as an appropriate timestep t for domain transfer. We demonstrate in Section IV-D the performance of using the timestep t thus chosen by this pre-selecting method.

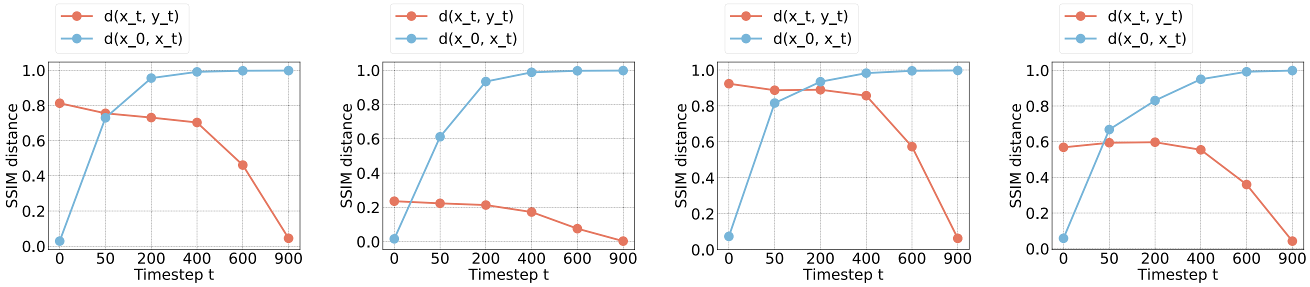
To summarize, we train the DMT module in the same way as a simple I2I task. First, we gradually apply the same diffusion forward process onto both the input condition and the desired output until a pre-selected timestep. Then, we train the function approximator f_θ using a reparameterization strategy to reformulate the objective function. We theoretically prove the feasibility of the simple DMT module and show that the approximator f_θ resembles the reverse process mean function approximator in DDPM [2]. We verify the efficiency of the DMT in Section IV with comprehensive experiments on a wide range of datasets, and provide the algorithms and the pseudo-codes in Appendix A, available online.

E. Further Discussion of DMT

Recall that we introduce the shared encoder by diffusing both x_0 and y_0 with the identical timestep t. To address the trade-off between the strength of content information and domain gap, we propose a strategy to automatically preset an adequate

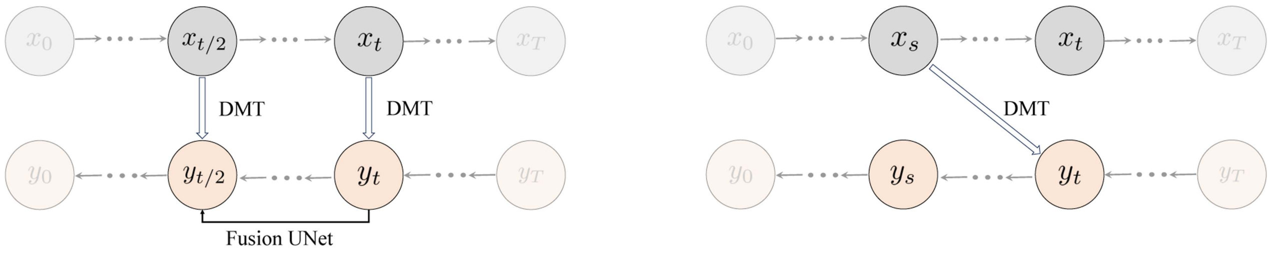
[2] This assumption is reasonable because the diffusion and denoising processes are reciprocal at the same time step, although in different domains.

[3] This assumption is reasonable because the larger the time step, the greater the complexity of forward diffusion and the lower the complexity of DMT.



(a) Portrait (b) AFHQ (c) CelebA-HQ (d) Edges2handbags

Fig. 3. *Analysis on the preset timestep, t.* Our DMT needs a pre-defined timestep to learn and perform the distribution shift. We plot the distance between (x_t, y_t) and (x_0, x_t) at different timesteps, which are shown in red and blue curves, respectively. When t increases, $d(x_t, y_t)$ decreases so that the distribution is easier to shift from x_t to y_t, while $d(x_0, x_t)$ increases so that the input condition signal is becoming less relevant because x_t is drifting away from the input x_0. Considering such a trade-off, we select the intersection as the practical choice of the timestep for DMT learning.



(a) (b)

Fig. 4. *Conceptual comparison* for (a) multi-step DMT and (b) asymmetric DMT. $\{x_t\}_{t=0}^{T}$ represent different states of the input from the source domain, while $y_T \to y_0$ stands for the denoising process of DDPM. Here, T denotes the total number of noise-adding steps in the diffusion process. Multi-step DMT combines the translation results of DMT at two different timesteps with an auxiliary fusion UNet and denoise to achieve the final output, while asymmetric DMT applies translation at different timestep pair (s, t). More discussions are addressed in Section III-E and *Supplementary Material*, available online.

timestep t^* to achieve equilibrium between the distances of (x_0, x_t) and (x_t, y_t). Therefore, one could reasonably consider to use 1) multi-step translation results from DMT to facilitate the denoising precess, or 2) diffusion processes with distinct timesteps for the source and target domains, as a strategy to mitigate trade-offs and achieve improved performance. In this subsection, we discuss these two interesting alternatives, by fusing the DMT results at multiple timesteps (e.g., t and $t/2$) (i.e., Fig. 4(a)), together with using the *asymmetric* timestep pair (s, t) (i.e., Fig. 4(b)), where x_0 and y_0 are diffused at timesteps s and t, $s \neq t$ respectively. Given the results analyzed in this section, we conclude that the former multi-step method significantly increases training time cost while degrading the FID performance, and that the latter more complicated pipeline practically coincides with our proposed DMT method, since the optimal timestep pair (s, t) appears to be the same.

To implement the multi-step DMT, due to the use of the vanilla DDPM, which is only capable of inputting a 3-channel input intermediate noisy image, we train an auxiliary UNet model to fuse the $y_{t/2}$ transformed from $x_{t/2}$ together with the $y'_{t/2}$ denoised from the y_t. However, we argue that the additional UNet significantly increases the training cost, while degrading the FID performance, due to additional error from the UNet. Detailed experimental setups and quantitative comparison are provided in *Supplementary Material*, available online.

As for the asymmetric setting, we define the disjoint distribution of the forward Markov process of $(x_{1:s}, y_{1:t})$ as below:

$$q(y_{1:t}, x_{1:s}|y_0, x_0) = \prod_{i=1}^{s} q(x_i|x_{i-1}) \prod_{j=1}^{t} q(y_j|y_{j-1}), \tag{20}$$

$$p_\theta(y_{0:t}, x_{1:s}|x_0) = p_\theta(y_t|x_s) \prod_{i=1}^{s} q(x_i|x_{i-1}) \prod_{j=1}^{t} q(y_{j-1}|y_j). \tag{21}$$

We first claim the feasibility of this pipeline, whose proofs are addressed in *Supplementary Material*, available online. Similar to Lemma 1, Theorems 1 and 2, we have

Lemma 2: The negative log-likelihood of $-\log p_\theta(y_0|x_0)$ has the following upper bound,

$$-\log p_\theta(y_0|x_0) \leqslant \mathbb{E}_q \left[\log \frac{q(y_{1:t}, x_{1:s}|y_0, x_0)}{p_\theta(y_{0:t}, x_{1:s}|x_0)} \right], \tag{22}$$

where $q = q(y_{1:t}, x_{1:s}|y_0, x_0)$.

We accordingly define the $\mathcal{L}_{VLB}$ as below:

$$\mathcal{L}_{CE} = \nmid\mathbb{E}_{q(y_0|x_0)} \left[\log p_\theta(y_0|x_0)\right] \tag{23}$$

$$\leqslant \mathbb{E}_{q(y_{0:t}, x_{1:s}|x_0)} \left[\log \frac{q(y_{1:t}, x_{1:s}|y_0, x_0)}{p_\theta(y_{0:t}, x_{1:s}|x_0)} \right] := \mathcal{L}_{VLB}. \tag{24}$$

Then we have the re-claimed Theorem 1:

Theorem 3 (Closed-Form Expression): The loss function in (23) has a closed-form representation. The training is equivalent to optimizing a KL-divergence up to a non-negative constant, i.e.,

$$\mathcal{L}_{VLB} = \mathbb{E}_{q(y_0, x_s|x_0)}\left[D_{KL}(q(y_t|y_0) \| p_\theta(y_t|x_s))\right] + C. \quad (25)$$

For the given closed-form expression in (25), the optimal $p_\theta(y_t|x_s)$ follows a Gaussian distribution and its mean μ_θ has an analytic form, as summarized in the Theorem 2 above.

Theorem 4 (Optimal Solution to (25)): The optimal $p_\theta(y_t|x_s)$ follows a Gaussian distribution with its mean being

$$\mu_\theta(x_s) = \sqrt{\bar{\alpha}_t} y_0. \quad (26)$$

By applying the diffusion forward process on both x_0 and y_0 with identical random noise at asymmetric timestep s and t, respectively, we have the following:

$$x_s = \sqrt{\bar{\alpha}_s} x_0 + \sqrt{1-\bar{\alpha}_s} z, \quad y_t = \sqrt{\bar{\alpha}_t} y_0 + \sqrt{1-\bar{\alpha}_t} z. \quad (27)$$

Theorem 4 reveals that μ_θ needs to approximate the expression $\sqrt{\bar{\alpha}_t} y_0$ with x_s as the only available input. Then we apply the following parameterization,

$$\mu_\theta(x_s) = f_\theta(x_s) - \sqrt{1-\bar{\alpha}_t} z, \quad (28)$$

where f_θ is a trainable function. The KL-divergence in (25) is optimized by minimizing the difference between the two means together with the variance Σ_θ of $p_\theta(y_t|x_s)$. Formally, we have the simplified objective:

$$\mathcal{L}_{s,t} = \mathbb{E}_q\left[\frac{1}{2(1-\bar{\alpha}_t)} \|f_\theta(x_s) - y_t\|^2\right]. \quad (29)$$

To determine an adequate timestep pair (s, t) for the asymmetric diffusion process, similar to the theoretical analysis about original DMT, the complexity of our I2I system is characterized by $C(S) = \max\{C(S_1), C(S_2), C(S_3)\}$. For I2I with the asymmetric DMT, the three sub-systems are 1) the forward diffusion process from x_0 to x_s with the complexity $f(s)$, 2) DMT from x_s to y_t with the complexity $g(s,t)$, and 3) the denoising process via pre-trained diffusion model from y_t to y_0 with the complexity $h(t)$. $f(s)$ and $h(t)$ are monotone w.r.t. s and t, respectively; but $g(s,t)$ does not have to be monotone. If $s \neq t$, the diffusion process from x_0 to x_s and denoising process from y_t to y_0 are no longer reciprocal, so we need to consider both $f(s)$ and $h(t)$. Then the complexity of $C(S)$ can be represented as $C(S) = C(s,t) = \max\{f(s), g(s,t), h(t)\}$. Our target is to search the timestep pair (s,t) minimizing $\min_{s,t} C(s,t)$. We have

$$\max_{i=1,2,3} d_i = \max\{\max\{d_1, d_2\}, d_3\} \quad (30)$$

$$= \max\left\{\frac{d_1+d_2}{2} + \frac{|d_1-d_2|}{2}, d_3\right\} \quad (31)$$

$$\geqslant \max\left\{\frac{d_1+d_2}{2}, d_3\right\} \quad (32)$$

$$\geqslant \frac{1}{3}\left(2 \cdot \frac{d_1+d_2}{2} + d_3\right) = \frac{1}{3}(d_1+d_2+d_3), \quad (33)$$

where the equality holds if and only if $|d_1 - d_2| = 0$ and $\frac{d_1+d_2}{2} = d_3$, i.e., $d_1 = d_2 = d_3$. That means $C(s,t) = \max\{f(s), g(s,t), h(t)\}$ reaches its minimum when $s = t$. In practice, we add the regularity term $\text{SSIM}(x_0, x_s) + \text{SSIM}(x_s, y_t) + \text{SSIM}(y_0, y_t)$ to help search the global minimum. Formally, we calculate the weighted sum of SSIM distances defined below, in which the smaller the result the better the performance.

$$\begin{aligned}\text{dist}(s,t) = {} & |\text{SSIM}(x_0, x_s) - \text{SSIM}(x_s, y_t)| \\ & + |\text{SSIM}(x_s, y_t) - \text{SSIM}(y_0, y_t)| \\ & + |\text{SSIM}(x_0, x_s) - \text{SSIM}(y_0, y_t)| \\ & + \lambda \text{SSIM}(x_0, x_s) \\ & + \lambda \text{SSIM}(x_s, y_t) \\ & + \lambda \text{SSIM}(y_0, y_t)). \end{aligned} \quad (34)$$

By setting the weight $\lambda = 0.5$, we acquire an appropriate timestep pair as in Table I. Notably, the preset timestep pair (s,t) of this generalized pipeline coincide with the original pipeline theoretically and empirically, i.e., the asymmetric timestep pair appears to be identical.

TABLE I
Ablation Study on the Preset Timestep Pair (s,t) in Our Proposed DMT on the Two I2I Tasks Under Different λ Defined in (34)

Stylization	Colorization	Segmentation	Sketch
$(s,t) = (50, 50)$	$(s,t) = (5, 5)$	$(s,t) = (200, 200)$	$(s,t) = (20, 20)$

SSIM is used to evaluate the distance between samples.

IV. Experiments

In this section, we evaluate the proposed DMT on four different I2I tasks: image stylization, colorization, segmentation to image, and sketch to image. We first show that the DMT is capable of mapping translation between the two domains of an I2I task in Section IV-B. Then, we compare the DMT with several representative methods to demonstrate its superior efficiency and performance in Section IV-C. Finally, we provide an ablation study on the effect of the timestep t for training in Section IV-D.

A. Experimental Setups

Datasets and Tasks: We train the I2I task on four datasets: our handcrafted Portrait dataset using CelebA-HQ by QMUPD [42], AFHQ [43], CelebA-HQ [44], and Edges2handbags [45], [46]. All the images are resized to 256 × 256 resolution. Our Portrait dataset consists of 27,000 images for training and 3,000 images for inference; all these images are generated from the CelebA-HQ dataset using a pretrained QMUPD model. The AFHQ dataset consists of 14,630 images for training and 1,500 images for inference, encompassing a variety of cats, dogs, and wild animal images. For the CelebA-HQ dataset, we randomly choose 27,000 images together with their segmentation masks as the paired training data, while the remaining 3,000 images are used as test data. As for Edges2handbags, we use all 138,567 images as training data and the 200-image test data for inference.

Evaluation Metrics: We use Fréchet Inception Distance (FID) [40], Structure Similarity Index Measure (SSIM) [41], LPIPS [39], $L1$ and $L2$ metrics to evaluate the fidelity of the generated images and how well the content information is kept after the translation. Besides, we compare all the methods in a user study, where users were asked to score the image quality from 1 to 5. We also compare the training and inference efficiency of all the methods by comparing the number of total training epochs, training speed for 1,000 images, and inference time for generating an image.

Baselines: We compare our proposed DMT algorithm with five representative I2I algorithms: Pix2Pix [7], TSIT [13], SPADE [14], QMUPD [42], and Palette [9]. The alternatives can be divided into two categories: GAN-based and DDPM-based algorithms. Pix2Pix is a classic cGAN-based method involving L_1 and adversarial loss. TSIT is a GAN-based versatile framework using specially designed normalization layers and coarse-to-fine feature transformation. SPADE is a GAN-based specially-designed framework for semantic image synthesis with spatially-adaptive normalization. QMUPD is also GAN-based, which is specially designed for portrait stylization by unpaired training. We train the model with paired data for fair comparison. Palette introduces the DDPM [2] framework into the I2I task and injects the input constraint to each step of the denoising process.

Implementation Details: We train the proposed DMT module on the platform of PyTorch [47], in a Linux environment with an NVIDIA Tesla A100 GPU. We set total timestep $T = 1000$ for all the experiments, the same setting as in [2]. We train the reverse denoising process of the DDPM using a U-Net backbone together with the Transformer sinusoidal embedding [48], [49], following [6]. The DDPM is frozen during the training of the DMT module. To train the DMT module, we use the Pix2Pix [7] and TSIT [13] model. We remove the discriminator model and train only the generator block to ensure that the translator f_θ has approximately the same functional form as the real mapping. Note that our DMT employs the DDPM denoising process during sampling, which employs hundreds of iterative function evaluations for denoising and can be time-consuming. Therefore, we apply DDIM [4] for acceleration, which realizes high-quality synthesis within 10 function evaluations (NFE = 10).

B. Qualitative Evaluation on Various Tasks

The process of inferring images with DMT consists of the following three simple steps.

1) We apply the forward diffusion process to the input image x_0 until the pre-selected timestep t to obtain x_t, which can be written as $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z_t$;
2) By obtaining the mean by the functional approximator f_θ according to (18), we infer the approximated y_t by adding another Gaussian noise;
3) Using y_t as the intermediate result, sampling with the given pre-trained DDPM by the reverse process achieves the required output.

We conducted four experiments to evaluate our proposed DMT on four datasets, i.e., our handcrafted Portrait dataset, AFHQ [43], CelebA-HQ [44], and Edges2handbags [45], [46]. In training, we use 40 epochs for the sketch-to-image task, and 60 epochs for the other three tasks. As shown in Fig. 2, our method is capable of learning the cross-domain translation mapping and generates high-quality images. For example, in the stylization task, the shared encoder is able to distinguish the two different forward diffusion processes of the two domains. In the other tasks, our method can still extract the input feature and generate photo-realistic images with high diversity even with little input condition information More results can be found in Appendix C, available online.

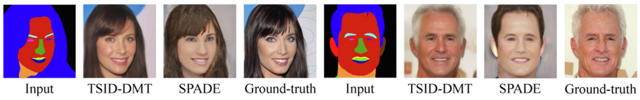


Fig. 5. *Qualitative comparison* between DMT and SPADE [14] on segmentation-to-image task. Our proposed DMT achieves better image quality and content consisitency compared with SPADE.

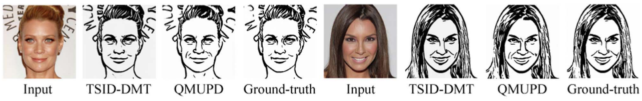


Fig. 6. *Qualitative comparison* between DMT and QMUPD [42] on image stylization task. Our proposed DMT achieves better image quality and content consisitency compared with QMUPD.

TABLE II
QUANTITATIVE COMPARISON BETWEEN DMT AND SPADE [14] ON SEGMENTATION-TO-IMAGE TASK

Method	FID↓	SSIM↑	LPIPS↓	L1↓	L2↓
SPADE (GAN)	66.55	0.140	0.487	0.413	0.285
Ours	**36.78**	**0.446**	**0.433**	**0.182**	**0.053**

FID, SSIM, LPIPS, L1, and L2 metrics are used to evaluate the image quality and content consistency, respectively.

TABLE III
QUANTITATIVE COMPARISON BETWEEN DMT AND QMUPD [42] ON IMAGE STYLIZATION TASK

Method	FID↓	SSIM↑	LPIPS↓	L1↓	L2↓
QMUPD (GAN)	12.81	0.660	0.248	0.268	0.392
Ours	**11.01**	**0.760**	**0.138**	**0.131**	**0.101**

FID, SSIM, LPIPS, L1, and L2 metrics are used to evaluate the image quality and content consistency, respectively.

C. Comparisons

We qualitatively and quantitatively compare our method with the four classic I2I methods: Pix2Pix [7], TSIT [13], SPADE [14], QMUPD [42], and the DDPM-based conditional generation method Palette [9]. First, we compare with SPADE [14]. It requires category-wise segmentation masks,

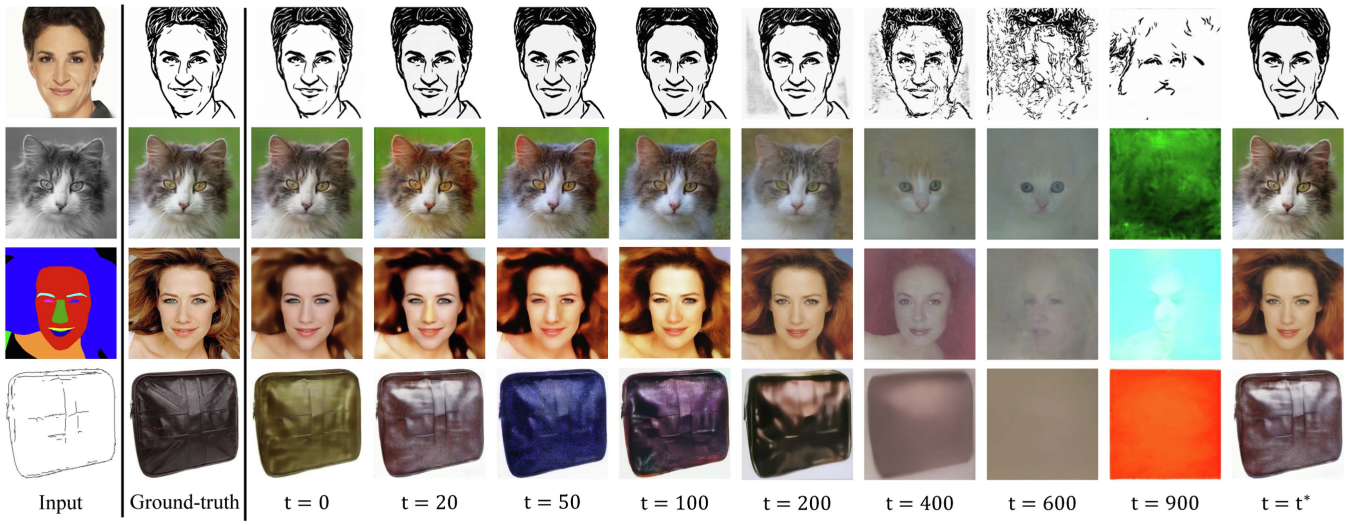


Fig. 7. *Qualitative results for ablation study* of the preset timestep t in our proposed DMT on the four I2I tasks. We observe that a smaller t helps in better retaining the content information from the input source, but suffers from a larger gap between the target domain and the source domain. The optimally selected timestep (t^*) for each of the four I2I tasks is given in Table V.

TABLE IV
QUANTITATIVE COMPARISON BETWEEN PALETTE [9], PIX2PIX [7], TSIT [13], AND OUR PROPOSED DMT. FID, SSIM, AND LPIPS ARE USED TO EVALUATE THE IMAGE QUALITY AND CONTENT PRESERVATION, RESPECTIVELY. BESIDES, WE INTRODUCE THE USER STUDY (*SCORE*) TO EVALUATE THE QUALITY OF THE SYNTHESIZED IMAGES

	Stylization				Colorization				Segmentation				Sketches						
Method	**FID↓**	**SSIM↑**	**LPIPS↓**	**Ep.**	**FID↓**	**SSIM↑**	**LPIPS↓**	**Ep.**	**FID↓**	**SSIM↑**	**LPIPS↓**	**Ep.**	**FID↓**	**SSIM↑**	**LPIPS↓**	**Ep.**	**Train**	**Infer.**	**Score**
Palette (DDPM)	17.16	0.663	0.366	2500	14.48	0.582	0.299	2450	40.77	0.092	0.521	1000	74.51	0.360	0.275	215	71s	21.63s	3.0
Pix2Pix (GAN)	19.14	0.630	0.260	60	17.50	**0.769**	0.263	60	70.98	0.105	0.542	60	77.80	0.524	0.306	40	25s	0.09s	1.7
Pix2Pix-DMT (Ours)	**10.81**	**0.703**	**0.183**	60	**17.44**	0.752	**0.263**	60	**65.26**	**0.137**	**0.534**	60	**76.75**	**0.527**	**0.306**	40	20s	0.31s	**3.5**
TSIT (GAN)	16.62	0.681	0.235	60	13.60	0.645	0.243	60	40.59	0.357	0.450	60	76.80	0.606	0.282	40	134s	0.11s	3.6
TSIT-DMT (Ours)	**11.01**	**0.760**	**0.138**	60	**13.03**	**0.684**	**0.180**	60	**36.78**	**0.446**	**0.433**	60	**74.37**	**0.687**	**0.255**	40	82s	0.48s	**4.4**

We also report the total number of training epochs (Ep.), training time for 1,000 images (Train), and inference time for a single image (Infer.) of each method.

limiting its application to most I2I tasks. Note that our proposed DMT introduces the shared encoder by gradually adding noise onto the original images, which corrupts the semantic information from the category-wise segmentation masks. Hence, we only compare with SPADE on segmentation-to-image task, without applying the DMT on top of it.

The results are shown in Fig. 5 and Table II.

We also compare with the specially-designed stylization algorithm QMUPD [42]. It introduces a quality metric guidance for portrait generation using unpaired training data. We train QMUPD with paired data for fair comparison, which reduces the training difficulty and achieves a stronger baseline. The results, presented in Fig. 6 and Table III, demonstrate that our approach achieves performance that is on par with, or even surpasses, existing standards.

Then, we compare with Palette [9] using the open source implementation.[4] As shown in Fig. 8, we observe that the results of Palette fail to extract the segmentation feature of CelebA-HQ and Edges2handbags dataset. Consequently, this leads to an inability to accurately generate details in the background of human images or replicate the horse pattern on the bags. As a comparison, our proposed DMT can generate high-quality images and preserve the semantic information of the input condition, even when given little input semantic information.

Next, we compare with Pix2Pix [7]. We observe that our method can generate images of much higher quality than the Pix2Pix method. For instance, the generated images of Pix2Pix suffer from severe artifacts over the facial region in the CelebA-HQ datasets, while our method consistently produces high-quality results. Moreover, the feature extraction performance is significantly improved by the shared encoder and the well-prepared DDPM model in our method.

We finally compare with TSIT [13]. Although TSIT introduces a coarse-to-fine feature transformation block and hence can synthesize high-quality images in most cases, it fails to produce results with sufficient and satisfying semantics and textures when given very little inference information (e.g., hair and forehead region of segmentation). In contrast, the results of DMT have clear boundaries at the forehead and hair region, together with rich texture.

The quantitative results are reported in Table IV, showing that our method has the best image fidelity (FID), the lowest

[4] https://github.com/Janspiry/Palette-Image-to-Image-Diffusion-Models

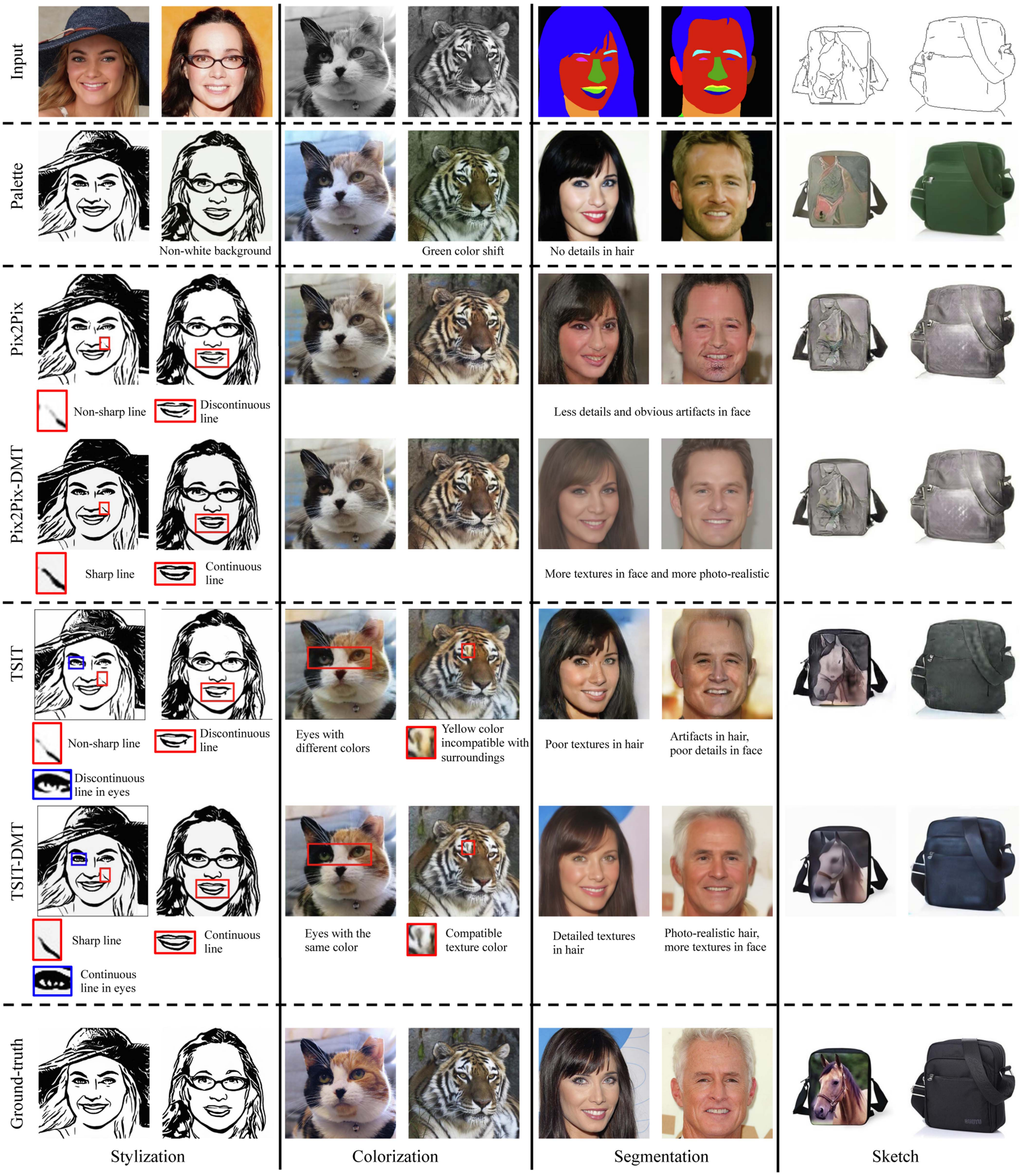


Fig. 8. *Qualitative comparison*. Our DMT achieves on par or better results than the three baseline methods Pix2Pix [7], Palette [9], TSIT [13] on the four I2I tasks, which are image stylization, image colorization, segmentation to image, and sketch to image. Significant differences are highlighted in red or blue boxes, and brief textual explanations are provided besides the boxes. The comparison on efficiency can be found in Table IV.

perceptual loss (LPIPS), and comparable structural similarity (SSIM). Regarding the training and inference speed, our method uses the smallest number of training epochs and has the fastest training speed for generating 1,000 images, because it only needs to train one translation module. The DMT is also 40x $\sim$ 80x faster than Palette [9] due to starting the sampling process at an intermediate step (4x $\sim$ 8x faster) and the use of the fast sampling algorithm DDIM ($\sim$10x faster).

D. Ablation Study on the Timestep for Domain Translation

In the DMT algorithm, we first gradually add noise for both x_0 and y_0 using a shared decoder until some preset timestep t. Here, the timestep t plays a critical role in the performance of the translator f_θ as well as the quality of the generated images. As discussed in Section III-D, we proposed a simple method to determine an adequate timestep before training, denoted by

TABLE V

Ablation Study on the Preset Timestep t in Our Proposed DMT on the Four I2I Tasks

	Stylization		Colorization		Segmentation		Sketches	
	($t^* = 50$)		($t^* = 5$)		($t^* = 200$)		($t^* = 20$)	
Method	**FID↓**	**SSIM↑**	**FID↓**	**SSIM↑**	**FID↓**	**SSIM↑**	**FID↓**	**SSIM↑**
TSIT-DMT ($t = 0$)	22.38	**0.827**	13.28	**0.708**	59.51	**0.522**	76.95	**0.729**
TSIT-DMT ($t = 20$)	**11.01**	0.760	13.90	0.568	47.00	0.486	74.37	0.687
TSIT-DMT ($t = 50$)	20.46	0.732	15.18	0.496	42.00	0.473	78.03	0.629
TSIT-DMT ($t = 100$)	39.13	0.674	16.38	0.394	37.22	0.460	80.81	0.668
TSIT-DMT ($t = 200$)	80.55	0.518	18.82	0.249	36.78	0.446	88.54	0.629
TSIT-DMT ($t = 400$)	110.97	0.301	114.08	0.085	50.79	0.251	126.56	0.307
TSIT-DMT ($t = 600$)	254.44	0.177	216.62	0.019	158.69	0.050	338.89	0.084
TSIT-DMT ($t = 900$)	301.77	0.051	337.37	0.028	213.81	0.000	371.87	0.004
TSIT-DMT ($t = t^*$)	20.46	0.732	**13.03**	0.684	**36.78**	0.446	**74.37**	0.687

FID and SSIM are used to evaluate the image quality and content preservation, respectively. Notably, the model trained on the timestep t^* , which is automatically selected by our strategy in Section III-D, achieves satisfactory performance for all the four tasks.

TABLE VI

Ablation Study of t Near t^* on AFHQ Dataset

t	0	5 (t^*)	10	15	25
FID↓	13.28	**13.03**	13.61	13.92	14.62
SSIM↑	**0.708**	0.684	0.680	0.620	0.537

The bold values show the best scores of FID and SSIM across different timestep t on AFHQ dataset.

TABLE VII

Ablation Study of t Near t^* on CelebA-HQ Dataset

t	180	190	195	200 (t^*)	205	210	220
FID↓	37.93	38.48	36.46	36.78	37.09	39.62	**36.04**
SSIM↑	0.450	0.438	**0.455**	0.446	0.420	0.418	0.432

The bold values show the best scores of FID and SSIM across different timestep t on CelebA-HQ dataset.

$t = t^*$, by pre-computing the distance between (x_0, x_t) and between (x_t, y_t). In this section, we compare the generation quality using different timesteps t and show that the timestep t^* selected using our method in Section III-D offers the optimal performance.

In Fig. 7, we observe that: 1) As the translation timestep t increases, the input condition provides weaker constraint to the output generation. For instance, the face poses of the results in row 1 and row 3 begin to change in an unwarranted way when $t > 400$; 2) When the translation timestep t is small, the translation mapping can hardly approximate the real distribution (e.g., the hair texture of the segmentation to image task in row 3, column 3).

We also present quantitative comparison results in Table V, from which we see the trade-off between the strength of the input condition and the difficulty of learning the translation mapping. Significantly, our method for selecting an appropriate timestep achieves performance comparable to using the optimal t shown in Table V. This confirms the effectiveness of our simple selection strategy.

We conduct further ablation study on the performance of timestep t near the preset timestep t^*, in order to demonstrate the strong robustness of our strategy. As shown in Tables VI and VII, despite the significant performance drop when using different timesteps, our strategy is still able to search an adequate timestep for DMT.

E. Limitations

Our DMT method has several limitations that are interesting avenues for future research. First, our algorithm is based on the assumption that both the forward and the reverse process satisfy the Markovian property, but this assumption holds only for the DDPM or its extension. Second, the DMT is designed to train with paired data due to its reliance on using Pix2Pix [7] or TSIT [13] module as the translation mapping f_θ. Hence, our method cannot be applied to unpaired training data and related I2I tasks. Third, our DMT is not applicable to tasks whose condition (source domain) and the target domain are almost identical. We briefly explain this limitation next. Following (13), when x_0 equals y_0, we have $q(y_0|x_0) = \delta_{x_0}(y_0)$, which is the Dirac distribution. Then, (13) becomes

$$\mathcal{L}_{CE} = \log p_\theta(x_0|x_0) = 0, \tag{35}$$

which is a constant independent of the model parameter θ. Therefore, the model cannot be optimized.

V. Conclusion

In this paper, we propose an efficient diffusion model translator, which bridges a well-prepared DDPM and the input inference. We provide theoretical proof to show the feasibility of using this simple module to accomplish the popular I2I task. By using our proposed practical method to pre-select an adequate timestep and applying the forward diffusion process until this timestep, we formulate the task as the learning process of a translation mapping, without relying on any retraining of the given DDPM. We conduct comprehensive experiments to show the high efficiency and the outstanding performance of our proposed algorithm.

References

[1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2256–2265.

[2] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 6840–6851.

[3] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–12.

[4] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–12.

[5] A. Q. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8162–8171.

[6] P. Dhariwal and A. Nichol, “Diffusion models beat GANs on image synthesis,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 8780–8794.

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[8] H. Sasaki, C. G. Willcocks, and T. P. Breckon, “UNIT-DDPM: Unpaired image translation with denoising diffusion probabilistic models,” 2021, *arXiv:2104.05358*.

[9] C. Saharia et al., “Palette: Image-to-image diffusion models,” in *Proc. Special Int. Group Comput. Graph. Interactive Techn. Conf.*, M. Nandigjav, N. J. Mitra, and A. Hertzmann, Eds., 2022, pp. 15:1–15:10.

[10] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, “ILVR: Conditioning method for denoising diffusion probabilistic models,” in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 14 347–14 356.

[11] X. Liu et al., “More control for free! Image synthesis with semantic diffusion guidance,” 2021, *arXiv:2112.05744*.

[12] T. Wang et al., “Pretraining is all you need for image-to-image translation,” 2022, *arXiv:2205.12952*.

[13] L. Jiang, C. Zhang, M. Huang, C. Liu, J. Shi, and C. C. Loy, “TSIT: A simple and versatile framework for image-to-image translation,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 206–222.

[14] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, “Semantic image synthesis with spatially-adaptive normalization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2332–2341.

[15] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, “SEAN: Image synthesis with semantic region-adaptive normalization,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5103–5112.

[16] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, “WaveGrad: Estimating gradients for waveform generation,” 2020, *arXiv: 2009.00713*.

[17] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “DiffWave: A versatile diffusion model for audio synthesis,” 2020, *arXiv: 2009.09761*.

[18] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, “Video diffusion models,” 2022, *arXiv:2204.03458*.

[19] J. Ho et al., “Imagen video: High definition video generation with diffusion models,” 2022, *arXiv:2210.02303*.

[20] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, “Image super-resolution via iterative refinement,” 2021, *arXiv:2104.07636*.

[21] H. Li et al., “SRDiff: Single image super-resolution with diffusion probabilistic models,” *Neurocomputing*, vol. 479, pp. 47–59, 2022.

[22] D. Watson, J. Ho, M. Norouzi, and W. Chan, “Learning to efficiently sample from diffusion probabilistic models,” 2021, *arXiv:2106.03802*.

[23] F. Bao, C. Li, J. Zhu, and B. Zhang, “Analytic-DPM: An analytic estimate of the optimal reverse variance in diffusion probabilistic models,” in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–12.

[24] A. Jolicoeur-Martineau, K. Li, R. Piché-Taillefer, T. Kachman, and I. Mitliagkas, “Gotta go fast when generating data with score-based models,” 2021, *arXiv:2105.14080*.

[25] L. Liu, Y. Ren, Z. Lin, and Z. Zhao, “Pseudo numerical methods for diffusion models on manifolds,” in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–11.

[26] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. S. Kudinov, and J. Wei, “Diffusion-based voice conversion with fast maximum likelihood sampling scheme,” in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–12.

[27] H. Tachibana, M. Go, M. Inahara, Y. Katayama, and Y. Watanabe, “Itô-Taylor sampling scheme for denoising diffusion probabilistic models using ideal derivatives,” 2021, *arXiv:2112.13339*.

[28] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, “DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, Art. no. 418.

[29] Z. Xiao, K. Kreis, and A. Vahdat, “Tackling the generative learning trilemma with denoising diffusion GANs,” in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–15.

[30] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 2014, *arXiv:1411.1784*.

[31] I. Goodfellow et al., “Generative adversarial nets,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[32] H. Dong, S. Yu, C. Wu, and Y. Guo, “Semantic image synthesis via adversarial learning,” in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 5706–5714.

[33] T. Kaneko, K. Hiramatsu, and K. Kashino, “Generative attribute controller with conditional filtered generative adversarial networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6089–6098.

[34] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, “Learning to generate images of outdoor scenes from attributes and semantic layouts,” 2016, *arXiv:1612.00215*.

[35] C. Ledig et al., “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.

[36] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, “Scribbler: Controlling deep image synthesis with sketch and color,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5400–5409.

[37] X. Wang and A. Gupta, “Generative image modeling using style and structure adversarial networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 318–335.

[38] Z. Zhang, Y. Song, and H. Qi, “Age progression/regression by conditional adversarial autoencoder,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5810–5818.

[39] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.

[40] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local nash equilibrium,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6629–6640.

[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[42] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, “Quality metric guided portrait line drawing generation from unpaired training data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 905–918, Jan. 2023, doi: 10.1109/TPAMI.2022.3147570.

[43] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, “StarGAN v2: Diverse image synthesis for multiple domains,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8188–8197.

[44] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.

[45] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, “Generative visual manipulation on the natural image manifold,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.

[46] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 1395–1403.

[47] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[48] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2015, pp. 234–241.

[49] A. Vaswani et al., “Attention is all you need,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

Mengfei Xia received the BS degree from the Department of Mathematical Science, Tsinghua University, Beijing, China, in 2020. He is currently working toward the PhD degree with the Department of Computer Science and Technology, Tsinghua University. His research interests include mathematical foundation in deep learning, image processing, and computer vision. He was the recipient of the Silver Medal twice in 30th and 31st National Mathematical Olympiad of China.

Yu Zhou is currently working toward the undergraduate degree with Zhili College, Tsinghua University, China. His research interests include deep learning and computer vision. He was the recipient of the Silver Medal twice in 35th and 36th National Olympiad in Informatics of China.

Ran Yi received the BEng and PhD degrees from Tsinghua University, China, in 2016 and 2021. She is an assistant professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Her research interests include computer vision, computer graphics, and computational geometry.

Yong-Jin Liu (Senior Member, IEEE) received the BEng degree from Tianjin University, China, in 1998, and the PhD degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2004. He is a professor with the Department of Computer Science and Technology, Tsinghua University, China. His research interests include computer vision, computer graphics, and computer-aided design.

Wenping Wang (Fellow, IEEE) received the PhD degree in computer science from the University of Alberta, in 1992. He is a professor of computer science with Texas A&M University. His research interests include computer graphics, computer visualization, computer vision, robotics, medical image processing, and geometric computing. He is or has been an journal associate editor of the *ACM Transactions on Graphics*, *IEEE Transactions on Visualization and Computer Graphics*, *Computer Aided Geometric Design*, and *Computer Graphics Forum* (CGF). He has chaired a number of international conferences, including Pacific Graphics, ACM Symposium on Physical and Solid Modeling (SPM), SIGGRAPH and SIGGRAPH Asia. He received the John Gregory Memorial Award for his contributions to geometric modeling. He is an ACM fellow.