Animating General Image with Large Visual Motion Model

Dengsheng Chen Xiaoming Wei Xiaolin Wei
Meituan
Beijing, China
{chendengsheng,weixiaoming,weixiaolin02}@meituan.com

Abstract

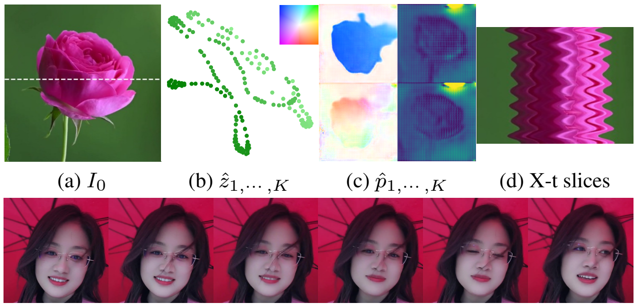
We present the pioneering Large Visual Motion Model (LVMM), meticulously engineered to analyze the intrinsic dynamics encapsulated within real-world imagery. Our model, fortified with a wealth of prior knowledge extracted from billions of image pairs, demonstrates promising results in predicting a diverse spectrum of scene dynamics. As a result, it can infuse any generic image with authentic dynamic effects, enhancing its visual allure.

Project page: https://github.com/densechen/LVMM.

1. Introduction

Recent progress in generative models [39], specifically conditional diffusion models [11, 22], and large-scale models [39], have substantially enhanced our capability to represent complex and rich distributions. These models have underscored the transformative potential of harnessing vast data and intensive training [14], exhibiting unparalleled proficiency in comprehending and generating human-like text, and creating visually rich and diverse images from textual descriptions. This has facilitated a variety of previously unachievable applications, such as text-conditioned generation of arbitrary, realistic image content [21]. The advent of these models [5, 19, 24], propelled by the availability of large-scale datasets [25] and advancements in training methodologies [4, 20], has ignited interest in probing other domains, including audio [38] and multimodal data [15].

In this paper, we present a novel Large Visual Motion Model (LVMM), specifically designed to proficiently predict local motion embedded within a given scene, thereby enhancing the dynamic appeal of a static image. The dynamism of the natural world is characterized by subtle changes even in seemingly static landscapes, influenced by various factors such as wind, water currents, and inherent rhythms. When observing a still image, we can envisage plausible motions that might have been occurring when the photograph was captured. This predictability is ingrained in our human perception of real scenes, i.e., we can imagine a distribution of natural motions conditioned on that image if there could have been multiple possible motions. Given the ease with which humans can envision these potential motions, an intriguing research question is to computationally model this motion distribution with a large-scale model.



(a) I_0 (b) $\hat{z}_{1,\cdots,K}$ (c) $\hat{p}_{1,\cdots,K}$ (d) X-t slices

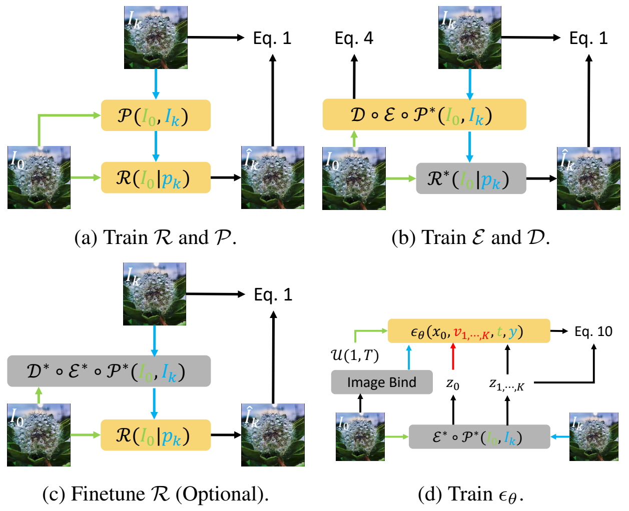
(e) Capability to animate faces, generating natural and realistic smiles.

Figure 1. Beginning with a reference image I_0 as depicted in Fig. 1a, the Large Visual Motion Model (LVMM) estimates a latent motion trajectory $\hat{z}_{1,\cdots,K}$ as shown by the t-SNE plot in Fig. 1b, utilizing the motion denoising model ϵ_θ. This trajectory is subsequently processed by the Motion Decoder $\mathcal{D}$, generating a sequence of optical flows $\hat{\delta}_{1,\cdots,K}$ (visualized in the left column of Fig. 1c) and intention maps $\hat{\omega}_{1,\cdots,K}$ (displayed in the right column of Fig. 1c). Ultimately, the Neural Image Renderer $\mathcal{R}$ transforms I_0 into a series of novel images $\hat{I}_{1,\cdots,K}$, guided by optical flows and intention maps. Fig 1d illustrates the resultant videos, employing space-time X-t slices across 300 frames (corresponding to the scanline shown in Fig. 1a).

The proposed LVMM excels in associating salient visual and motion patterns, thereby accurately predicting local motion trajectory, as shown in Fig. 1. It comprises two components: the motion rendering model and the motion diffusion model. The former extracts a latent motion vector from the scene and reconstructs the target image. The latter generates suitable motion trajectories from the given scene and feeds them to the motion rendering model to produce realistic dynamic effects.

Our primary contribution is the pioneering proposal and design of a large-scale model dedicated to visual motion, the effectiveness of which has been empirically validated.

2. Overview


I_k
Eq. 1
$\mathcal{P}(I_0, I_k)$
I_0
$\mathcal{R}(I_0|p_k)$
$\hat{I}_k$
Eq. 4
$\mathcal{D} \circ \mathcal{E} \circ \mathcal{P}^*(I_0, I_k)$
$\mathcal{R}^*(I_0|p_k)$
$\mathcal{D}^* \circ \mathcal{E}^* \circ \mathcal{P}^*(I_0, I_k)$
$\epsilon_\theta(x_0, v_{1,\cdots,K}, t, y)$
Eq. 10
$\mathcal{U}(1,T)$
Image Bind
z_0
$z_{1,\cdots,K}$
$\mathcal{E}^* \circ \mathcal{P}^*(I_0, I_k)$

(a) Train $\mathcal{R}$ and $\mathcal{P}$.

(b) Train $\mathcal{E}$ and $\mathcal{D}$.

(c) Finetune $\mathcal{R}$ (Optional).

(d) Train ϵ_θ.

Figure 2. Training phase. Modules denoted in gray or marked with an asterisk ($*$) in the upper right corner indicate that these modules do not undergo parameter updates. Conversely, modules highlighted in orange undergo parameter updates. The color of the connecting lines directly corresponds to the color of the variables.

Given a single reference image I_0, our goal is to synthesize a video sequence $\hat{I}_{1,\cdots,K}$ of length K. This sequence is designed to capture local dynamics such as the sway of vegetation or the flicker of candle flames under wind influence, as well as human emotional expressions like joy, anger, sorrow, and pleasure.

Our proposed framework consists of two components: a motion rendering model (Sec. 3.1) and a motion diffusion model (Sec. 3.2). The motion rendering model is a sophisticated system comprising a neural image renderer $\mathcal{R}$, a motion flow predictor $\mathcal{P}$, and a motion encoder-decoder pair $\mathcal{E}$ and $\mathcal{D}$. The neural image renderer $\mathcal{R}$ transforms the reference image I_0 into the target image I_k by utilizing the motion flow p predicted by $\mathcal{P}$. A motion flow p is composed of an optical flow δ and an intention map ω, which respectively corresponds to low-frequency and high-frequency motion dynamics. The motion encoder $\mathcal{E}$ maps p into a latent motion space, where even the most complex motion flows can be represented by a motion vector z. The motion decoder $\mathcal{D}$ then converts the motion vector back into the motion flow space.

Through rigorous training, we discovered that the motion vector z in the latent motion space demonstrates a higher degree of regularity compared to motion flow space, as depicted in Fig. 1b. Intriguingly, the motion vector can encapsulate motion rules in two segments: those associated with visual features v (visual segments) and those unrelated u (motion segments). By retaining the visual segments and replacing the motion segments with those from different scenes, we can achieve cross-scene motion transfer (Para. 3.2.1).

Exploiting this property, the motion diffusion model can concentrate on learning motion segments u that are unrelated to visual features, thereby significantly simplifying the task. The motion denoising model ϵ_θ, can generate a latent motion trajectory $\hat{u}_{1,\cdots,K}$ based on the provided image I_0, thus producing dynamic image effects (Sec. 3.3).

3. Large Visual Motion Model

3.1. Motion Rendering Model

3.1.1 Neural Image Renderer and Motion Flow Predictor

Our approach commences with the deployment of two images, namely the reference image I_0 (Fig. 3a) and the driving image I_k (Fig. 3b). The concurrent training of the motion flow predictor $\mathcal{P}$ and the neural image renderer $\mathcal{R}$ is the initial step. The aim is to steer the motion portrayed by the motion flow such that the image $\hat{I}_k$ rendered from I_0 closely mirrors the actual driving image I_k. The training pipeline is illustrated in Fig. 2a. This process can be mathematically expressed as:

$$\arg\min_{\mathcal{R},\mathcal{P}} \mathbb{E}[\| \; I_k - \underbrace{\mathcal{R}(I_0|\mathcal{P}(I_0, I_k))}_{\hat{I}_k} \; \|_2^2] \tag{1}$$

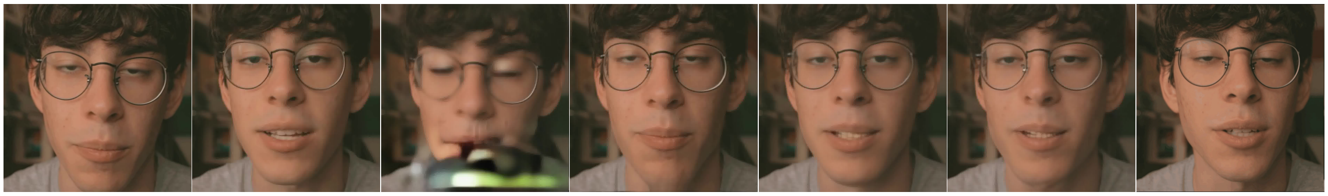
Design of Motion Flow Predictor $\mathcal{P}$ Our design incorporates a network architecture akin to that proposed by Siarohin et al. [26]. We introduce two key modifications to enhance the estimation of a wide range of motion scenarios across large-scale datasets.

Initially, the local similarity of optical flow can result in a loss of generalization capability across different scenarios if $\mathcal{P}$ is directly instructed to predict a motion flow p in a pixel-wise manner. To avoid this, we downsample I_0 and I_k by a factor of 0.25 prior to feeding them to $\mathcal{P}$, as follows:

$$p := \mathcal{P}(I_0', I_k') \tag{2}$$

where $I_0', I_k' \in \mathcal{R}^{\frac{H}{4} \times \frac{W}{4}}$ are the downsampled images. This approach still produces satisfactory results, while significantly reducing the computational overhead of the motion flow predictor.

Secondly, the motion flow predictor $\mathcal{P}$ tends to assign an optical flow δ with a non-negligible value to most stationary points in the background, as shown in the left column of Fig. 1c. This tendency severely hinders the model's ability to capture subtle local movements, as there will be a large number of relatively stationary regions in image pair



(a) I_0 (b) I_k (c) $\mathcal{R}(I_0|\delta_x, \delta_y)$ (d) $\mathcal{F}(I_0|\delta_x, \delta_y, \omega)$ (e) $\mathcal{R}(I_0|\delta_x, \delta_y, \omega)$ (f) $\mathcal{R}(I_0|\hat{p}_k)$ (g) $\mathcal{R}^*(I_0|\hat{p}_k)$.

Figure 3. Ablation study of the motion accretion model. For a detailed explanation, please refer to Sec. 4.4.

$\langle I_0, I_k \rangle$. Specifically, during backpropagation, these relatively stationary regions contribute a large amount of gradient that is not practically meaningful, preventing $\mathcal{P}$ from learning more detailed local motion information, as shown in Fig. 3c. To address this issue, we predict an intention map $\omega \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4}}$ to represent the motion intention of each region along with optical flow δ. When ω approaches zero, it indicates that the current region tends to be stationary, and its gradient will be reduced by ω during backpropagation, thereby avoiding gradient pollution. As shown in the right column of Fig. 1c, the predicted intention map indicates many high-frequency information, such as the contour of the object, while the optical flow represents more low-frequency information, such as the motion trend of the main body.

In summary, the motion flow $p \in \mathbb{R}^{3 \times \frac{H}{4} \times \frac{W}{4}}$ can be viewed as composed of two parts: the optical flow $\delta \in \mathbb{R}^{2 \times \frac{H}{4} \times \frac{W}{4}}$ along the x and y dimensions, respectively, and an intention map $\omega \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4}}$ that represents the motion tendency at each location.

Design of Neural Image Renderer $\mathcal{R}$ We denote our neural image renderer as $\mathcal{R}$. The initial step in the process involves the construction of a multi-scale feature pyramid, F_{I_0}, for the image I_0. Following this, a warp function $\mathcal{F}$ is applied to each feature map within F_{I_0}, leading to the generation of a warped feature pyramid, F'_{I_0}:

$$F'_{I_0} = \mathcal{F}(F_{I_0}, \delta') \odot \omega' \tag{3}$$

In the above equation, δ', ω' represent the interpolated optical flow and the intention map respectively. These components are specifically designed to align with the spatial configuration of the feature in F_{I_0} across multiple scales. $\odot$ denotes an element-wise multiplication operation. In the final phase, $\mathcal{R}$ generates an estimated driving image, $\hat{I}_k$, which is based on the warped feature pyramid F'_{I_0}, rather than F_{I_0}.

The direct application of the motion flow to the feature pyramid F_{I_0} offers several benefits. Primarily, it significantly reduces the likelihood of image distortion and degradation that could potentially occur when the motion flow is applied within the pixel space (Fig. 3d). Furthermore, the neural image renderer can leverage the prior knowledge encapsulated within large-scale data to compensate for a variety of missing features, as depicted in Fig. 3e. Lastly, it can generate more meaningful gradients across different scales, thereby augmenting the learning capability of the motion flow predictor.

3.1.2 Motion Variational Autoencoder (VAE)

In a formalized manner, our objective is to train a Motion Variational Autoencoder (VAE) that minimizes the error as defined in the equation below:

$$\arg\min_{\mathcal{E},\mathcal{D}} \underbrace{\mathbb{E}[\| \, p_k - \mathcal{D} \circ \mathcal{E}(p_k) \, \|_2^2]}_{\text{motion flow regularization}} + \underbrace{\mathbb{E}[\| \, I_k - \hat{I}_k \, \|_2^2]}_{\text{pixel regularization}} \tag{4}$$

where, $p_k = \mathcal{P}^*(I_0, I_k), \hat{I}_k = \mathcal{R}^*(I_0|\mathcal{D} \circ \mathcal{E}(p))$. $*$ denotes that the model parameters are fixed at this stage. The training pipeline is depicted in Fig. 2b.

Eq. 4 comprises two constraints: motion flow regularization and pixel regularization. The loss value of the former, also referred to as the reconstruction error of the VAE network, ensures optimal accuracy in the encoding and decoding process. However, an exclusive emphasis on motion flow regularization can lead to *model collapse*. Specifically, motion flow does not possess a unique deterministic solution, implying that the motion flow between I_0 and I_k is not unique and can accommodate multiple plausible solutions. Highly similar scenes may yield significantly different predicted motion flows. Furthermore, as discussed in Para. 3.1.1, $\mathcal{P}$ may generate a plethora of irrelevant motion features for points that are essentially stationary. Motion flow regularization necessitates the motion decoder $\mathcal{D}$ to accurately reproduce the predicted motion flow, which deviates from our primary objective. Our goal is to ensure that the image $\hat{I}_k$ rendered by the neural image renderer closely resembles I_k. The introduction of pixel regularization, which directly imposes constraints on $\hat{I}_k$, effectively mitigates this issue, as demonstrated in Fig. 3f.

3.1.3 Fine-tuning Neural Image Renderer (Optional)

To further alleviate the inherent reconstruction error in the Variational Autoencoder (VAE), we propose an optional step of fine-tuning the Neural Image Renderer. The training pipeline is illustrated in Fig. 2c. This is accomplished by minimizing the following objective function:

$$\arg\min_{\mathcal{R}} \mathbb{E}[\| I_k - \mathcal{R}(I_0|\mathcal{D}^* \circ \mathcal{E}^* \circ \mathcal{P}^*(I_0, I_k)) \|_2^2] \tag{5}$$

Based on our empirical observations, this fine-tuning step might not be essential for scenarios with lower complexity. Nevertheless, for scenes rich in local motion details, such as those influenced by facial expressions, this additional fine-tuning process can significantly improve the quality of the reconstructed details, as demonstrated in Fig. 3g.

3.2. Motion Diffusion Model

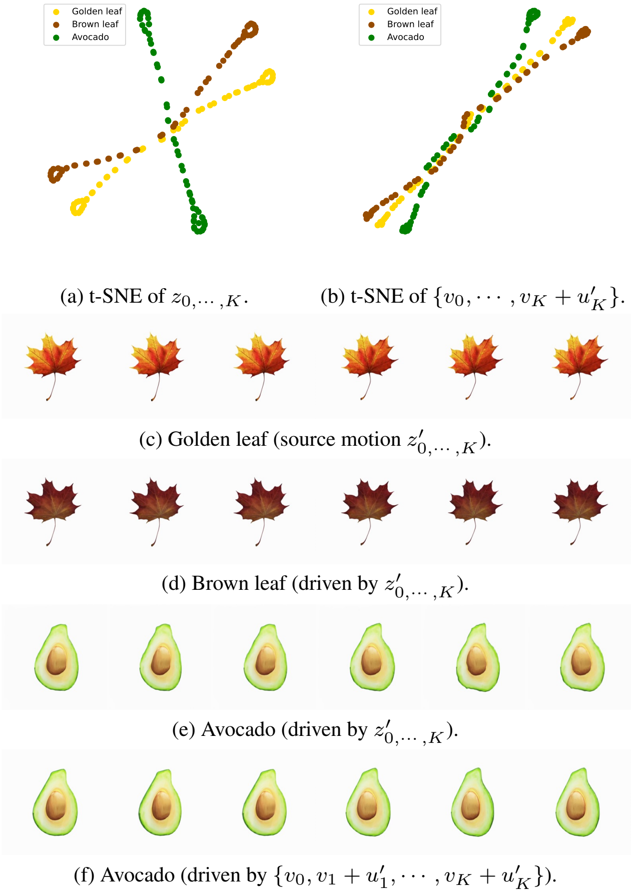
Conceptually, we could train a motion diffusion model to directly capture the distribution of motion flow p, as p symbolizes the motion within the scene. However, we observe that when two images exhibit no motion (i.e., $I_0 = I_k$), p_k is significantly non-zero and fluctuates with different I_0. This implies that the predicted motion flow p not only encapsulates motion information but also integrates visual features.

Although the amalgamation of motion information and visual features might not present a substantial problem on small-scale data, it considerably impairs the diffusion model's capacity to exploit the benefits of large-scale data. This is attributed to the fact that when training on large-scale data, the surplus visual features obstruct the model from abstracting a unified motion law. Conversely, on a small-scale dataset, the model is entirely capable of memorizing all the information.

3.2.1 Decomposition of the Neural Motion Vector

The challenge of disentangling motion information from visual features in the motion flow space prompts us to investigate potential solutions within the latent motion space. We propose a hypothesis, devise a solution based on this hypothesis, and subsequently verify the hypothesis through rigorous experimentation.

Hypothesis We propose that the neural motion vector $z_k = \mathcal{E} \circ \mathcal{P}(I_0, I_k)$ can be decomposed into $z_k = u_k + v_k$, where u_k and v_k denote the motion-related and visual-related components, respectively. For any given pair $\langle I_0, I_k \rangle$, our objective is to independently compute the components u_k and v_k, which could be advantageous for subsequent applications.



(a) t-SNE of $z_{0,\cdots,K}$.

(b) t-SNE of $\{v_0, \cdots, v_K + u'_K\}$.

(c) Golden leaf (source motion $z'_{0,\cdots,K}$).

(d) Brown leaf (driven by $z'_{0,\cdots,K}$).

(e) Avocado (driven by $z'_{0,\cdots,K}$).

(f) Avocado (driven by $\{v_0, v_1 + u'_1, \cdots, v_K + u'_K\}$).

Figure 4. Cross-Scene Motion Transfer (For optimal understanding, we recommend viewing the video provided on our homepage).

Solution We observe that for $\langle I_0, I_0 \rangle$, $z_0 = u_0 + v_0$. In this scenario, the component u_0 is absent, as there is no motion information for an image concerning itself. Therefore, we deduce that $v_0 = z_0$. Moreover, we find that for a fixed I_0 and any I_k, as I_0 remains constant, we can approximate v_k using v_0 (since we are estimating local minor movements, the visual feature disparity between I_k and I_0 is not significant). Consequently, we can compute $u_k = z_k - v_0$. In summary, we derive the following system of equations:

$$\begin{cases} z_k = \mathcal{P}(I_0, I_k) \\ v_k = z_0 \\ u_k = z_k - v_k \end{cases} \tag{6}$$

Validation To validate our proposed hypothesis, we carefully constructed three distinct scenarios using computer simulation. Each scenario involves different objects: Golden leaf, Brown leaf, and Avocado, all of which follow identical motion trajectories. Notably, the Golden leaf and Brown leaf share the same geometric shapes but differ in their color textures. The hypothesis suggests that the neural motion vector z can be effectively decomposed into two components: the motion-related component u (*motion segment*) and the visual-related component v (*visual segment*). We assign the latent motion vector of the Golden leaf as the source motion, and all variables related to it are marked with a prime symbol in the upper right corner. Its actual motion process is demonstrated in Fig. 4c[1].

Initially, we present the latent motion trajectories under the three scenarios in the form of a t-SNE plot, as depicted in Fig. 4a. It is evident from the figure that these three trajectories do not overlap. More specifically, we observe that despite the Golden leaf and Brown leaf differing merely in their texture maps, there is a slight difference between their latent motion trajectories. The latent motion trajectory of the Avocado significantly deviates from the other two. If we attempt to directly drive the Brown leaf and Avocado using $z'_{1,\cdots,K}$, the Brown leaf can still achieve a relatively good animation effect (Fig. 4d). However, the Avocado will exhibit noticeable deformation in the central core area, making it appear shriveled and not plump (Fig. 4e). The upper pulp area also exhibits distortion and deformation. This strongly suggests that the latent motion vector z encapsulates not only motion information but also visual features.

Then, we calculate the visual component $v_{0,K}$ for each scenario. After that, we use Eq. 6 to replace the visual component u'_k in $z'_{0,\cdots,K}$, which results in a new latent motion trajectory, $\{v_0, v_1 + u'_1, \cdots, v_K + u'_K\}$. In Fig. 4b, we illustrate the t-SNE graph of these new three trajectories. When compared with the trajectory of the Golden leaf, they show notable consistency, particularly for the Avocado. Afterwards, we try to animate the Avocado scenario using $\{v_0, v_1 + u'_1, \cdots, v_K + u'_K\}$ (See Fig. 4f). It's clear that the core of the avocado remains full throughout the motion process, and the avocado as a whole doesn't experience any illogical deformation. This strongly indicates that our proposed method can effectively separate visual features and motion information.

3.2.2 Background on Diffusion Models

The foundation of our proposed motion diffusion model is built upon the denoising diffusion probabilistic models (DDPM) [11, 28, 31].

The *forward* process in DDPM generates a Markov chain $\mathbf{x}_1, \ldots, \mathbf{x}_T$ by progressively incorporating Gaussian noise into $\mathbf{x}_0$, a sample drawn from the data distribution. This process follows a variance schedule $\beta_1, \ldots, \beta_T$, as depicted below:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \ , \tag{7}$$

In this equation, the variances β_t are kept constant. When β_t is minimal, the posterior $q(\mathbf{s}_{t-1}|\mathbf{x}_t)$ can be precisely approximated by a diagonal Gaussian [18, 28]. Moreover, if the length of the chain T is sufficiently large, $\mathbf{x}_T$ can be closely approximated by a standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$. These insights suggest that the actual posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ can be estimated by $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, defined as follows:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t), \sigma_t^2\mathbf{I}) \ , \tag{8}$$

where the variances σ_t are also constants.

The Reverse Procedure of DDPM The sampling process, also referred to as the reverse procedure of Denoising Diffusion Probabilistic Models (DDPM), is initiated by generating samples $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0)$ from Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This is followed by a progressive reduction of noise through a Markov chain of $\mathbf{x}_{T-1}, \mathbf{x}_{T-2}, \ldots, \mathbf{x}_0$ utilizing the learned $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$.

To train $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, Gaussian noise ϵ is added to $\mathbf{x}_0$ to generate samples $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)$. Subsequently, a model ϵ_θ is trained to predict ϵ using the mean-squared error loss as follows:

$$\arg\min_{\epsilon_\theta} \mathbb{E}_{t\sim\mathcal{U}(1,T),\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[||\epsilon - \epsilon_\theta(\mathbf{x}_t, t)||^2] \ , \tag{9}$$

Here, the time step t is uniformly sampled from $\{1, \ldots, T\}$. The $\mu_\theta(\mathbf{x}_t)$ in Eq. 8 can be derived from $\epsilon_\theta(\mathbf{x}_t, t)$ to model $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ [11].

The denoising model ϵ_θ is typically implemented via a time-conditioned U-Net [23] with residual blocks [9] and self-attention layers [33]. The time step t is conveyed to ϵ_θ by the sinusoidal position embedding [33]. For conditional generation, i.e., sampling $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0|y)$, a y-conditioned model $\epsilon_\theta(\mathbf{x}_t, t, y)$ can be learned [18, 22].

3.2.3 Motion Diffusion Model

We commence by encoding the input video clips $I_{0,\cdots,K}$ into $z_{0,\cdots,K}$ using a proficiently trained motion rendering model. As discussed in Para. 3.2.1, directly setting $\mathbf{x}_0 = \text{cat}[z_0, \cdots, z_K]$ as the training target for the motion diffusion model would necessitate the simultaneous memorization of the visual segment and motion segment, thereby complicating the training process. We discovered that using v_k as supplementary information and concatenating it with z_k for input while still directly predicting z_k yields superior results compared to directly predicting the motion component by setting $\mathbf{x}_0 = \text{cat}[u_0, \cdots, u_K]$. Concurrently, other additional information is converted into the condition vector y via a feature encoder, such as CLIP [19] or ImageBind [8]. In this scenario, we chose to employ ImageBind to encode I_0 and the optional text into y.

$\mathbf{x}_0$ is progressively transformed into a standard Gaussian noise volume $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ by integrating Gaussian noise

[1] We strongly recommend observing the motion process through the video provided on our homepage

via the DDPM forward process. Conditioned on y, the denoising model $\epsilon_\theta(\mathbf{x}_t, t, y)$ is trained to predict the added noise ϵ in $\mathbf{x}_t$ based on a conditional 3D U-Net [7] with the subsequent loss:

$$\arg\min_{\epsilon_\theta} \mathbb{E}_{t\sim\mathcal{U}(1,T),\mathbf{x}_0\sim q(z_{1,\cdots,K}),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[||\epsilon - \epsilon_\theta(\mathbf{x}_t, t, y)||^2] \ , \tag{10}$$

where the time step t is uniformly sampled from $\{1, \ldots, T\}$. ϵ_θ is further employed in the DDPM reverse sampling process to output $\hat{\mathbf{x}}_0 = \mathrm{cat}[\hat{z}_0, \cdots, \hat{z}_1]$. The training pipeline is depicted in Fig. 2d.

3.3. LVMM Inference

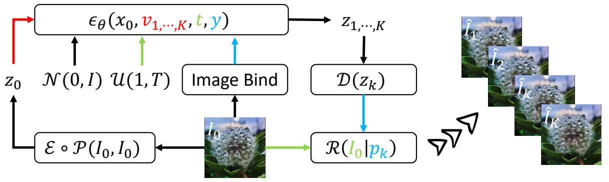


Figure 5. Inference pipeline. Starting with a given reference image I_0, a series of video frames $\hat{I}_{1,\cdots,K}$ is ultimately rendered through neural image renderer $\mathcal{R}$.

As demonstrated in Fig. 5, given an image I_0, we initially compute the corresponding latent motion $z_0 := \mathcal{E} \circ \mathcal{P}(I_0, I_0)$ and condition y, which subsequently produces $v_0 = z_0$ and $v_k = v_0$ from Eq. 6. Conditioned on y, a volume of randomly sampled Gaussian noise $\mathbf{n}$ is progressively denoised by ϵ_θ through the DDPM reverse sampling process to generate the latent motion sequence $\hat{z}_{1,\cdots,K}$. Following this, a video clip $\hat{I}_{1,\cdots,K}$ can be rendered via $\hat{I}_k = \mathcal{R}(I_0|\mathcal{D}(\hat{z}_k))$. It is important to note that the optical flow predictor $\mathcal{P}$ and motion encoder $\mathcal{E}$ are only employed once to compute z_0, and are not used in subsequent steps.

4. Experiments

4.1. Pretraining of Large Visual Motion Model

We have successfully trained two high-performing pretrained models, namely “LVMM-General” and “LVMM-Facial”, on the WebVid10M [2] and CelebV-HD [40] datasets respectively. The experiments were deployed on 32 A100-80GB GPUs, and to ensure the stability of the training, we utilized 32-bit floating-point numbers.

Specifically, the “LVMM-General” model was initially trained on the WebVid10M dataset. We dedicated approximately one week each to complete Phase- 2a and Phase- 2b of the training. Given the presence of watermarks in the WebVid10M data, we randomly selected a subset of data from the HDVILA-100M [37] dataset for Phase- 2c training to eliminate this prior in the Neural Image Renderer $\mathcal{R}$. This process took roughly 12 hours. Before the training of the notion denoising model, we precomputed the required training data, i.e., the latent motion trajectories $z_{0,\cdots,K}$. We utilized 128 V100-32GB GPUs and spent over a week preprocessing the data. Ultimately, it took us two weeks to complete Phase- 2d training.

To enhance the LVMM’s capability of modeling facial motion features, we retrained the “LVMM-General” parameters on the CelebV-HD facial dataset. Specifically, we spent approximately two weeks completing all computations and training, including data processing, to obtain the “LVMM-Facial” model.

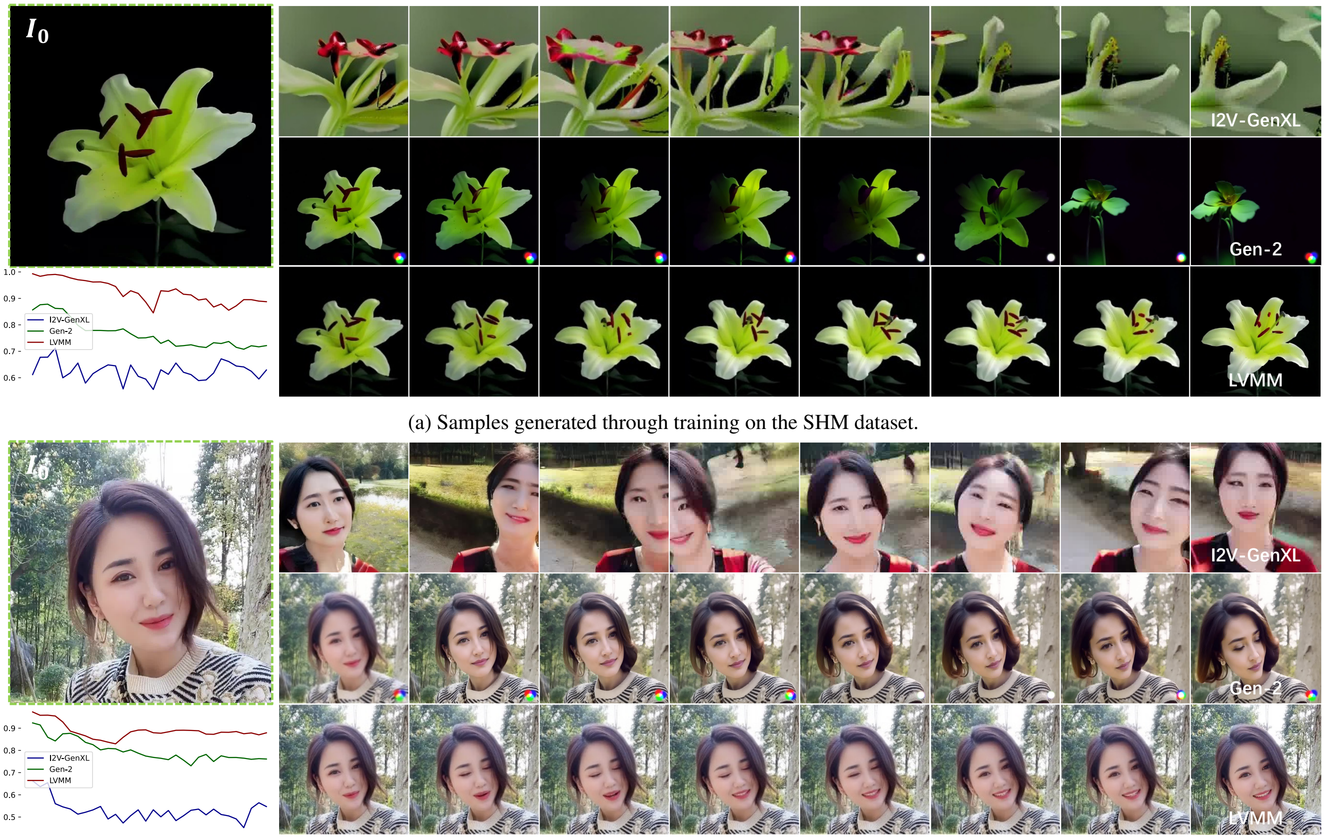
For more detailed information on the experimental setup, model parameters, computational efficiency, etc., please refer to the supplementary materials.

4.2. Quantitative Evaluation

In this section, we validate two aspects through relevant evaluation metrics: a) the superiority of the LVMM model architecture, and b) the significant performance gain brought by large-scale data training.

Dataset. We conducted experiments primarily on three specific task datasets, namely MUG [1], MHAD [6], and NATOPS [30]. The MUG Facial Expression Dataset comprises 1,009 videos featuring 52 subjects, each exhibiting one of seven distinct expressions: anger, disgust, fear, happiness, neutral, sadness, and surprise. The MHAD Human Action Dataset includes 861 videos of 27 actions performed by 8 subjects, covering a wide range of human movements, such as sports actions, hand gestures, daily activities, and training exercises. The NATOPS Aircraft Handling Signal Dataset consists of 9,600 videos of 20 subjects performing 24 body-and-hand gestures used for communication with U.S. Navy pilots.

Baseline. We benchmark LVMM against four competitive baseline models, namely the GAN-based I2V model **ImaGINator** [35], video diffusion models **VDM** [12], a variant of image latent diffusion models **LDM** [36], and the latent flow diffusion models **LFDM**. We maintain the same experimental configuration as in LFDM for a fair comparison. For sampling, we employ a 1000-step DDPM for LDM and LFDM. Given the slow DDPM sampling in the large latent space of VDM (40 × 64 × 64 × 3), we utilize a 200-step DDIM [29] to expedite the sampling process. The training image resolution for LVMM is 512 × 512, and the generated latent motion trajectory is 40 × 16 × 16 × 4. We use a 250-step DDIM sampling strategy for LVMM, and any potential additional text conditions will be encoded through ImageBind and concatenated with the features of I_0. In the experiments, we thoroughly trained each model on the corresponding dataset to ensure convergence.



(a) Samples generated through training on the SHM dataset.

(b) Samples generated through training on the FMM dataset.

Figure 6. Qualitative comparison between I2V-GenXL and Gen-2. In the lower left corner of Fig. 6a and Fig. 6b, we have plotted the similarity of CLIP features between each frame in the video sequence generated by each method and the given reference image I_0. This clearly demonstrates that LVMM is capable of stably generating video sequences highly relevant to the given content.

Evaluation Metrics Consistent with preceding research [10,12,13,27], we utilize the Fréchet Video Distance (**FVD**) [32] to evaluate the *visual quality*, *temporal coherence*, and *sample diversity* of videos synthesized by various methods. To quantify the extent to which a synthesized video aligns with the class condition y (*condition accuracy*) and the provided image I_0 (*subject relevance*), we also adapt two FVD variants, as proposed in [3]: class conditional FVD (**cFVD**) and subject conditional FVD (**sFVD**). Both cFVD and sFVD calculate the distance between the feature distributions of real and synthesized videos under identical class conditions y or subject images I_0, respectively. We initially compute the cFVD and sFVD for each condition y and image I_0 and subsequently report their mean and variance as the final results.

Quantitative Results Table 1 presents quantitative comparisons between our method and the baselines on our test set of unseen video clips. As we maintain the same experimental settings as LFDM, we directly use the data from that paper in the table. In our experimental setup, to ensure an accurate estimation of the feature distributions, we generate 10,000 videos for LVMM under consideration to compute our statistics. The data in the table demonstrates that even when training LVMM directly from scratch on the relevant dataset, our method can achieve superior results, which fully illustrates the superiority of the LVMM structure design. Moreover, the model obtained by fine-tuning based on ”LVMM-Facial” significantly outperforms previous single-image animation baselines in terms of both image and video synthesis quality. This suggests that the videos generated with the assistance of the pre-trained model are more realistic and temporally coherent. Visual comparison results can be found in the supplementary material.

4.3. Qualitative Visualization

In this section, we aim to further elucidate the proficiency of the Local Visual Motion Model (LVMM) in deducing local motion from visual features. Despite the extensive pre-training of LVMM on large-scale datasets, the existing datasets, be it WebVid10M or CelebV-HD, do not

Model	MUG			MHAD			NATOPS		
	FVD↓	cFVD↓	sFVD↓	FVD↓	cFVD↓	sFVD↓	FVD↓	cFVD↓	sFVD↓
ImaGINator [35]	170.73	257.46±62.88	319.37±95.23	889.48	1406.56±260.70	1175.74±327.99	721.17	1122.13±150.74	1042.69±416.16
VDM [12]	108.02	182.90±69.56	213.59±97.70	295.55	531.20±104.25	398.09±121.16	169.61	410.71±105.97	350.59±125.03
LDM [22]	123.88	196.49±66.99	236.26±76.08	280.26	515.29±125.70	427.03±112.31	251.72	506.40±125.08	491.37±231.85
LFDM [17]	27.57	77.86±20.27	108.36±39.60	152.48	339.63±52.88	242.61±28.50	160.84	376.14±106.13	324.45±116.21
LVMM †	23.47	70.32±19.45	102.78±38.45	143.67	315.72±45.34	235.89±23.40	157.92	365.78±104.25	305.67±113.89
LVMM ‡	15.34	54.63±16.13	84.23±27.12	124.56	275.89±34.21	204.75±19.76	144.46	305.12±99.53	277.78±105.07

Table 1. Quantitative comparison. "†" denotes training model from scratch, "‡" signifies fine-tuning the "LVMM-Facial" weights.

offer scenes with distinct motion patterns, thereby impeding the training of high-caliber model parameters. As a result, models trained on these datasets merely serve as pretraining parameters and lack direct applicability.

To rectify the issue of training data, we strive to develop new datasets encapsulating precise and distinct motion information apt for optical flow estimation. We have devised the Simple Harmonic Motion dataset (**SHM**), targeting the omnipresent simple harmonic motion in nature, and the Facial Muscle Movements dataset (**FMM**), concentrating on muscle motion during facial expression alterations. The SHM encompasses numerous close-up, high-definition images of flora swaying in the wind, with objects persistently visible throughout the motion, thereby facilitating superior optical flow computation. The dataset ultimately includes approximately 1500 videos ranging from a few seconds to several minutes. Conversely, the FMM comprises high-definition short videos featuring frontal human portraits. Apart from a rich assortment of facial expressions, all other parts of the videos, including the background, remain as static as possible. This dataset eventually includes video clips of approximately 100 distinct individuals. It is worth noting that scenes with severe appearance alterations during motion are omitted from these two datasets. Supplementary material provides additional information.

Our work is primarily juxtaposed with other large-scale data-trained video generation models, namely I2V-GenXL [16, 34] and Gen-2. I2V-GenXL is currently the sole publicly available video generation model, for which we utilized the official project code [2] to generate videos. For Gen-2, we employed their provided API service [3] for video generation. Since both models utilize undisclosed training data and other implementation details remain unknown, we solely provide a visual comparison of the videos generated by different methods herein.

We fine-tuned "LVMM-General" and "LVMM-Facial" on SHM and FMM for approximately one week, respectively, as depicted in Fig. 6a and Fig. 6b. As can be discerned, compared to directly predicting video streams, our video generation approach by predicting latent motion trajectory can more accurately adhere to the given image content and generate more detailed and controllable motion processes, thereby fully demonstrating the superiority of our algorithm.

4.4. Ablation Study

We visually demonstrate the role of each component in the Motion Rendering Model, primarily corroborating the viewpoints discussed in Section 3.1. Given the input reference image I_0 (Fig. 3a), we aim for the Neural Image Renderer $\mathcal{R}$ to render some features not visible in I_0, such as the clear teeth shown in Fig. 3b. If we attempt to predict only the optical flow $\delta_{x,y}$, we find that this results in a noticeable image blur (Fig. 3c). If we predict the intention map ω but directly use the warp function $\mathcal{F}$ for image rendering, this can generate some motion effects on the input image but fails to accomplish complex motion information, such as opening the subject's mouth (Fig. 3d). Our final result is shown in Fig. 3e. With rendering based on the neural image renderer, we can achieve more complex facial feature transformations, such as opening the mouth or closing the eyes. Our Motion Encoder and Decoder structure can compress the optical flow into a latent motion vector with imperceptible errors (Fig. 3f). By further fine-tuning the Neural Image Renderer $\mathcal{R}$ (based on the reconstructed optical flow $\hat{p}$), we can enhance our image rendering capabilities. As shown in Fig. 3g, the teeth features in the subject's lip area become noticeably clearer (note: these teeth are not visible in the input image I_0). This effectively demonstrates the necessity of each component within the LVMM architecture. The samples are sourced from the CelebV-HQ dataset [41].

5. Discussion and Conclusion

Limitations Although a generalized Motion Diffusion Model has been trained, it is yet incapable of directly producing high-quality motion trajectories encompassing arbitrary motion forms in a zero-shot manner.

Conclusion We have introduced the Large Visual Motion Model (LVMM), empowering it to learn from large-scale prior data. LVMM exhibits the capability of capturing local motion trends across various real-world scenarios, generating high-quality motion trajectories from provided images, and rendering realistic dynamic effects.

[2] https://modelscope.cn/models/damo/Image-to-Video

[3] https://research.runwayml.com/gen2

References

[1] Niki Aifanti, Christos Papachristou, and Anastasios Delopoulos. The mug facial expression database. In *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, pages 1–4. IEEE, 2010. 6

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*, 2021. 6

[3] Yaniv Benny, Tomer Galanti, Sagie Benaim, and Lior Wolf. Evaluation metrics for conditional image generation. *International Journal of Computer Vision*, 129(5):1712–1731, 2021. 7

[4] Zhengda Bian, Hongxin Liu, Boxiang Wang, Haichen Huang, Yongbin Li, Chuanrui Wang, Fan Cui, and Yang You. Colossal-ai: A unified deep learning system for large-scale parallel training. *arXiv preprint arXiv:2110.14883*, 2021. 1

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[6] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Utdmhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE International conference on image processing (ICIP)*, pages 168–172. IEEE, 2015. 6

[7] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 6

[8] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023. 5

[9] 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. 5

[10] 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. 7

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 5

[12] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022. 6, 7, 8

[13] Yaosi Hu, Chong Luo, and Zhenzhong Chen. Make it move: controllable image-to-video generation with text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18219–18228, 2022. 7

[14] Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. Gpt-4 passes the bar exam. *Available at SSRN 4389233*, 2023. 1

[15] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. Multimodal foundation models: From specialists to general-purpose assistants. *arXiv preprint arXiv:2309.10020*, 1, 2023. 1

[16] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 8

[17] Haomiao Ni, Changhao Shi, Kai Li, Sharon X Huang, and Martin Renqiang Min. Conditional image-to-video generation with latent flow diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18444–18455, 2023. 8

[18] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 5

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 5

[20] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020. 1

[21] Mr D Murahari Reddy, Mr Sk Masthan Basha, Mr M Chinnaiahgari Hari, and Mr N Penchalaiah. Dall-e: Creating images from text. *UGC Care Group I Journal*, 8(14):71–75, 2021. 1

[22] 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. 1, 5, 8

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[24] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1

[25] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 1

[26] Aliaksandr Siarohin, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13653–13662, 2021. 2

[27] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3636, 2022. 7

[28] 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. 5

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 6

[30] Yale Song, David Demirdjian, and Randall Davis. Tracking body and hands for gesture recognition: Natops aircraft handling signals database. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pages 500–506. IEEE, 2011. 6

[31] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 5

[32] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 7

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5

[34] Xiang* Wang, Hangjie* Yuan, Shiwei* Zhang, Dayou* Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *arXiv preprint arXiv:2306.02018*, 2023. 8

[35] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. Imaginator: Conditional spatio-temporal gan for video generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1160–1169, 2020. 6, 8

[36] Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. Latent image animator: Learning to animate images via latent space navigation. *arXiv preprint arXiv:2203.09043*, 2022. 6

[37] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 6

[38] Chenshuang Zhang, Chaoning Zhang, Sheng Zheng, Mengchun Zhang, Maryam Qamar, Sung-Ho Bae, and In So Kweon. A survey on audio diffusion models: Text to speech synthesis and enhancement in generative AI. *CoRR*, abs/2303.13336, 2023. 1

[39] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. 1

[40] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. Celebv-hq: A large-scale video facial attributes dataset. In *European conference on computer vision*, pages 650–667. Springer, 2022. 6

[41] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. CelebV-HQ: A large-scale video facial attributes dataset. In *ECCV*, 2022. 8