Motion Optimization with Feature Decoupling for Image Animation

Anonymous submission

Abstract

Image animation converts the motion from driving videos to static source images. Most of the unsupervised works suffer from two fatal defects: 1) Localized single scale motion flows induced by local bias of convolutions make it impossible to model large displacement; 2) Motion flows and identity features are interwoven which hinders semantic understanding about the underlying motion mechanism. In this work, a framework is proposed for image-to-image motion transfer that synergistically combines motion flow optimization with motion-decoupled representation according to theory. The Swin-Transformer is exploited to construct a new motion optimization module based on hierarchical attention toward global/local details to optimize locally/long-range correlated information simultaneously, theoretically avoiding the constraint of CNN's receptive field. Meanwhile, a motion feature decoupling approach exploiting learning orthogonal basis vectors to discriminate identity space from motion space as well as an optimal motion attention fusion strategy coupled together are designed, theoretically offering a solution to decode identity/motion flows in the constrained space. Finally, an image-animation pipeline built upon these frameworks for optimizing both semantically and geometrically integrated flows is implemented, which could be widely used. Extensive experimental results demonstrated that our method achieved excellent performance with 5.0% AKD gain on TaiChiHD, 4.1% on Fashion, 9.0% L1 distance improvement on MGIF, etc., outperforming the state-of-the-art methods on all motion metrics.

Introduction

Image animation transfers the motion of objects in a driving video to static objects in a source image. It has recently been heavily used in game production, fashion design, movie special effects(Naruniec et al. 2020), etc., and other numerous application areas are catching the attention of computer vision researchers.

Traditional model-based methods (Chan et al. 2019; Doukas, Zafeiriou, and Sharmanska 2021; Ha et al. 2020; Ren et al. 2020; Siarohin et al. 2018; Thies et al. 2016; Zhu et al. 2019) adopt the prior knowledge of objects and structures such as keypoints, 3D models, domain labels to assist motion transfer. Generally speaking, these methods depend on some pre-trained models to extract the corresponding structural information, so they are dependent on specific targets such as face (Doukas, Zafeiriou, and Sharmanska 2021; Ha et al. 2020; Thies et al. 2016) and body (Chan et al. 2019; Ren et al. 2020; Siarohin et al. 2018; Zhu et al. 2019). It greatly restricts their versatility and flexibility, which is very difficult to adapt to the motion transfer requirements of any object in the open scene.

In recent years, several unsupervised methods have been proposed that can realize end-to-end motion transfer without using object priors (Wiles, Koepke, and Zisserman 2018; Siarohin et al. 2019a,b, 2021; Wang et al. 2024; Zhao and Zhang 2022). Usually, they take two randomly chosen images from a video frame as inputs: one for source image and another for driving image. Some existing works introduce three special modules – a keypoint detector, dense motion network and generator; then calculate the perceptual loss between reconstructed image and driving image to iteratively update the motion representation progressively. For example, FOMM (Siarohin et al. 2019b) captures motion info by predicting multiple keypoints and its Jacobian matrices simultaneously, achieving good results. However, since affine transform is linear, it cannot represent complex local nonlinear motions. Instead, TPSMM (Zhao and Zhang 2022) uses non-linear thin-plate-spline (TPS) transform and introduces a multi-scale feature fusion generator to achieve better motion transferring results. Compared with traditional model-based methods, they can transfer motion to any object instead of specific ones.

Although there are some improvements compared with previous methods, the unsupervised methods share two strongly connected inherent weaknesses at the same time: Firstly, the convolutional operation is used to estimate the motion parameters, but it is impossible to capture the long-range relationship due to the locality bias of convolutions, so only coarse-grained single-scale motion flows could be obtained, which will limit the precision of large-displacement motion capturing. Secondly, the lack of supervision makes it extremely hard to get well-decoupled features representations, this leads to entangled feature space in which identity and motion information are mixed together, resulting in poor generation ability for more semantically meaningful motion transfers.

Considering these related problems, we propose a theoretical scheme based on the synergy of hierarchy attention and explicit feature space decoupling to solve them. The design idea of our motion optimization method is to choose Swin-

Transformer according to theory; Hierarchical attention can realize multi-scale motion modeling without sacrificing calculation efficiency. The Shifted Window mechanism ensures that local detail acquisition and large motion dependency model are solved simultaneously under the condition of directly solving the receptive field constraint of convolutions. At the same time, the orthogonal basis vector mode adopted in our feature decoupling method makes identity and motion semantic information have an ideal separation: namely, it realizes semantically meaningful, identity-insensitive motion feature representation under the identity preservation constraint. It has become a multiplicative relationship instead of an additive relationship after good combination with its own advantages. After the optimal motion flow provides an effective geometric consistent structure, it improves the effect brought by the decoupled motion feature flow which guides it in terms of semantics. So as to form a perfect complementary and fused relationship, which constitutes our motion attention fusion component.

Our main contributions are summarized as follows:

- A new theoretical method to solve the dual demands of geometric conformity and semantic richness in motion transfer by combining motion optimization and feature decoupling methods.
- A motion optimization module based on Swin-Transformer is proposed, which can obtain the accurate multi-scale motion flow by capturing local fine-grained details and global motion dependency better.
- Propose a motion feature decoupling scheme based on learnable orthogonal basis vectors and a motion attention fusion module combining geometric and semantic information for better motion transfer accuracy.
- Extensive experimental validation demonstrates significant improvements over state-of-the-art approaches across multiple datasets, with particularly notable gains in motion-related metrics.

Related Work

Motion Transfer

The previous methods all need explicit structure representations. They extract the pose information by existing pose estimators or keypoint detectors and then input them to image generation tasks. That is, 2D landmarks (Chan et al. 2019; Ha et al. 2020; Ren et al. 2020; Siarohin et al. 2018; Zhu et al. 2019) and 3D models (Doukas, Zafeiriou, and Sharmanska 2021; Thies et al. 2016), manually annotated or extracted by other pre-trained models, are used for transferring motion target objects like faces and human bodies. The above methods can transfer human poses or facial expressions effectively but depend on other models according to certain objects. Recently, unsupervised methods based on video reconstruction tasks use self-supervision to train frameworks end-to-end (Wiles, Koepke, and Zisserman 2018; Siarohin et al. 2019a,b, 2021; Wang et al. 2024; Zhao and Zhang 2022). Among these methods, Monkey-Net (Siarohin et al. 2019a) proposes the first model-free motion transfer method for arbitrary objects via constructing a motion flow from aligned keypoints to distort source image features and move them toward corresponding positions with respect to the pose. FOMM (Siarohin et al. 2019b) uses a first-order motion model that introduces local affine transformation parameters at each keypoint to predict motion. An occlusion-aware generator that predicts occlusion maps inside a dense motion network to repair occluded areas is also utilized. MRAA (Siarohin et al. 2021) defines regions modeling different parts of an object and introduces a background motion predictor to estimate background motion. TPSMM (Wang et al. 2024) replaces the affines in FOMM with thin plate spline (TPS) transformations increasing the flexibility of the motion model and introduces multi-scale occlusion maps. CPAB (Zhao and Zhang 2022) uses continuous piecewise affine transformations to model motion transforming the image segmentally so it better preserves original features during the process of motion transfer. These methods suffer from two drawbacks: (1) Convolution-based operations cannot effectively model long-range dependencies so they may perform poorly when dealing with large displacement motions; (2) Since they are unpaired training methods, it is hard to get good-decoupled feature representation, resulting in entangled identity and motion information. Differently from those above approaches only considering geometry aspect or semantic one separately, we provide a unified framework solving both problems simultaneously.

Motion Optimization

There are many ways to calculate and optimize optical flow. RAFT (Teed and Deng 2020) calculates the correlation of all feature pixel pairs in the input image pair, and then updates the flow field prediction through GRU's module; but it uses CNN. It is impossible to get global information when iteration occurs, making it difficult to model large movement size. The Transformer architecture has been introduced into vision problems in recent years (Vaswani et al. 2017). Flowformer (Huang et al. 2022) uses a method that segments the traditional 4D flow cost volume into cost tokens, and then alternates between groups for transformation self-attention global coding and iterative decoding by ConvGRU for estimating the optical flow. Although the matching scheme adopts the full scale computing cost-volume, its fixed self attention range will inevitably cause high costs. We choose computationally efficient and more powerful than normal transformer Swin-transformer (Liu et al. 2021), but unlike transformers, swin-transformers do not have quadratic complexity, but multi-level structures based on hierarchy combined with shift windows can capture different scales of characteristics at each level linearly, solving the problem of the conflict of computational ability limited by other solutions due to the need for capturing a larger number of parameters resulting in a greater amount of calculations required for the global perspective.

Feature Space Decoupling

In their previous work(Zhu et al. 2025), they have confirmed the efficiency of decoupling feature space for motion transfer in face-specific situations. They separated the identity

subspace and the motion subspace based on orthogonal basis vectors to realize face animation.

We also adopt the idea of using an orthogonal basis vector, but here we provide some new ideas: generalizing feature space decoupling from specific faces to any objects, novelly combined with Transformer-based motion optimization with our own motion attention fusion part, and theoretically explained its effectiveness through our geometry semantics decomposition theory.

Method

The overall view of our method is illustrated in Figure 1. Our model extends TPSMM(Zhao and Zhang 2022) by adding a motion flow optimization module and designing a motion attention fusion module to fuse the decoupled motion feature vector and warped feature.

Flow based Motion Estimation

The affine transform (Siarohin et al. 2019b, 2021) and thin plate spline transform (Zhao and Zhang 2022) are commonly used motion models. The flow method usually aligns a source image to a series of driving video frames by selecting one driving frame from the driving video sequence as reference for aligning at each step. Then selected motion fields will be applied on source image in sequence to synthesize the transferred video.

Mathematically, based on the thin plate transform, given a source image $\mathbf{S}$ and a driving frame $\mathbf{D}$, we first use the keypoint detector $mathcalD$ to predict their keypoints respectively:

$$P_{\boldsymbol{S}}^{i}=\mathcal{D}(\boldsymbol{S}) \quad P_{\boldsymbol{D}}^{i}=\mathcal{D}(\boldsymbol{D}), \quad i=1,2,\ldots,N, \tag{1}$$

Where $P_{\boldsymbol{X}}^{k} \in \mathcal{R}^{2\times 1}$ are the keypoints of image $\boldsymbol{X}$. With a keypoint detector, we can predict $K \times N$ keypoints for $\boldsymbol{S}$ and $\boldsymbol{D}$, where K represents the number of TPS transformations. It is unsupervised. Each pair of the N keypoints leads to one TPS transformation from $\boldsymbol{S}$ to $\boldsymbol{D}$. Given the keypoints, the parameters $A_k \in \mathcal{R}^{2\times 3}, W_{ki} \in \mathcal{R}^{2\times 1}$ of the TPS transformations can be estimated. Therefore, one of the TPS transformations can be written as:

$$\mathcal{T}_k(z)=A_k\left[\begin{array}{c} z \\ 1 \end{array}\right]+\sum_{i=1}^{N} W_{ki} U(\left\|z-P_{\boldsymbol{D}}^{ki}\right\|_2), \tag{2}$$

Where z are the coordinates of any pixel in an image. $U = r^2 \log r^2$ is the radial basis function describing the influence of each keypoint on the pixel at z.

In addition to the K transformations, the motion flow of the background can also be expressed as:

$$\mathcal{T}_{bg}(z)=A_{bg}\left[\begin{array}{c} z \\ 1 \end{array}\right], \tag{3}$$

Where $A_{bg} \in \mathcal{R}^{2\times 3}$ is an affine transformation matrix predicted by the background motion predictor proposed by (Siarohin et al. 2021).

Finally, in order to obtain the final motion flow, we introduce a weight factor M_k,$k = 0, \ldots, k$, to combine all different transformations. Thus, the dense motion flow of any pixel z can be written as:

$$\mathcal{T}_{\boldsymbol{S}\rightarrow\boldsymbol{D}}(z)=M_0(z)\mathcal{T}_{bg}(z)+\sum_{k=1}^{K} M_k(z)\mathcal{T}_k(z). \tag{4}$$

We treat Eq.(4) as a rough motion flow estimation. Henceforth, we denote it by $\mathcal{F}_{init} = \mathcal{T}_{\boldsymbol{S}\rightarrow\boldsymbol{D}}(z)$, for short, if not specified otherwise.

Motion Flow Optimization

The previous calculation result of the low-resolution motion flow $\mathcal{F}_{init}$, which reflects the rough motion of the object due to the low resolution. It cannot acquire sufficient global feature information because of the limitation of convolving operation, so its motion modeling ability is limited. More importantly, insufficient multi-scale feature fusion capability is caused by the lack of high-resolution details.

The main reason why existing methods have these limitations is that it is limited by the convolution operation. In general, if a CNN has L layers and a kernel size of k, then its receptive field expands at most to $1 + L(k - 1)$, increasing linearly with depth; this is a crucial bottleneck for motion prediction requiring long-range dependencies like large-displacement motions. Our Swin-Transformer based Motion Transformer model takes advantage of the hierarchical attention structure between windows with our new cross-window shift scheme: we allow each attention query head to attend all positions within an image theoretically, i.e., a full-receptive-field structure.

Inspired by recent visual works, using two window sizes, 8, 4 allows us to model small- and medium-sized motion scales differently. In theory, one should consider it when predicting finer-grained local motions or coarser-grained global movements; thus different sized windows are used. Using windows sizes 8 and 4 gives a range of different scales -

$$\mathcal{F}_{multi}=\sum_{s\in\{4,8\}} \mathcal{W}_s * \text{SwinAttn}_s(\mathcal{F}_{init}) \tag{5}$$

Where $\mathcal{W}_s$ represents learned weights for each scale s.

Inspired by the advantages of Swin-transformer (Liu et al. 2021) and super-resolution methods (Sun et al. 2023; Ge, Gong, and Yu 2018), we propose a motion flow optimizing module to address these issues. The process is illustrated in Fig. 2. Our module takes two inputs: the coarse motion flow $\mathcal{F}_{init} \in \mathcal{R}^{h\times w\times 2}$ and the warped source image feature $\boldsymbol{F}_w \in \mathcal{R}^{c\times h\times w}$. We first bilinear interpolate the coarse motion flow and the feature map to size $H \times W$, then swap their channels, and concatenate them along the channel dimension, which is our module's input $\boldsymbol{T} \in \mathcal{R}^{C\times H\times W}$. Because input features with different scales have different resolutions, if we simply flatten high-level feature maps for compensation, the computation cost will be very high. Here we use window partition; assume that the input has a feature map with shape (C, H^i, W^i) at scale i, and split it

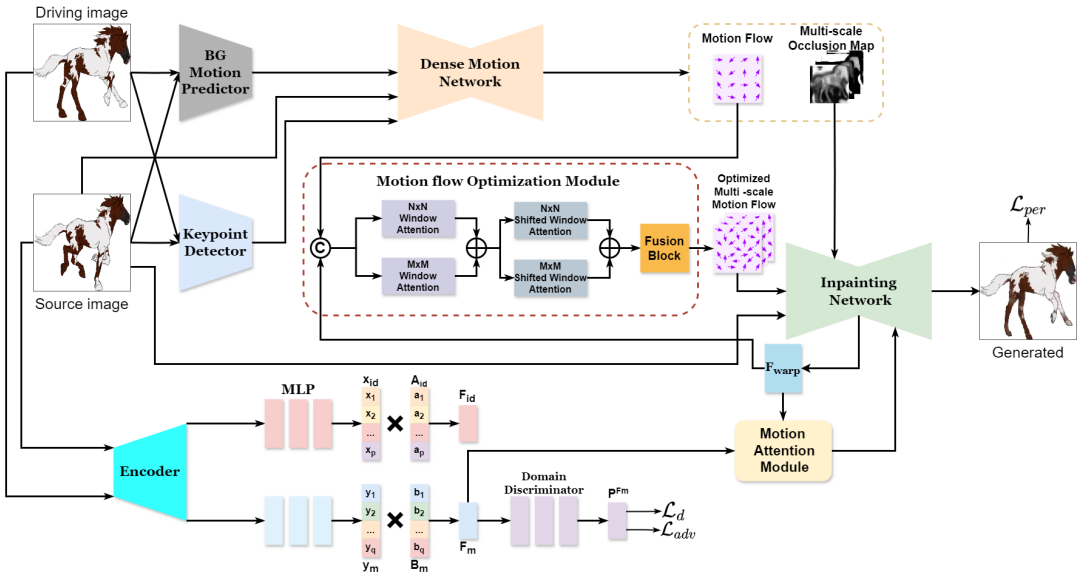

Figure 1: Our model overview. The coarse motion flow output by the Dense Motion Network will be sent to the motion flow optimization module to obtain the multi-scale optimized motion flow, and the motion feature vector separated by decoupling will be sent to the motion attention module with the warped features for feature fusion.

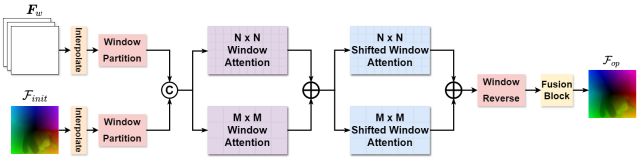

Figure 2: The illustration of our motion optimization module. We learn diverse features through two Swin-transformer blocks with different window sizes.

into pieces with shape $(H^i/h_p, W^i/w_p)$, so that the feature map's shape becomes $(C \times H^i/h_p \times W^i/w_p, h_p, w_p)$, linearly project it into C dimensions, generate more compact input feature maps $\boldsymbol{T}^i \in \mathcal{R}^{C \times h_p \times w_p}$ at scale i. We set $h_p = 32$ and $w_p = 32$. The motion flow optimization module contains four swin-transformer blocks. Each block uses shifted window attention to generate multi-scale optimized motion flows. To combine different features, we design two windows' attention branches with sizes of 8 and 4 respectively. Given the input $\boldsymbol{T}^i$, we compute the attention features with each window size of 8 and 4, add them together, and send the added features into the next stage. Two more shifted window attentions are applied in this way. Set the stride of the shifted window to half of the window size. Finally, we restore the added features through window reverse. Through linear projection and reshape operations, we can obtain the original feature maps (C, H^i, W^i). We feed these restored features into the fusion module. Formally, we denote the process by:

$$
\begin{aligned}
&x_1^i = \mathcal{S}_1(\boldsymbol{T}^i), \quad x_2^i = \mathcal{S}_2(\boldsymbol{T}^i), \quad x_{12}^i = x_1^i + x_2^i, \\
&x_3^i = \mathcal{S}_1^{shifted}(x_{12}^i) \quad x_4^i = \mathcal{S}_2^{shifted}(x_{12}^i), \\
&X^i = Window\ Reverse(x_3^i + x_4^i), \\
&\mathcal{F}_{op}^i = FusionBlock(X^i),
\end{aligned}
\tag{6}
$$

Where S_1 and S2 are the Swin-Transformer blocks with different window size. $S_1^{shifted}$ and $S_2^{shifted}$ represent the shifted window attention block, respectively. FusionBlock contains two 3x3 convolution blocks and one ReLU layer. Finally, we can get the optimized motion flow F_{op}^i at scale i.

Decoupling Identity and Motion Subspaces

To obtain the semantically useful motion representation out of the unsupervised observation is a blind source separation problem. Since the given feature space F includes both ID and motion information, we would like to separate them by multiplying a vector G on their linear combination. To avoid the redundancies between themselves, each orthogonal basis vectors must not be related with any linears relation to others; therefore, those ID and motion subspaces will remain independent.

Thus referring to the paper(Zhu et al. 2025) to decouple the feature space, let's use a group of learnable orthogonal basis vectors set as $\mathbf{C} = \{\mathbf{a}_1, \ldots, \mathbf{a_p}, \mathbf{b_1}, \ldots, \mathbf{b_q}\}$, where a portion of the basis vectors represents the identity subspace $\mathbf{A_{id}} = \{\mathbf{a}_1, \ldots, \mathbf{a_p}\}$, and the rest represents the motion subspace $\mathbf{B_m} = \{\mathbf{b}_1, \ldots, \mathbf{b_q}\}$, any two basis vectors satisfy the constraint:

$$
\langle \mathbf{c_i}, \mathbf{c_j} \rangle = 0 \text{ if } i \neq j; 1 \text{ if } i = j. \tag{7}
$$

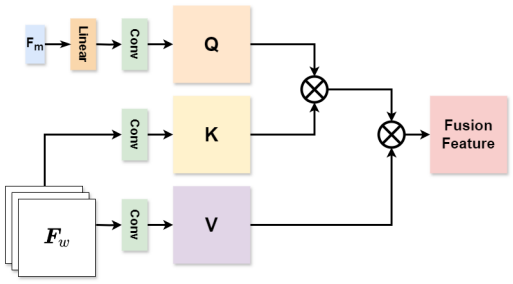

Figure 3: The illustration of our Motion Attention Module. Here we use linear layer, 1x1 convolution and multiplication between matrices.

C is learnable and Gram-Schmidt is conducted every time in the forward step to ensure orthogonality. We use a parameter-sharing encoder to extract features from the source image and the driving image, respectively, and generate a set of weight coefficient vectors $\mathbf{x_{id}} = \{x_1, \ldots, x_p\}$ and $\mathbf{y_m} = \{y_1, \ldots, y_q\}$ respectively through a three-layer MLP. Therefore, the feature space can be expressed as:

$$\mathbf{F} = \mathbf{x_{id}}\mathbf{A_{id}} + \mathbf{y_m}\mathbf{B_m} = \sum_{i=1}^{p} x_i \mathbf{a_i} + \sum_{i=1}^{q} y_i \mathbf{b_i}, \quad (8)$$

$\mathbf{F_{id}} = \mathbf{x_{id}}\mathbf{A_{id}}, \mathbf{F_m} = \mathbf{y_m}\mathbf{B_m}$ are the identity feature vector and motion feature vector, respectively.

Eliminating identity from motion subspace In order to make the motion subspace does not contain the identity information of the driving image, we use a domain discriminator composed of three layers MLP to remove it. We input the motion feature vector $\mathbf{F_m}$ into the domain discriminator and get the prediction label:

$$P^{\mathbf{F_m}} = D(\mathbf{F_m}), \quad (9)$$

Where $D(\cdot)$ is the domain discriminator, $P^{\mathbf{F_m}} \in \mathcal{R}^N$ is the predicted identity label, N is the number of identities, and we optimize the domain discriminator with cross entropy loss:

$$\mathcal{L}_d = CE(P, P^{\mathbf{F_m}}), \quad (10)$$

Where $\mathcal{L}_{CE}(\cdot)$ is the cross entropy. Let P be the label of driving image, we use domain loss $\mathcal{L}_d$ as adversarial loss $\mathcal{L}_{adv}$ with a negative value weight λ_d to remove the identity term from the motion space; that is, using $\mathcal{L}_{adv}$ to optimize D. Thus it is equal to optimizing the domain discriminable loss $\mathcal{L}_d$ by multiplying $-\lambda_d$ on $\mathcal{L}_d$ and training the decoupled generative model to make the optimal output has almost zero domain discriminability w.r.t. the source video. This way, we blank or erase (if discriminant ability is sufficient but there is still some residual information) the motion representation. For keeping consistency in terms of motions between driving image and generated one, another latent regression loss $\mathcal{L}_r$ will act as guidance for the corresponding motion subspace.

$$\mathcal{L}_r = \left\| \hat{\mathbf{F}}_\mathbf{m} - \mathbf{F_m} \right\|_1, \quad (11)$$

Where $\hat{\mathbf{F}}_\mathbf{m}$ is the extracted motion subspace of the generated image; because we only need to obtain the motion feature vector, we do not consider the identity subspace.

Motion feature vector fusion The coupling of the optimized motion flows and the decoupled motion features corresponds to the complementation information fusion rule: that is, the motion flow means "where to look", while the decoupled feature means "what to look". Our fusion module's attention mechanism could be regarded as an adaptive feature combination mode where the geometric information and semantic information are optimally matched according to the local area.

After applying the aforementioned method to decouple the motion feature vector F_m, its rich motion information and semantic information were obtained, without identity information interference representing more detailed motions. In order to be able to fuse with unsupervised methods, we specially designed a motion attention module. We used cross-attention to fuse F_m and the warped feature Fw. First, the linear layer reduced Fm to C dimension; then Fm expanded into 3 channels, and finally expanded along the last two channels into the same-resolution feature map $\bar{F}_m \in R^{C\times H\times W}$ compared with Fw. The warpped gradient feature map $\bar{F}_m$ worked as ***query*** here, and meanwhile the warpped feature map Fw worked as both ***key*** and ***value***. It can be written as:

$$\mathbf{F_s} = \text{Softmax}\left(\left(\mathbf{W}_q \bar{\mathbf{F}}_\mathbf{m}\right) \left(\mathbf{W}_k \mathbf{F_w}\right)^T \right) \times \left(\mathbf{W}_v \mathbf{F_w}\right), \quad (12)$$

Where $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v$ are different learnable weight matrices for feature mapping to the QKV space; this process is shown in Figure 3. The fused feature F_s carries motion and semantic information and can facilitate better motion transfer.

Training

We train our method end-to-end. Like the previous methods (Siarohin et al. 2019a), we use perceptual loss as a major Loss function according to the pre-trained VGG-19 network (Johnson, Alahi, and Fei-Fei 2016). For the driving image $\boldsymbol{D}$ and generated image $\widetilde{\boldsymbol{D}}$, the perceptual loss is given by:

$$\mathcal{L}_{per} = \sum_i \sum_l \left\| \phi_l\left(\boldsymbol{D}_i\right) - \phi_l\left(\widetilde{\boldsymbol{D}}_i\right) \right\|_1, \quad (13)$$

Where ϕ_l is the feature extractor of the first layer of VGG-19 network, and i denotes the index of features with different resolutions. Similar to TPSMM (Zhao and Zhang 2022), we also employ a warp loss as:

$$\mathcal{L}_{\text{warp}} = \sum_i \left| \mathcal{T}\left(E_i\left(\boldsymbol{S}\right)\right) - E_i\left(\boldsymbol{D}\right) \right|, \quad (14)$$

Where $\boldsymbol{S}$ is source image, E_i denotes the $i-th$ layer encoder in generator. Equivariance loss is used to constrain the keypoint Detector:

$$\mathcal{L}_{eq} = \left| E_{kp}\left(\mathcal{T}_{rand}\left(\boldsymbol{S}\right)\right) - \mathcal{T}_{rand}\left(E_{kp}\left(\boldsymbol{S}\right)\right)\right|, \quad (15)$$

Where E_{kp} is Keypoint Detector and $\mathcal{T}_{rand}$ denotes a random nonlinear TPS transformation.

We retain the loss $\mathcal{L}_{bg}$ of TPSMM. Overall, the total training loss function is:

$$\begin{aligned}\mathcal{L} = \lambda_{per}\mathcal{L}_{per} + \lambda_{warp}\mathcal{L}_{warp} + \lambda_{eq}\mathcal{L}_{eq} + \lambda_{bg}\mathcal{L}_{bg} \\ + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{d}\mathcal{L}_{d} + \lambda_{r}\mathcal{L}_{r}\end{aligned}, \quad (16)$$

Where λ_{per}, λ_{warp}, λ_{eq}, λ_{bg}, λ_{adv}, λ_{d}, λ_{r} are hyperparameters.

Experiments

In this section, we evaluate our approach on benchmark datasets and conduct ablation experiments to analyze the performance of each module.

Datasets

We train and evaluate our method on four datasets from different object categories, including human bodies, faces and pixel animals. The datasets are as follows:

- TaiChiHD (Siarohin et al. 2019b): The dataset contains 2867 training videos and 253 test videos. People from different backgrounds are practicing Taichi in the video dataset, so it is difficult due to the large range of motion. We resize all videos to a size of 256×256.
- FashionVideo (Zablotskaia et al. 2019): The dataset consists of 500 training videos and 100 test videos that contain many different models wearing different clothes performing similar poses. These videos were also resized to a size of 256×256.
- VoxCeleb (Nagrani, Chung, and Zisserman 2017) is a talking head dataset containing 19522 training 525 test videos. All videos are resized to a 256×256 resolution.
- MGIF (Siarohin et al. 2019a) is a cartoon animal dataset containing 900 training videos and 100 test videos. All videos are resized to a 256×256 resolution.

Experiment Settings

Evaluation Protocols Since there is no ground-truth video, we use the previous methods to do quantitative experiments based on a video reconstruction task. In this case, the first frame of each test video is used as the driving image, while the others are considered as source images to drive the reconstructed video for quantitative comparison with the test video.

Evaluation Metrics We use the same quantitative metrics as previous work:

- $\mathcal{L}_1$: Average $\mathcal{L}_1$ distance between the generated and driving image pixel values.
- Average Keypoint Distance (AKD): This index indicates the pose of an image detected by a pre-trained keypoint detector (Bulat and Tzimiropoulos 2017; Cao et al. 2017). It firstly employs a pre-trained keypoint detector to detect keypoints for both driving and generated images and then obtains the average distance of each key point.
- Missing Keypoint Rate (MKR): MKR indicates the proportion of keypoints extracted from the keypoint detecting model (Bulat and Tzimiropoulos 2017; Cao et al. 2017) but presented in driving image but not on generated image.
- Average Euclidean Distance (AED): This index reflects how close a fake video frame generated is relative to real video frames (extracted from the corresponding feature space) which have been labeled through a pre-trained network (Amos et al. 2016; Hermans, Beyer, and Leibe 2017). In other words, it illustrates how similar the identity of images has kept.

Implementation Details We choose $K = 10$ and $N = 5$, meaning that we use ten TPS transformations, and each needs five control points. We model both identity space and motion space with 20 basis vectors, where they have the same size of 512 in dimension. We use PyTorch and Adam to update parameters. We train for 100 epochs using one GeForce RTX 4090 card starting from a learning rate of 2×10^{-4}. We configure our hyperparameters as $\lambda_{\text{per}} = 10$, $\lambda_{\text{warp}} = 10$, $\lambda_{\text{eq}} = 10$, $\lambda_{\text{bg}} = 10$, $\lambda_{\text{adv}} = 10$, $\lambda_{d} = -1$, and $\lambda_{r} = 10$.

Quantitative Evaluation

Video reconstruction To quantify the comparison with FOMM, MRAA and TPSMM, we show the quantitative results on 4 different datasets in Table 1. Our model obtains more improvements on motion-related AKD metrics and structure-related MKR metric on the TaiChiHD, Fashion, and VoxCeleb datasets respectively due to our motion optimization and motion decoupled representation. Besides, our method gets better improvement on the L1 metric by MGIF which contains various animal species with different shapes and huge motion ranges. We can also synthesize videos from large displacement motions. Some cross-identity image animation results on four datasets are shown in Figure4 compared with TPSMM. Our method outperforms TPSMM and the result on MGIF dataset by our method is significantly better than TPSMM.

Image Animation

Ablation Study We analyze the effect of our modules in this part. We design some ablation experiments on MGIF and TaiChiHD datasets: 1) w/o DR (Decoupled Representation), 2) w/o MOM (Motion Optimization Modules). The results are shown in Table 2. Its second row validates the necessity of using motion-decoupled representation to achieve a satisfactory result in motion synthesis. Since there are many kinds of animal postures in MGIF, we need more motion information provided by disentangled motion feature vectors for faithful motion transfer with different postures in MGIF. Third row demonstrates the usefulness of our proposed motion flow optimization module capturing more effective global dependency and can be utilized to model large displacement motions that frequently appear in MGIF and Tai Chi sequences in TaiChiHD respectively. Therefore, compared with other baselines without the proposed motion flow optimization module, adopting our motion flow

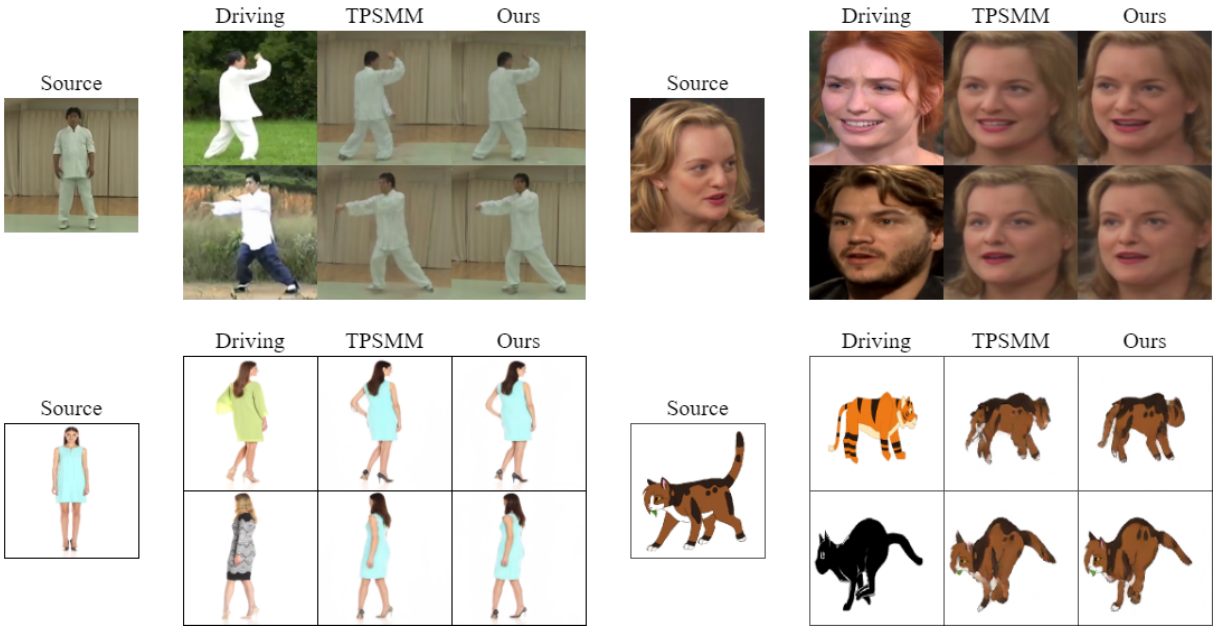

Figure 4: Cross-identity image animation task compared with TPSMM: TaiChiHD (top left), VoxCeleb (top right), Fashion (bottom left), MGIF (bottom right).

	TaiChiHD			Fashion			VoxCeleb			MGIF
	$\mathcal{L}_1$	(AKD, MKR)	AED	$\mathcal{L}_1$	(AKD, MKR)	AED	$\mathcal{L}_1$	AKD	AED	$\mathcal{L}_1$
FOMM	0.063	(6.86, 0.036)	0.179	0.013	(1.131, 0.006)	0.059	0.041	1.29	0.135	0.0264
MRAA	0.048	(5.41, 0.025)	0.149	0.012	(1.106, 0.005)	0.059	0.040	1.29	0.136	0.0274
TPSMM	0.045	(4.57, 0.018)	0.151	**0.011**	(0.845, 0.005)	**0.056**	**0.039**	1.22	0.125	0.0212
Ours	**0.044**	(**4.34**, **0.017**)	**0.148**	**0.011**	(**0.810**, **0.004**)	**0.056**	**0.039**	**1.20**	**0.123**	**0.0193**

Table 1: Quantitative comparison of video reconstruction task with the state of the art on four different datasets.

	MGIF	TaiChiHD		
	$\mathcal{L}_1$	$\mathcal{L}_1$	(AKD, MKR)	AED
TPSMM	0.0212	0.0454	(4.57, 0.018)	0.151
Ours w/o DR	0.0200	0.0450	(4.38, **0.017**)	0.150
Ours w/o MOM	0.0196	0.0446	(4.40, 0.018)	**0.148**
Ours	**0.0193**	**0.0444**	(**4.34**, **0.017**)	**0.148**

Table 2: Ablation study for video reconstruction on MGIF and TaiChiHD.

optimization module is highly beneficial to the final motion transfer results.

Conclusion

In this paper, we presented a novel theoretical framework for unsupervised motion transfer that addresses two fundamental limitations of existing methods: the inability to capture long-range motion dependencies and the difficulty in obtaining semantically decoupled feature representations. Our key insight is that effective motion transfer requires the synergistic combination of geometric consistency and semantic richness. Our optimization module represents the first application of hierarchical attention mechanisms to motion flow estimation in unsupervised settings. The multi-window design specifically addresses the multi-scale nature of motion patterns, while the integration with orthogonal feature decoupling provides unprecedented semantic control. Extensive experiments across four diverse datasets demonstrate that our approach achieves significant improvements over state-of-the-art methods, with particularly notable gains in motion-related metrics.

While our method achieves strong performance, future work could explore extension to video-to-video motion transfer, handling of more complex motion patterns like fluid dynamics, and integration with 3D motion models for enhanced realism. Our theoretical framework and technical contributions provide both practical improvements and theoretical insights that can benefit the broader computer vision community.

References

Amos, B.; Ludwiczuk, B.; Satyanarayanan, M.; et al. 2016. Openface: A general-purpose face recognition library with mobile applications. *CMU School of Computer Science*, 6(2): 20.

Bulat, A.; and Tzimiropoulos, G. 2017. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE international conference on computer vision*, 1021–1030.

Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7291–7299.

Chan, C.; Ginosar, S.; Zhou, T.; and Efros, A. A. 2019. Everybody dance now. In *Proceedings of the IEEE/CVF international conference on computer vision*, 5933–5942.

Doukas, M. C.; Zafeiriou, S.; and Sharmanska, V. 2021. Headgan: One-shot neural head synthesis and editing. In *Proceedings of the IEEE/CVF International conference on Computer Vision*, 14398–14407.

Ge, W.; Gong, B.; and Yu, Y. 2018. Image super-resolution via deterministic-stochastic synthesis and local statistical rectification. *ACM Transactions on Graphics (TOG)*, 37(6): 1–14.

Ha, S.; Kersner, M.; Kim, B.; Seo, S.; and Kim, D. 2020. Marionette: Few-shot face reenactment preserving identity of unseen targets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 10893–10900.

Hermans, A.; Beyer, L.; and Leibe, B. 2017. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.

Huang, Z.; Shi, X.; Zhang, C.; Wang, Q.; Cheung, K. C.; Qin, H.; Dai, J.; and Li, H. 2022. Flowformer: A transformer architecture for optical flow. In *European conference on computer vision*, 668–685. Springer.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 694–711. Springer.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Nagrani, A.; Chung, J. S.; and Zisserman, A. 2017. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*.

Naruniec, J.; Helminger, L.; Schroers, C.; and Weber, R. M. 2020. High-resolution neural face swapping for visual effects. In *Computer Graphics Forum*, volume 39, 173–184. Wiley Online Library.

Ren, Y.; Yu, X.; Chen, J.; Li, T. H.; and Li, G. 2020. Deep image spatial transformation for person image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7690–7699.

Siarohin, A.; Lathuilière, S.; Tulyakov, S.; Ricci, E.; and Sebe, N. 2019a. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2377–2386.

Siarohin, A.; Lathuilière, S.; Tulyakov, S.; Ricci, E.; and Sebe, N. 2019b. First order motion model for image animation. *Advances in neural information processing systems*, 32.

Siarohin, A.; Sangineto, E.; Lathuiliere, S.; and Sebe, N. 2018. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3408–3416.

Siarohin, A.; Woodford, O. J.; Ren, J.; Chai, M.; and Tulyakov, S. 2021. Motion representations for articulated animation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13653–13662.

Sun, Y.; Zhao, D.; Yin, Z.; Huang, Y.; Gui, T.; Zhang, W.; and Ge, W. 2023. Correspondence transformers with asymmetric feature learning and matching flow super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 17787–17796.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, 402–419. Springer.

Thies, J.; Zollhofer, M.; Stamminger, M.; Theobalt, C.; and Nießner, M. 2016. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2387–2395.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, H.; Liu, F.; Zhou, Q.; Yi, R.; Tan, X.; and Ma, L. 2024. Continuous piecewise-affine based motion model for image animation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 5427–5435.

Wiles, O.; Koepke, A.; and Zisserman, A. 2018. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European conference on computer vision (ECCV)*, 670–686.

Zablotskaia, P.; Siarohin, A.; Zhao, B.; and Sigal, L. 2019. Dwnet: Dense warp-based network for pose-guided human video generation. *arXiv preprint arXiv:1910.09139*.

Zhao, J.; and Zhang, H. 2022. Thin-plate spline motion model for image animation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3657–3666.

Zhu, L.; Chen, Y.; Liu, X.; Li, T. H.; and Li, G. 2025. Learning Semantic Facial Descriptors for Accurate Face Animation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Zhu, Z.; Huang, T.; Shi, B.; Yu, M.; Wang, B.; and Bai, X. 2019. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2347–2356.