ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2024
M. Skouras and H. Wang
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 8

Pose-to-Motion: Cross-Domain Motion Retargeting with Pose Prior

Qingqing Zhao[1], Peizhuo Li[2], Wang Yifan[1], Sorkine-Hornung Olga[2], Gordon Wetzstein[1]

[1]Stanford University, USA
[2]ETH Zurich, Switzerland

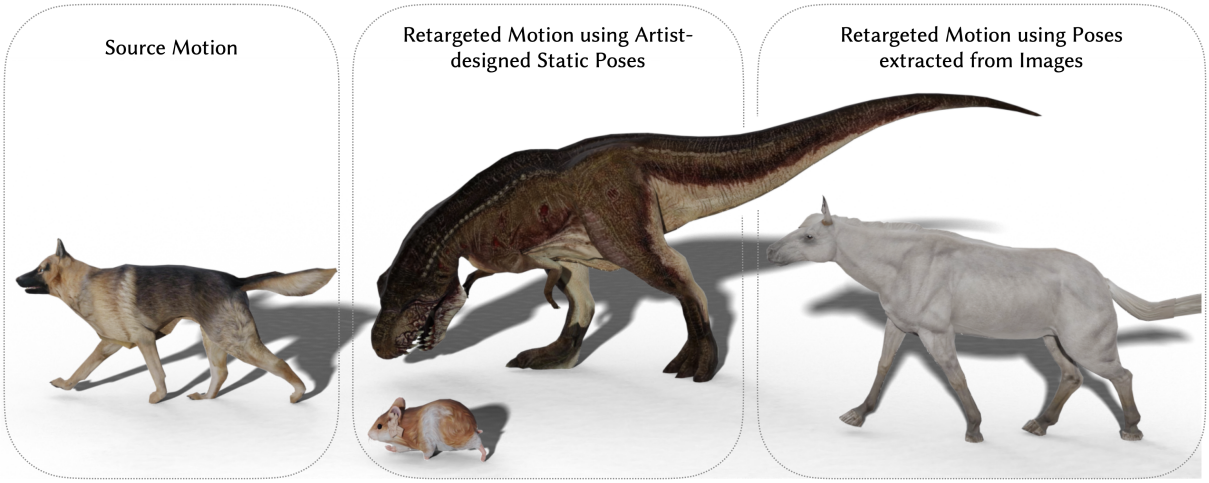

Figure 1: *Our generative motion retargeting framework enables the motion dynamics of one creature to be transferred to another in a plausible manner. For this purpose, motion capture data of the source creature is transferred to the target while taking a few static body poses of the target into account, among other constraints. This allows us to transfer the motion dynamics of tame and cooperative animals, such as dogs, to more exotic creatures, such as horses, rodents, or carnivorous dinosaurs, for which motion capture data may be difficult to obtain but individual body poses are readily available.*

Abstract

Creating plausible motions for a diverse range of characters is a long-standing goal in computer graphics. Current learning-based motion synthesis methods rely on large-scale motion datasets, which are often difficult if not impossible to acquire. On the other hand, pose data is more accessible, since static posed characters are easier to create and can even be extracted from images using recent advancements in computer vision. In this paper, we tap into this alternative data source and introduce a neural motion synthesis approach through retargeting, which generates plausible motion of various characters that only have pose data by transferring motion from one single existing motion capture dataset of another drastically different characters. Our experiments show that our method effectively combines the motion features of the source character with the pose features of the target character, and performs robustly with small or noisy pose data sets, ranging from a few artist-created poses to noisy poses estimated directly from images. Additionally, a conducted user study indicated that a majority of participants found our retargeted motion to be more enjoyable to watch, more lifelike in appearance, and exhibiting fewer artifacts. Our code and dataset can be accessed here.

CCS Concepts

• ***Computing methodologies*** → ***Motion processing;***

1. Introduction

The ability to generate plausible motion across a diverse array of characters is a crucial aspect of creating immersive and engaging experiences in this digital era, and is vital to a wide range of applications including augmented reality, cinematography, and education.

Recently, motion retargeting from unpaired motion data has emerged as a promising approach to address these needs [GYQ*18, VYCL18, ALL*20, VCH*21, ZWK*23]. Powered by cycle-consistent generative adversarial networks [ZPIE17, GPAM*14, LBK17], these approaches have moved away from traditional approaches, which require skeleton-level correspondence [Gle98, CK00, MBBT00, PW99, TK05, VCH*21, ZWK*23] or pose-level correspondence [SP04, BVGP09, SOL13, WPP14, CYÇ15, AMYB17], successfully demonstrating the capability to transfer motion between different skeletal structures using unpaired motion data [VYCL18, ALL*20, GYQ*18, DAS*20], as long as they are topologically similar. However, these approaches largely depend on having symmetric data, i.e., a similar amount of high-quality motion data from both the source and target domains is required, which can be challenging and sometimes impractical to obtain, especially for unique non-humanoid characters.

To this end, we propose Pose-to-Motion, which leverages static *pose* data from the target domain to tackle the fundamental challenge posed by the scarcity of high-quality motion data. Compared to motion data, typically acquired through extensive motion capture (MoCap) sessions, pose data is more accessible, and can be obtained by, e.g., contemporary computer vision techniques [LTV*22, WLJ*23, SMH24, ZKBWB19], analyzing the fossils of extinct creatures, or by artist creation [Tru22, BB22].

Our method leverages an *asymmetric* CycleGAN, transforming source domain motion data to target domain pose data and vice versa, effectively allowing us to "project" motion onto our target characters using solely their pose data (Fig. 3). This cycle is further refined by synthesizing plausible root transformations using soft constraints, overcoming the root ambiguity problem arising from the lack of motion data in the target domain. While neither cycle consistency nor adapted soft constraints are novel concepts, applying them to asymmetric data within the realm of motion retargeting offers a new and effective solution to the unique challenges we face in our task. This approach specifically addresses the large domain gap between motion clips in the source domain and static poses in the target domain. As we demonstrate in section 4, our method is able to generate plausible motion for a wide range of subjects by combining the motion prior from another domain, where MoCap data has been captured a-priori, and the pose prior of the subject observed from static poses, even when the pose data is small or noisy.

In summary, this paper makes the following primary contributions:

1. We propose a novel motion-retargeting approach for motion synthesis, which leverages pose data from the target domain to tackle the fundamental challenge posed by the scarcity of high-quality motion data.
2. We overcome the root ambiguity issue unique in the pose-to-motion setting by adapting a combination of existing networks and regularizations.
3. We present a detailed analysis and comparison against existing retargeting approaches, showing state-of-the-art results in terms of motion quality and versatility across a wide range of characters.

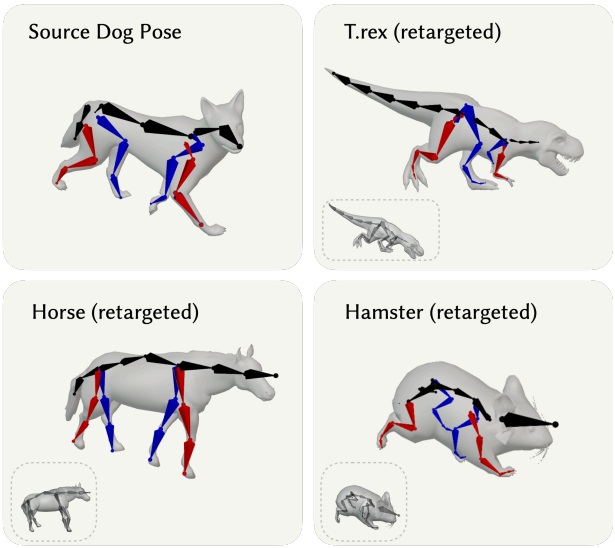

Figure 2: ***Motion retargeting to versatile characters.*** *Given a small set of artist-created posed animals (e.g. t.rex, hamster) or noisy poses derived from 2D images (horse), our method successfully transfers the dog motion to these animals despite significant differences in their bone structures. We include images of the closest instance in the training data at the lower left corner, highlighting the preservation of key attributes during the motion retargeting process. Notably, the elongated tail of the T.rex, the arched spine of the hamster, and the forward-bending knee of horse are all preserved even though the source dog pose does not contain these characteristics. Please refer to the supplementary video for additional qualitative evaluation of the motion clips.*

2. Related work

Motion Retargeting. As one of the pioneering works, [Gle98] proposed to solve the kinematics constraint of two topologically identical skeletons with a space-time optimization problem. [LS99, CK00] further employ per-frame inverse kinematics (IK) for retargeting, followed by a smooth process while preserving high-frequency details. [MBBT00] explore the possibility of using an intermediate skeleton to retarget motion between skeletons with different numbers of bones. In addition to simple kinematics constraints, [PW99] introduce dynamics constraints to the spacetime optimization and achieves better realism of the source motion sequence. [TK05] take a different approach by modeling the retargeting problem as a state estimation on a per-frame Kalman filter and further improve the realism of generated motion.

However, those methods are limited to retargeting between motions with skeletons containing limited differences in bone proportion, thus are unable to handle retargeting between different creatures. Since the desired motion gaits and the correspondence between motions cannot be inferred solely from bone proportions in these cases, especially for drastically different creatures like humanoid and quadrupeds, [BVGP09] propose to exploit a few sparse mesh pairs to transfer poses between different creatures using feature extraction and extrapolation. [YAH10] also exploits paired poses and is able to retarget motion to a different creature. [SOL13] demonstrate the ability to control a target creature with human motion, given paired motion examples. [WPP14] model the locomo-

tion of different creatures with the same skeletal structure with a physically-based optimization method, and is able to capture the gait with a few examples. However, it requires delicate handcrafted design and is limited to locomotion. [IAF09] utilizes Gaussian processing and probabilistic inference to map motion from one control character to a different target character, but it requires artists' edits as training data.

[CYÇ15] use paired pose and mesh to retarget motion from humans to different meshes. Although they are able to perform retargeting between different creatures, at least several paired poses or motions are required as guidance. Besides, when applying pose transfer method to motion in a frame-by-frame manner, the high-frequency details of motion and the temporal coherence are not well preserved, and the missing global translation information leads to severe foot skating artifacts. The same issue also applies to the generative model for poses [PCG*19] learned from a large dataset, making motion generation with only pose prior extremely challenging. An interesting exception is the work of [AMYB17], which requires only a manually assigned part correspondence to achieve motion style transferring between different creatures. [RWY*23, YSI*23, PALVdP18, PCZ*20] use Deep RL to generate physics-based retargeting methods leveraging a physics simulator.

Neural Motion Processing. With the progress of deep learning, deep neural networks are applied to motion process and synthesis tasks [AWL*20, YYB*23, LAZ*22, SGXT20, LYRK21, TCL23, TRG*22, YSI*23, ZLAH23, SMK22, HYNP20, KAS*20, SZKS19, HSK16], including recurrent neural networks (RNNs) [FLFM15, AAC22], convolutional neural networks (CNNs) [HSKJ15, HSK16]. As for motion retargeting, [JKY*18] apply a U-Net structure to paired motion data to solve the problem. Villegas et al. [VYCL18] use cycle-consistency adversarial training [ZPIE17] on a RNN for retargeting, and drops the requirement for paired motion datasets. Dong et al. [DAS*20] use cycle-consistency training to transform adult motion capture data to the style of child motion, trained on a small number of sequences of unpaired motions from both domains. PMnet [LCC19] opts for CNNs and achieve better performance. With the proposed skeleton-aware networks, [ALL*20] can retarget among skeletons with different yet homeomorphic topologies. [LWJ*22] bypass the usage of adversarial training and use an iterative solution with a motion autoencoder. At the same time, directly transferring poses without any correspondence information pose-wise and geometry-wise is made possible with neural networks [GYQ*18, LYS*22]. More recent works keep exploring the possibility of better retargeting results by incorporating skinning constraints introduced by the geometry [VCH*21, ZWK*23]. Note those methods require motion dataset on both source and target skeleton for training, but the difficulty of acquiring high-quality and comprehensive motion dataset greatly limits their usage. We demonstrate that we can achieve similar performance as skeleton-networks [ALL*20] on the Mixamo dataset [Ado20] in Section 4.1, while our model is trained only with a pose dataset for the target character.

3. Method

3.1. Data representation.

We inherit the representation for pose and motion from prior work [ALL*20], which we briefly recap below. Given a character's skeleton with J joints, its pose is represented by a vector $P \in \mathbb{R}^{6J}$, which defines the relative joint rotations in the kinematic tree, with each rotation represented by a 6-dimensional vector [ZBL*19].

A character's motion consists of a sequence of poses $[P_n]_{n=1}^N$ and root transformations $[R_n]_{n=1}^N$, where R_n is composed of root orientation $\theta_r \in \mathbb{R}^6$ and velocity $v_r \in \mathbb{R}^3$. The root transformation is handled as a special structure connected to the root node. As such, the overall representation of motion can be denoted by $M \in \mathbb{R}^{T\times(6(J+1)+3)}$, where T is the number of frames in the sequence. Note that the root displacement, x_r, can be computed from root velocity using the forward Euler method $x_r(t+1) = x_r(t) + v_r(t)$.

In the following, we denote the motion and pose from a domain $\mathcal{Q} \in \{\mathcal{S}, \mathcal{T}\}$ as $M^{\mathcal{Q}}$ and $P^{\mathcal{Q}}$ respectively.

3.2. Asymmetric Cycle-consistency Learning

Our model builds on the foundation of the "classic" design of a symmetric motion retargeting network [ASL*18, ALL*20, GYQ*18], which adopts a CycleGAN framework [ZPIE17] to maximize the likelihood of the output motion in the distribution of the target motion data. In the absence of target motion data, we aim to transfer the source motion such that *each frame* in the output adheres to the pose priors observed in the target domain's pose data. More concretely, our objective is to learn a mapping $\mathcal{G} : \mathcal{S} \to [\mathcal{T}_n]_{n=1}^N$

$$\arg\max_{\mathcal{G}} p_{\mathcal{T}}\left(\tilde{P}_n^{\mathcal{T}}\right) \text{ s.t. } \left[\tilde{P}_n^{\mathcal{T}}\right]_{n=1}^N = \mathcal{G}\left(M^{\mathcal{S}}\right). \tag{1}$$

Since the data in the source and target are unpaired, i.e. pose-level correspondence is absent, we adopt a CycleGAN [ZPIE17] framework, following the approach of prior works addressing unpaired motion retargeting [ASL*18, ALL*20, GYQ*18].

This forms an *asymmetric* cycle as shown in Fig. 3. The first half maps a given source motion $M^{\mathcal{S}}$ to a set of poses and root transformations (discussed in section 3.2) in the target domain:

$$\left(\left[\tilde{P}_n^{\mathcal{T}}\right]_{n=1}^N, \left[\tilde{R}_n^{\mathcal{T}}\right]_{n=1}^N\right) = \mathcal{G}\left(M^{\mathcal{S}}\right), \tag{2}$$

whereas the other half maps the outputs back to motion in the source domain:

$$\tilde{M}^{\mathcal{S}} = \mathcal{F}\left(\left[\tilde{P}_n^{\mathcal{T}}\right]_{n=1}^N, \left[\tilde{R}_n^{\mathcal{T}}\right]_{n=1}^N\right). \tag{3}$$

Correspondingly, a pose discriminator $\mathcal{D}_P$ and a motion discriminator $\mathcal{D}_M$ distinguish the outputs of the two half cycles against real pose and motion samples, respectively.

The asymmetric CycleGAN can be supervised in the same way as in prior work [ALL*20], which includes a Wasserstein adversar-

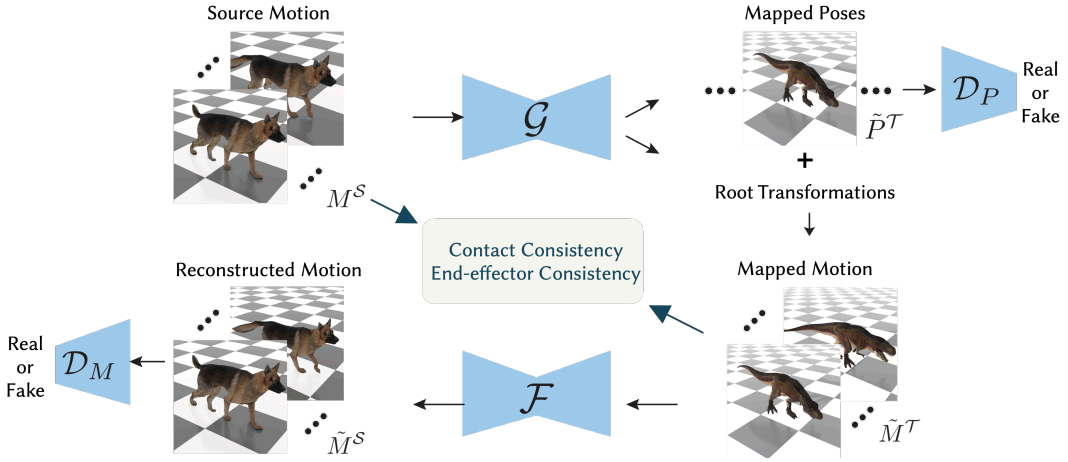

Figure 3: ***Method overview.*** *Our method builds on an asymmetric CycleGAN. The first half of the cycle maps a motion sequence from the source domain $M^{\mathcal{S}}$ to a sequence of poses $\left[\tilde{P}_n^{\mathcal{T}}\right]_{n=1}^N$ and root transformations $[R_n]_{n=1}^N$ in the target domain; the individual poses are compared against the pose dataset of the target domain using a pose discriminator $\mathcal{D}_P$. The other half of the cycle maps the sequence of poses and root transformations ($\tilde{M}^{\mathcal{T}}$) back to a motion sequence in the source domain $\tilde{M}^{\mathcal{S}}$, which is supervised with a reconstruction loss and an adversarial loss using a motion discriminator. The contact and end-effector consistency implicitly regulates the root prediction, leading to more realistic motion.*

ial loss with gradient penalty [GAA*17] and a reconstruction loss:

$$\begin{aligned}\mathcal{L}_{\text{cycle}} =& \mathcal{L}_{\text{GAN}}(\mathcal{G},\mathcal{D}_P)+\lambda_{\text{GP}}\mathcal{L}_{\text{GP}}(\mathcal{D}_P)+ && \text{first cycle}\\ & \mathcal{L}_{\text{GAN}}(\mathcal{F}\circ\mathcal{G},\mathcal{D}_M)+\lambda_{\text{GP}}\mathcal{L}_{\text{GP}}(\mathcal{D}_M)+ && \text{second cycle}\\ & \lambda_{\text{recon}}\mathcal{L}_{\text{recon}}(\mathcal{G},\mathcal{F}), && (4)\end{aligned}$$

where $\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{\text{GP}}$ are the standard Wasserstein adversarial loss and gradient penalty , and $\mathcal{L}_{\text{recon}}$ is the reconstruction loss defined as the L2 distance between the input source motion and the remapped source motion, $\left\|\tilde{M}^{\mathcal{S}}-M^{\mathcal{S}}\right\|_2^2$.

The asymmetric CycleGAN framework is agnostic specific architecture of $\mathcal{G}$ and $\mathcal{F}$. We choose the design from prior work [ALL*20, PCG*19, LAZ*22], which is constructed of multiple layers of skeleton-aware operators [ALL*20] to account for different joint hierarchies. The motion discriminator, $\mathcal{D}_M$, also uses skeleton-aware operators and adopts a patch-wise classification to reduce overfitting [LAZ*22]. The pose-level discriminator, $\mathcal{D}_P$, consists of $J+1$ discriminators, one for each joint rotation and another one for all rotations [DRC*22].

3.3. Root Transformation

The asymmetric CycleGAN framework generates reasonable motions by combining the rough trajectory from the source motion and the pose prior from the target. However, these generations often suffer from artifacts such as foot sliding and jittering. The reason is that the root transformations in the target domain has been neglected in the objective defined in eq. (1), leading to unresolved ambiguity when mapping the root transformation. One trivial (but wrong) solution is to have an identity mapping from the source root motion to the target root motion with scaling, yet this solution leads to various artifacts as shown section 4.2, due to the negligence of changes in the bone size, skeleton structure, e.t.c.

To address this issue, we propose to a.) predict root transformations $\left[\tilde{R}_n^{\mathcal{T}}\right]_{n=1}^N$ for the target domain, and b.) employ a set of soft constraints, described below, to effectively regulate these predicted roots and alleviate the root ambiguity issue. Although these soft constraints do not directly supervise the root transformations, they promote consistency between the generated and source motion from various complementary perspectives. This guidance helps the root predictions converge towards more realistic and plausible solutions.

Contact Consistency. This constraint ensures that the contact-to-ground patterns in the retargeted motion match those of the input source motion. In a more intuitive sense, we determine contacts by examining the velocity of the foot joints. The hypothesis is that when the velocity is close to zero, it indicates a contact point. Although the relative distance from foot to floor could serve as a useful metric, the changing floor position is not available in our dataset. Therefore, we rely solely on velocity thresholding for contact identification. Therefore, we enforce the retargeted motion to maintain contact whenever the source motion does. We manually specify the correspondence of the "feet" end-effectors between the characters to ensure contact consistency. Specifically, using the shorthand M_n for the motion at frame n, we can write the velocity of a specific joint as $v_j(M_n) = FK_j(M_n) - FK_j(M_{n-1})$, where FK denotes the forward kinematics function that converts joint angles into joint positions $x \in \mathbb{R}^{3J}$. Then, one can express this constraint using the loss

$$\begin{aligned}\mathcal{L}_{\text{con}} = \frac{1}{N|\Phi|}\sum_{j\in\Phi}\sum_{n=1}^{N}\left\|v_j(\tilde{M}_n^{\mathcal{T}})\right\|_2^2 s_j(M_n^{\mathcal{S}}) \text{ with}\\ s_j(M_n^{\mathcal{S}}) = \mathbb{1}\left[\left\|v\left(M_n^{\mathcal{S}}\right)\right\|_2 < \epsilon\right],\end{aligned} \quad (5)$$

where Φ represents the set of foot joints and $s_j(M_n^{\mathcal{S}})$ is the reference contact label from source motion, and ϵ is the velocity threshold to define contact.

method	J. Angle Err.↓	Root Rel J. Pos. Err.↓	Global J. Pos. Err. ↓	Mean J. Pos. Jitter ($\times 10^2$) ↓	Contact Consis ↑
frame-level SA-Net	7.12	0.53	3.88	0.81	86.6%
motion-level SA-Net	11.06	0.67	0.83	0.52	84.1%
Pose-to-Motion (ours)	**6.71**	**0.52**	**0.81**	**0.49**	**91.4%**

Table 1: ***Quantitative evaluation on Mixamo.*** *We compare our approach with the frame-level and motion-level Skeleton-Aware Network [ALL*20], which perform motion retargeting on frame-by-frame and sequence-by-sequence basis, respectively. Our method leverages pose information and further uses the proposed root estimation techniques to achieve more accurate global joint position and higher-quality motion with less jittering and more consistent ground contact.*

End-Effectors Consistency. End-effectors are the terminal points of a skeleton structure that are commonly used to interact with the real world. End-effector consistency takes advantage of the fact that homeomorphic skeletons share a common set of end-effectors, and encourages that their normalized velocities from the source and retargeted motions are consistent. Enforcing this constraint helps prevent common retargeting artifacts like foot sliding [ALL*20]. Formally, this constraint is formulated using the following loss

$$\mathcal{L}_{\text{ee}} = \mathbb{E}_{M^{\mathcal{S}} \sim P^{\mathcal{S}}} \frac{1}{|\Theta|} \sum_{j \in \Theta} \left\| \frac{v_j\left(\tilde{M}^{\mathcal{T}}\right)}{h_{Tj}} - \frac{v_j(M^{\mathcal{S}})}{h_{Sj}} \right\|_2^2. \tag{6}$$

Here, Θ denotes the end-effector joints, and $h_j^{\mathcal{S}}$ and $h_j^{\mathcal{T}}$ correspond to the lengths of the kinematic chains from the root to the end-effector j in the source and target domain, respectively.

Furthermore, under the assumption that the rest poses P_0 in the source and target domain are similar, we require the end-effectors of the source and retargeted motion at every frame n to exhibit comparable offsets to their rest poses. This objective is based on the premise that if one character's end-effector has moved in a specific direction (relative to its rest pose), the retargeted character should have its corresponding end-effector positioned similarly. We manually specify the correspondence between the end-effectors of the characters based on their semantic correspondences. We compute the offsets and this relative end-effector loss using:

$$o(M_n) = FK(M_n) - P_0 \tag{7}$$

$$\mathcal{L}_{\text{ee,r}} = \mathbb{E}_{M^{\mathcal{S}} \sim P^{\mathcal{S}}} \frac{1}{|\Theta|} \sum_{j \in \Theta} \left\| \frac{o_j\left(\tilde{M}_n^{\mathcal{T}}\right)}{h_j^{\mathcal{T}}} - \frac{o_j(M^{\mathcal{S}})}{h_j^{\mathcal{S}}} \right\|_2^2. \tag{8}$$

In summary, our overall learning objective is

$$\mathcal{L} = \lambda_{\text{cycle}}\mathcal{L}_{\text{cycle}} + \lambda_{\text{con}}\mathcal{L}_{\text{con}} + \lambda_{\text{ee}}\mathcal{L}_{\text{ee}} + \lambda_{\text{ee,r}}\mathcal{L}_{\text{ee,r}}. \tag{9}$$

4. Experiments and Evaluations

We evaluate our approach using three distinct datasets: first the Mixamo dataset, which is a large-scale paired character–motion dataset. Since this dataset provides paired data, it allows us to evaluate the retargeting motions against ground truth for quantitative assessment. Second, we employ an animal dataset, using a large-scale dog MoCap data from [ZSKS18] as the source domain, along with a smaller animated animal dataset from [Tru22], containing approximately 1000 frames, as the target domain. This particular setup serves as a stress test to evaluate how our model handles scenarios where the target domains have limited data and highly distinct topologies. Lastly, we extract 3D poses from a horse image dataset [WLJ*23] as the target domain and use the dog MoCap dataset as the source domain. This experiment demonstrates our method's capability to learn from accessible but noisy data (in this case, extracted 3D poses obtained from images). To evaluate our results qualitatively, please refer to the supplementary video.

4.1. Mixamo dataset

We conduct a quantitative evaluation on the Mixamo dataset [Ado20], which consists of characters with unique skeletal structures, each performing the same set of 2,400 motion clips. To emulate a target domain with only pose data, we extract individual pose from the target motion clips, removing the temporal information including global translation. 20% of the data is reserved for testing in both the source and target domains. For our quantitative evaluation, we select two distinct characters from the Mixamo dataset (Aj and Mousey), each possessing five primary limbs (two hands, two feet, and a head).

4.1.1. Baselines

We compare our method with a state-of-the-art symmetric motion retargeting Skeleton-Aware Network [ALL*20] (SA-Net). Our model differs from the baseline in the asymmetric design and additional loss terms $\mathcal{L}_{ee,r}$ and $\mathcal{L}_{\text{con}}$ described in 3.3, which we incorporated to improve root transformation and motion realism in absence of motion data.

We investigate two variations of the SA-Net. The first, denoted as motion-level SA-Net, is the original SA-Net, which necessitates motion data from both domains, trained with all the losses described in [ALL*20]. The second variation is a frame-level SA-Net. In this setup, both domains consist of poses, $P^{\mathcal{S}}$ and $P^{\mathcal{T}}$, and two generators and discriminators are trained for translating poses between these domains. During retargeting, frame-by-frame decoding is employed. For root transformation, we approximate the translation from the source root's velocity after scaling it by the skeleton size, and the rotation is directly copied from the source to the target. This variation serves to quantify the benefit of motion prior. We train the baseline models for the source and target characters from scratch following the author recommended protocol.

4.1.2. Motion Reconstruction.

Table 1, presents the quantitative evaluation of the different methods described above. We use commonly employed metrics for assessing motion reconstruction quality [JYG*22].

(1) **Mean Joint Angle Error**: Calculates the joint angle difference (in degrees, represented as axis angles) between the retargeted and ground truth joint angles.
(2) **Mean Root Relative Joint Position Error**: Calculates the MSE between the local joint positions of the retargeted and

ground truth motions after removing the global root translation and rotation. The error is normalized by the skeleton's height and multiplied by 1000.

(3) **Mean Global Joint Position Error**: Measures the MSE of the global joint positions between the retargeted and ground truth motions. The error is normalized by the skeleton's height and multiplied by 1000.

(4) **Mean Joint Position Jitter**: Estimates joint position jitter by computing the third derivative (jerk) of the global joint position. A lower value indicates smoother motion, which is generally more desirable.

(5) **Contact Consistency Score**: Calculates the ratio of consistent contacts made between the source and target domains. A contact is considered consistent if the contact state (contact or no contact) determined by eq. (5) is the same in both the source and retargeted motion. A higher Contact Consistency Score indicates better contact consistency.

Our method compares favorably against both baselines across all metrics. Frame-level SA-Net performs on par in terms of joint angles and relative joint position (see the first two columns), indicating that the relative joint positions can be sufficiently estimated from pose information. Our method leverages this information and further uses the proposed root estimation techniques to achieve more accurate global joint position and higher-quality motion with less jittering and consistent ground contact (see the last three columns). Motion-level SA-Net is similar to ours in terms of the Mean Global Joint Position Error, but is worse in the terms of pose-level retargeting accuracy (see Mean Joint Angle Error and Mean Root Relative Joint Position Error in column 2 and 3) and contact consistency (see the last column). This can be attributed to the relative end-effector loss, $\mathcal{L}_{ee,r}$ and the contact consistency loss $\mathcal{L}_{\text{con}}$, which specifically focuses on the relative joint positioning and contact consistency.

4.2. Animal dataset

We further assess the robustness and versatility of our method by applying it to the challenging task of retargeting animal motion, specifically from dogs to two drastically different animals, T.rexes and hamsters. As the source domain, we use a large-scale dog MoCap dataset from [ZSKS18] consisting of 30 minutes of unstructured dog motion encapsulating various locomotion modes. In contrast, the target domain was comprised of a small number of short motion clips of T.rexes and hamsters from the Turebone dataset [Tru22], from which we extract individual poses as our training data in the target domain. This evaluation setup presents a high level of complexity due to the large domain gap between the source and target domains. We manually specify the end-effectors' correspondences based on semantic meaning. For the T.rex, we mapped its two hind legs to the dog's two hind legs for contact consistency loss and mapped all five limbs for the end-effector consistency loss.

4.2.1. Motion Quality

We use the same baselines as in the Mixamo dataset. Both are retrained on this dataset using the same hyperparameters as in the original implementation when possible. Since there is no ground truth motion data, we evaluate motion quality metrics: Mean Joint Position Jitter and Contact Consistency Score.

method	Mean J. Pos. Jitter ($\times 10^2$) ↓			Contact Consis ↑		
	T.rex	Hamster	Horse	T.rex	Hamster	Horse
frame-level SA-Net	5.37	8.87	2.19	86.2%	83.5%	77.2%
motion-level SA-Net	**0.68**	**0.46**	-	89.8%	**94.6%**	-
Pose-to-Motion (ours)	1.08	1.33	**0.88**	**92.6%**	91.1%	**81.4%**

Table 2: ***Quantitative evaluation for zoo and horse datasets.*** *Our method largely outperforms frame-level SA-Net [ALL*20] in terms of both joint position jitter and contact consistency. While the motion-level SA-Net [ALL*20] is capable of generating smooth motion, its qualitative results degrade significantly, as shown in Fig. 4.*

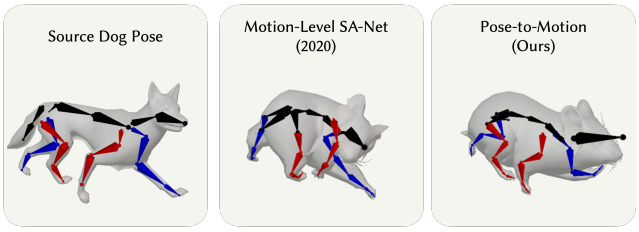

Figure 4: ***Comparison with motion-level SA-Net on the animal dataset.*** *Pose-to-Motion achieves more plausible poses compared to the motion-level SA-Net, despite the absence of target motion information. We believe the reason is that motion-to-motion mapping is a much harder task requiring significantly more amount of data and diversity for convergence. For more qualitative evaluations, please watch our supplementary video.*

As shown in Table 2 and Fig. 2 (T.rex and Hamster), despite the large domain gap and the very limited amount of training poses (600 frames for hamster and 7000 frames for T.rex), our method is able to synthesize high-quality motion. In contrast, the frame-level SA-Net exhibits high jittering and poorer contact consistency. While the motion-level SA-Net appears smoother motion, it has noticeably less realistic pose as shown in Fig. 4. Please check additional qualitative evaluations in our supplementary video.

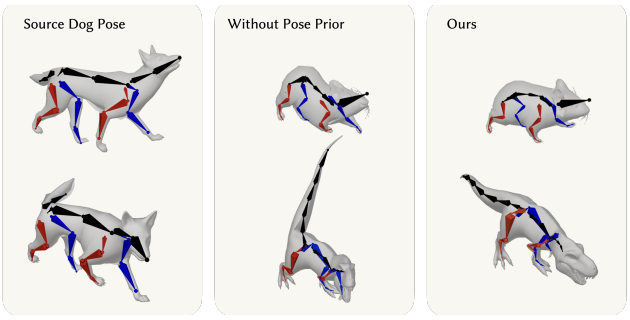

Figure 5: ***Preservation of pose characteristics through adversarial training.*** *We compare the retargeting with and without adversarial training. The latter relies solely on the end-effector and reconstruction loss to establish pose correspondence, thus unable to leverage any pose prior from the target domain, leading to unrealistic and out-of-distribution retargeting results, such as the hamster's head and hip, as well as the T.rex's tail being bent upwards in an unnatural way.*

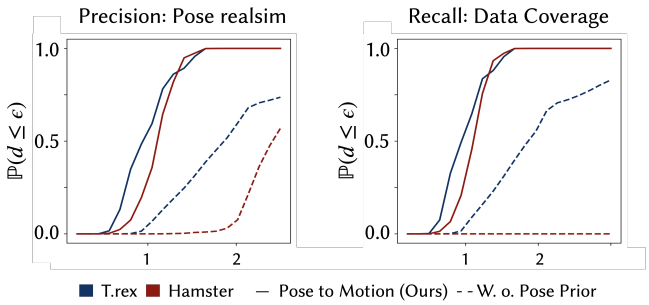

Figure 6: ***Precision and Recall:*** *Empirical estimation of data coverage and realism of retargeted poses using precision (left) and recall (right). For both plots, higher values indicate better performance. Across all testing scenarios, our approach (Pose-to-Motion) consistently achieved higher values, indicating better pose realism and coverage.*

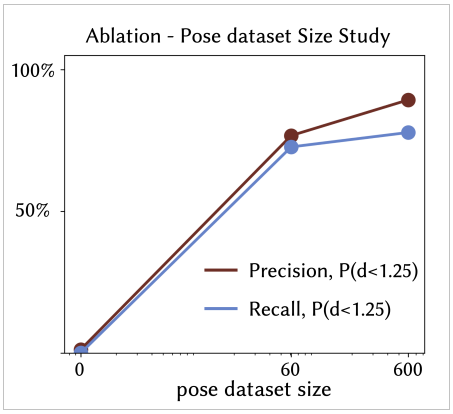

Figure 7: ***Dataset size ablation:*** *In this table, we examine the impact of dataset size on the final performance of our approach. We present precision and recall for training sessions with different pose dataset sizes, 0% (0 poses), 10% (60 poses), and 100% (600 poses). We observe that even with 60 poses only, our approach is able to learn meaningful pose features, and yields results with relatively high precision and recall.*

4.2.2. Pose Realism and Data Coverage.

One of our primary goals is to preserve the diversity and peculiarities of the poses in the target domain. The retargeted pose should ideally span the entire space of realistic target poses without including extraneous poses. To evaluate how well our approach achieves this, we employ *Precision* and *Recall* to assess pose realism and data coverage respectively [DRC*22]. Given K retargeted poses, precision evaluates the ratio of "accurate" predictions. A retargeted pose is considered accurate if the Mean Root Relative Joint Position Error with at least one sample in the target pose dataset is smaller than a threshold ϵ. On the other hand, recall measures the ratio of "covered" training poses over the size of the training dataset. A training pose is considered covered if the Mean Root Relative Joint Position Error with at least one sample among the retargeted poses is smaller than a threshold ϵ.

For both, we use $K = 8000$ and plot the precision/recall as a cumulative distribution $\mathbb{P}(d \leq \epsilon)$ in Fig. 6. Omitting pose prior by removing the adversarial losses ($\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{\text{GP}}$) leads to a significant deterioration in precision and recall. This highlights the effectiveness of GAN training in generating realistic and diverse retargeted poses that cover the entire distribution of target poses. Fig. 5 visually illustrates this effect. In the absence of adversarial losses, the retargeted poses retain traits from the source domain but appear unnatural in the target domain.

4.2.3. Data efficiency

One advantage of our method is data efficiency: besides not requiring hard-to-acquire motion data, our model is able to effectively capture the character-specific pose features from very few poses. We demonstrate this advantage with the Hamster character, comparing the outcomes of training with 0%, 1%, 10%, 40% and 100% random samples of the total 600 poses. Note that the 0% data scenario corresponds to the case where no pose prior is available, i.e. only the end-effector loss described in Section 3.3 is utilized during the training process. Since the motion prior remains consistent, we focus on how well the character-specific pose features are captured under the varying pose data size using precision and recall metrics, defined in Section 4.2.2, which measures pose realism and coverage respectively. We observe that even with 60 poses, our method was able to learn meaningful pose features, yielding results that exhibited both high pose realism and coverage, as shown in Fig. 7.

4.3. Horse dataset from Images

Our approach is evaluated using pose datasets obtained from images. To extract 3D poses from a diverse collection of horse images, we utilize the unsupervised method called MagicPony [WLJ*23]. The dataset used in our evaluation consisted of approximately 10,000 images, capturing various horse poses from different viewing angles. These extracted 3D poses from MagicPony served as our target domain pose data. We further augmented our dataset by flipping the left and right limbs of the horse. For the source domain, we use the same dog MoCap dataset described in Section 4.2. The purpose of this experiment was to demonstrate the robustness and versatility of our approach when applied to readily available but potentially noisy image-derived datasets. Fig. 8 visually illustrates the pipeline. As evident in figs. 2 and 8, the retargeted horse poses match the dog poses in the source domain, while at the same time preserving the important features unique to horses, e.g., forward-bending knee, less upright head, and smaller strides.

5. User Study

To evaluate the synthesized motion holistically, we also conduct a user study in which we focus on the general appeal, the alignment with the target pose, and the realism of the motion itself. We rendered 9 motion clips for 3 characters - Hamster and T.rex (see section 4.2), and Horse section 5, and compare it with commercial motion processing software - MotionBuilder [Aut]. MotionBuilder employs inverse kinematics (IK) for motion retargeting without pose priors. We use MotionBuilder following the standard procedure recommended by professionals, with default settings. Note that MotionBuilder requires manually setting correspondences between skeletons using a template skeleton, a step that our approach eliminates. The clips included in our study are chosen adhering to

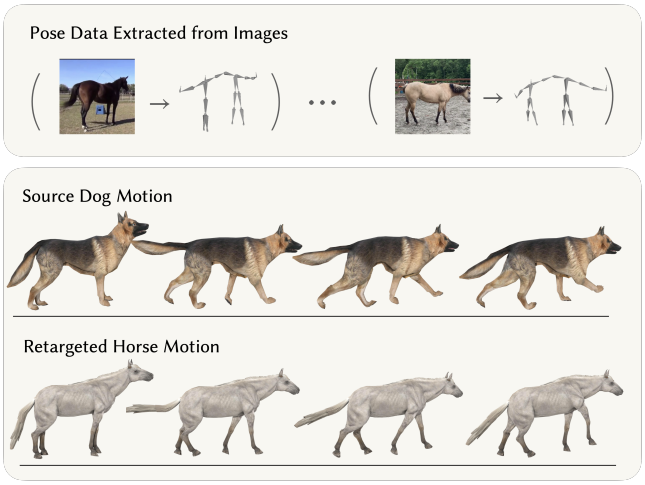

Figure 8: ***Retargeting using noisy pose data estimated from 2D images.*** *We extract pose priors from a noisy pose dataset generated from state-of-the-art 3D reconstructions method developed in the vision community, and demonstrate that it is possible to synthesize coherent and plausible horse motions by retargeting a dog motion sequence to the horse domain, essentially enabling conditional 2D-to-4D synthesis.*

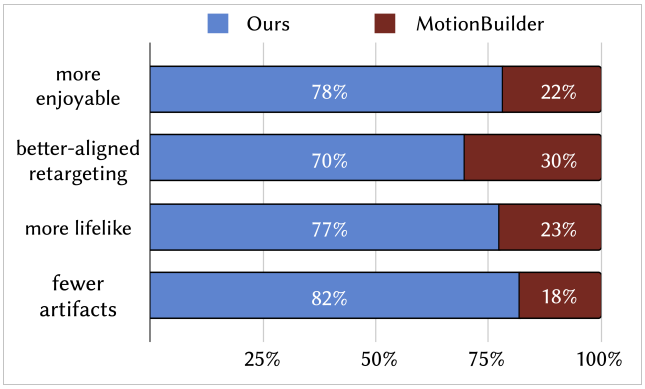

Figure 9: ***User Study:*** *We conducted a user study to evaluate the quality of retargeting achieved through our method, comparing it with the commercial software MotionBuilder. In this study, nine motion clips were generated for three distinct characters — Hamster, T.rex, and Horse — with 26 users participating in total. The specific questions posed to the participants during the study are detailed in section 5. Generally, the feedback indicated that participants preferred the retargeting results of our method, finding them to be more enjoyable to watch (78%), noticing that they produced fewer artifacts (82%), exhibited greater lifelikeness (77%), and better aligned with the source motion (70%).*

the principle of representative sampling, specifically we sampled 9 motion clips, three per character, from different motion categories (walking, running, and turning). Within each category, the target motion is randomly sampled to minimize bias.

For each motion clip, we ask human subjects the following questions:

- **Q1:** Which one is more pleasing to watch?
- **Q2:** Which adapted animation on the right captures the essence of the original animation displayed on the left side of the video more effectively?
- **Q3:** Which adapted animation on the right exhibits better smoothness, lifelikeness, and overall visual appeal?
- **Q4:** Which adapted animation on the right shows fewer noticeable issues, such as overlapping body parts or unnatural movement of feet?

The anonymous participants were volunteers recruited via email from both within and outside our university. Among 26 participants, 78% found our results more pleasing to watch, 82% reported observing fewer artifacts, 77% noted an increase in lifelikeness, and 70% recognized a closer alignment with the source motion, as depicted in fig. 9. We employed a one-tailed hypothesis test with a null hypothesis - user satisfaction is 50% or lower. The resulting p-value of 0.00467, below the standard 0.05 significance level, suggests a higher user satisfaction rate with our results [GSR*16]. Our user study shows that our approach, which leverages pose priors, leads to more lifelike, smoother and artifact-free animations, with an enhanced overall user experience compared to the traditional IK-based retargeting (MotionBuilder). The default results from MotionBuilder disregard pose priors, and integrating these priors requires substantial labor and extensive tuning by a professional artist for each motion clip. In contrast, our approach learns automatically from the pose data. We have included the details of the user study, the interface design, and a sample Google Form here for reference.

6. Implementation Details

Our retargeting network $\mathcal{G}$ and $\mathcal{F}$ follows SA-Net [ALL*20] leveraging skeleton-aware operators to take account of the skeleton structures. The pose-level discriminator $\mathcal{D}_P$ in our network comprises $J+1$ discriminators: one for each of the J joint rotations and an additional one for all rotations combined. Each discriminator is structured as a fully connected neural network [DRC*22].

Our network and training framework is implemented in PyTorch and trained on an NVIDIA GeForce GTX Titan Xp GPU (12 GB). We train all our models for 50 epochs using the Adam optimizer [KB14] and the loss terms described in Sec. 3. During training, we employ a fixed temporal window with $T = 64$, and during inference, our temporal window can be arbitrarily long due to the convolutional nature of our model.

We use the same set of hyperparameters λ_{cycle}, λ_{con}, λ_{ee}, and $\lambda_{\text{ee,r}}$ for all examples. The code, hyperparameters, and the dataset can be found in this anonymous GitHub page.

7. Discussion and Conclusion

Our work tackles the challenging task of synthesizing plausible motion in the absence of reference motion data. We propose a novel motion synthesis approach that leverages static pose data by projecting the motion prior from another domain with MoCap data and hallucinating plausible root joint movement. Through extensive experiments on a variety of datasets, we demonstrate that the proposed method can generate high-quality motion sequences that are both plausible and diverse despite significantly different skeleton

topologies, sizes, and proportions, and even outperforms motion-to-motion retargeting in the low-data regime.

Limitation. While we demonstrated that pose data can provide extremely useful priors for motion synthesis, there are some limitations that inevitably arise from the lack of reference motion data in the target domain. As we transfer the motion prior from the source domain, the generated motion can contain motion traits from the source domain that are unrealistic or physically infeasible for the target domain. For example, dogs have specific gaits that are different from those of horses, our method is not able to account for such differences. Similarly, the motion prior from the source domain may not be able to capture the full range of motion of the target domain. One promising venue for future work is to combine pose and limited motion priors to generate more realistic motion, addressing the missing motion prior and integrating it to enhance the synthesis process.

Conclusion. In this paper, we introduced a neural-based motion synthesis approach through retargeting, leveraging static pose data from the target domain to overcome the restrictive requirement of high-quality motion data. Our approach opens up new possibilities for motion synthesis in domains where motion data is scarce or unavailable. By utilizing the latest advancements in related fields such as computer vision, our method can potentially stimulate new applications, such as 2D-to-4D generation, to create new engaging and interactive experiences in entertainment, education, telecommunications and beyond.

8. Acknowledgment

This work is partially funded by the ERC Consolidator Grant No. 101003104 (MYCLOTH), Google, and an SNF Postdoc Mobility fellowship.

References

[AAC22] ANDREOU N., ARISTIDOU A., CHRYSANTHOU Y.: Pose representations for deep skeletal animation. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 155–167. 3

[Ado20] ADOBE SYSTEMS INC.: Mixamo. https://www.mixamo.com, 2020. Accessed: 2020-10-10. 3, 5

[ALL*20] ABERMAN K., LI P., LISCHINSKI D., SORKINE-HORNUNG O., COHEN-OR D., CHEN B.: Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 62–1. 2, 3, 4, 5, 6, 8

[AMYB17] ABDUL-MASSIH M., YOO I., BENES B.: Motion style retargeting to characters with different morphologies. *Comp. Graph. Forum 36* (2017), 86–99. 2, 3

[ASL*18] ABERMAN K., SHI M., LIAO J., LISCHINSKI D., CHEN B., COHEN-OR D.: Deep video-based performance cloning. *arXiv preprint arXiv:1808.06847* (2018). 3

[Aut] AUTODESK:. URL: https://www.autodesk.com/products/motionbuilder/overview. 7

[AWL*20] ABERMAN K., WENG Y., LISCHINSKI D., COHEN-OR D., CHEN B.: Unpaired motion style transfer from video to animation. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 64–1. 3

[BB22] BRODT K., BESSMELTSEV M.: Sketch2pose: Estimating a 3d character pose from a bitmap sketch. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 1–15. 2

[BVGP09] BARAN I., VLASIC D., GRINSPUN E., POPOVIĆ J.: Semantic deformation transfer. *ACM Trans. Graph. 28*, 3 (July 2009). 2

[CK00] CHOI K.-J., KO H.-S.: Online motion retargetting. *The Journal of Visualization and Computer Animation 11*, 5 (2000), 223–235. 2

[CYÇ15] CELIKCAN U., YAZ I. O., ÇAPIN T. K.: Example-based retargeting of human motion to arbitrary mesh models. *Comp. Graph. Forum 34* (2015), 216–227. 2, 3

[DAS*20] DONG Y., ARISTIDOU A., SHAMIR A., MAHLER M., JAIN E.: Adult2child: Motion style transfer using cyclegans. In *Proceedings of the 13th ACM SIGGRAPH Conference on Motion, Interaction and Games* (2020), pp. 1–11. 2, 3

[DRC*22] DAVYDOV A., REMIZOVA A., CONSTANTIN V., HONARI S., SALZMANN M., FUA P.: adversarial parametric pose prior. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2022). 4, 7, 8

[FLFM15] FRAGKIADAKI K., LEVINE S., FELSEN P., MALIK J.: Recurrent network models for human dynamics. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 4346–4354. 3

[GAA*17] GULRAJANI I., AHMED F., ARJOVSKY M., DUMOULIN V., COURVILLE A. C.: Improved training of wasserstein gans. *Advances in neural information processing systems 30* (2017). 4

[Gle98] GLEICHER M.: Retargetting motion to new characters. In *Proc. 25th annual conference on computer graphics and interactive techniques* (1998), ACM, pp. 33–42. 2

[GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. *Advances in Neural Information Processing Systems 27* (2014). 2

[GSR*16] GREENLAND S., SENN S. J., ROTHMAN K. J., CARLIN J. B., POOLE C., GOODMAN S. N., ALTMAN D. G.: Statistical tests, p values, confidence intervals, and power: a guide to misinterpretations. *European journal of epidemiology 31* (2016), 337–350. 8

[GYQ*18] GAO L., YANG J., QIAO Y.-L., LAI Y.-K., ROSIN P. L., XU W., XIA S.: Automatic unpaired shape deformation transfer. *ACM Trans. Graph. 37* (2018), 237:1–237:15. 2, 3

[HSK16] HOLDEN D., SAITO J., KOMURA T.: A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph. 35*, 4 (July 2016), 138:1–11. 3

[HSKJ15] HOLDEN D., SAITO J., KOMURA T., JOYCE T.: Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs* (November 2015), ACM, pp. 18:1–18:4. 3

[HYNP20] HARVEY F. G., YURICK M., NOWROUZEZAHRAI D., PAL C.: Robust motion in-betweening. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 60–1. 3

[IAF09] IKEMOTO L., ARIKAN O., FORSYTH D.: Generalizing motion edits with gaussian processes. *ACM Transactions on Graphics (TOG) 28*, 1 (2009), 1–12. 3

[JKY*18] JANG H., KWON B., YU M., KIM S. U., KIM J.: A variational u-net for motion retargeting. In *SIGGRAPH Asia 2018 Posters* (2018). 3

[JYG*22] JIANG Y., YE Y., GOPINATH D., WON J., WINKLER A. W., LIU C. K.: Transformer inertial poser: Real-time human motion reconstruction from sparse imus with simultaneous terrain generation. In *SIGGRAPH Asia 2022 Conference Papers* (2022), pp. 1–9. 5

[KAS*20] KAUFMANN M., AKSAN E., SONG J., PECE F., ZIEGLER R., HILLIGES O.: Convolutional autoencoders for human motion infilling. In *2020 International Conference on 3D Vision (3DV)* (2020), IEEE, pp. 918–927. 3

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 8

[LAZ*22] LI P., ABERMAN K., ZHANG Z., HANOCKA R., SORKINE-HORNUNG O.: Ganimator: Neural motion synthesis from a single sequence. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 138. 3, 4

[LBK17] LIU M.-Y., BREUEL T., KAUTZ J.: Unsupervised image-to-image translation networks. *Advances in neural information processing systems 30* (2017). 2

[LCC19] LIM J., CHANG H. J., CHOI J. Y.: Pmnet: Learning of disentangled pose and movement for unsupervised motion retargeting. In *BMVC* (2019), vol. 2, p. 7. 3

[LS99] LEE J., SHIN S. Y.: A hierarchical approach to interactive motion editing for human-like figures. In *Proc. 26th annual conference on computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 39–48. 2

[LTV*22] LI R., TANKE J., VO M., ZOLLHÖFER M., GALL J., KANAZAWA A., LASSNER C.: Tava: Template-free animatable volumetric actors. In *European Conference on Computer Vision* (2022), Springer, pp. 419–436. 2

[LWJ*22] LI S., WANG L., JIA W., ZHAO Y., ZHENG L.: An iterative solution for improving the generalization ability of unsupervised skeleton motion retargeting. *Computers & Graphics 104* (2022), 129–139. 3

[LYRK21] LI R., YANG S., ROSS D. A., KANAZAWA A.: Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 13401–13412. 3

[LYS*22] LIAO Z., YANG J., SAITO J., PONS-MOLL G., ZHOU Y.: Skeleton-free pose transfer for stylized 3d characters. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II* (2022), Springer, pp. 640–656. 3

[MBBT00] MONZANI J.-S., BAERLOCHER P., BOULIC R., THALMANN D.: Using an intermediate skeleton and inverse kinematics for motion retargeting. In *Computer Graphics Forum* (2000), vol. 19, Wiley Online Library, pp. 11–19. 2

[PALVdP18] PENG X. B., ABBEEL P., LEVINE S., VAN DE PANNE M.: Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG) 37*, 4 (2018), 1–14. 3

[PCG*19] PAVLAKOS G., CHOUTAS V., GHORBANI N., BOLKART T., OSMAN A. A. A., TZIONAS D., BLACK M. J.: Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2019). 3, 4

[PCZ*20] PENG X. B., COUMANS E., ZHANG T., LEE T.-W., TAN J., LEVINE S.: Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784* (2020). 3

[PW99] POPOVIĆ Z., WITKIN A.: Physically based motion transformation. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 11–20. 2

[RWY*23] REDA D., WON J., YE Y., VAN DE PANNE M., WINKLER A.: Physics-based motion retargeting from sparse inputs. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 6*, 3 (2023), 1–19. 3

[SGXT20] SHIMADA S., GOLYANIK V., XU W., THEOBALT C.: Physcap: Physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics (ToG) 39*, 6 (2020), 1–16. 3

[SMH24] SHOOTER M., MALLESON C., HILTON A.: Digidogs: Single-view 3d pose estimation of dogs using synthetic training data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2024), pp. 80–89. 2

[SMK22] STARKE S., MASON I., KOMURA T.: Deepphase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 1–13. 3

[SOL13] SEOL Y., O'SULLIVAN C., LEE J.: Creature features: online motion puppetry for non-human characters. In *Proc. SCA '13* (2013). 2

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 399–405. 2

[SZKS19] STARKE S., ZHANG H., KOMURA T., SAITO J.: Neural state machine for character-scene interactions. *ACM Transactions on Graphics 38*, 6 (2019), 178. 3

[TCL23] TSENG J., CASTELLON R., LIU K.: Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 448–458. 3

[TK05] TAK S., KO H.-S.: A physically-based motion retargeting filter. *ACM Trans. Graph. 24*, 1 (2005), 98–117. 2

[TRG*22] TEVET G., RAAB S., GORDON B., SHAFIR Y., COHEN-OR D., BERMANO A. H.: Human motion diffusion model. *arXiv preprint arXiv:2209.14916* (2022). 3

[Tru22] TRUEBONES MOTIONS ANIMATION STUDIOS: Truebones, 2022. Accessed: 2022-1-15. URL: https://truebones.gumroad.com/. 2, 5, 6

[VCH*21] VILLEGAS R., CEYLAN D., HERTZMANN A., YANG J., SAITO J.: Contact-aware retargeting of skinned motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 9720–9729. 2, 3

[VYCL18] VILLEGAS R., YANG J., CEYLAN D., LEE H.: Neural kinematic networks for unsupervised motion retargetting. In *Proc. IEEE CVPR* (2018), pp. 8639–8648. 2, 3

[WLJ*23] WU S., LI R., JAKAB T., RUPPRECHT C., VEDALDI A.: MagicPony: Learning articulated 3d animals in the wild. 2, 5, 7

[WPP14] WAMPLER K., POPOVIĆ Z., POPOVIĆ J.: Generalizing locomotion style to new animals with inverse optimal regression. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 1–11. 2

[YAH10] YAMANE K., ARIKI Y., HODGINS J. K.: Animating non-humanoid characters with human motion data. In *Proc. SCA '10* (2010). 2

[YSI*23] YUAN Y., SONG J., IQBAL U., VAHDAT A., KAUTZ J.: Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 16010–16021. 3

[YYB*23] YIN W., YIN H., BARAKA K., KRAGIC D., BJÖRKMAN M.: Dance style transfer with cross-modal transformer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2023), pp. 5058–5067. 3

[ZBL*19] ZHOU Y., BARNES C., LU J., YANG J., LI H.: On the continuity of rotation representations in neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019), pp. 5745–5753. 3

[ZKBWB19] ZUFFI S., KANAZAWA A., BERGER-WOLF T., BLACK M. J.: Three-d safari: Learning to estimate zebra pose, shape, and texture from images" in the wild". In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 5359–5368. 2

[ZLAH23] ZHANG Z., LIU R., ABERMAN K., HANOCKA R.: Tedi: Temporally-entangled diffusion for long-term motion synthesis. *arXiv preprint arXiv:2307.15042* (2023). 3

[ZPIE17] ZHU J.-Y., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. ICCV 2017* (2017), pp. 2223–2232. 2, 3

[ZSKS18] ZHANG H., STARKE S., KOMURA T., SAITO J.: Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–11. 5, 6

[ZWK*23] ZHANG J., WENG J., KANG D., ZHAO F., HUANG S., ZHE X., BAO L., SHAN Y., WANG J., TU Z.: Skinned motion retargeting with residual perception of motion semantics & geometry. *arXiv preprint arXiv:2303.08658* (2023). 2, 3