MimiCAT: Mimic with Correspondence-Aware Cascade-Transformer for Category-Free 3D Pose Transfer

Supplementary Material

This supplementary material provides additional technical details and presents further visualized results supporting the main paper. The content is organized as follows: First, the notations used throughout the paper is summarized in Section A1. Section A2 describes the details of our pose prior model. Section A3 presents further implementation details. Section A4 includes additional qualitative visualizations and a user study to strengthen our results. Finally, Section A5 discusses the limitations of our method and outlines potential directions for future work.

A1. Notation Table

For clarity and ease of reference, the key notations used throughout the paper are summarized in Tab. A1.

A2. Details of Pose Prior Transformer

In this section, we provide technical details of the pose prior model, including how we address the challenges arising from varying keypoint lengths and diverse rotation behaviors across characters, as well as the training objectives.

A2.1. Architecture of Pose Prior Model

We observe that without proper priors or regularization, the pose transfer model often suffers from severe degeneration such as collapse. However, directly learning unified rotation representations across characters with highly diverse geometries and skeletal structures is non-trivial. To address this, we leverage our motion dataset to train a probabilistic prior model that explicitly captures the likelihood of skeletal structures together with their associated rotations.

Formally, the rotation of each keypoint is represented by a quaternion $\mathbf{q} \in \mathbb{S}^3$ [5]. One option is to parameterize this distribution using the Bingham distribution [3]. However, the Bingham distribution requires strong constraints on its parameters, which limits model expressiveness [8]. Instead, we follow previous works [7, 8, 10] to map quaternions to attitude matrices $\mathbf{R} = A(\mathbf{q}) \in \mathrm{SO}(3)$ and model them using the *matrix-Fisher distribution*, which defines a probability density over $\mathrm{SO}(3)$:

$$p(\mathbf{R}_k \mid \mathbf{F}_k) = \frac{1}{c(\mathbf{F}_k)} \exp\left(\mathrm{tr}(\mathbf{F}_k^\top \mathbf{R}_k)\right), \tag{A1}$$

where $\mathbf{F}_k \in \mathbb{R}^{3\times 3}$ is the distribution parameter of the k-th keypoint, and $c(\mathbf{F}_k)$ is the normalization constant.

Similar to the cascade-transformer design of MimiCAT, to capture the joint distribution of rotations across all keypoints of arbitrary characters, we design a transformer-based pose prior model $\mathcal{F}$. It estimates $p(\bar{\mathbf{C}}, \mathbf{C}; \mathbf{f}_{\bar{\mathbf{V}}}, \mathbf{f}_{\mathbf{V}}) = \prod_{k=1}^{K} p(\mathbf{R}_k \mid \mathbf{F}_k)$, where $\bar{\mathbf{C}}$ and $\mathbf{C}$ denote canonical and posed keypoints, and $\mathbf{f}_{\bar{\mathbf{V}}}, \mathbf{f}_{\mathbf{V}}$ are geometry features extracted from the corresponding meshes.

Specifically, we concatenate $\mathbf{f}_{\bar{\mathbf{V}}}$ and $\mathbf{f}_{\mathbf{V}}$ and project them through the shape projector to obtain the shape tokens $f_{\mathbf{M}} \in \mathbb{R}^{l_{\mathcal{E}} \times d_c}$. For the keypoints tokens, we concatenate the canonical and posed keypoint coordinates $\bar{\mathbf{C}}$ and $\mathbf{C}$, and map them into a d_c-dimensional latent representation $f_{\mathbf{C}} \in \mathbb{R}^{K\times d_c}$ via keypoint encoder. The concatenated tokens $[f_{\mathbf{M}}, f_{\mathbf{C}}]$ are then fed into transformer blocks, which applies attention mechanism to model interactions among keypoints while conditioning on global ge-

Table A1. **Summary of important notations.** A consolidated reference of the key variables and symbols used in MimiCAT, grouped according to the modules introduced in the main paper.

Notation	Description
$\bar{\mathbf{V}}^{\text{src}}$	vertices of source character in canonical pose
$\mathbf{V}^{\text{src}}$	vertices of posed source character
N^{src}	number of vertices in source character
$\bar{\mathbf{V}}^{\text{tgt}}$	vertices of target character in canonical pose
$\hat{\mathbf{V}}^{\text{tgt}}$	vertices of posed target character (predicted)
N^{tgt}	number of vertices in target character
$\bar{\mathbf{C}}^{\text{src}}$	canonical keypoints of source character
$\bar{\mathbf{C}}^{\text{tgt}}$	canonical keypoints of target character
K_1	number of keypoint of source character
K_2	number of keypoint of target character
$f(\mathbf{V}^{\text{A}}, \bar{\mathbf{V}}^{\text{A}}, \bar{\mathbf{V}}^{\text{B}}) \to \hat{\mathbf{V}}^{\text{B}}$	transferring pose from character A to B
$\mathcal{E}$	pretrained shape encoder
$\mathbf{f}_{\bar{\mathbf{V}}^{\text{src}}}$	shape feature of canonical source character
$\mathbf{f}_{\mathbf{V}^{\text{src}}}$	shape feature of posed source character
$\mathbf{f}_{\bar{\mathbf{V}}^{\text{tgt}}}$	shape feature of canonical target character
$\delta_{\mathbf{f}}$	residual shape feature of source character
$\mathcal{F}$	pose prior transformer
$A(\mathbf{q})$	attitude matrix of quaternion $\mathbf{q}$
$f_{\mathbf{M}}$	shape tokens for $\mathcal{F}$
$f_{\mathbf{C}}$	keypoint tokens for $\mathcal{F}$
$\hat{\mathbf{F}}$	parameters of matrix-Fisher distribution
$c(\mathbf{F}_k)$	distribution normalization constant w.r.t. $\mathbf{F}_k$
$\mathcal{G}$	correspondence transformer
$g_{\mathbf{M}}$	shape tokens for $\mathcal{G}$
$g_{\mathbf{C}}$	keypoint tokens for $\mathcal{G}$
$\mathbf{g}^{\text{src}}$	shape-aware latent representations of source keypoints
$\mathbf{g}^{\text{tgt}}$	shape-aware latent representations of target keypoints
$\mathbf{A}$	learnable weight for affinity matrix of $\mathcal{G}$
$\mathbf{S}$	similarity matrix of source and target keypoints
$\mathbf{f}_k$	CLIP latent feature of keypoint $\mathbf{c}_k$
$\mathbf{S}_{\text{cos}}$	CLIP-based cosine similarity of keypoint pairs
$\mathbf{M}$	doubly stochastic correspondence matrix
$\mathcal{H}$	pose transfer transformer
$h_{\mathbf{M}}$	shape tokens for $\mathcal{H}$
$h_{\mathbf{C}}$	keypoint tokens for $\mathcal{H}$
$\mathbf{T}^{\text{src}} = \{\mathbf{T}_1^{\text{src}}, \cdots, \mathbf{T}_{K_1}^{\text{src}}\}$	per-keypoint transformations for source character
$\mathbf{T}^{\text{tgt}} = \{\hat{\mathbf{T}}_1^{\text{tgt}}, \ldots, \hat{\mathbf{T}}_{K_2}^{\text{tgt}}\}$	per-keypoint transformations for target character
$\mathbf{q}_i$	rotation quaternion of i-th keypoint
$\mathbf{t}_i$	translation vector of i-th keypoint

Figure A1. **Additional pose examples from PokeAnimDB.** PokeAnimDB contains diverse and high-quality character poses covering a wide spectrum of species and morphological structures. From *left* to *right*, we present representative pose samples across characters.

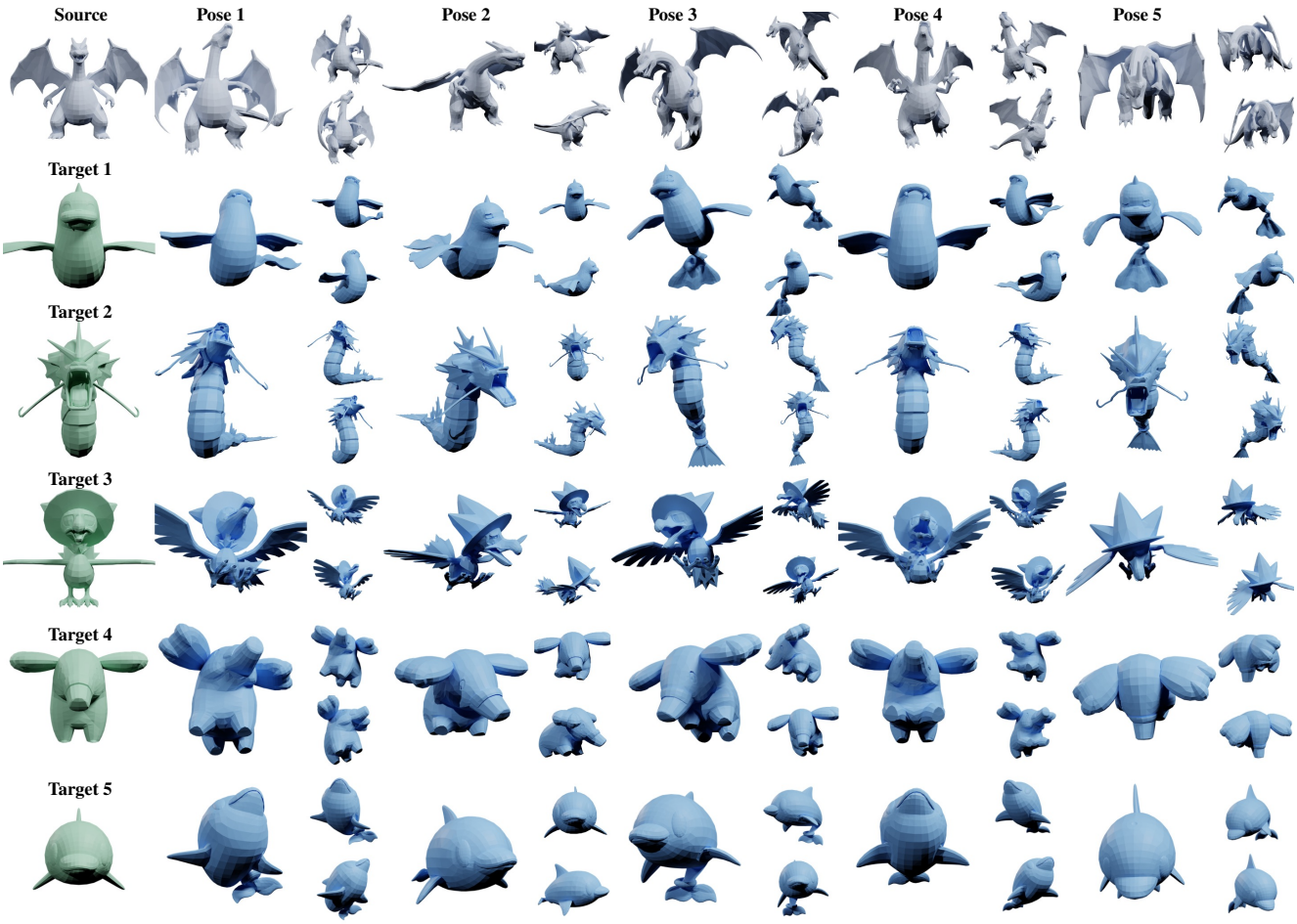

Figure A2. **Qualitative results of MimiCAT (part I).** We present pose transfer results across a wide range of character categories, with each example rendered from three viewpoints. From *left* to *right*: the canonical character followed by its transferred results under five different poses. The *1st* row shows the source character and the five input poses; the *2nd–6th* rows show the corresponding transferred poses for each target character.

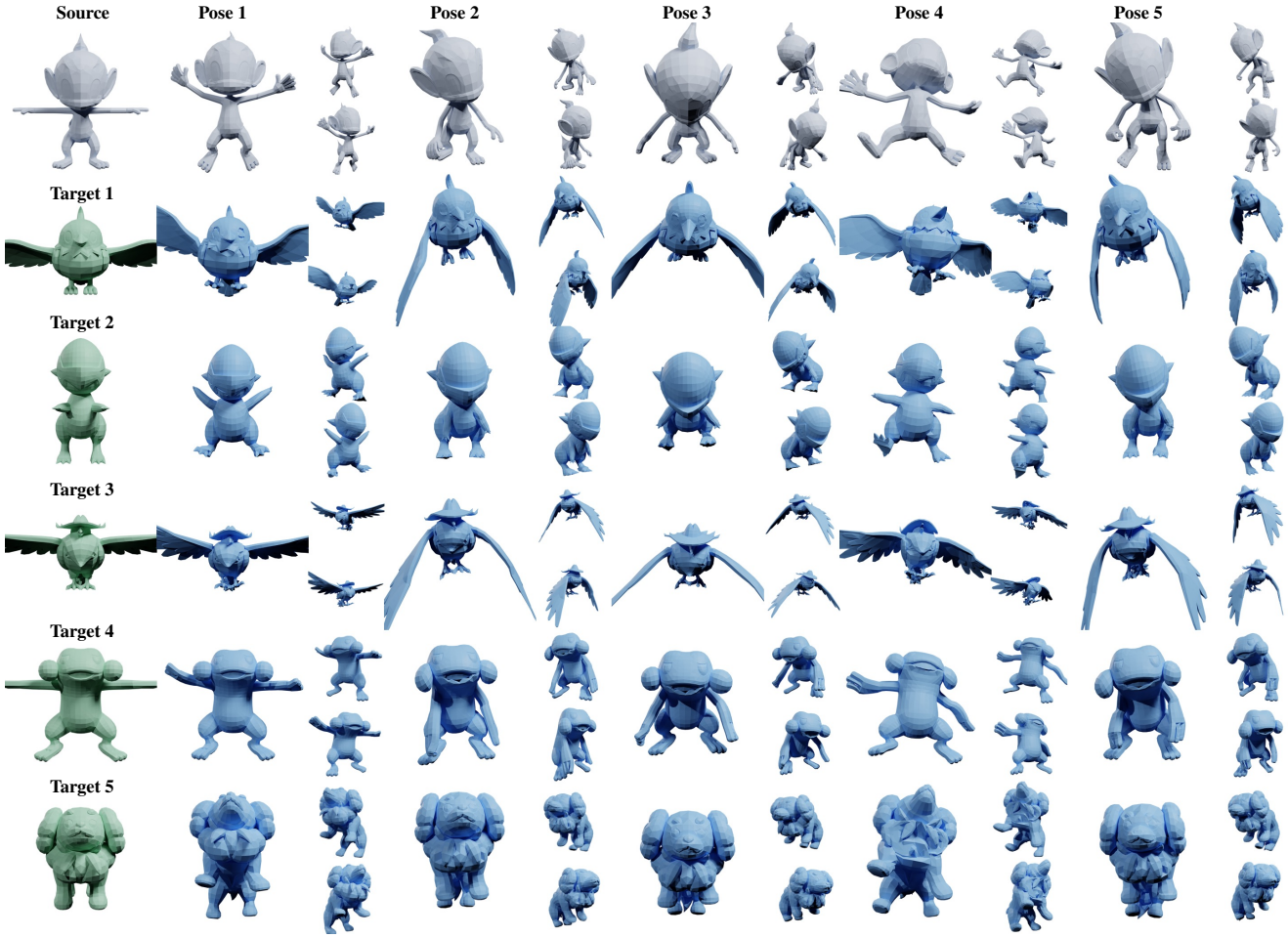

Figure A3. **Qualitative results of MimiCAT (part II).** We present pose transfer results across a wide range of character categories, with each example rendered from three viewpoints. From *left* to *right*: the canonical character followed by its transferred results under five different poses. The *1st* row shows the source character and the five input poses; the *2nd–6th* rows show the corresponding transferred poses for each target character.

ometry. Finally, an MLP decoder maps the latent representations to a set of matrix-Fisher distribution parameters $\{\hat{\mathbf{F}}_1, \cdots, \hat{\mathbf{F}}_K\}$, where each $\hat{\mathbf{F}}_k$ models the rotation distribution of the k-th keypoint.

A2.2. Training Objective Functions

Following previous works [8], $\mathcal{F}$ is trained with the negative log-likelihood (NLL) of the ground-truth rotations, with pose sampled from PokeAnimDB and Mixamo [1]:

$$\mathcal{L}_{\text{NLL}} = \sum_{k=1}^{K} \left(\log c(\mathbf{F}_k) - \text{tr}\left(\mathbf{F}_k^\top A(\mathbf{q}_k) \right) \right). \quad \text{(A2)}$$

Additionally, we adopt differentiable rejection sampling [4] to draw n candidate rotations from the predicted distributions. Combined with the estimated translation vectors, the sampled characters $\hat{\mathbf{V}}$ are reconstructed via linear blend skinning and supervised against the ground-truth $\mathbf{V}$ using a reconstruction loss:

$$\mathcal{L}_{\text{sample}} = \frac{1}{n} \sum_{i=1}^{n} \left\| \hat{\mathbf{V}}^{(i)} - \mathbf{V} \right\|_2^2, \ \mathbf{q}_k^{(i)} \sim p\left(A(\mathbf{q}_k) \mid \mathbf{F}_k \right). \quad \text{(A3)}$$

The pose prior transformer is trained using AdamW [6] with an initial learning rate of $1e{-}4$, a mini-batch size of 256, for a total of 5 epochs. As such, once the probability model is trained, it predicts the distribution parameters for a given canonical–posed keypoint pair. During pose transfer, we regularize the predicted rotations by maximizing their likelihood under these learned distributions, ensuring that the estimated joint rotations remain plausible for the given character geometry.

A3. Implementation Details

A3.1. Model Details of MimiCAT

For all modules–the pose prior transformer $\mathcal{F}$, the correspondence transformer $\mathcal{G}$, and the pose transfer transformer $\mathcal{H}$–we adopt a similar architectural design. The keypoint encoder and shape projector first map their respective inputs into 256-dimensional latent tokens, keypoint encoder is implemented with a 2-layer MLP and shape projector is a linear layer, with hidden dimension of $1{,}024$.

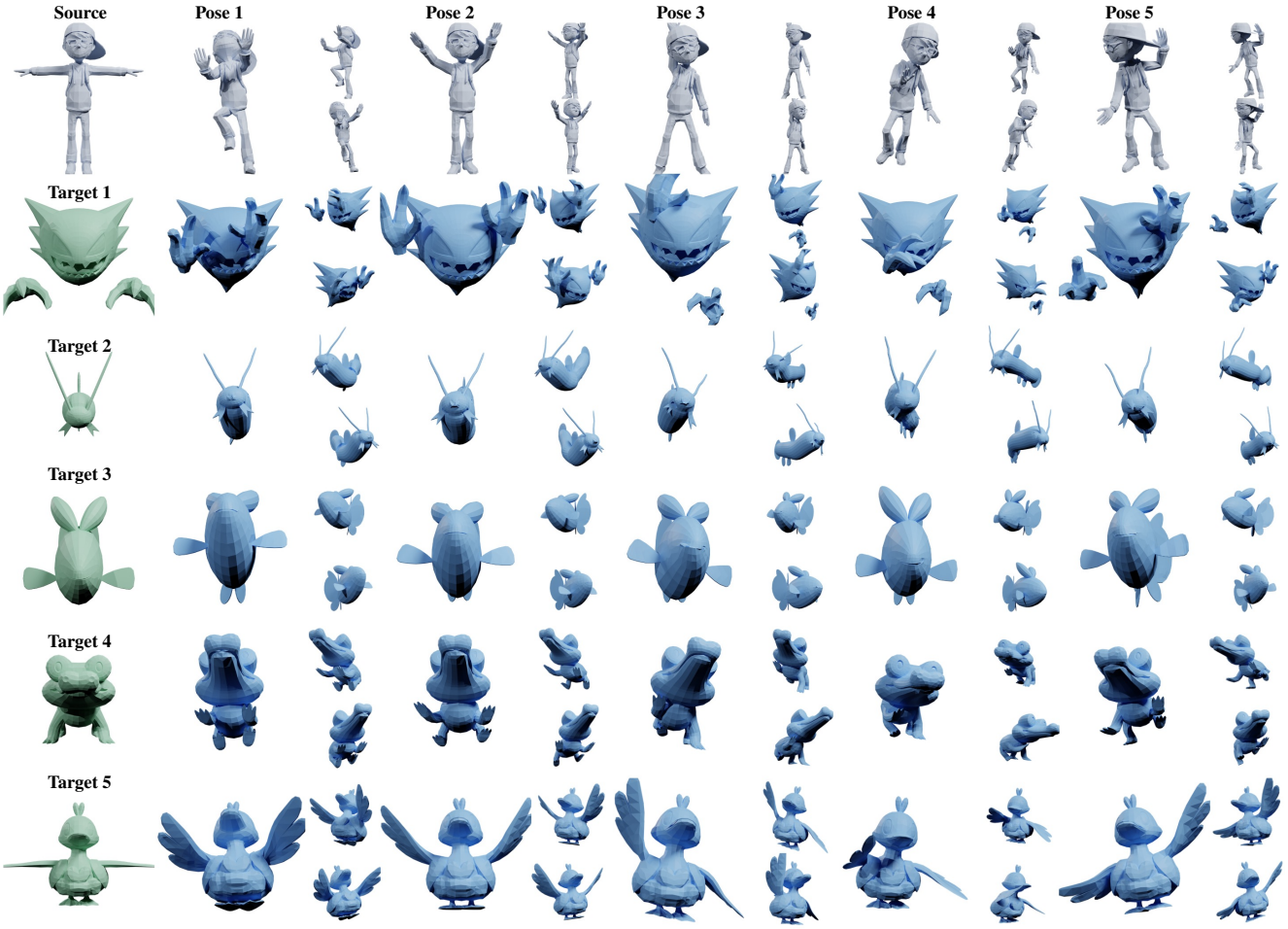

Figure A4. **Qualitative results of MimiCAT (part III).** We present pose transfer results across a wide range of character categories, with each example rendered from three viewpoints. From *left* to *right*: the canonical character followed by its transferred results under five different poses. The *1st* row shows the source character and the five input poses; the *2nd–6th* rows show the corresponding transferred poses for each target character.

For $\mathcal{F}$, $\mathcal{G}$, and $\mathcal{H}$, each module adopts a 6-layer stacked transformer encoder, where every layer comprises a multi-head self-attention (MHSA) module (with 8 heads) followed by a 2-layer MLP. The MLP uses a hidden dimension of 2,048. The distribution decoder of $\mathcal{F}$ is a 2-layer MLP with a hidden dimension of 128 and nonlinear activation. For the correspondence module $\mathcal{G}$, the learnable weights $\mathbf{A}$ is parameterized with a hidden dimension of 256. The transformation decoder of $\mathcal{H}$ is implemented as an MLP with a hidden dimension of 256.

A3.2. Details of Dataset Split

For the AMASS dataset, we follow the standard protocol in prior works [5, 12] and split the motions into training and validation sets. For Mixamo, we use 97 characters for training and 11 for testing. For PokeAnimDB, we split the dataset into 780 training characters, 109 validation characters, and 86 test characters. Across these sources, we use a total of 4.21 million pose samples to train the pose prior transformer $\mathcal{F}$. For the correspondence transformer $\mathcal{G}$, we construct $384k$ canonical-pose pairs for training. To train the pose transfer transformer $\mathcal{H}$, we sample $100k$ source-target character pairs from the training split at each epoch, drawing random pose for each pair during every iteration.

A4. Additional Visualized Results

A4.1. Details of PokeAnimDB

As stated in the main paper, PokeAnimDB provides character-level motion sequences, forming a large-scale corpus of diverse 3D character poses. Each character is associated with a set of predefined motion clips spanning various action categories. On average, a character contains approximately 30 actions, with the number ranging from 3 to 102. Fig. A1 further illustrates the diversity of poses and character types captured in our dataset.

A4.2. Visualized Results from MimiCAT

In Fig. A2, A3, and A4, we present additional qualitative pose transfer results produced by MimiCAT. For each

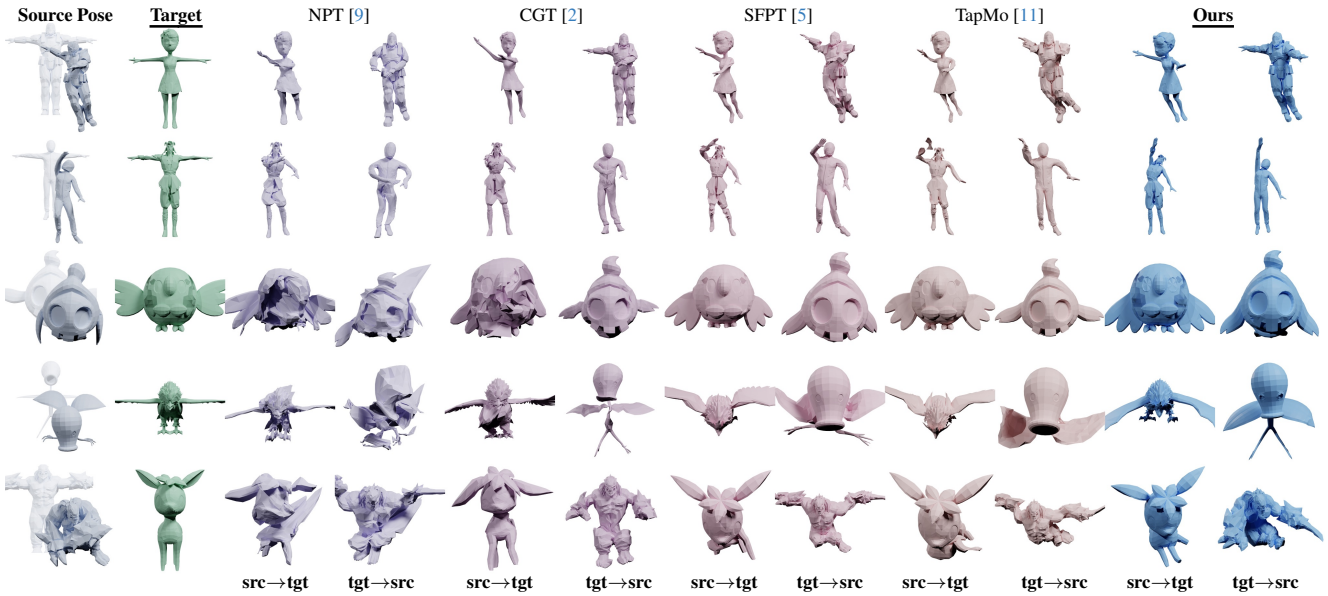

Figure A5. **Qualitative cycle-consistency comparisons with existing methods.** From *left* to *right*: source character, target character, and bidirectional pose transfer results (source→target and target→source) produced by different methods. MimiCAT consistently yields higher-quality transfers with more realistic poses and fewer distortions.

source character, we transfer 5 distinct poses to 5 target characters spanning diverse categories. The results further demonstrate that MimiCAT can reliably transfer poses across structurally different characters, faithfully preserving geometric details while capturing the pose characteristics from the source characters.

A4.3. Cycle Consistency Comparison

As a supplement to our quantitative evaluation, we provide additional visual comparisons of pose transfer quality in Fig. A5. For each method, we show both the transferred target poses and the corresponding cycled-source reconstructions. The first two rows show examples of humanoid-to-humanoid transfer, where we also include cycle-consistency results for reference. The remaining three rows illustrate a variety of cross-category transfer cases, covering challenging scenarios with large geometric and topological discrepancies. These visualizations qualitatively confirm our quantitative findings: MimiCAT achieves the smallest PMD while preserving mesh smoothness (highest ELS), producing more plausible transfer results than prior baselines.

A4.4. User Study

In Tab. A2, we report a user study evaluating human perception of pose transfer quality in comparison with baseline methods. We recruited 50 participants with diverse technical expertise in computer vision and graphics via *Prolific* to assess 20 transfer pairs. Participants rated each method on a 1–5 scale in terms of pose similarity and geometric quality, and selected the best-performing method for each pair. The results show that our method achieves the highest average scores in both pose similarity (4.076) and geometric quality (4.102), and is chosen as the best method in 60.0% of the votes. These findings are consistent with our quantitative evaluations and overall benchmark rankings.

Table A2. **Perceptual user study comparisons with existing methods.** We ask participants to assess samples across two primary dimensions: pose similarity and geometric quality.

Rating (1–5)	NPT [9]	CGT [2]	SFPT [5]	TapMO [11]	**Ours**
Pose Similarity↑	1.884	2.310	3.364	3.292	**4.076**
Geo. Quality↑	1.556	2.516	3.531	3.605	**4.102**

A5. Limitation & Future Work

Although MimiCAT achieves state-of-the-art performance compared to existing baselines, it still has several limitations. In this section, we discuss the limitations of our work and outline several future research directions.

First, our framework relies on keypoints and skin weights predicted by pretrained models. Errors in this stage may propagate to the downstream pose transfer pipeline and negatively affect the final results. In the future, we plan to explore an optimization framework that jointly updates the skin weights, keypoints, and target transformations, such that they coherently contribute to the final transfer quality.

Second, as the exploration of efficient transformer architectures is beyond the scope of this work, MimiCAT adopts computationally expensive vanilla attention implementations. An important extension would be to incorporate more efficient attention mechanisms (*e.g.*, linear or sparse attention) to reduce computational cost while maintaining–or potentially improving–the quality of pose transfer.

Finally, while we demonstrate that MimiCAT can serve as a plug-and-play module for zero-shot text-to-any-

character motion generation, the current pipeline does not explicitly enforce temporal consistency across frames. Incorporating temporal modeling could significantly improve motion-level coherence and stability. In future work, we plan to leverage the dataset introduced in this paper to further advance 4D generation and general motion synthesis.

References

[1] Mixamo. Online service by Adobe., 2025. Accessed: Jan 2025. 3

[2] Haoyu Chen, Hao Tang, Zitong Yu, Nicu Sebe, and Guoying Zhao. Geometry-contrastive transformer for generalized 3D pose transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 258–266, 2022. 5

[3] Igor Gilitschenski, Roshni Sahoo, Wilko Schwarting, Alexander Amini, Sertac Karaman, and Daniela Rus. Deep Orientation Uncertainty Learning based on a Bingham Loss. In *International Conference on Learning Representations*, 2020. 1

[4] John T Kent, Asaad M Ganeiber, and Kanti V Mardia. A new method to simulate the Bingham and related distributions in directional data analysis with applications. *arXiv preprint arXiv:1310.8110*, 2013. 3

[5] Zhouyingcheng Liao, Jimei Yang, Jun Saito, Gerard Pons-Moll, and Yang Zhou. Skeleton-Free Pose Transfer for Stylized 3D Characters. In *European Conference on Computer Vision*, pages 640–656, 2022. 1, 4, 5

[6] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. 3

[7] David Mohlin, Josephine Sullivan, and Gérald Bianchi. Probabilistic orientation estimation with matrix fisher distributions. *Advances in Neural Information Processing Systems*, 33:4884–4893, 2020. 1

[8] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Hierarchical kinematic probability distributions for 3D human shape and pose estimation from images in the wild. In *IEEE/CVF International Conference on Computer Vision*, pages 11219–11229, 2021. 1, 3

[9] Jiashun Wang, Chao Wen, Yanwei Fu, Haitao Lin, Tianyun Zou, Xiangyang Xue, and Yinda Zhang. Neural pose transfer by spatially adaptive instance normalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5831–5839, 2020. 5

[10] Yingda Yin, Yingcheng Cai, He Wang, and Baoquan Chen. FisherMatch: Semi-Supervised Rotation Regression via Entropy-based Filtering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11164–11173, 2022. 1

[11] Jiaxu Zhang, Shaoli Huang, Zhigang Tu, Xin Chen, Xiaohang Zhan, Gang YU, and Ying Shan. TapMo: Shape-aware Motion Generation of Skeleton-free Characters. In *International Conference on Learning Representations*, 2024. 5

[12] Keyang Zhou, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Unsupervised Shape and Pose Disentanglement for 3D Meshes. In *European Conference on Computer Vision*, pages 341–357. Springer, 2020. 4