Ultrafast and Controllable Online Motion Retargeting for Game Scenarios

TIANZE GUO, State Key Laboratory of CAD&CG, Zhejiang University, China
ZHEDONG CHEN, Zhejiang University, China
YI JIANG, State Key Laboratory of CAD&CG, Zhejiang University, China
LINJUN WU, State Key Laboratory of CAD&CG, Zhejiang University, China
XILEI WEI, MoreFun Studios, Tencent, China
LANG XU, MoreFun Studios, Tencent, China
YESHUANG LIN, MoreFun Studios, Tencent, China
HE WANG, AI Centre, University College London, United Kingdom
XIAOGANG JIN*, State Key Laboratory of CAD&CG, Zhejiang University, China

Fig. 1. Our online motion retargeting approach effectively mitigates penetrations and preserves contact fidelity, all while maintaining the overall semantic integrity of the original motion. Furthermore, it enables precise control over attached objects, such as ensuring that a spear strikes its intended target accurately. For each pose pair, the source character is depicted on the left and the retargeted result on the right.

Geometry-aware online motion retargeting is crucial for real-time character animation in gaming and virtual reality. However, existing methods often rely on complex optimization procedures or deep neural networks, which constrain their applicability in real-time scenarios. Moreover, they offer limited control over fine-grained motion details involved in character interactions, resulting in less realistic outcomes. To overcome these limitations, we propose a novel optimization framework for ultrafast, lightweight motion retargeting with joint-level control (i.e., controls over joint position, bone orientation, etc,). Our approach introduces a semantic-aware objective grounded in a spherical geometry representation, coupled with a bone-length-preserving algorithm that iteratively solves this objective. This formulation preserves spatial relationships among spheres, thereby maintaining motion semantics, mitigating interpenetration, and ensuring contact. It is lightweight and computationally efficient, making it particularly suitable for time-critical real-time deployment scenarios. Additionally, we incorporate a heuristic optimization strategy that enables rapid convergence and precise joint-level control. We evaluate our method against state-of-the-art approaches on the Mixamo dataset, and experimental results demonstrate that it achieves comparable performance while delivering an order-of-magnitude speedup.

CCS Concepts: • **Computing methodologies** → **Motion processing**.

Additional Key Words and Phrases: Character Animation, Motion Retargeting

ACM Reference Format:
Tianze Guo, Zhedong Chen, Yi Jiang, Linjun Wu, Xilei Wei, Lang Xu, Yeshuang Lin, He Wang, and Xiaogang Jin. 2025. Ultrafast and Controllable Online Motion Retargeting for Game Scenarios. *ACM Trans. Graph.* 44, 6 (December 2025), 11 pages. https://doi.org/10.1145/3763351

*Corresponding author.

Authors' Contact Information: Tianze Guo, 22421091@zju.edu.cn, State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China; Zhedong Chen, zdchen17@zju.edu.cn, Zhejiang University, Hangzhou, China; Yi Jiang, 3210103803@zju.edu.cn, State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China; Linjun Wu, 12321232@zju.edu.cn, State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China; Xilei Wei, 396591712@qq.com, MoreFun Studios, Tencent, Shenzhen, China; Lang Xu, xuwave631@gmail.com, MoreFun Studios, Tencent, Shenzhen, China; Yeshuang Lin, amberyslin@tencent.com, MoreFun Studios, Tencent, Shenzhen, China; He Wang, realcrane@gmail.com, AI Centre, University College London, London, United Kingdom; Xiaogang Jin, jin@cad.zju.edu.cn, State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China.

ACM 1557-7368/2025/12-ART
https://doi.org/10.1145/3763351

1 Introduction

Motion retargeting has long been a challenge in computer animation, dating back to several decades [Choi and Ko 1999; Gleicher 1998]. In recent years, the problem has revived in the deep learning era [Villegas et al. 2018], with data-driven approaches [Cheynel et al. 2025; Ye et al. 2024; Zhang et al. 2023b] significantly advancing the fidelity of retargeted motions. By integrating character geometry representations into semantic-aware constraints, these methods achieve improved preservation of motion semantics and mitigation of artifacts such as self-penetrations and mismatched contacts.

However, the game scenarios require *online* deployment, which means runtime calculation at a rate no lower than the frame rate of the game environment, as well as high quality results without the need for post-processing. Unfortunately, recent geometry-aware methods continue to exhibit three critical limitations that constrain their applicability in game scenarios. (i) Current methods are too computationally expensive to support real-time execution given the limited computing resources available in interactive video games. Unlike offline approaches, online performance is the only viable solution in scenarios where the specific retargeting task cannot be anticipated, such as in interactive gaming applications. (ii) They lack precise control over frequent and subtle environmental interactions, particularly in accurate contact handling between character mesh and external environments, such as opening a door without penetration between the character's hand and the door. (iii) They typically operate on characters with fixed-topology skeletons and neglect attached objects, while the motion of attached objects, such as the aiming target when holding a rifle, is critical for preserving motion semantics during retargeting.

These limitations motivate us to develop an ultrafast and controllable geometry-aware motion retargeting framework tailored for interactive gaming environments. This goal introduces two primary challenges. The first is achieving real-time performance under limited computational resources. At runtime, most system resources are allocated to rendering and core gameplay logic, necessitating a lightweight online solution while still capable of preserving motion semantics and mitigating artifacts. The second challenge lies in enabling fine-grained control without compromising overall motion semantics. High-fidelity environmental interactions demand joint-level or even vertex-level accuracy, whereas semantic preservation requires adjusting the entire body to satisfy overall motion constraints. Unifying these constraints without conflict or degradation in performance remains a challenge.

To address these challenges, we propose a novel optimization framework that formulates the retargeting task as a conditioned optimization problem, where a geometry-aware objective minimizes the semantic difference, such as misplaced end-effectors and unmaintained contact between the original and retargeted motions, and joint-level conditions facilitate fine-grained control. To achieve real-time performance, we further develop a heuristic optimization strategy tailored for fast convergence.

Our method builds upon a hybrid representation that jointly models the character and its attached objects. This representation combines skeletal joints with a set of sampling spheres extracted from the input mesh. Each sphere is associated with specific joints to maintain a consistent mapping between kinematic structure and surface geometry. Based on this, we define an optimization objective that maintains spatial relationships among the spheres, which not only preserves motion semantics but also mitigates mesh interpenetration and preserves contact. Complementing this, joint-level conditions impose spatial constraints on joints and associated spheres, enabling precise control in interactive scenarios. We solve the optimization problem using a heuristic algorithm that updates joint positions iteratively while preserving bone lengths and that allows direct joint manipulation. Our method is rotation-free— it avoids computing forward kinematics or Jacobians, which significantly reduces computational cost. By integrating these components, our framework achieves ultrafast performance with minimal computational overhead, making it suitable for online platforms. In addition, it supports highly controllable motion generation, enabling customized interactions with environments. Experimental results demonstrate that our method achieves comparable performance to state-of-the-art approaches while delivering an order-of-magnitude speedup. In summary, the contributions of this paper are as follows:

- An ultrafast and controllable motion retargeting method that simultaneously ensures semantic preservation, precise control, and online performance.
- A sphere-based geometric representation and a corresponding simplified optimization objective that enables the preservation of motion semantics at low computational cost.
- A rotation-free optimization method that bypasses forward kinematics and Jacobian computations while remaining compatible with precise and fine-grained controls.

2 Related Works

2.1 Motion Retargeting

Motion retargeting aims to transfer motion from a source character to a target character with different skeletons and geometry. Early approaches formulate motion retargeting as a space-time optimization problem [Gleicher 1998], utilizing specialized optimization techniques (e.g., inverse kinematics) to ensure precise control over end-effectors [Choi and Ko 1999; Lee and Shin 1999]. However, these methods often rely on carefully hand-designed constraints tailored to specific motions, making them cumbersome to deploy. To address these limitations, some semantic-aware spatial structure methods [Al-Asqhar et al. 2013; Ho et al. 2010; Ho and Shum 2013; Kim et al. 2016] are proposed for automatic optimization. Deep learning-based methods further enhance the efficiency of holistic and automatic retargeting through semantic preservation constraints [Aberman et al. 2020; Jang et al. 2018; Lim et al. 2019; Mourot et al. 2023; Villegas et al. 2018]. While effective, these methods generally neglect the geometry of characters, leading to frequent contact mismatches and severe self-penetrations. Several geometry-aware optimization methods leverage different geometries to eliminate these artifacts [Basset et al. 2019; Ho et al. 2010; Wang et al. 2023], but they suffer from high computational cost due to complex objectives. Recent approaches improve efficiency by leveraging latent optimization [Villegas et al. 2021], geometry-aware networks [Zhang et al. 2023b], and vision-language fine-tuning [Zhang et al. 2024]. Reinforcement learning-based methods leverage physically simulated

characters to achieve more realistic motion control [Reda et al. 2023] and environmental interaction [Zhang et al. 2023a]. Nevertheless, the computational overhead of physical simulation poses significant challenges for real-time deployment in games.

State-of-the-art methods [Cheynel et al. 2025; Ye et al. 2024] adopt sampling-point-based strategies, enforcing semantic consistency via point clouds sampled on the character surface. While dense sampling [Ye et al. 2024] offers better geometric coverage at high computational cost, sparse alternatives [Cheynel et al. 2025] are more efficient but may fail to mitigate artifacts. Despite their ability to preserve whole-body semantics, these methods often lack precise control over fine-grained interactions. In practice, inverse kinematics-based methods remain the dominant solution in game engines for retargeting with high-fidelity spatial constraints. To date, no unified geometry-aware framework integrates semantic preservation and fine-grained control for real-time game scenarios.

2.2 Character Geometry Representations

In motion retargeting, character geometry is commonly employed to handle self-penetrations and preserve contacts. While original meshes provide accurate geometric representations [Liu et al. 2018; Wang et al. 2023], their high computational cost and indeterminate topology render them unsuitable for real-time processing. SMPL models [Loper et al. 2015] have been leveraged to enable optimization-based, geometry-aware retargeting [Basset et al. 2019; Jin et al. 2017], but are restricted to regular humanoid forms. Bounding-volume-based methods offer greater generalizability [Ho et al. 2010; Zhang et al. 2023b], typically attaching volumes such as bounding boxes to bones. Point clouds provide high flexibility and generalizability for simplifying character meshes [Biswas et al. 2021; Jang et al. 2024; Ye et al. 2024; Zhang et al. 2023b]; however, balancing the sampling density to minimize computational overhead while mitigating artifacts remains a significant challenge.

In addition to motion retargeting, simplified character geometries also apply to other fields, such as rendering and cloth simulation. Spherical approximation methods for soft shadowing offer low computational overhead and plausible shadows [Ren et al. 2006; Wang et al. 2006]. A mixed geometry approach using capsules and spheres has been employed to accelerate cloth simulations [Wu et al. 2018]. The sparse sphere-based geometries have the advantage of sparse sampling points that are computationally efficient and easy to manipulate with skinning methods. Furthermore, spheres provide better coverage of the character mesh due to their volume and mitigate the risk of missing self-penetrations on small body parts.

3 Method

3.1 Overview

The standard motion retargeting task is formulated as follows: Given a source character A with skeleton representation S^A, geometry representation G^A and motion sequence Q; a target character B with skeleton S^B and geometry G^B; and a set of interaction controls $\{C_i\}$ (manually defined constraints on parameters such as joint positions for precise control of the environmental interactions, details are in Sec. 3.6), the objective is to synthesize a plausible motion sequence $\hat{Q}$ for the target character:

$$\hat{Q} = F\left(Q, S^A, S^B, G^A, G^B, \{C_i\}\right). \tag{1}$$

The retargeted motion $\hat{Q}$ should preserve the semantics of the source motion and satisfy the given controls while mitigating artifacts such as self-penetrations and mismatched contacts. In this paper, we assume the source animation is of high quality and contains few artifacts, which is certainly satisfactory for hand-made animations. In our method, the character skeleton S is represented by joint offsets in the T-pose $\{\boldsymbol{s}_j\}_{j=1}^{J}$, where J is the number of joints. The geometry G is represented by a set of sampling spheres $\{g_i\}_{i=1}^{N}$, where N is the number of spheres. The motion Q is defined as a sequence of poses over time: $\{\boldsymbol{q}_t\}_{t=1}^{T}$, where each pose $\boldsymbol{q}_t \in \mathbb{R}^{J\times 4}$ denotes quaternion joint rotations. As an online framework, our method computes the retargeted pose $\hat{\boldsymbol{q}}_t$ for each input pose $\boldsymbol{q}_t$ in a streaming manner. Unless otherwise specified, we use $\boldsymbol{q}$ and $\boldsymbol{q}_t$ interchangeably to refer to a pose at a specific frame.

To solve this problem, we first design a semantic-preserving objective based on our novel spherical geometry representation (Sec. 3.2). To solve this objective, we begin with a preprocessing step for initialization (Sec. 3.3), followed by a heuristic rotation-free strategy to optimize the joint positions efficiently (Sec. 3.4) and two ways to recover joint rotations from that yield the retargeted pose (Sec. 3.5). Finally, we seamlessly integrate precise interaction control into the framework (Sec. 3.6). The following sections will describe the details.

3.2 Semantic-preserving Objective

In this section, we first introduce our novel spherical geometry representation, which provides a simplified yet effective way to model character geometry. Then, we introduce the semantic-preserving objective, which aims to preserve motion semantics while mitigating artifacts such as self-penetrations and mismatched contacts.

3.2.1 Spherical Geometry Representation. State-of-the-art retargeting methods [Cheynel et al. 2025; Ye et al. 2024] represent human geometry with sampling points on the skin, which have two inherent drawbacks: First, due to the sparsity of the sampling points, a large number of points are required to eliminate penetrations. Second, these points require normal directions to distinguish the inner and outer sides of the mesh, leading to inevitable rotational calculations. In contrast, our rotation-free spherical geometry representation overcomes these limitations.

Specifically, we adopt a hybrid representation that models the kinematic structure via skeletal joint positions and approximates surface geometry using a set of sampling spheres $\{g_i\}_{i=0}^{N}$ extracted from the input mesh. Each sphere is associated with a bone, defined by its parent joint $J_p(g_i)$ and child joint $J_c(g_i)$, thereby maintaining a consistent mapping between kinematic structure and surface geometry.

For a given joint with index j, we define two sets of indices of associated spheres: the *parent spheres* $\mathbb{G}_p(j) = \{i | J_c(g_i) = j\}$ and the *child spheres* $\mathbb{G}_c(j) = \left\{i | J_p(g_i) = j\right\}$. Each sampling sphere is represented as $g_i = (\boldsymbol{c}_i, \boldsymbol{c}_i^l, r_i)$, where $\boldsymbol{c}_i$ donates the position of sphere center in world coordinates, $\boldsymbol{c}_i^l$ is the position of sphere center

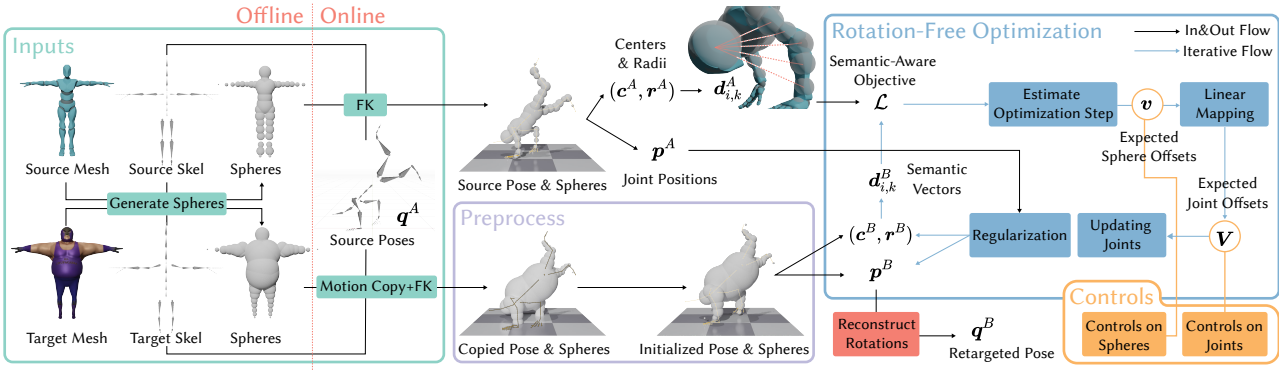

Fig. 2. **Overview of our framework.** The pipeline takes the skeletons and source poses, together with sampling spheres extracted from character mesh as input, followed by preprocessing and iterative optimization steps. These stages successively adjust the joint positions $\boldsymbol{p}^B$ and sphere positions $\boldsymbol{c}^B$ of the target character. Finally, the retargeted joint rotations $\boldsymbol{q}^B$ are reconstructed based on the optimized joint positions. Spatial controls are incorporated into each iteration by manipulating the expected offsets of joints ($\boldsymbol{V}$) and spheres ($\boldsymbol{v}$), enabling precise control over motion details.

in local coordinates relative to parent joint, and r_i is radius. To ensure semantic consistency between the source and target characters, the source and target sphere sets g_i^A and g_i^B must contain the same number of spheres and maintain identical joint associations, that is, $J^A(g_i^A) = J^B(g_i^B)$ for both parent and child joints.

We propose an automated method to generate sampling spheres, thereby avoiding manual construction. Inspired by [Ren et al. 2006; Wang et al. 2006; Wu et al. 2018], we iteratively optimize each sphere g_i by minimizing its external volume $V_{out}(g_i)$, defined as:

$$V_{out}(g_i) = V_{g_i} \wedge (\neg V_{mesh}), \tag{2}$$

where V_{g_i} denotes the volume of sphere g_i, V_{mesh} represents the volume of character's mesh. Minimizing $V_{out}(g_i)$ yields a compact approximation of the mesh surface using the set of sampling spheres. Further details can be found in the supplementary material.

3.2.2 Objective Definition. The semantic-aware objective operates purely on the geometric representation to preserve motion semantics while mitigating self-penetrations and mismatched contacts. To this end, we introduce *semantic vectors* that represent spatial relationships between sampling spheres, effectively capturing the motion semantics, as shown in Fig. 3. For the source character, the semantic vectors $\boldsymbol{d}_{i,k}^A$ between spheres i and k are defined as the relative offset between the nearest points on their respective surfaces:

$$\boldsymbol{d}_{i,k}^A = \left(\boldsymbol{c}_k^A - \boldsymbol{c}_i^A\right) \cdot \max\left(\frac{||\boldsymbol{c}_k^A - \boldsymbol{c}_i^A|| - (r_i^A + r_k^A)}{||\boldsymbol{c}_k^A - \boldsymbol{c}_i^A||}, \varepsilon\right), \tag{3}$$

where ε is a small positive constant, and $||\cdot||$ denotes the Euclidean length of a vector. We apply the $\max(\cdot)$ operation to ensure the semantic vectors do not flip in cases where self-penetration occurs in the source character. For the target character, the semantic vectors are defined as the relative offset between two auto-sampled points that comes from projecting the centers of spheres g_i^B and g_k^B onto their respective surface along the direction of $\boldsymbol{d}_{i,k}^A$ and $-\boldsymbol{d}_{i,k}^A$:

$$\boldsymbol{d}_{i,k}^B = \left(\boldsymbol{c}_k^B - r_k^B \cdot \frac{\boldsymbol{d}_{i,k}^A}{||\boldsymbol{d}_{i,k}^A||}\right) - \left(\boldsymbol{c}_i^B + r_i^B \cdot \frac{\boldsymbol{d}_{i,k}^A}{||\boldsymbol{d}_{i,k}^A||}\right). \tag{4}$$

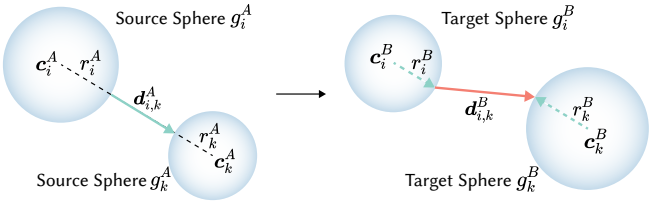

Fig. 3. Illustration of the *semantic vectors* $\boldsymbol{d}_{i,k}^A$ and $\boldsymbol{d}_{i,k}^B$, representing the spatial relationships between sampling spheres for the source and target characters, respectively.

Matching such pairwise semantic vectors between the target and source characters preserves motion semantics and mitigates artifacts. To better capture the semantic significance of different vectors, each semantic vector is assigned a composite weight incorporating: (i) spatial proximity in the source pose, measured by $w_{i,k}^{dist}$; (ii) a penetration mask $m_{i,k}^{pen}$ that excludes physically implausible configurations due to self-penetration. The overall weighting mechanism prioritizes contact preservation and reduces self-penetration. Detailed definitions are in the supplementary material.

The pairwise semantic discrepancies are aggregated to compute the semantic loss $\boldsymbol{\mathcal{L}}_i$ for each sphere g_i:

$$\boldsymbol{\mathcal{L}}_i = \sum_{k=1}^{N} \left(\boldsymbol{d}_{i,k}^B - \boldsymbol{d}_{i,k}^A\right) \cdot \frac{w_{i,k}^{dist} \cdot m_{i,k}^{pen}}{\sum_{k=1}^{N} w_{i,k}^{dist} \cdot m_{i,k}^{pen}}. \tag{5}$$

We define the semantic-aware objective as minimizing the semantic loss for all spheres:

$$\underset{\boldsymbol{p}_j}{argmin} \sum_{i=1}^{N} ||\boldsymbol{\mathcal{L}}_i||^2, \tag{6}$$

where $\boldsymbol{p}_j$ is the position of joints in the retargeted pose that determines the location of spheres.

3.2.3 Mapping Between Joints And Spheres. To maintain the consistent mapping between kinematic structure and surface geometry,

we establish relations between the joint positions and sphere locations and optimize Eq. 6 by iteratively updating joint positions. The relations are defined as linear mappings between the offsets of joints and spheres during the iterative optimization, which offers an efficient approximation of the precise position relations. For a given sphere g_i, let $\boldsymbol{V}_p$ and $\boldsymbol{V}_c$ denote the offsets of its parent and child joints, respectively. The offset of the sphere is approximated as a linear interpolation between these two joint offsets:

$$\boldsymbol{v}_i = \boldsymbol{V}_p \cdot (1 - l_i) + \boldsymbol{V}_c \cdot l_i, \tag{7}$$

where l_i is a coefficient computed by projecting the center c_i^l of sphere g_i onto the corresponding bone:

$$l_i = \frac{\boldsymbol{c}_i^l \cdot \boldsymbol{s}_c}{||\boldsymbol{s}_c||^2}, \text{where } c = J_c(g_i). \tag{8}$$

For a sphere g_i with a desired offset $\boldsymbol{v}_i$, computing the exact offsets of its parent and child joints constitutes a complex inverse kinematics (IK) problem. For efficiency, we heuristically approximate the corresponding joint offsets $\boldsymbol{V}_p$ and $\boldsymbol{V}_c$ as:

$$\boldsymbol{V}_p = \frac{1 - l_i}{l_i^2 + (1 - l_i)^2} \cdot \boldsymbol{v}_i, \boldsymbol{V}_c = \frac{l_i}{l_i^2 + (1 - l_i)^2} \cdot \boldsymbol{v}_i. \tag{9}$$

This approximation satisfies the linear interpolation constraint defined in Eq. 7, along with two boundary conditions: (i) $\boldsymbol{V}_c = 0$ and $\boldsymbol{V}_p = v_i$ for a sphere overlaps with its parent joint ($l_i = 0$); (ii) $\boldsymbol{V}_c = v_i$ and $\boldsymbol{V}_p = 0$ for a sphere overlaps with its child joint ($l_i = 1$).

In practice, a single bone may be associated with multiple spheres; thus, the final offset for a joint j is computed as the average of all mapped offsets from the associated spheres:

$$\boldsymbol{V}_j = \frac{1}{|\mathbb{G}_c(j)| + |\mathbb{G}_p(j)|} \cdot \left(\sum_i \frac{(1 - l_i) \cdot \boldsymbol{v}_i}{l_i^2 + (1 - l_i)^2} + \sum_k \frac{l_k \cdot \boldsymbol{v}_k}{l_k^2 + (1 - l_k)^2} \right), \tag{10}$$

where $|\cdot|$ is the cardinal number of a set, $i \in \mathbb{G}_c(j)$ is the indices of child spheres and $k \in \mathbb{G}_p(j)$ is the indices of parent spheres. While our approximated mappings ignore twist rotations around the bone axis, they remain effective for the retargeting task, as such twist rotations tend to be approximately invariant throughout the retargeting process.

3.3 Preprocessing

A well-initialized solution is critical to the effective optimization of Eq. 6. To initialize the input for our retargeting framework, we begin with motion copying from the source character, where the global root motion is simply copied. However, this naive copying often results in undesirable physical artifacts, such as ground penetrations or floating behavior. To mitigate these issues, we propose a heuristic preprocessing that ensures a physically plausible initialization.

The preprocessing consists of three sequential steps. First, the ground plane is estimated in world space, and the target character is positioned accordingly: it is placed on the ground if the source character is in contact with the ground (e.g., standing poses), or aligned to the same vertical height if airborne (e.g., jumping motions). Second, the target character's joints are heuristically adjusted to restore valid ground contact and eliminate ground penetrations. Finally, we apply the FABRIK [Aristidou and Lasenby 2011] to refine joint positions and ensure bone lengths are strictly preserved. An illustrative example is shown in Fig. 4, and detailed algorithmic formulations are provided in the supplementary material.

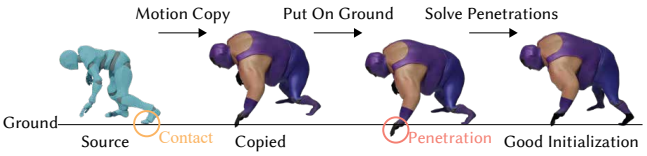

Fig. 4. Illustration of the preprocessing pipeline.

3.4 Rotation-Free Optimization

An intuitive method to optimize Eq. 6 is to update joint rotations through Jacobian matrix [Choi and Ko 1999] or gradient descent frameworks [Cheynel et al. 2025]. However, the efficiency of these methods is constrained by three primary limitations: (i) highly nonlinear gradients, (ii) forward-kinematics (FK) requirements for each iteration, and (iii) constrained step sizes due to potential Jacobian singularities. A more efficient strategy is to produce rotation-free quadratic optimizations [Ho et al. 2010] with additional bone length constraints. The bone length constraints, as difficult quadratic equations, can be integrated into quadratic objectives through Jacobian-based approximation [Ho et al. 2010] or gradient descent frameworks [Tang et al. 2022; Usman et al. 2022]. However, they are not precise enough and may subsequently lead to inaccurate interactions. For precisely preserving bone lengths, we propose an efficient iterative optimization strategy to solve Eq. 6 that takes bone lengths as conditions. Each iteration can be decomposed into four distinct steps: (i) Estimate the expected offsets of spheres linearly. (ii) Map the expected sphere offsets to corresponding joint offsets. (iii) Update the joint positions using a rotation-free algorithm that preserves bone lengths. (iv) Adjust the sphere positions by mapping the updated joint offsets back to sphere offsets.

3.4.1 Estimating Optimization Step. For efficiency, we adopt a heuristic strategy that estimates the update step for each sphere independently, rather than computing the overall derivatives. For a sphere g_i^B with semantic loss $\boldsymbol{\mathcal{L}}_i$, we approximate its expected offset as:

$$\boldsymbol{v}_i = \boldsymbol{\mathcal{L}}_i. \tag{11}$$

Subsequently, the expected offset for joint j is derived through Eq. 10.

The derivation of Eq. 11 begins by rewriting Eq. 5:

$$\boldsymbol{\mathcal{L}}_i = \sum_{k=1}^{N} \left(\boldsymbol{c}_k^B + \boldsymbol{C} \right) \cdot W - \boldsymbol{c}_i^B, \tag{12}$$

where $\boldsymbol{C}$ including the source spheres and $\boldsymbol{W}$ including the weights are constants in the current iteration. Although the semantic loss $\boldsymbol{\mathcal{L}}_i$ is a linear function of both $\boldsymbol{c}_i^B$ and the positions of other spheres $\left\{ \boldsymbol{c}_k^B | k = 1, 2, ...N \wedge k \neq i \right\}$, we approximate the optimization by treating $\boldsymbol{c}_i^B$ as the sole variable and fixing the others. Assuming $\boldsymbol{\mathcal{L}}_i = 0$ after applying offset $\boldsymbol{v}_i$ on sphere g_i, we have:

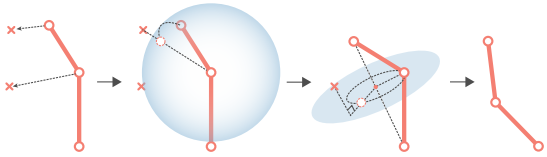

Fig. 5. Illustration of updating a 3-link.

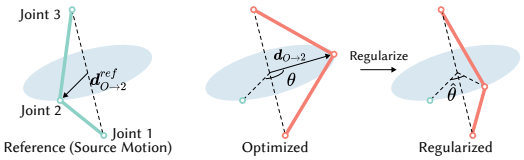

Fig. 6. Illustration of the regularization step that constrains the size of θ.

$$\mathbf{0} = \sum_{k=1}^{N} \left(\boldsymbol{c}_k^B + \boldsymbol{C} \right) \cdot W - \left(\boldsymbol{c}_i^B + \boldsymbol{v}_i \right), \tag{13}$$

which directly leads the approximation in Eq. 11.

3.4.2 Updating Joints. We propose a heuristic algorithm to update joint positions through $\{\boldsymbol{v}_i\}$. We model limbs as linkages, starting at a fixed root (e.g., shoulders) and ending at end-effectors (e.g., hands). For example, the shoulder-elbow-hand chain forms a 3-link, while the pelvis-knee-ankle-toe is a 4-link. A K-link is defined by the joint positions $\left\{ \boldsymbol{p}_j | j = 1, 2, ..., K \right\}$, satisfying the bone length constraint:

$$||\boldsymbol{p}_{j+1} - \boldsymbol{p}_j|| = ||\boldsymbol{s}_{j+1}||, j = 1, 2, ..., K-1. \tag{14}$$

Given a set of desired offsets $\boldsymbol{V}_j$ for each joint $j \leq K$ along a kinematic link (with $\boldsymbol{V}_1 = 0$ for the root), the objective is to compute updated joint positions $\widetilde{\boldsymbol{p}}_j$ such that the displacement $\widetilde{\boldsymbol{p}}_j - \boldsymbol{p}_j$ approximates the target offset $\boldsymbol{V}_j$, while strictly preserving bone lengths. The algorithm for updating K-links($K \geq 3$) proceeds as follows:

(1) End-effector update: Compute the target position of the K-th joint as $\hat{\boldsymbol{p}}_K = \boldsymbol{p}_K + \boldsymbol{V}_K$. Then, project $\hat{\boldsymbol{p}}_K$ onto the spherical surface centered at the $K-1$-th joint with radius $||\boldsymbol{s}_K||$ to obtain the updated joint position.
(2) Intermediate joint updates (applicable for $K \geq 4$): For each joint j from $K-1$ down to 3, compute the target position $\hat{\boldsymbol{p}}_j = \boldsymbol{p}_j + \boldsymbol{V}j$ and project it onto the sphere centered at $\boldsymbol{p}_{j+1}$ with radius $||\boldsymbol{s}_j||$.
(3) Second joint update: After other joints are located, only one degree of freedom is left, i.e., rotating around the axis connecting the root and the third joint. Similar to the previous step, we first calculate a target $\hat{\boldsymbol{p}}_2$, and project it onto the circle to obtain the updated joint location.

Noted that step (2) may produce invalid joint positions $\boldsymbol{p}_j$, causing $||\boldsymbol{p}_j - \boldsymbol{p}_1|| > \sum_{i=2}^{j} ||\boldsymbol{s}_i||$, and needs additional handling. The illustration for common cases $K = 3$ is shown in Fig. 5. More details with the pseudo-code are presented in the supplementary material.

While efficient, the algorithm is less flexible and may struggle to reach certain target positions as the number of joints increases. Given that motion semantics are primarily conveyed through the limbs [Zhang et al. 2023b], we restrict this algorithm to limb joints, leaving the torso adjustments to preprocessing.

3.4.3 Regularization. Human joint motions are subject to biomechanical constraints, which help maintain naturalness and prevent implausible poses. While these constraints can be manually specified as joint rotation limits [Aristidou and Lasenby 2011], manually setting dozens of constraints is time-consuming and impractical. Instead, we propose an efficient strategy that regularizes the retargeted motion by leveraging the source motion, which inherently satisfies such constraints.

For any 3-link chain, we define its *twist* as the rotation of the intermediate joint around the axis linking both ends. Similar to a single bone, the change in the twist of 3 links is small in the retargeting task. This insight inspired us to regularize the optimized result by constraining the twist change θ of 3-link chains:

$$\hat{\theta} = \theta \cdot \left(1 - \sin\left(\frac{\theta}{2} \right) \right). \tag{15}$$

The regularization process, illustrated in Fig. 6, avoids excessive twists while allowing smaller ones, satisfying two boundary conditions: $\hat{\theta} \to \theta$ for $\theta \to 0$ and $\hat{\theta} \to 0$ for $\theta \to \pi$. Instead of the precise rotation-relevant solution, we employ two rotation-free approximations that achieve similar effects with greater efficiency:

(1) Approximate θ as the angle between $\boldsymbol{d}_{O\to 2}^{ref}$ and $\boldsymbol{d}_{O\to 2}$.
(2) Approximate the interpolation between 0 and θ by a linear interpolation between $\boldsymbol{d}_{O\to 2}^{ref}$ and $\boldsymbol{d}_{O\to 2}$.

The detailed pseudo-code is provided in the supplementary material.

3.5 Reconstructing Rotations

After the last optimization iteration, joint rotations are reconstructed from joint positions in two ways, which respectively minimize the global and local twist change of bones. The first strategy, i.e, the one minimizing change of global twists, is performed by a minimal rotation from the motion copied orientation to the optimized orientation:

$$\hat{\boldsymbol{q}}_j^{glo} = \text{between}\left(\boldsymbol{q}_j^{glo} \cdot \boldsymbol{s}_c, \boldsymbol{p}_c - \boldsymbol{p}_j \right) \cdot \boldsymbol{q}_j^{glo}, \tag{16}$$

where the subscript c refers to the child of joint j, $\boldsymbol{s}_j$ is the relative offset of c as mentioned in Sec. 3.1, $\boldsymbol{p}_j, \boldsymbol{p}_c$ are the optimized global joint position, $\boldsymbol{q}_j, \hat{\boldsymbol{q}}_j$ indicate the motion copied and retargeted joint rotation respectively, and between$(A \cdot B)$ is a quaternion operation calculating the rotation from direction A to B.

The second strategy, minimizing the change of local twists, is performed by a minimal rotation in local space:

$$\hat{\boldsymbol{q}}_j^{loc} = \text{between}\left(\boldsymbol{q}_j^{loc} \cdot \boldsymbol{s}_c, \left(\boldsymbol{q}_p^{glo} \right)^{-1} \cdot \left(\boldsymbol{p}_c - \boldsymbol{p}_j \right) \right) \cdot \boldsymbol{q}_j^{loc}, \tag{17}$$

where the subscript p refers to the parent joint of j.

Users can combine the use of the two strategies for joint-wise control over rotation reconstruction. By default, we use the first strategy for the root and the second for the other joints. Pseudo-code is provided in the supplementary material.

3.6 Interaction Control

3.6.1 Attached Objects. Although often overlooked, attached objects—such as weapons—are common for game characters. Our method supports attaching arbitrary objects to end-effectors (e.g., a sword in hand) by treating them as additional bones with uniform spherical representations, as illustrated in Fig. 7. The optimization process flexibly accommodates the dynamic attachment and removal of such objects. Once modeled as spheres, these items are seamlessly incorporated into the semantic-preservation optimization, effectively preventing artifacts such as collisions or penetrations with the character body.

3.6.2 Spatial Interactions. Most interactions with the environment are defined through spatial constraints, such as foot placement or contact with specific locations. These constraints are especially critical in motions involving weapons. For example, consider a tall warrior wielding a spear as the source character and a shorter dwarf as the target; despite the height difference, the spear is expected to strike the same point in space. Our framework integrates spatial interactions seamlessly by directly constraining the expected offsets of joints ($\{\boldsymbol{V}_j\}$) and spheres ($\{\boldsymbol{v}_i\}$).

For a joint-level interaction defined at a global position $\boldsymbol{P}_j$ with joint j, we enforce $\boldsymbol{V}_j = \boldsymbol{P}_j - \boldsymbol{p}_j$ during optimization. Similarly, for a sphere-level interaction defined at $\boldsymbol{P}_i$, we constrain $\boldsymbol{v}_i = \boldsymbol{P}_i - \boldsymbol{c}_i$. These constraints ensure precise control when the targets are within reach. However, due to skeletal differences, some targets may lie outside the reachable region and can only be approximated.

3.6.3 Applications. To reduce the need for cumbersome manual configurations, we implement several common applications using the aforementioned control mechanisms. The first is to avoid penetrations into the ground. We check if any sphere i that penetrates into the ground, then constrain the expected offset by:

$$\boldsymbol{v}_i = -\min\left(\text{height}\left(\boldsymbol{c}_i^B\right) - r_i^B, 0\right) \cdot \boldsymbol{n}_{gnd}, \tag{18}$$

where $\boldsymbol{n}_{gnd}$ is the normal direction of the ground.

The second is to keep footsteps. Footstep is defined by the horizontal offset of the foot joint:

$$\boldsymbol{F}_f = \left(\boldsymbol{p}_{t,f} - \boldsymbol{p}_{t-1,f}\right) - \boldsymbol{n}_{gnd} \cdot \left(\left(\boldsymbol{p}_{t,f} - \boldsymbol{p}_{t-1,f}\right) \cdot \boldsymbol{n}_{gnd}\right), \tag{19}$$

where t indicates frame index, f is a foot joint. We constrain the footstep of the target character by referring to the source character:

$$\boldsymbol{V}_f^B = \boldsymbol{p}_{t-1,f}^B + \boldsymbol{F}_f^A - \boldsymbol{p}_{t,f}^B. \tag{20}$$

The third is to preserve the "hit" point of a weapon. To preserve impact locations for weapon-based interactions, we treat the interactive point (e.g., a spearhead) as a weapon joint. The hit position is preserved by matching the global position of this joint:

$$\boldsymbol{V}_w^B = \boldsymbol{p}_w^A - \boldsymbol{p}_w^B, \tag{21}$$

where w refers to the extra weapon joint.

4 Experiments and Results

4.1 Dataset

We construct our test dataset using characters and animations from Mixamo [Adobe 2025]. To better reflect the characteristics of game scenarios, we carefully select four game-style characters as source characters and eight game-style characters with significant skeletal and geometric differences as target characters. We manually select a subset of 157 game-style animations with minimal artifacts.

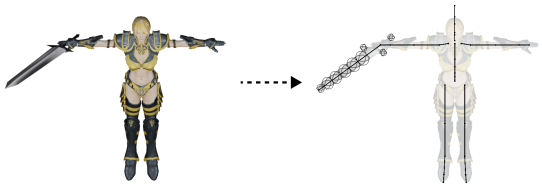

Fig. 7. Attaching a sword to a character by extending the right arm with an extra weapon bone. The sword is represented by sampling spheres.

Table 1. **Comparison of inference speed in experimental environment** †indicates the methods evaluated using GPUs, while all other methods are tested on CPUs.

Method	Platform	Fps↑
ReConForM† [Cheynel et al. 2025]	PyTorch	67
R2ET [Zhang et al. 2023b]	PyTorch	268
MeshRet† [Ye et al. 2024]	PyTorch	30
Ours	Numpy	2,204
Ours	C++	22,431

4.2 Inference Speed

We compare our method with existing geometry-aware retargeting approaches by measuring average per-frame processing time. For ReConForM [Cheynel et al. 2025] and MeshRet [Ye et al. 2024], we adopt GPU-based framerates reported in their respective publications. Other methods are parallelized for optimal CPU usage on an Intel i9-13900K. Despite this, our CPU-only implementation achieves significantly higher speeds. As shown in Tab. 1, our method achieves an order-of-magnitude speedup, making it highly suitable for real-time applications. To reflect the real performance in engineering, we tested the per-frame time cost of our method and the fastest geometry-aware method R2ET, in Unity deployment. Our method costs 0.13ms for each frame, while the R2ET costs 3.92ms. More details about the Unity deployment are in the supplementary material.

4.3 Quantitative Results

Tab. 2 presents comparisons between our method and baseline methods, including state-of-the-art geometry-aware approaches MeshRet [Zhang et al. 2023b], R2ET [Ye et al. 2024] and a skeleton-only method SAN [Aberman et al. 2020]. Our method achieves superior performance in contact preservation (Con.), and demonstrates competitive results on the self-penetration rate (Pen.), which quantifies the proportion of vertices engaged in self-intersections. It is worth noting that relatively higher MSE and MSE^{lc} values do not necessarily imply degraded motion semantic preservation, as MSE measures joint position error with respect to the source motion and fails to capture geometry-level motion semantics. Further discussion can be found in the supplementary material.

Table 2. **Quantitative comparisons with baseline methods. Bold** and underline denote best and second best, respectively.

Methods	MSE ↓	MSE^{lc} ↓	Pen.(%) ↓	Con. ↓
Source	-	-	3.715	-
Copy	0.069	0.045	5.114	0.959
SAN	**0.069**	**0.045**	4.674	1.974
R2ET	0.087	0.048	**4.375**	1.274
MeshRet	0.280	0.072	5.282	0.912
Ours	0.108	0.050	4.915	**0.394**
Ours w/o Regu.	0.108	0.050	4.837	0.349

Table 3. **User study across Semantic Preservation, Motion Quality and Smoothness.**

Methods	Semantic Preservation	Motion Quality	Smoothness
ReConForM	3.219	2.707	3.129
R2ET	3.667	2.624	2.710
MeshRet	3.4	2.912	3.152
Ours	**4.424**	**4.164**	**4.207**

4.4 Qualitative Results

The qualitative results in Figs. 10 and 11 highlight the effectiveness of our approach. As shown in Fig. 11, our method effectively preserves motion semantics, mitigates geometric penetrations, and ensures accurate contact. In contrast, SAN fails to eliminate geometric artifacts as a skeleton-only method. R2ET suffers from inconsistencies between the skeleton-aware and geometry-aware modules, resulting in jitter and unnatural poses. MeshRet is unable to produce satisfactory results in our setting, and we attribute this to the significant domain gap between its motion-captured, photorealistic training dataset and our stylized game characters and motions. Fig. 10 further demonstrates our method's ability to preserve motion semantics with attached objects and to support fine-grained control over environmental interactions—such as the precise positioning of a spearhead or the orientation of a sword—capabilities that remain challenging for current deep learning-based methods.

4.5 User Study

We conducted a user study to evaluate the perceived performance of our method against baseline approaches. A total of 20 participants rated the results of four methods across fourteen motion clips, involving four source characters and eight target characters. Participants were asked to assess each result on a 1-5 scale for Semantic Preservation, Motion Quality, and Smoothness. Further details of the user study are provided in the supplementary material. The results presented in Tab. 3 demonstrate our method's competitiveness across all criteria.

4.6 Ablation Study

4.6.1 Efficiency. We evaluate the computational efficiency of our heuristic joint update method (Sec. 3.4.2) against the standard Jacobian-based optimization. In the Jacobian-based approach, the joint rotations are iteratively updated using the Jacobian matrix, followed by forward kinematics to recompute joint positions. We first compare

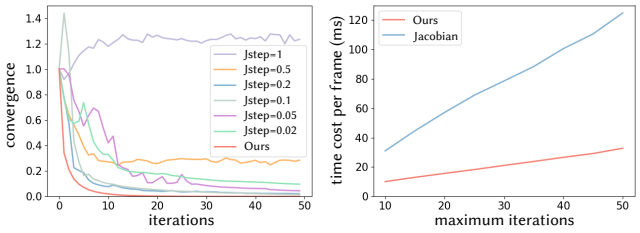

Fig. 8. Comparison between our method and the Jacobian-based approach in terms of convergence speed (left) and per-frame computational cost (right). "Jstep" denotes the optimization step size used in the Jacobian-based method.

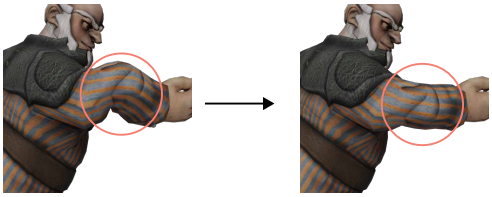

Fig. 9. Visualization of the regularization process, which corrects the erroneous pose shown on the left.

the convergence speed between our method and the Jacobian-based method by measuring the relative change in joint positions per iteration. Specifically, for i-th iteration, we compute the joint position change divided by the change in the first iteration as an indicator of convergence:

$$Conv(i) = \frac{\max_j ||\boldsymbol{p}_j^i - \boldsymbol{p}_j^{i-1}||}{\max_j ||\boldsymbol{p}_j^1 - \boldsymbol{p}_j^0||}, i = 1, 2, ..., maxiters. \quad (22)$$

Smaller values indicate better convergence. We also measure the average time per frame with the maximum iterations ranging from 10 to 50. As shown in Fig. 8, our method converges faster and requires less computation per iteration, resulting in an order-of-magnitude speedup overall. Detailed experiments about Jacobian-based optimization can be found in the supplementary material.

4.6.2 Regularization. We evaluate the qualitative results without applying regularization, as shown in the bottom of Tab. 2. While the unregularized variant achieves slightly higher scores, the lack of biomechanical constraints leads to physically implausible poses and reduced perceptual quality, as demonstrated in Fig. 9.

4.6.3 Sphere Configuration. In the main experiments, we use $N = 40$ spheres for geometry representation. Eight spheres are assigned to the torso (i.e., body and head), and thirty-two spheres are assigned to the limbs, with eight for each limb. An ablation study was conducted to evaluate how the number of spheres affects the speed and motion quality. As shown in Tab. 4, the quality improves as N increases. However, when $N > 40$, the metrics improve little while the speed degrades. Further details of the sphere configuration are provided in the supplementary material.

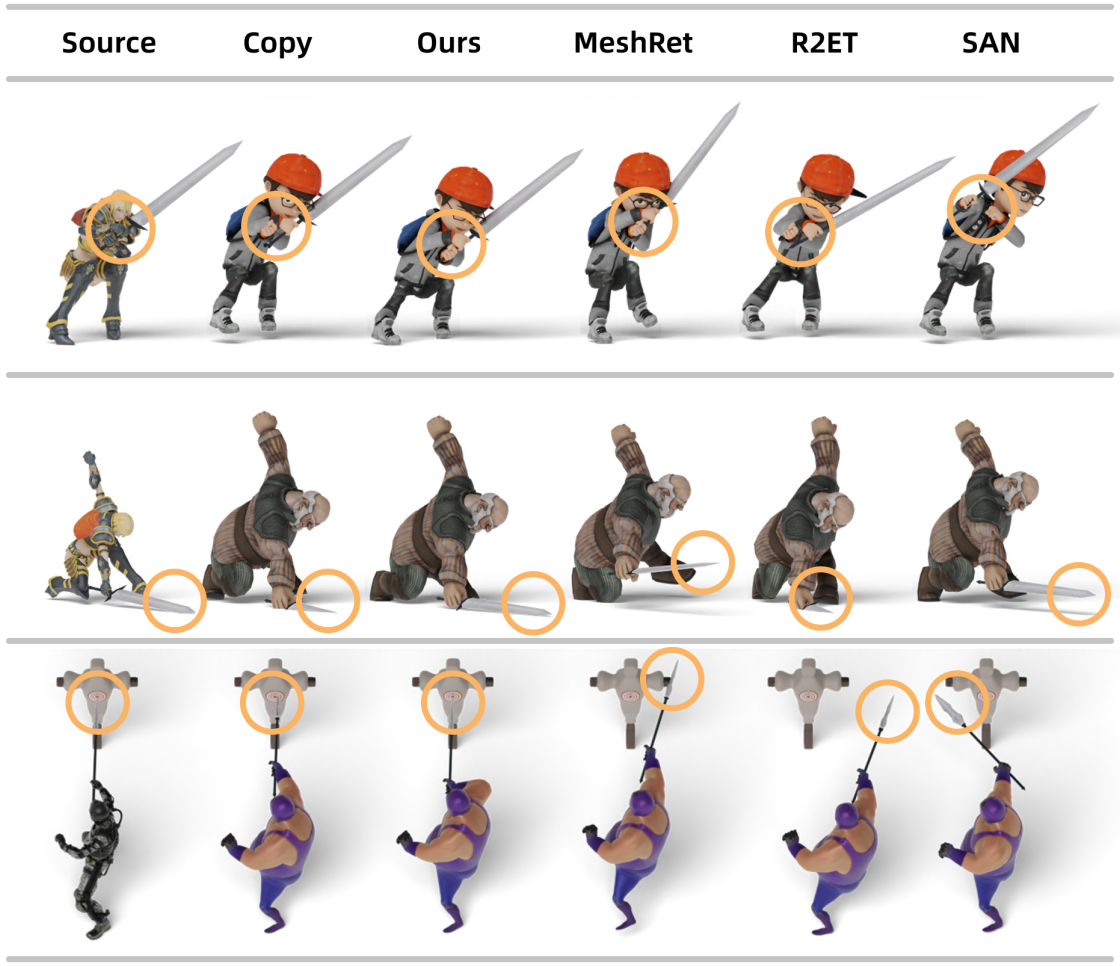

Fig. 10. Qualitative comparison with baseline methods. Our approach preserves motion semantics involving attached objects more effectively and enables precise control over interactions with the environment.

Table 4. **Quality metrics and speed under different sphere configuration.** For each configuration, "A+B" means A spheres assigned to torso and B spheres assigned to limbs.

Configuration	MSE ↓	MSE^{lc} ↓	Pen.(%) ↓	Con. ↓	Fps↑
8+32	0.108	0.050	4.915	0.394	2204
8+40	0.108	0.050	4.901	0.379	2033
16+40	0.108	0.050	4.815	0.416	2062
6+14	0.108	0.050	4.766	0.635	2376
16+40	0.108	0.050	4.856	0.398	1992

5 Limitations and Conclusions

Our method operates under the minimal twist assumption, which holds when the source and target characters share the same skeletal topology and T-pose, as empirically validated in the supplementary material. While effective in typical settings, this assumption may lead to suboptimal performance in extreme cases. Additionally, although the spherical representation enhances efficiency, its geometric approximation makes it less effective at detecting fine-grained self-penetrations, such as those between fingers.

In conclusion, we propose an ultrafast, lightweight online motion retargeting method with joint-level control, capable of real-time performance even in highly resource-constrained environments such as online games. Our approach introduces a semantic-aware objective grounded in a spherical geometry representation, combined with a heuristic, rotation-free optimization strategy that bypasses forward kinematics and Jacobian computations, significantly simplifying the optimization pipeline and yielding an order-of-magnitude speedup. Furthermore, our method is inherently scalable: it supports extensions such as attaching objects and enables precise joint-level control to preserve interaction semantics. Its efficiency and flexibility make it well-suited for real-time applications in interactive game scenarios. In addition, our method can be used as a postprocessing tool for motion generation methods [Tang et al. 2024; Tevet et al. 2023; Wu et al. 2025].

Fig. 11. Qualitative comparison with baseline methods. Our approach effectively maintains the motion semantics, ensures accurate contact preservation and significantly reduces geometric interpenetration.

Acknowledgments

Xiaogang Jin was supported by the National Natural Science Foundation of China (Grant: 62472373) and Key R&D Program of Zhejiang (Grant: 2024C01069).

References

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-aware networks for deep motion retargeting. *ACM Trans. Graph.* 39, 4, Article 62 (Aug. 2020), 14 pages. doi:10.1145/3386569.3392462

Adobe. 2025. *Miaxmo.* https://www.mixamo.com/

Rami Ali Al-Asqhar, Taku Komura, and Myung Geol Choi. 2013. Relationship descriptors for interactive motion adaptation. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Anaheim, California) *(SCA '13)*. Association for Computing Machinery, New York, NY, USA, 45–53. doi:10.1145/2485895.2485905

Andreas Aristidou and Joan Lasenby. 2011. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models* 73, 5 (2011), 243–260. doi:10.1016/j.gmod.2011.05.003

Jean Basset, Stefanie Wuhrer, Edmond Boyer, and Franck Multon. 2019. Contact Preserving Shape Transfer For Rigging-Free Motion Retargeting. In *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games* (Newcastle upon Tyne, United Kingdom) *(MIG '19)*. Association for Computing Machinery, New York, NY, USA, Article 24, 10 pages. doi:10.1145/3359566.3360075

Sourav Biswas, Kangxue Yin, Maria Shugrina, Sanja Fidler, and Sameh Khamis. 2021. Hierarchical Neural Implicit Pose Network for Animation and Motion Retargeting. arXiv:2112.00958 [cs.CV] https://arxiv.org/abs/2112.00958

Théo Cheynel, Thomas Rossi, Baptiste Bellot-Gurlet, Damien Rohmer, and Marie-Paule Cani. 2025. ReConForM : Real-time Contact-aware Motion Retargeting for more Diverse Character Morphologies. arXiv:2502.21207 [cs.GR] https://arxiv.org/abs/2502.21207

Kwang-Jin Choi and Hyeong-Seok Ko. 1999. On-line motion retargetting. In *Proceedings. Seventh Pacific Conference on Computer Graphics and Applications (Cat. No.PR00293)*. 32–42. doi:10.1109/PCCGA.1999.803346

Michael Gleicher. 1998. Retargetting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 33–42. doi:10.1145/280814.280820

Edmond S. L. Ho, Taku Komura, and Chiew-Lan Tai. 2010. Spatial relationship preserving character motion adaptation. *ACM Trans. Graph.* 29, 4, Article 33 (July 2010), 8 pages. doi:10.1145/1778765.1778770

Edmond S. L. Ho and Hubert P. H. Shum. 2013. Motion adaptation for humanoid robots in constrained environments. In *2013 IEEE International Conference on Robotics and Automation*. 3813–3818. doi:10.1109/ICRA.2013.6631113

Hanyoung Jang, Byungjun Kwon, Moonwon Yu, Seong Uk Kim, and Jongmin Kim. 2018. A variational U-Net for motion retargeting. In *SIGGRAPH Asia 2018 Posters* (Tokyo, Japan) *(SA '18)*. Association for Computing Machinery, New York, NY, USA, Article 1, 2 pages. doi:10.1145/3283289.3283316

Inseo Jang, Soojin Choi, Seokhyeon Hong, Chaelin Kim, and Junyong Noh. 2024. Geometry-Aware Retargeting for Two-Skinned Characters Interaction. *ACM Trans. Graph.* 43, 6, Article 203 (Nov. 2024), 17 pages. doi:10.1145/3687962

Taeil Jin, Meekyung Kim, and Sung-Hee Lee. 2017. Motion retargeting to preserve spatial relationship between skinned characters. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '17)*. Association for Computing Machinery, New York, NY, USA, Article 25, 2 pages. doi:10.1145/3099564.3106647

Yeonjoon Kim, Hangil Park, Seungbae Bang, and Sung-Hee Lee. 2016. Retargeting Human-Object Interaction to Virtual Avatars. *IEEE Transactions on Visualization and Computer Graphics* 22, 11 (2016), 2405–2412. doi:10.1109/TVCG.2016.2593780

Jehee Lee and Sung Yong Shin. 1999. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 39–48. doi:10.1145/311535.311539

Jongin Lim, Hyung Jin Chang, and Jin Young Choi. 2019. PMnet: learning of disentangled pose and movement for unsupervised motion retargeting. In *Proceedings of the 30th British Machine Vision Conference (BMVC 2019)*. British Machine Vision Association, BMVA. 30th British Machine Vision Conference (BMVC 2019) ; Conference date: 09-09-2019 Through 12-09-2019.

Zhiguang Liu, Antonio Mucherino, Ludovic Hoyet, and Franck Multon. 2018. Surface based motion retargeting by preserving spatial relationship. In *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games* (Limassol, Cyprus) *(MIG '18)*. Association for Computing Machinery, New York, NY, USA, Article 7, 11 pages. doi:10.1145/3274247.3274507

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.* 34, 6, Article 248 (Oct. 2015), 16 pages. doi:10.1145/2816795.2818013

Lucas Mourot, Ludovic Hoyet, François Le Clerc, and Pierre Hellier. 2023. HuMoT: Human Motion Representation using Topology-Agnostic Transformers for Character Animation Retargeting. arXiv:2305.18897 [cs.GR] https://arxiv.org/abs/2305.18897

Daniele Reda, Jungdam Won, Yuting Ye, Michiel van de Panne, and Alexander Winkler. 2023. Physics-based Motion Retargeting from Sparse Inputs. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 3, Article 33 (Aug. 2023), 19 pages. doi:10.1145/3606928

Zhong Ren, Rui Wang, John Snyder, Kun Zhou, Xinguo Liu, Bo Sun, Peter-Pike Sloan, Hujun Bao, Qunsheng Peng, and Baining Guo. 2006. Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Trans. Graph.* 25, 3 (July 2006), 977–986. doi:10.1145/1141911.1141982

Xiangjun Tang, He Wang, Bo Hu, Xu Gong, Ruifan Yi, Qilong Kou, and Xiaogang Jin. 2022. Real-time controllable motion transition for characters. *ACM Trans. Graph.* 41, 4, Article 137 (July 2022), 10 pages. doi:10.1145/3528223.3530090

Xiangjun Tang, Linjun Wu, He Wang, Yiqian Wu, Bo Hu, Songnan Li, Xu Gong, Yuchen Liao, Qilong Kou, and Xiaogang Jin. 2024. Decoupling Contact for Fine-Grained Motion Style Transfer. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) *(SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 54, 11 pages. doi:10.1145/3680528.3687609

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. 2023. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=SJ1kSyO2jwu

Ben Usman, Andrea Tagliasacchi, Kate Saenko, and Avneesh Sud. 2022. MetaPose: Fast 3D Pose From Multiple Views Without 3D Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6759–6770.

Ruben Villegas, Duygu Ceylan, Aaron Hertzmann, Jimei Yang, and Jun Saito. 2021. Contact-Aware Retargeting of Skinned Motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 9720–9729.

Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural Kinematic Networks for Unsupervised Motion Retargetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8639–8648.

Haoyu Wang, Shaoli Huang, Fang Zhao, Chun Yuan, and Ying Shan. 2023. HMC: Hierarchical Mesh Coarsening for Skeleton-free Motion Retargeting. arXiv:2303.10941 [cs.CV] https://arxiv.org/abs/2303.10941

Rui Wang, Kun Zhou, John Snyder, Xinguo Liu, Hujun Bao, Qunsheng Peng, and Baining Guo. 2006. Variational sphere set approximation for solid objects. *Vis. Comput.* 22, 9 (Sept. 2006), 612–621. doi:10.1007/s00371-006-0052-0

Linjun Wu, Xiangjun Tang, Jingyuan Cong, He Wang, Bo Hu, Xu Gong, Songnan Li, Yuchen Liao, Yiqian Wu, Chen Liu, and Xiaogang Jin. 2025. Semantically Consistent Text-to-Motion with Unsupervised Styles. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 56, 10 pages. doi:10.1145/3721238.3730641

Nannan Wu, Dongliang Zhang, Zhigang Deng, and Xiaogang Jin. 2018. Variational Mannequin Approximation Using Spheres and Capsules. *IEEE Access* 6 (2018), 25921–25929. doi:10.1109/ACCESS.2018.2837013

Zijie Ye, Jia-Wei Liu, Jia Jia, Shikun Sun, and Mike Zheng Shou. 2024. Skinned Motion Retargeting with Dense Geometric Interaction Perception. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 125907–125934. https://proceedings.neurips.cc/paper_files/paper/2024/file/e3ed7183233afa8e5485ff8f6c3f18b1-Paper-Conference.pdf

Haodong Zhang, Zhike Chen, Haocheng Xu, Lei Hao, Xiaofei Wu, Songcen Xu, Zhensong Zhang, Yue Wang, and Rong Xiong. 2024. Semantics-aware Motion Retargeting with Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2155–2164.

Jiaxu Zhang, Junwu Weng, Di Kang, Fang Zhao, Shaoli Huang, Xuefei Zhe, Linchao Bao, Ying Shan, Jue Wang, and Zhigang Tu. 2023b. Skinned Motion Retargeting With Residual Perception of Motion Semantics & Geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 13864–13872.

Yunbo Zhang, Deepak Gopinath, Yuting Ye, Jessica Hodgins, Greg Turk, and Jungdam Won. 2023a. Simulation and Retargeting of Complex Multi-Character Interactions. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 65, 11 pages. doi:10.1145/3588432.3591491