

EquiFusion: Kinematics-Agnostic Human Motion Prediction via Equivariant Latent Diffusion

Cecilia Curreli^{1,2} Florian Hofherr^{1,2} Dominik Muhle^{1,2}
 Abhishek Saroha^{1,2} Riccardo Marin^{1,2} Daniel Cremers^{1,2}

¹ Technical University of Munich, Munich, Germany
² Munich Center for Machine Learning, Munich, Germany



Fig. 1: EquiFusion. We introduce the first model for stochastic human motion prediction that generalizes to unseen skeleton parameterization, i.e. *kinematics*. While previous methods require a trained instance for each dataset or better kinematics, with a single model we unlock training on multiple datasets and inference on motion parametrized with different kinematics. EquiFusion is the first SHMP model to handle **zero-shot** novel kinematics and occluded limbs without explicit training, while achieving state-of-the-art results on established benchmarks.

Abstract. Existing Stochastic 3D Human Motion Prediction models are fundamentally constrained by hard-coding the skeleton kinematics, severely limiting generalization, preventing cross-dataset training, and requiring complex data retargeting. We introduce *EquiFusion*, the first kinematics-agnostic model to solve this bottleneck, implementing a latent diffusion model with a permutation equivariant architecture. EquiFusion treats the kinematics’ connectivity as an explicit input parameter, ensuring its internal computations are inherently agnostic to joint ordering and graph structure. This novel design enables truly cross-dataset generalization to unseen kinematics and unlocks novel zero-shot directions, such as motion prediction from partial or occluded observations

and targeted limb generation. EquiFusion achieves state-of-the-art results on major benchmarks, being up to 75% more compact than previous kinematics-specific methods, while achieving faster training and inference. EquiFusion thus establishes a new, flexible standard for robust human motion prediction. Model and training code available on our [project page](#).

1 Introduction

Predicting future human motion from past observations is a core component of human intelligence and a prerequisite for mature spatial AI. Due to the inherent ambiguity of human intent, the field has shifted from deterministic toward Stochastic Human Motion Prediction (SHMP), which models a distribution of diverse, physically plausible futures from an observation, with applications including human-robot collaboration, autonomous navigation, and augmented telepresence. While recent advances in generative models [13, 42, 98] have further popularized SHMP probabilistic formulations [9, 18, 23, 108], a fundamental bottleneck remains: kinematics rigidity. Diverse datasets (e.g., Human3.6M [46], AMASS [76], and more recently Nymeria [75]) often employ different motion-capturing technologies, resulting in diverse skeleton parametrizations *i.e.* *kinematics*. Motion retargeting [8, 43, 57], *i.e.*, converting a motion to a different kinematics parametrization, is often not possible without introducing errors and drifts [66, 116, 117]. The downstream impact on SHMP methods is tremendous. Previous works [9, 18, 23, 26, 78, 108, 136] have hard-coded kinematics as a network structural prior, resulting in a plethora of skeleton-specific networks. Training a network for every different kinematics is impractical, inefficient, and does not generalize to new skeletal configurations.

We break these dataset-specific boundaries by addressing settings involving heterogeneous kinematic structures. We propose two *zero-shot kinematics* inference tasks for SHMP: predicting motion for (i) full-body kinematics unseen at training time, and (ii) kinematics with occlusions or missing parts (*i.e.*, partial observations). Both these tasks address real-world use cases and important milestones toward mature spatial AI and foundation human motion models. To date, this has been unaddressed in the SHMP literature, while deterministic HMP solutions (*e.g.*, hand-crafted training to address specific occlusions or skeletal conversions [21, 127]) are impractical and require manual engineering. Can we provide such flexibility for SHMP?

In this paper, we answer this question by proposing the first kinematics-agnostic method for SHMP. To handle arbitrary kinematics chains, a model’s learned weights must be independent of the cardinality of the joint set. Most existing works violate this principle by employing joint-dependent weights or temporal-only transformers that treat joints as feature channels. We identify permutation equivariance as the key mathematical property to achieve this independence. We thus present EquiFusion, the first kinematics-agnostic approach to SHMP, implemented as an equivariant latent diffusion model that explicitly

incorporates the kinematics connectivity as a model input. By design, we achieve robustness to partial and unseen kinematics without requiring explicit exposure to such corruptions during training, and we demonstrate this through extensive experiments. Our design allows us to break the single-dataset barrier, enabling a single model to be trained simultaneously on heterogeneous datasets such as AMASS and Nymeria for the first time, unlocking the future potential of “foundation models” in human motion. By leveraging the inductive bias of equivariance with formal guarantee [12, 73], EquiFusion achieves state-of-the-art performance on standard benchmarks even when trained on single-dataset distributions while being 75% *smaller* than the closest competitor [23], and significantly faster in training and inference. These improvements are particularly remarkable in zero-shot scenarios, where our model demonstrates a performance boost of at least 25% across all metrics and up to 70% in terms of fidelity. Our contributions are:

- We formalize and analyze for the first time the challenge of zero-shot kinematics SHMP, introducing two novel tasks of high relevance for the real-world setting, while opening new applications and use cases: training on datasets having multiple kinematics, zero-shot handling of occlusion, and zero-shot limb generation without ad-hoc training.
- We identify joint-ordering permutation equivariance as a key property to enable kinematics-agnostic SHMP and propose a principled architecture that inherently satisfies this requirement.
- We present EquiFusion, which not only achieves zero-shot kinematics by design but outperforms existing SHMP methods in extensive experiments. Our method achieves state-of-the-art results on standard benchmarks (AMASS, Human3.6M) with 75% fewer parameters than the closest competitor. Additionally, our model’s complexity remains invariant to the number of joints, kinematics, or datasets, offering superior scalability.

2 Related Works

2.1 Human Kinematics Representations

In SHMP, human motions are represented as the temporal evolution of body joints; we refer to them as skeleton kinematics $\mathcal{K}$, which are generally inherited from the Motion Capture (MoCap) system used to register the human movement. Hence, depending on the MoCap system, the kinematic system can differ in the number and positions of the joints, resulting in a plethora of different formats. Examples are: the AMASS [76] dataset uses the SMPL [72] format, and is parametrized by 22 joints; H36M [46] kinematics comprises 17 joints and does not include the foot joints; Nymeria [75] has been captured by META’s ARIA devices [29] and parametrized with XSens [1] by 23 joints, exhibiting different spine and hip modeling from the aforementioned formats (see Fig. 1 for visualizations). With the increasing availability and affordability of wearable sensors [29, 41, 94, 120], new kinematic configurations are expected to emerge [33, 132], making it crucial to address multiple kinematics, since differences in kinematic structure hinder data combination during training [29].

2.2 Stochastic Human Motion Prediction

Human Motion Prediction (HMP) aims to predict future motion from past observations. While deterministic HMP [22, 60, 62] forecasts a single future, Stochastic HMP (SHMP) predicts multiple plausible ones. SHMP is an ill-posed problem with multiple solutions, which has been modeled by different generative approaches involving GANs [10, 54, 69], VAEs [15, 38, 78, 119, 131], diverse sampling [26, 130, 136], and lately denoising diffusion models [9, 18, 23, 100, 108, 124]. All such models are trained and evaluated independently for each kinematics $\mathcal{K}^i$, which means that each requires ad hoc training, computational waste, hyperparameter engineering, and lacks generalization at inference time to novel kinematics $\mathcal{K}^j$ with $j \neq i$, or occluded data (*i.e.*, partial kinematics). Digesting motions in different kinematics formats has been tackled in other computer vision tasks, such as human pose estimation from video [104], 2D-to-3D keypoints lifting [24], unconditional animal motion generation [34], motion classification [57], or character animation via text [45, 68]. Occluded motions and training on multiple kinematics have already been investigated by deterministic HMP, but to the best of our knowledge, only explicitly, by directly training for occlusions [21, 127] or for multiple skeletons without supporting zero-shot inference on new kinematics [103]. Yet, no method has been proposed for SHMP, which is the core of our work and a mandatory stage towards foundation models for SHMP. We provide an extended discussion on existing methods in the Secs. A.2 to A.4. We will build our framework on top of a newly designed permutation equivariant diffusion model. Diffusion methods equivariant to SE(3) or permutation groups have provided promising results for the task of drug molecule generation [55, 88, 105], but they haven’t found broad application in computer vision, especially not in HMP or SHMP to date. A reasonable alternative to enable flexibility of existing approaches would be solving for skeleton retargeting, and we discuss such techniques in the next section.

2.3 Kinematics Conversion and Motion Retargeting

As current SHMP approaches do not support inference on kinematics $\mathcal{K}^j$ different from the training kinematics $\mathcal{K}^i$, the only way to allow inference on $\mathcal{K}^j$ is to first convert the input motion from kinematics $\mathcal{K}^j$ to the same motion with $\mathcal{K}^i$. This conversion challenge is commonly addressed as *skeleton retargeting*. The large literature on this topic ranges from robotics [27, 31, 135] to computer graphics [8, 44, 57, 64] and specializes in isomorphic [3, 19, 27, 30, 37, 48, 56, 66, 86, 87, 109, 116, 117], homeomorphic [2, 44, 137], and non-homeomorphic graphs [17, 43, 49, 51, 57, 63, 65, 85, 121]. For the SHMP use case, only *non-homeomorphic* techniques can be considered, since kinematics can have different end-effectors and a varying number of joints. As not all approaches are applicable to SHMP’s data structure, in this paper, we consider the retargeting approach from Holden *et al.* [43], which is already established in SHMP by HumanMAC [18]. More recent approaches for non-homeomorphic retargeting on robotics and character animation are not applicable to our case. These require joint rotations of end-effectors and reference poses [57], skinning [65],

meshes [101, 121], or both [49] – even without the skeleton itself [65, 121] – which are not always available in SHMP. Or, they present a too high reconversion error (ca. 100mm [85] or 90mm [17], in both cases, code is not available). We discuss retargeting approaches and each graph category in detail in Sec. A.1, and here we maintain a high-level contextualization, taking as an example the AMASS and H36M skeletons depicted in Fig. 1. These kinematics are non-homeomorphic to each other, so the conversion is not bijective and causes: **(a) Error accumulation**, as converting H36M (17 joints) to AMASS (22 joints) requires adding non-existent feet (with errors of 22.18 mm or 2.27 mm depending on the conversion direction), **(b) Distribution shift**: additionally, this kinematics conversion shifts the input to a distribution that differs from the one seen by the SHMP models at train time, resulting in additional noise for the models. As SHMP approaches achieve precision up to 70 mm, the additional retargeting error is not negligible.

2.4 Human Pose Representations

Another relevant aspect of SHMP is the motion parametrization. In SHMP, motion sequences are conventionally represented as trajectories of 3D joint coordinates in Euclidean space [9, 26, 78, 119, 124, 136]. While such representation is flexible and a natural output of upstream pipelines [16, 71, 106, 125], *e.g.* human tracking in videos [90, 95, 99, 141], it is under-constrained and allows non-realistic predictions. This often results in limb stretching or jitter, requiring subsequent SHMP approaches to directly measure [23] and address [9, 23, 26] this issue. Instead of making the model learn from data what we already know *i.e.* human bones have fixed length, we propose to include this prior directly in the parametrization. Intuitively, consistent limb lengths are guaranteed by design in rotation-based representation, where the joint positions are expressed as angles from a canonical pose, and the bone lengths are fixed along a sequence. Among the many possible representations for rotations [35], some common choices are quaternions [102] or axis-angles [59, 72, 110, 111] as in SMPL [72]. However, rotations also tie the pose representation to a specific skeleton kinematic chain and rest pose, which can cause instabilities [9]. In this work, we adopt a spatial representation, where every joint is encoded by the relative direction from its parent, *i.e.* its *bone direction*. In this way, predictions are guaranteed to have perfectly consistent bones by design, and networks do not have to learn this additional prior. This representation is general and can be applied to both any SHMP kinematics and any SHMP method.

3 Methodology

3.1 Problem Definition

In stochastic human motion prediction (SHMP), a motion sequence is represented as a trajectory $\mathbf{M} \in \mathbb{R}^{T \times J \times 3}$, describing the temporal evolution of J body joints over T frames. Given an observed motion $\mathbf{X} = \mathbf{M}_{0:T_P} \in \mathbb{R}^{T_P \times J \times 3}$

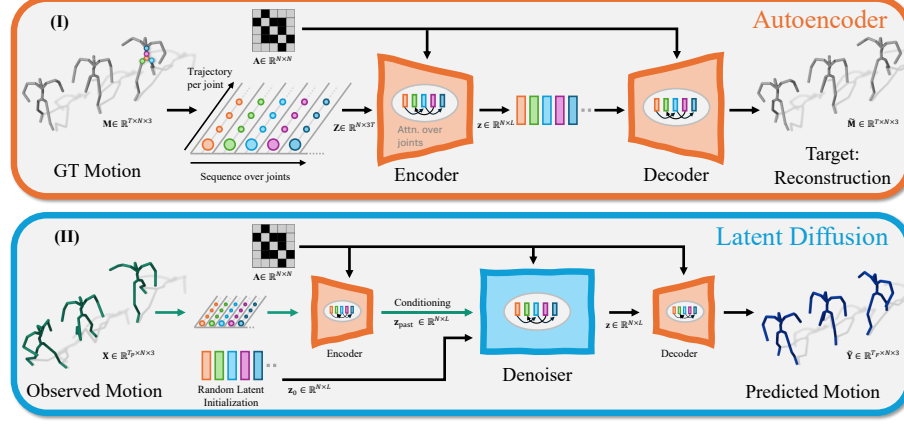


Fig. 2: EquiFusion is the first skeleton-agnostic model for SHMP, implemented as a novel equivariant latent diffusion model that adds the adjacency matrix A of the motion kinematics as input. (I) A transformer autoencoder learns a latent space z equivariant to joint order permutation. (II) A denoiser predicts future motion in this latent domain conditioned on the observed past X .

of length T_P , the objective is to generate multiple plausible future trajectories $\hat{Y} \in \mathbb{R}^{T_F \times J \times 3}$ over a prediction horizon of T_F frames. Each motion sequence is described with respect to a kinematic structure $\mathcal{K}$, as already discussed in Sec. 2.1. Formally, we define a kinematics $\mathcal{K} := (V_{\tau}, E)$ as an undirected graph with semantically labeled vertices V_{τ} and edges E . We refer to nodes as joints and edges as limbs or bones, and we define the cardinality $|\mathcal{K}|$ equal to the number of joints. Each vertex $v_{\tau,i}$ represents a joint with a connected semantic convention label that distinguishes among joint types and kinematics conventions. The kinematic configuration $\mathcal{K}$ is fixed within each dataset across all motion sequences, so we usually refer to a kinematics with the name of the dataset: *e.g.* $\mathcal{K}^A$ for AMASS [76], $\mathcal{K}^H$ for H36M [46], and $\mathcal{K}^N$ for Nymeria [75], where different individuals in the same dataset have the same kinematics $\mathcal{K}$ but different bone lengths. Typically, models are evaluated on the test set of the dataset they were trained on, *in-domain*, and within the same kinematics. The popular mesh SMPL model, which underlies the AMASS kinematics, has led to smaller data collections [80, 113] that share the kinematics $\mathcal{K}^A$. These datasets are occasionally used for evaluation on out-of-distribution motions, i.e., zero-shot motion for SHMP. For the first time in SHMP, we investigate **zero-shot kinematics**, the case where inference kinematics differ from those seen at training. While both evaluations are theoretically independent, the nature of the datasets leads to zero-shot kinematics, which often implies zero-shot motion. Zero-shot kinematics also covers *partial* kinematics, representing limbs that are occluded or missing due to physical impairments. While partiality is highly relevant for real-world applications, data collection is not straightforward, and there are no specific SHMP datasets to date. Partial motions are thus usually investigated by masking limbs of motions parametrized with existing full-body kinematics.

3.2 Kinematics-Agnostic SHMP

While kinematics-agnostic models have been investigated in other domains, such as character animation from text [34, 68], motion classification [57], and robotics [7], the problem of addressing multiple skeleton kinematics has not been addressed or formalized in SHMP so far. Existing SHMP approaches assume a single, fixed skeleton kinematics inherited from the training dataset [9, 18, 23, 26, 36, 44, 108, 136]. Instead, we are interested in a model that natively supports 1) training on multiple kinematics, and 2) inference on novel kinematics, partial or full-body, not seen at training time, i.e., in a zero-shot kinematics setting. We call such a model *kinematics-agnostic*.

Formally, let $\mathbb{K}_{\text{train}} = \{\mathcal{K}^1, \mathcal{K}^2, \dots, \mathcal{K}^S\}$ be the set of kinematic settings seen at training time by a method f , S the number of train skeletons, and $\mathbb{K}_{\text{test}}$ the set of kinematics used for evaluation. When we perform evaluation on a motion under the kinematic setting $\mathcal{K}' \in \mathbb{K}_{\text{test}}$ with $\mathbb{K}_{\text{train}} \cap \mathbb{K}_{\text{test}} = \emptyset$, we regard this as a zero-shot scenario. Previous SHMP approaches have been limited to $N = 1$ and $\mathbb{K}_{\text{train}} = \mathbb{K}_{\text{test}}$ due to their kinematics-specific design decisions, requiring a trained instance for each kinematics. While zero-shot kinematics may be achieved at least in some cases through overly convoluted engineering, we advocate for a method that achieves zero-shot kinematics by design.

Intuitively, the core requirement is that the model must accommodate kinematic chains with an arbitrary number of joints. We formalize this as a constraint on the learnable parameters Θ .

Lemma 1 *To handle arbitrarily sized kinematic chains $\mathcal{K}$, the learned parameters Θ of a model cannot be dependent on the number of joints i.e. the cardinality of the input $\mathcal{K}$:*

$$\frac{d|\Theta|}{dJ} = 0 \quad \text{i.e.} \quad |\Theta| \in O(1) \quad (1)$$

Indeed, in the trivial case $\mathbf{W} \in \mathbb{R}^{J \times J}$, the weights do not generalize to any new $|\mathcal{K}'| > J$. And generally, allocating independent parameters $\mathbf{W}_j$ for each joint $j \in 1 \dots J$ lets the parametrization Θ grow with J and change whenever the kinematic chain changes, violating the Lemma. Yet previous works in both deterministic [62, 79, 139] and stochastic HMP [23, 102, 108] intentionally learn joint-dependent weights to extract strong dataset- [139] and kinematics-specific [23, 102] priors, and thus cannot satisfy Lemma 1 (we prove this by counterexample for each architecture class in Sec. B.2).

We recognize that this requirement is naturally fulfilled by models f that are permutation equivariant $f(\mathbf{P}\mathbf{X}) = \mathbf{P}f(\mathbf{X})$ with respect to arbitrary joint reordering $\mathbf{P} \in \mathbb{R}^{J \times J}$.

Theorem 2. *Let $o(\mathbf{X}) = \mathbf{W}\mathbf{X}\mathbf{G}$ be a general network operation for feature extraction on an input $\mathbf{X} \in \mathbb{R}^{J \times F}$. If $o(\mathbf{X})$ is permutation equivariant under joint reordering, i.e. $\mathbf{P}o(\mathbf{X}) = o(\mathbf{P}\mathbf{X})$ for any permutation $\mathbf{P} \in \mathbb{R}^{J \times J}$, then the number of learned parameters $|\Theta|$ is constant in J , and Lemma 1 is satisfied.*

In other words, weights must be shared across all joints, and the model must behave consistently under any reordering of these instances. Permutation equivariance is therefore sufficient, though not necessary, for kinematics-agnosticism. We report the full proof in Sec. C.1. Since equivariance is preserved under composition, a network f built entirely from equivariant operations o is end-to-end permutation equivariant. With this motivation, we decide to implement our kinematics-agnostic solution as a permutation equivariant model. While we already mentioned that previous work do not fulfill the Lemma, we also prove numerically and mathematically in Sec. C.4 that their architectural designs are not equivariant. We present an intuitive high level explanation on why this is the case in Sec. B.1 and in the next section.

3.3 EquiFusion

Overview. Based on the previously presented findings on permutation equivariance fulfilling the premise of a kinematics-agnostic model, we implement EquiFusion as an equivariant latent diffusion model (Fig. 2). We design a novel end-to-end equivariant framework consisting of (i) an autoencoder mapping motion sequences $\mathbf{M}$ to and from the latent space $\mathbf{z} \in \mathbb{R}^{J \times L}$, and (ii) a denoiser that predicts future motions in the latent domain $\mathbf{z}_\theta$ conditioned on the embedding $\mathbf{z}_{past} = \mathbf{e}(\mathbf{X})$ of the input past $\mathbf{X}$.

Equivariant Latent Diffusion The generative process of diffusion models [42] is known to be computationally expensive in input space [28, 91]. To gain in efficiency we operate in a lower-dimensional latent space [9, 23] and opt for latent diffusion models (LDM) [98]. While previous approaches learn a mapping $f_{\mathcal{K}}(\mathbf{X}) = \tilde{\mathbf{Y}}$ for a fixed skeletal kinematics $\mathcal{K}$, we support operations on different kinematics out-of-the-box, by making the connectivity $\mathbf{A}$ of the kinematics $\mathcal{K}$ an explicit input to the model. We thus design a novel framework architecture that is end-to-end formally permutation equivariant w.r.t. its inputs:

$$f(\mathbf{X}, \mathbf{A}) = \tilde{\mathbf{Y}}, \quad \text{with} \quad f(\mathbf{P}\mathbf{X}, \mathbf{P}\mathbf{A}\mathbf{P}^\top) = \mathbf{P}f(\mathbf{X}, \mathbf{A}). \quad (2)$$

Equivariance in LDMs requires an equivariant denoiser and a latent space that preserves input permutations [67, 112, 118]. Although this space can be learned non-deterministically (e.g., via VAEs [98]), we adopt a deterministic approach to improve training stability [133]. At inference, new latent variables are sampled from a univariate Gaussian distribution. Since this sampling is i.i.d., sample-wise equivariance is guaranteed only in a deterministic setting where the noise is fixed and permuted accordingly. Indeed, in generative models, permutation equivariance holds at a distribution level. We provide a detailed discussion in Sec. C.3. Differently from previous LDM in SHMP [9, 23], we implement the autoencoder as a transformer rather than a recurrent network, gaining in inference speed. We follow the training paradigm of LDM [98] adapted to SHMP [9, 23]. Further details and equations in Sec. D.

EquiFusion’s Architecture. In looking for a permutation equivariant (PEQ) architecture, the reader may already be thinking that Graph Convolution Networks (GCN) [52] naturally fulfill the requirement [50]. However, while many SHMP models are based on GCN [9, 23, 62, 102, 108], no one of them enjoys PEQ. The reason is that independently extracted features are not aggregated according to the connectivity of the graph [52], but according to learned weights $\mathbf{W} \in \mathbb{R}^{J \times J}$ that explicitly depend on the number and position of joints [23, 62, 96]. Such practice in deterministic [139] and SHMP [23, 102, 108] showed advantages in leveraging dataset-specific priors (*e.g.* action [139] or joint-specific [23, 102]). However, such implicit bias is detrimental in our case. We advocate for a kinematics-agnostic formulation aligning with [52]. Specifically, for an initial input $\mathbf{Z} \in \mathbb{R}^{J \times 3T}$, we implement a convolution c as:

$$c(\mathbf{Z}, \mathbf{A}) = \mathbf{W}\mathbf{Z}\mathbf{G}_1 + \mathbf{Z}\mathbf{G}_0 + \mathbf{b}, \quad \text{with} \quad \mathbf{W} = \mathbf{D}^{-1}\mathbf{A} \quad (3)$$

where the weights $\mathbf{G}_0, \mathbf{G}_1 \in \mathbb{R}^{F_i \times F_o}$ learn to extract features for a joint itself or its neighbours respectively, $\mathbf{b} \in \mathbb{R}^{F_o}$ is learned, $\mathbf{D}$ the normalizing degree matrix [52] of $\mathbf{A}$, F_i and F_o are the input and output feature dimensions. This operation fulfills equivariance not only with respect to a vector $\mathbf{X}$, but also with respect to the input matrix $\mathbf{A}$ [112], which excludes otherwise compelling layers [134]. We want to consider the skeleton graph on a global scale in addition to the local scale of Eq. (3), and do so via Graph Attention (GAT) [115] or multi-head self-attention [114] with joints as token dimension.

$$\text{Att}(\mathbf{Z}, \mathbf{A})_h = \text{softmax} \left(c_Q(\mathbf{Z}, \mathbf{A}) c_K(\mathbf{Z}, \mathbf{A})^\top / \sqrt{F_o} \right) c_V(\mathbf{Z}, \mathbf{A}) \quad (4)$$

Particularly, we employ the operation in Eq. (3) to compute $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ for attention on each of the h heads. Building on these operations, we design a novel end-to-end equivariant architecture. Noticeably, by definition, attention layers [114] without positional encodings are PEQ along their token dimension. But in transformer-based SHMP approaches [9, 18, 26, 78, 108, 119, 136], or approaches that treat joints as features [18, 61, 100], tokens are reserved for the time dimension instead of the joints (against Lemma 1). We provide mathematical proof that our architecture is end-to-end equivariant in Sec. C.2, together with experimental results on equivariance for the baselines. To the best of our knowledge, an end-to-end PEQ architecture over joint orderings has not been demonstrated before for SHMP.

Additional Advantages. While equivariance may be learned via extensive augmentation (as for molecule generation [4, 123]), we chose to fulfill it mathematically by design, reducing model and learning complexity [12, 73]: on just a single dataset [46], our kinematics-agnostic approach achieves state-of-the-art results with 75% fewer parameters than the latest baseline SkelDiff [23], requiring around half of the training and inference time (see Fig. 3 and discussion in Sec. 4.2). Furthermore, our equivariance property allows us to train natively on multiple datasets $\mathbb{K}_{\text{train}} = \{\mathcal{K}^A, \mathcal{K}^N\}$ leveraging for the first time the two largest SHMP datasets at once, AMASS and Nymeria.

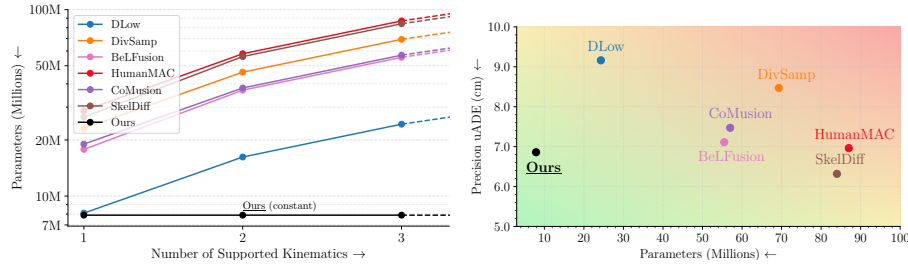


Fig. 3: State-of-the-art with 75% fewer parameters: (left) baselines scale with the number of supported datasets i.e. kinematics, we do not; (right) we achieve SOTA precision, averaged over three datasets [46, 75, 76], with a single model instance.

3.4 Directions as Motion Parametrization

Motion representation is a critical choice in SHMP, especially for our aims of ensuring compatibility across disparate datasets and motion formats. Despite its importance, this remains underinvestigated in SHMP literature, where 3D joint absolute coordinates (Sec. 2.4) remain the de facto standard. While flexible, this representation suffers from limb stretching and physical inconsistencies [23, 26]. Conversely, joints’ rotation angles relative to the kinematic chain parent (as in SMPL [72]) preserve the body structure by design, but require a canonical pose, which is not always available. We propose a robust middle ground by representing the joints as the relative direction vector w.r.t. the parent joint, *i.e.* $\mathbf{L}_t \in \mathbb{R}^{B \times 3}$, where each limb i is defined by the vector between a joint i and its parent.

$$\mathbf{L}_t^i = \mathbf{M}_t^i - \mathbf{M}_t^{\text{parent}(i)}, \quad i = 1, \dots, B. \quad (5)$$

We report the details in Sec. D.3. This parametrization has several advantages: 1) it is easily derived from both positions and angles pose representations, facilitating cross-dataset training; 2) it eliminates limb stretching by rescaling relative distances between joints based on past observations at inference time; 3) it remains numerically stable during training [11] without singularities typical of rotation spaces [35, 102]. We use the average limb length of the training data as a heuristic for generating missing limbs. Empirical results in Tab. 5 demonstrate that this representation improves realism and diversity metrics also for other SHMP methods, without requiring any change to existing SHMP architectures.

4 Experiments

4.1 Experimental Settings

Datasets. We follow the evaluation settings of [9, 18, 23, 108], and include an additional dataset [75]: in addition to the protocols involving the highly diverse AMASS (A) dataset [76], and the widely employed, but involving only 7 subjects, H36M (H) [46], we also include Nymeria (N) [75], after adapting it to SHMP.

Table 1: Evaluation of zero-shot kinematics on H36M ($\mathcal{K}^H$) [46]. We support inference on novel kinematics out-of-the-box, while previous approaches require additional kinematics conversion [18, 43]. Baselines are trained on AMASS [76] (A, $\mathcal{K}^A$). We are the first to natively support multiple kinematics and thus present a model trained additionally on Nymeria [75] (N, $\mathcal{K}^N$). The best results are highlighted in **bold**, second-best are underlined. Conventional SHMP metrics rank identically, see Tab. 19.

Units Method	$\mathcal{K}_{\text{new}}$	Precision ↓			MM GT ↓		Div ↑	Realism ↓		Body Real ↓	
		cm	cm	deg°	cm	cm	m	—	—	%	%
		$u\text{ADE}$	$u\text{FDE}$	MAE	$u\text{MMA}$	$u\text{MMF}$	$u\text{APD}$	CMD	FID	str	jit
ZeroVel	✓	11.77	17.88	6.753	13.74	18.56	0.000	22.822	-	0.00	0.00
TPK [119]+ [43]	✗	13.81	16.13	22.276	14.60	16.22	1.469	10.051	3.773	19.55	0.46
DLow [136]+ [43]	✗	12.71	14.80	21.887	13.60	14.97	2.060	9.204	2.875	20.26	0.53
GSPS [78]+ [43]	✗	9.29	11.91	8.107	10.85	12.35	2.069	7.409	1.735	11.51	0.38
DivSamp [26]+ [43]	✗	9.27	12.61	8.374	11.42	13.27	4.210	47.783	5.629	18.47	1.01
BeLFusion [9]+ [43]	✗	9.24	11.62	8.200	10.93	12.16	1.305	8.031	1.195	9.81	0.34
CoMusion [108]+ [43]	✗	10.07	12.15	21.066	12.49	12.96	<u>2.070</u>	8.587	1.426	15.98	0.51
SkelDiff [23]+ [43]	✗	10.81	14.99	14.947	12.75	15.42	0.992	7.616	5.252	11.25	0.28
EquiFusion(A)	✓	<u>7.86</u>	<u>10.47</u>	<u>5.861</u>	10.66	<u>11.61</u>	1.973	7.061	0.691	0.00	0.00
EquiFusion(A+N)	✓	7.71	10.21	5.683	10.58	11.36	1.797	<u>7.349</u>	0.504	0.00	0.00

Details of this process are reported in Sec. G.2, but overall the final size is comparable to AMASS and the quality of the motions is less dynamic. While the notation $\mathcal{K}$ denotes exclusively kinematics, the letters A,H,N denote training data, automatically implying the corresponding kinematics. See Sec. 3.1 and Sec. 2.1 for details on kinematics. We test models trained on A for the zero-shot motion scenario of MoYoga ($\mathcal{K}^A$) [113].

Metrics. Conventionally employed metrics [9, 23, 136] can be categorized into precision, diversity, realism, and body realism (see Sec. F for extensive definitions). Current metrics ADE, FDE, APD cannot be used to compare methods across datasets because they are dependent on the number of joints and cannot be measured in meters i.e. they include a cofactor $\sim \sqrt{J}$. Hence we introduce corresponding revised, unified versions $u\text{ADE}$, $u\text{FDE}$, $u\text{APD}$ measured in centimeters ($u\text{ADE}$, $u\text{FDE}$ with their multimodal counterpart) or meters ($u\text{APD}$). Note that $u\text{ADE}$ is mathematically equivalent to the mean per-joint projection error (MPJPE), widely employed in other vision tasks. Since they differ by just a cofactor, unified metrics rank identical to conventional metrics, which are provided for every table in the appendix.

Baselines. SOTA baselines [9, 18, 23, 26, 78, 108, 119, 136] do not support novel kinematics out-of-the-box, while we do so by design. To enable evaluation in a zero-shot kinematics setting for baselines, as described in Sec. 2.3, we perform kinematics conversions following Holden *et al.* established by [18]. We include the ZeroVelocity baseline (ZeroVel), a competitive algorithmic baseline that repeats the last frame of the past observation for all future frames [136].

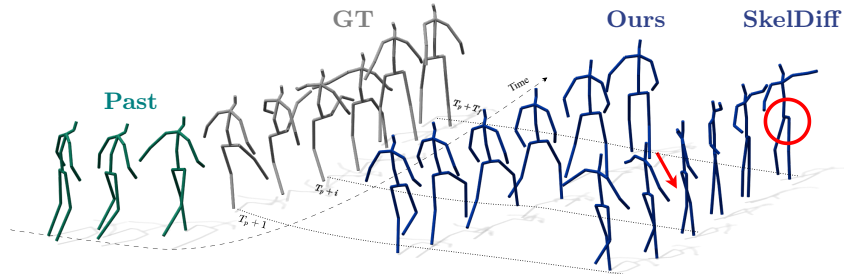


Fig. 4: Qualitatives for zero-shot kinematics on H36M($\mathcal{K}^H$) [46]. We report the prediction closest to the GT for our method and SkelDiff [23], which does not support novel skeletons natively and is thus combined with a retargeting procedure [43]. While this challenging dynamic kick is not reproduced by either works, our prediction is realistic and coherent with the observation. SkelDiff is unable to generate a semantically close motion and cannot recover from the degradation introduced by retargeting.

4.2 Comparison

Zero-Shot full-body Kinematics. In the main body of this paper, we concentrate on the realistic case that the SHMP model is trained on the kinematic $\mathcal{K}^A$ corresponding to the dataset with the largest data amount available, AMASS [76], and inference is performed on a skeleton kinematics $\mathcal{K}^H$ with fewer joints and a smaller dataset, H36M [46]. This is the most advantageous setting for a SHMP model, as it is trained with a stronger prior (we discuss the more challenging, reverse scenario in Sec. H.2). We report quantitative results in Tab. 1. Current SOTA approaches do not support this zero-shot kinematics scenario on $\mathcal{K}^H$ out-of-the-box, and the input motion must first be converted via retargeting [43] to the kinematics supported by the model (see discussion in Sec. 2.3). Isolating the retargeting error from the baseline error completely is not possible, but via triangle inequality we estimate an upper bound. We see that baselines can recover from input retargeting: SkelDiff’s total error 10.81 cm is around half of the upper bound 22 cm (further analysis in Sec. H.3). Our model instead, Tab. 1, can operate natively on any kinematics and achieves the best results, with an improvement up to 27% for precision metrics, and 72% for realism. As already discussed by previous works [9, 23], evaluating the multidimensional HMP problem is complex, and the numerous metrics are often complementary: a high diversity scores (APD) can be originated by unrealistic and ill-posed motions, as shown by the poor realism and body realism results of the VAE-based method DivSamp [26]. When considering the latest diffusion method, SkelDiff [23], our generated motions are almost twice as diverse. Additionally, our motion representation as bone direction delivers perfect body realism by definition, and we discuss it further in Sec. 4.2. A qualitative result can be seen in Fig. 4.

Multi-dataset Training. Our key ability to train on multiple kinematics translates into the ability to train on any dataset simultaneously. We leverage for the first time the two largest SHMP datasets, AMASS (A) and Nymeria (N), and report results for our method trained on the combination of both (A+N). As expected,

Table 2: Out-of-distribution MoYoga [113] for models trained on AMASS [76].

Method	$\mathcal{K}$ train	MoYoga ($\mathcal{K}^A$)						
		Precision ↓			Div↑	Real↓	B. Real↓	
		ADE	FDE	MAE	APD	CMD	str	jit
ZeroVel	-	0.709	1.187	7.954	0.000	20.333	0.00	0.00
SkelDiff [23]	A	0.567	0.892	8.048	13.304	15.710	7.25	0.29
EquiFusion	A	<u>0.501</u>	0.780	<u>6.982</u>	<u>13.153</u>	<u>11.961</u>	0.00	0.00
EquiFusion	A+N	0.492	<u>0.786</u>	6.596	12.467	7.095	<u>0.00</u>	<u>0.00</u>

Table 3: Occlusion of a random limb (leg or arm) at inference on AMASS test set.

Method	$\mathcal{K}$ train	AMASS ($\mathcal{K}^A$)					
		Precision ↓			Div ↑	B. Real ↓	
		ADE	FDE	MAE	APD	str	jit
SkelDiff [23]	A	-	-	-	-	-	-
SkelDiff [23]+rp	A	0.574	0.727	6.996	8.890	8.15	0.27
SkelDiff [23]+sl	A	0.567	0.683	<u>7.162</u>	<u>9.274</u>	5.64	0.23
EquiFusion	A	<u>0.553</u>	<u>0.618</u>	10.190	10.152	<u>0.00</u>	<u>0.00</u>
EquiFusion	A+N	0.499	0.553	8.777	9.099	0.00	0.00

training on an additional kinematic type (i.e. $\mathcal{K}^N$) and more data unlocks better performance. Our gains in this cross-dataset setting do not come from increased data diversity alone: we see that only 2% of our 29%ADE improvement and 27% of our 90% FID come from the multi-dataset training. The only difference between EquiFusion(A), trained only on AMASS, and EquiFusion(A+N) is the training data, while the architecture and hyperparameters remain the same. We find the fact remarkable, highlighting the need for methods that natively reason over any dataset and kinematics and opening future discussion on the impact of different motion distributions on training.

Scalability to Any Kinematics. Fig. 3 highlights the impact on scaling and efficiency of our method. The kinematics-agnostic property grants us two advantages: 1) the number of training parameters does not increase with the number of joints J of the training kinematics $\mathcal{K}$ (e.g. SkelDiff requires more parameters for AMASS compared to H36M as AMASS has more joints); 2) we scale constantly with the number of supported datasets or kinematics because a single instance of our method can handle all kinematics, full-body or partial, that currently exists or will be released in the future. Instead, as shown on the left plot of Fig. 3, baselines require a new instance for each kinematics. Therefore, when comparing instances on a single dataset (AMASS), we are 75% more compact (70% on H36M) than the latest baseline SkelDiff [23], while when considering three datasets (right plot of Fig. 3), we are 90% more compact. Our inference and training time are halved, despite training on multiple datasets (Sec. H.1).

Out-of-distribution Motions on MoYoga ($\mathcal{K}^A$). In Tab. 2 we report quantitative evaluation on the MoYoga dataset [113], containing out-of-distribution motions recorded via MoCap. Since this dataset shares the same kinematics as the training dataset AMASS ($\mathcal{K}^A$), baselines allow testing without retargeting. Our architecture outperforms the most recent and competitive baseline, SkelDiff [23], showcasing the stronger generalization capability of our model and again the advantages of using a richer prior. This is best highlighted in the qualitative results provided on our webpage.

Zero-Shot Partial Kinematics. Performing SHMP with partial input skeletons in a zero-shot setting, without training for it specifically, constitutes an exciting opportunity opened by our method. We also find it particularly relevant for applications, as it allows, depending on the upstream acquisition (MoCap or video), to represent uncertainty or occlusions. EquiFusion natively supports

Table 4: Ablations for methods trained on multiple datasets, AMASS($\mathcal{K}^A$) and Nymeria($\mathcal{K}^N$), tested on AMASS($\mathcal{K}^A$) as usual for single-kinematics and on H36M($\mathcal{K}^H$) for zero-shot kinematics.

Method	AMASS ($\mathcal{K}^A$)			H36M ($\mathcal{K}^H$)		
	ADE↓	MAE ↓	CMD↓	ADE↓	MAE ↓	CMD↓
Modified [23]	0.500	7.007	19.233	0.670	12.065	31.461
Ours w/o Eq	0.539	6.974	22.294	0.467	7.245	9.559
Ours	0.512	<u>6.525</u>	19.699	0.380	<u>5.685</u>	8.086
Ours on $\mathbf{P} + \mathbf{P}_\epsilon$	0.512	<u>6.525</u>	19.699	0.380	<u>5.685</u>	8.086
Ours on $\mathbf{P}$	0.513	6.529	19.686	0.380	5.680	<u>8.088</u>
Ours+S	<u>0.508</u>	6.456	<u>19.492</u>	0.381	5.708	8.526

Table 5: Models trained on AMASS [76] with two different motion parametrization: the proposed bone directions $\mathbf{L}$ or the conventional 3D positions $\mathbf{M}$.

Mot.		AMASS ($\mathcal{K}^A$)					
		Precision ↓			Div ↑	Real ↓	B. Real ↓
		ADE	FDE	MAE	APD	CMD	str jit
SkelDiff [23]	$\mathbf{M}$	0.480	0.545	6.124	9.456	11.417	3.15 0.20
SkelDiff [23]	$\mathbf{L}$	0.496	0.546	6.193	9.960	9.143	0.00 0.00
Ours	$\mathbf{M}$	0.501	0.561	6.551	8.348	13.963	3.58 0.27
Ours	$\mathbf{L}$	0.498	0.559	6.173	8.413	12.530	0.00 0.00

flexible kinematics, and so also this scenario. For quantitative evaluation, we randomly remove full limbs (i.e., legs or arms consisting of three joints) from every input sequence in the AMASS test set and pass them to the networks (Tab. 3). To let the most recent and competitive baselines operate in this case, we adopt two heuristics. In the first, we complete the missing information with the one from the rest pose (+rp), simulating a “mask” effect. In the second step, we replicate the motion observed from the symmetric counterpart (+sl), resulting in more realistic motions. Despite the symmetric completion enhancing the diversity of predictions and improving body realism, we observe that precision is still lacking. Our model instead predicts a reasonable future regardless of the missing part. Also in this case, relying on a wider motion prior (A+N) helps overcoming the missing information. Qualitative results are presented on our website. Apart from its applicative relevance to face occlusions, we also foresee our flexibility handy for tackling marginalized categories, such as individuals with diverse body types and those with missing limbs.

Conventional Single-kinematics SHMP. Beyond cross-kinematics, our approach achieves state-of-the-art competitive performance on the in-domain, same kinematics evaluation of conventional SHMP benchmarks, *i.e.* on the designed test set of each dataset, for the kinematics belonging to that dataset. Besides AMASS [76] in Tab. 17 and H36M [46] in Tab. 16, we additionally train and evaluate the latest SHMP diffusion models on Nymeria in Tab. 18. The precision $uADE$ averaged over all three datasets is reported in Fig. 3. Remarkably, while different methods require manual tuning of hyperparameters for each dataset [9, 18, 23, 108], our method employs only a single configuration.

Ablations. We first validate our approach in Tab. 4. We show that a model whose weights are learned in dependence of joints, despite being trained with data augmentation on multiple kinematics on the same amount of data (A+N), does achieve competitive performance on the training kinematics (A), but fails at zero-shot kinematics on H36M. This shows in current settings, data alone does not guarantee kinematics-agnostic models. See Sec. G.1 for how we adapt the most competitive baseline [23] to multidataset training. Breaking our equivariance property by inserting positional encoding (*Ours w/o Eq*) leads to failure at zero-shot kinematics: equivariance guarantees kinematics-agnostic capabili-

ties. We verify that our model remains equivariant under permutations of joints, adjacency, and noise (*Ours on $\mathbf{P} + \mathbf{P}\epsilon$*). We empirically validate end-to-end equivariance under stochastic sampling (i.e. we permute joints and adjacency on the same noise realization and compare outputs) as (*Ours on $\mathbf{P}$*). Furthermore, providing explicit joint semantics (e.g., limb side or type) is non-trivial (permutation equivariance must hold) and yields no gains (*Ours+S*); the motion’s temporal information alone suffices to distinguish limbs, as human degrees of freedom are invariant. In Tab. 5 we validate the choice of using bone directions as motion representation, showing its contribution to both our method and the baseline SkelDiff. In both cases, such a representation has a limited impact on precision and diversity, while leading to improved realism by 10% and achieving perfectly consistent limb length by design. Further metrics and more detailed ablations are reported in the appendix.

5 Conclusion

Limitations and Future Work We introduced a novel motion parameterization, bone directions, whose formulation guarantees perfect adherence to bone length constraints. However, this representation handles only missing joints which are leaves of the kinematic chain; it does not support, e.g., a missing elbow when the hand joint is present. While such occlusions can be represented naturally in the input adjacency matrix, extending bone directions to handle them remains open. Beyond this, several broader directions warrant exploration. Training on multiple datasets with differing motion distributions raises open questions about how data quality, particularly the prevalence of dynamic movements, affects performance at inference time in both in-domain and cross-dataset scenarios. Finally, future work may integrate skeletons beyond human kinematics, investigating bipedal and quadrupedal animal motion.

Conclusions We presented EquiFusion, a novel and fundamentally more generalizable approach to Stochastic 3D Human Motion Prediction (SHMP). By formulating the skeleton kinematics as an explicit input and designing an end-to-end permutation-equivariant architecture for a latent diffusion model, we successfully sever SHMP models’ reliance on graph structures hard-coded at training time. This paradigm enables a single model to generalize zero-shot to unseen datasets and eliminates the need for expensive and inaccurate data retargeting. Beyond resolving the generalization bottleneck, EquiFusion achieves state-of-the-art performance with remarkably enhanced efficiency and supports novel tasks such as partial motion prediction and targeted limb generation. This represents a fundamental shift from kinematics-specific models toward a truly kinematics-agnostic, generalizable SHMP framework.

Acknowledgments This work was supported by the European Research Council (ERC) Advanced Grant SIMULACRON. Thanks to Maolin Gao and Felix Wimbauer for proofreading, Thomas Dagès for the detailed and constructive suggestions, Stefania Zunino and the CVG team for their unwavering support.

References

1. Movella XSens MVN Link motion capture, <https://www.movella.com/products/motion-capture/xsens-mvn-link> 3
2. 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), 62–1 (2020) 4, 2
3. Aberman, K., Wu, R., Lischinski, D., Chen, B., Cohen-Or, D.: Learning character-agnostic motion for motion retargeting in 2d. *arXiv preprint arXiv:1905.01680* (2019) 4, 2
4. Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A.J., Bambrick, J., et al.: Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature* **630**(8016), 493–500 (2024) 9
5. Adeli, V., Ehsanpour, M., Reid, I., Niebles, J.C., Savarese, S., Adeli, E., Rezaatoughi, H.: Tripod: Human trajectory and pose dynamics forecasting in the wild. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 13390–13400 (2021) 3
6. Aksan, E., Kaufmann, M., Cao, P., Hilliges, O.: A spatio-temporal transformer for 3d human motion prediction. In: *2021 International Conference on 3D Vision (3DV)*. pp. 565–574. *IEEE* (2021) 3
7. AlAttar, A., Chappell, D., Kormushev, P.: Kinematic-model-free predictive control for robotic manipulator target reaching with obstacle avoidance. *Frontiers in Robotics and AI* **9**, 809114 (2022) 7
8. Alibeigi, M., Ahmadabadi, M.N., Araabi, B.N.: A fast, robust, and incremental model for learning high-level concepts from human motions by imitation. *IEEE Transactions on Robotics* **33**(1), 153–168 (2016) 2, 4
9. Barquero, G., Escalera, S., Palmero, C.: Belfusion: Latent diffusion for behavior-driven human motion prediction. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 2317–2327 (2023) 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 21, 24, 26, 27, 28, 30, 31, 32, 35
10. Barsoum, E., Kender, J., Liu, Z.: Hp-gan: Probabilistic 3d human motion prediction via gan. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. pp. 1418–1427 (2018) 4
11. Bie, X., Guo, W., Leglaive, S., Girin, L., Moreno-Noguer, F., Alameda-Pineda, X.: Hit-dvae: Human motion generation via hierarchical transformer dynamical vae. *arXiv preprint arXiv:2204.01565* (2022) 10, 18
12. Bietti, A., Venturi, L., Bruna, J.: On the sample complexity of learning under geometric stability. *Advances in neural information processing systems* **34**, 18673–18684 (2021) 3, 9
13. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., et al.: Video generation models as world simulators. *OpenAI Blog* **1**(8), 1 (2024) 2
14. Cai, Y., Huang, L., Wang, Y., Cham, T.J., Cai, J., Yuan, J., Liu, J., Yang, X., Zhu, Y., Shen, X., et al.: Learning progressive joint propagation for human motion prediction. In: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII* 16. pp. 226–242. *Springer* (2020) 3
15. Cai, Y., Wang, Y., Zhu, Y., Cham, T.J., Cai, J., Yuan, J., Liu, J., Zheng, C., Yan, S., Ding, H., et al.: A unified 3d human motion synthesis model via condi-

- tional variational auto-encoder. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11645–11655 (2021) 4
16. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR (2017) 5
17. Cao, Z., Liu, B., Li, S., Zhang, W., Chen, H.: G-dream: Graph-conditioned diffusion retargeting across multiple embodiments. arXiv preprint arXiv:2505.20857 (2025) 4, 5, 2
18. Chen, L.H., Zhang, J., Li, Y., Pang, Y., Xia, X., Liu, T.: Humanmac: Masked motion completion for human motion prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9544–9555 (2023) 2, 4, 7, 9, 10, 11, 14, 16, 22, 24, 26, 30, 31, 32
19. Choi, K.J., Ko, H.S.: Online motion retargetting. The Journal of Visualization and Computer Animation 11(5), 223–235 (2000) 4, 2
20. Ci, H., Wu, M., Zhu, W., Ma, X., Dong, H., Zhong, F., Wang, Y.: Gfpose: Learning 3d human pose prior with gradient fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4800–4810 (2023) 6
21. Cui, Q., Sun, H.: Towards accurate 3d human motion prediction from incomplete observations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4801–4810 (2021) 2, 4, 5
22. Cui, Q., Sun, H., Yang, F.: Learning dynamic relationships for 3d human motion prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6519–6527 (2020) 4, 3
23. Curreli, C., Muhle, D., Saroha, A., Ye, Z., Marin, R., Cremers, D.: Nonisotropic gaussian diffusion for realistic 3d human motion prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1871–1882 (2025) 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 24, 26, 27, 28, 30, 31, 32, 33, 35
24. Dabhi, M., Jeni, L.A., Lucey, S.: 3d-lfm: Lifting foundation model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10466–10475 (2024) 4, 6
25. Dang, L., Nie, Y., Long, C., Zhang, Q., Li, G.: Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11467–11476 (2021) 3
26. Dang, L., Nie, Y., Long, C., Zhang, Q., Li, G.: Diverse human motion prediction via gumbel-softmax sampling from an auxiliary space. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 5162–5171 (2022) 2, 4, 5, 7, 9, 10, 11, 12, 8, 15, 16, 24, 26, 27, 28, 30, 31, 32
27. Delhaisse, B., Esteban, D., Roza, L., Caldwell, D.: Transfer learning of shared latent spaces between robots with similar kinematic structure. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 4142–4149. IEEE (2017) 4, 2
28. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems 34, 8780–8794 (2021) 8
29. Engel, J., Somasundaram, K., Goesele, M., Sun, A., Gamino, A., Turner, A., Talattof, A., Yuan, A., Souti, B., Meredith, B., et al.: Project aria: A new tool for egocentric multi-modal ai research. arXiv preprint arXiv:2308.13561 (2023) 3
30. Feng, A., Huang, Y., Xu, Y., Shapiro, A.: Automating the transfer of a generic set of behaviors onto a virtual character. In: Motion in Games: 5th International Conference, MIG 2012, Rennes, France, November 15–17, 2012. Proceedings 5. pp. 134–145. Springer (2012) 4, 2

31. Figuera, X., Park, S., Ahn, H.: Redefining data pairing for motion retargeting leveraging a human body prior. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4800–4807. IEEE (2024) [4](#)
32. Fragkiadaki, K., Levine, S., Felsen, P., Malik, J.: Recurrent network models for human dynamics. In: Proceedings of the IEEE international conference on computer vision. pp. 4346–4354 (2015) [3](#)
33. Fritsche, O., Camacho, S., Hossain, M.S.B., Halpenny, T., Archniegas, C., Dranetz, J., Hadley, D., Guo, Z., Choi, H.: Ultra-mocap: A multimodal imu and semg dataset for upper body joint kinematics analysis. Authorea Preprints (2025) [3](#)
34. Gat, I., Raab, S., Tevet, G., Reshef, Y., Bermano, A.H., Cohen-Or, D.: Anytop: Character animation diffusion with any topology. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. pp. 1–10 (2025) [4](#), [7](#), [6](#)
35. Geist, A.R., Frey, J., Zhobro, M., Levina, A., Martius, G.: Learning with 3d rotations, a hitchhiker’s guide to so (3). arXiv preprint arXiv:2404.11735 (2024) [5](#), [10](#), [18](#)
36. Gil, O., Sanfeliu, A.: Human motion trajectory prediction using the social force model for real-time and low computational cost applications. In: Iberian Robotics conference. pp. 235–247. Springer (2023) [7](#)
37. Gleicher, M.: Retargetting motion to new characters. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques. pp. 33–42 (1998) [4](#), [2](#)
38. Gu, C., Yu, J., Zhang, C.: Learning disentangled representations for controllable human motion prediction. Pattern Recognition **146**, 109998 (2024) [4](#)
39. Gui, L.Y., Wang, Y.X., Liang, X., Moura, J.M.: Adversarial geometry-aware human motion prediction. In: Proceedings of the european conference on computer vision (ECCV). pp. 786–803 (2018) [3](#)
40. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2255–2264 (2018) [17](#)
41. Guzov, V., Chibane, J., Marin, R., He, Y., Saracoglu, Y., Sattler, T., Pons-Moll, G.: Interaction replica: Tracking human–object interaction and scene changes from human motion. In: 2024 International Conference on 3D Vision (3DV). pp. 1006–1016. IEEE (2024) [3](#)
42. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) [2](#), [8](#), [13](#), [17](#), [20](#)
43. Holden, D., Saito, J., Komura, T.: A deep learning framework for character motion synthesis and editing. ACM Transactions on Graphics (TOG) **35**(4), 1–11 (2016) [2](#), [4](#), [11](#), [12](#), [22](#), [26](#), [27](#), [28](#), [32](#)
44. Hu, L., Zhang, Z., Zhong, C., Jiang, B., Xia, S.: Pose-aware attention network for flexible motion retargeting by body part. IEEE Transactions on Visualization and Computer Graphics (2023) [4](#), [7](#), [2](#)
45. Huang, Z., Feng, H., Sun, Y.T., Guo, Y.C., Cao, Y.P., Sheng, L.: Animax: Animating the inanimate in 3d with joint video-pose diffusion models. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. pp. 1–13 (2025) [4](#)
46. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments (2014), <http://vision.imar.ro/human3.6m>. [2](#), [3](#), [6](#), [9](#), [10](#), [11](#), [12](#), [14](#), [26](#), [27](#), [30](#), [31](#), [32](#)

47. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on spatio-temporal graphs. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 5308–5317 (2016) [3](#)
48. Jang, H., Kwon, B., Yu, M., Kim, S.U., Kim, J.: A variational u-net for motion retargeting. In: *SIGGRAPH Asia 2018 Posters*. pp. 1–2 (2018) [4](#), [2](#)
49. Jang, I., Choi, S., Hong, S., Kim, C., Noh, J.: Geometry-aware retargeting for two-skinned characters interaction. *ACM Transactions on Graphics (TOG)* **43**(6), 1–17 (2024) [4](#), [5](#), [2](#)
50. Keriven, N., Peyré, G.: Universal invariant and equivariant graph neural networks. *Advances in neural information processing systems* **32** (2019) [9](#)
51. Kim, W., Li, T., Ha, S.: Moreflow: Motion retargeting learning through unsupervised flow matching. *arXiv preprint arXiv:2509.25600* (2025) [4](#), [2](#)
52. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016) [9](#), [4](#)
53. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 2252–2261 (2019) [3](#)
54. Kundu, J.N., Gor, M., Babu, R.V.: Bihmp-gan: Bidirectional 3d human motion prediction gan. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 33, pp. 8553–8560 (2019) [4](#)
55. Laabid, N., Rissanen, S., Heinonen, M., Solin, A., Garg, V.: Equivariant denoisers cannot copy graphs: Align your graph diffusion models. In: *International Conference on Learning Representations*. vol. 2025, pp. 100256–100294 (2025) [4](#)
56. Lee, J., Shin, S.Y.: A hierarchical approach to interactive motion editing for human-like figures. In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. pp. 39–48 (1999) [4](#), [2](#)
57. Lee, S., Kang, T., Park, J., Lee, J., Won, J.: Same: Skeleton-agnostic motion embedding for character animation. In: *SIGGRAPH Asia 2023 Conference Papers*. pp. 1–11 (2023) [2](#), [4](#), [7](#), [6](#)
58. Li, C., Zhang, Z., Lee, W.S., Lee, G.H.: Convolutional sequence to sequence model for human dynamics. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 5226–5234 (2018) [3](#)
59. Li, C., Chibane, J., He, Y., Pearl, N., Geiger, A., Pons-Moll, G.: Unimotion: Unifying 3d human motion synthesis and understanding. In: *2025 International Conference on 3D Vision (3DV)*. pp. 240–249. IEEE (2025) [5](#)
60. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 3595–3603 (2019) [4](#), [3](#)
61. Li, M., Chen, S., Liu, Z., Zhang, Z., Xie, L., Tian, Q., Zhang, Y.: Skeleton graph scattering networks for 3d skeleton-based human motion prediction. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 854–864 (2021) [9](#), [3](#)
62. Li, M., Chen, S., Zhao, Y., Zhang, Y., Wang, Y., Tian, Q.: Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 214–223 (2020) [4](#), [7](#), [9](#), [3](#)
63. Li, P., Starke, S., Ye, Y., Sorkine-Hornung, O.: Walkthedog: Cross-morphology motion alignment via phase manifolds. In: *ACM SIGGRAPH 2024 Conference Papers*. pp. 1–10 (2024) [4](#), [2](#)

64. Li, T., Won, J., Clegg, A., Kim, J., Rai, A., Ha, S.: Ace: Adversarial correspondence embedding for cross morphology motion retargeting from human to nonhuman characters. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–11 (2023) [4](#)
65. Liao, Z., Yang, J., Saito, J., Pons-Moll, G., Zhou, Y.: Skeleton-free pose transfer for stylized 3d characters. In: European Conference on Computer Vision. pp. 640–656. Springer (2022) [4](#), [5](#), [2](#)
66. Lim, J., Chang, H.J., Choi, J.Y.: Pmnet: Learning of disentangled pose and movement for unsupervised motion retargeting. In: 30th British Machine Vision Conference (BMVC 2019). British Machine Vision Association, BMVA (2019) [2](#), [4](#)
67. Lin, P., Chen, P., Jiao, R., Mo, Q., Cen, J., Huang, W., Liu, Y., Huang, D., Lu, Y.: Equivariant diffusion for crystal structure prediction. arXiv preprint arXiv:2512.07289 (2025) [8](#), [13](#), [14](#)
68. Liu, Q., Lv, K., Dong, K., Xue, J., Niu, Z., Wang, J.: Text-to-any-skeleton motion generation without retargeting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12926–12936 (2025) [4](#), [7](#), [6](#)
69. Liu, Z., Lyu, K., Wu, S., Chen, H., Hao, Y., Ji, S.: Aggregated multi-gans for controlled 3d human motion prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 2225–2232 (2021) [4](#)
70. Liu, Z., Wu, S., Jin, S., Liu, Q., Lu, S., Zimmermann, R., Cheng, L.: Towards natural and accurate future motion prediction of humans and animals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10004–10012 (2019) [3](#)
71. Lohit, S., Anirudh, R., Turaga, P.: Recovering trajectories of unmarked joints in 3d human actions using latent space optimization. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2342–2351 (2021) [5](#)
72. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia) **34**(6), 248:1–248:16 (Oct 2015) [3](#), [5](#), [10](#)
73. Lyle, C., van der Wilk, M., Kwiatkowska, M., Gal, Y., Bloem-Reddy, B.: On the benefits of invariance in neural networks. arXiv preprint arXiv:2005.00178 (2020) [3](#), [9](#)
74. Ma, L., Rabbany, R., Romero-Soriano, A.: Graph attention networks with positional embeddings. In: Pacific-Asia conference on knowledge discovery and data mining. pp. 514–527. Springer (2021) [35](#)
75. Ma, L., Ye, Y., Hong, F., Guзов, V., Jiang, Y., Postyeni, R., Pesqueira, L., Gamino, A., Baiyya, V., Kim, H.J., et al.: Nymeria: A massive collection of multi-modal egocentric daily motion in the wild. In: European Conference on Computer Vision. pp. 445–465. Springer (2024) [2](#), [3](#), [6](#), [10](#), [11](#), [26](#), [32](#)
76. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: Amass: Archive of motion capture as surface shapes. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5442–5451 (2019) [2](#), [3](#), [6](#), [10](#), [11](#), [12](#), [13](#), [14](#), [15](#), [26](#), [28](#), [31](#)
77. Mao, W., Liu, M., Salzmann, M.: History repeats itself: Human motion prediction via motion attention. In: Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 474–489. Springer (2020) [3](#)
78. Mao, W., Liu, M., Salzmann, M.: Generating smooth pose sequences for diverse human motion prediction. In: Proceedings of the IEEE/CVF International Con-

- ference on Computer Vision. pp. 13309–13318 (2021) 2, 4, 5, 9, 11, 7, 8, 16, 26, 27, 28, 30, 31, 32
79. Mao, W., Liu, M., Salzmann, M., Li, H.: Learning trajectory dependencies for human motion prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9489–9497 (2019) 7, 3
80. von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: European Conference on Computer Vision (ECCV) (sep 2018) 6
81. Martinez, J., Black, M.J., Romero, J.: On human motion prediction using recurrent neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2891–2900 (2017) 3
82. Martínez-González, A., Villamizar, M., Odobez, J.M.: Pose transformers (potr): Human motion prediction with non-autoregressive transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2276–2284 (2021) 3
83. Medjaoui, O., Desai, K.: Hr-stan: High-resolution spatio-temporal attention network for 3d human motion prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2540–2549 (2022) 3
84. Mo, C.A., Hu, K., Long, C., Yuan, D., Siu, W.C., Wang, Z.: Pumps: Skeleton-agnostic point-based universal motion pre-training for synthesis in human motion tasks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14496–14506 (2025) 6
85. Mourot, L., Hoyet, L., Clerc, F.L., Hellier, P.: Humot: human motion representation using topology-agnostic transformers for character animation retargeting. arXiv preprint arXiv:2305.18897 (2023) 4, 5, 2
86. Musoni, P., Marin, R., Melzi, S., Castellani, U.: A functional skeleton transfer. Proceedings of the ACM on Computer Graphics and Interactive Techniques 4(3), 1–15 (2021) 4, 2
87. Musoni, P., Marin, R., Melzi, S., Castellani, U.: Reposing and retargeting unrigged characters with intrinsic-extrinsic transfer. In: STAG. pp. 21–30 (2021) 4, 2
88. Nan, X., Liu, X., You, X., Du, Y., Ji, C., Song, J.: Pgdif: A physics-guided equivariant diffusion model for structure-based drug design. In: 2025 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). pp. 687–692 (2025) 4
89. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning. pp. 8162–8171. PMLR (2021) 20
90. Park, J.S., Manocha, D.: Hmpo: Human motion prediction in occluded environments for safe motion planning. arXiv preprint arXiv:2006.00424 (2020) 5
91. Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.M., Rothchild, D., So, D., Texier, M., Dean, J.: Carbon emissions and large neural network training. arXiv preprint arXiv:2104.10350 (2021) 8
92. 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
93. Pavlo, D., Grangier, D., Auli, M.: Quaternet: A quaternion-based recurrent model for human motion. arXiv preprint arXiv:1805.06485 (2018) 3
94. Petrov, I.A., Guзов, V., Marin, R., Aksan, E., Chen, X., Cremers, D., Beeler, T., Pons-Moll, G.: Echo: Ego-centric modeling of human-object interactions. In: Proceedings of the European conference on computer vision (ECCV) (2026) 3

95. Phu, K.A., Hoang, V.D., et al.: Predicting occluded skeletal joints via tracking-based feature extraction. *Neurocomputing* p. 131004 (2025) [5](#)
96. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* **1**(2), 3 (2022) [9](#), [17](#)
97. Rempe, D., Birdal, T., Hertzmann, A., Yang, J., Sridhar, S., Guibas, L.J.: Humor: 3d human motion model for robust pose estimation. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 11488–11499 (2021) [3](#)
98. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2022) [2](#), [8](#), [16](#), [17](#)
99. Roudsarabi, N., Behrad, A.R.: Solving occlusion problem in 3d human motion reconstruction. In: *2008 International Symposium on Telecommunications*. pp. 701–706. IEEE (2008) [5](#)
100. Saadatnejad, S., Rasekh, A., Mofayez, M., Medghalchi, Y., Rajabzadeh, S., Moradan, T., Alahi, A.: A generic diffusion-based approach for 3d human pose prediction in the wild (2023) [4](#), [9](#)
101. Saito, J., Li, J., de Ruyter, M., Guerrero, M., Lim, E., Hassani, E., Ribera, R.B., Moon, H., Dadala, M., Di Lucca, M., et al.: Soma: Unifying parametric human body models. *arXiv preprint* (2026) [5](#)
102. Salzmann, T., Pavone, M., Ryll, M.: Motron: Multimodal probabilistic human motion forecasting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 6457–6466 (2022) [5](#), [7](#), [9](#), [10](#), [4](#), [18](#), [20](#)
103. Sáráandi, I., Hermans, A., Leibe, B.: Learning 3d human pose estimation from dozens of datasets using a geometry-aware autoencoder to bridge between skeleton formats. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. pp. 2956–2966 (2023) [4](#), [5](#)
104. Sáráandi, I., Pons-Moll, G.: Neural localizer fields for continuous 3d human pose and shape estimation. *Advances in Neural Information Processing Systems* **37**, 140032–140065 (2025) [4](#), [3](#)
105. Schneuing, A., Harris, C., Du, Y., Didi, K., Jamasb, A., Igashov, I., Du, W., Gomes, C., Blundell, T.L., Lio, P., et al.: Structure-based drug design with equivariant diffusion models. *Nature Computational Science* **4**(12), 899–909 (2024) [4](#)
106. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: *CVPR* (2017) [5](#)
107. Sosnovik, I., Moskalev, A., Smeulders, A.: Disco: accurate discrete scale convolutions. *arXiv preprint arXiv:2106.02733* (2021) [5](#)
108. Sun, J., Chowdhary, G.: Comusion: Towards consistent stochastic human motion prediction via motion diffusion. *European Conference on Computer Vision* (2024) [2](#), [4](#), [7](#), [9](#), [10](#), [11](#), [14](#), [8](#), [15](#), [16](#), [17](#), [24](#), [26](#), [27](#), [28](#), [30](#), [31](#), [32](#), [35](#)
109. Tak, S., Ko, H.S.: A physically-based motion retargeting filter. *ACM Transactions on Graphics (ToG)* **24**(1), 98–117 (2005) [4](#), [2](#)
110. Tang, J., Yang, H., Chen, T., Hu, J.F.: Stochastic human motion prediction with memory of action transition and action characteristic. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 1883–1893 (2025) [5](#)
111. Tevet, G., Gordon, B., Hertz, A., Bermano, A.H., Cohen-Or, D.: Motionclip: Exposing human motion generation to clip space. In: *European Conference on Computer Vision*. pp. 358–374. Springer (2022) [5](#)
112. Thiede, E.H., Hy, T.S., Kondor, R.: The general theory of permutation equivariant neural networks and higher order graph variational encoders. *arXiv preprint arXiv:2004.03990* (2020) [8](#), [9](#), [13](#), [14](#)

113. Tripathi, S., Müller, L., Huang, C.H.P., Omid, T., Black, M.J., Tzionas, D.: 3D human pose estimation via intuitive physics. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4713–4725 (2023) [6](#), [11](#), [13](#)
114. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017) [9](#)
115. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017) [9](#)
116. 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. pp. 9720–9729 (2021) [2](#), [4](#)
117. Villegas, R., Yang, J., Ceylan, D., Lee, H.: Neural kinematic networks for unsupervised motion retargeting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8639–8648 (2018) [2](#), [4](#)
118. Wad, T., Sun, Q., Pranata, S., Jayashree, K., Zhang, H.: Equivariance and invariance inductive bias for learning from insufficient data. In: European Conference on Computer Vision. pp. 241–258. Springer (2022) [8](#), [13](#), [14](#)
119. Walker, J., Marino, K., Gupta, A., Hebert, M.: The pose knows: Video forecasting by generating pose futures. In: Proceedings of the IEEE international conference on computer vision. pp. 3332–3341 (2017) [4](#), [5](#), [9](#), [11](#), [7](#), [8](#), [26](#), [27](#), [28](#), [30](#), [31](#), [32](#)
120. Wang, H., Basu, A., Durandau, G., Sartori, M.: A wearable real-time kinetic measurement sensor setup for human locomotion. *Wearable technologies* **4**, e11 (2023) [3](#)
121. Wang, J., Li, X., Liu, S., De Mello, S., Gallo, O., Wang, X., Kautz, J.: Zero-shot pose transfer for unriggerd stylized 3d characters. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8704–8714 (2023) [4](#), [5](#), [2](#)
122. Wang, J., Yang, F., Gou, W., Li, B., Yan, D., Zeng, A., Gao, Y., Wang, J., Zhang, R.: Freeman: Towards benchmarking 3d human pose estimation in the wild (2023) [2](#)
123. Wang, Y., Hu, K., Gupta, S., Ye, Z., Wang, Y., Jegelka, S.: Understanding the role of equivariance in self-supervised learning. *Advances in Neural Information Processing Systems* **37**, 127483–127510 (2024) [9](#)
124. Wei, D., Sun, H., Li, B., Lu, J., Li, W., Sun, X., Hu, S.: Human joint kinematics diffusion-refinement for stochastic motion prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 6110–6118 (2023) [4](#), [5](#), [7](#), [16](#)
125. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: CVPR (2016) [5](#)
126. Wimmer, T., Golkov, V., Dang, H.N., Zaiss, M., Maier, A., Cremers, D.: Scale-equivariant deep learning for 3d data. *arXiv preprint arXiv:2304.05864* (2023) [5](#)
127. Xu, C., Tan, R.T., Tan, Y., Chen, S., Wang, X., Wang, Y.: Auxiliary tasks benefit 3d skeleton-based human motion prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9509–9520 (2023) [2](#), [4](#), [5](#)
128. Xu, G., Tao, J., Li, W., Duan, L.: Learning semantic latent directions for accurate and controllable human motion prediction. *European Conference on Computer Vision* (2024) [8](#)
129. Xu, M., Wang, Q., Wen, Z., Thien, P.D., Li, Z., Zhang, N., He, X., Zhao, W., Gong, K., Zhang, M.: Necromancer: Breathing life into skeletons via bvh animation. *arXiv preprint* (2026) [6](#)

130. Xu, S., Wang, Y.X., Gui, L.Y.: Diverse human motion prediction guided by multi-level spatial-temporal anchors. In: European Conference on Computer Vision. pp. 251–269. Springer (2022) [4](#)
131. Yan, X., Rastogi, A., Villegas, R., Sunkavalli, K., Shechtman, E., Hadap, S., Yumer, E., Lee, H.: Mt-vae: Learning motion transformations to generate multi-modal human dynamics. In: Proceedings of the European conference on computer vision (ECCV). pp. 265–281 (2018) [4](#)
132. Yang, Z., Leite, C.S., Xiao, Y.: Strengthsense: A dataset of imu signals capturing everyday strength-demanding activities. arXiv preprint arXiv:2511.02027 (2025) [3](#)
133. Yao, J., Yang, B., Wang, X.: Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15703–15712 (2025) [8](#), [35](#)
134. Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., Liu, T.Y.: Do transformers really perform badly for graph representation? Advances in neural information processing systems **34**, 28877–28888 (2021) [9](#), [35](#)
135. Yoon, T., Kang, D., Kim, S., Cheng, J., Ahn, M., Coros, S., Choi, S.: Spatio-temporal motion retargeting for quadruped robots. IEEE Transactions on Robotics (2025) [4](#)
136. Yuan, Y., Kitani, K.: Dlow: Diversifying latent flows for diverse human motion prediction. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16. pp. 346–364. Springer (2020) [2](#), [4](#), [5](#), [7](#), [9](#), [11](#), [8](#), [15](#), [24](#), [26](#), [27](#), [28](#), [30](#), [31](#), [32](#), [35](#)
137. Zhang, H., Chen, Z., Xu, H., Hao, L., Wu, X., Xu, S., Zhang, Z., Wang, Y., Xiong, R.: Semantics-aware motion retargeting with vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2155–2164 (2024) [4](#), [2](#)
138. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. IEEE transactions on pattern analysis and machine intelligence **46**(6), 4115–4128 (2024) [6](#)
139. Zhong, C., Hu, L., Zhang, Z., Ye, Y., Xia, S.: Spatio-temporal gating-adjacency gcnn for human motion prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6447–6456 (2022) [7](#), [9](#)
140. Zhou, A., Yang, K., Burns, K., Cardace, A., Jiang, Y., Sokota, S., Kolter, J.Z., Finn, C.: Permutation equivariant neural functionals. Advances in neural information processing systems **36**, 24966–24992 (2023) [5](#)
141. Zhou, L., Meng, X., Liu, Z., Wu, M., Gao, Z., Wang, P.: Human pose-based estimation, tracking and action recognition with deep learning: A survey. arXiv preprint arXiv:2310.13039 (2023) [5](#)
142. Zhu, W., Qiu, Q., Calderbank, R., Sapiro, G., Cheng, X.: Scaling-translation-equivariant networks with decomposed convolutional filters. Journal of machine learning research **23**(68), 1–45 (2022) [5](#)

Table of Contents

A	Extended Discussion on Related Works and Positioning	2
A.1	Motion Retargeting	2
A.2	Stochastic 3D Human Motion Prediction (SHMP)	3
A.3	Deterministic Approaches to 3D Human Motion Prediction	5
A.4	Beyond our Task: Human Pose Estimation, Text-to-Motion, Unconditional Generation, and more	5
B	About Lemma 1: Weights Dependent on the Kinematics Cardinality . .	7
B.1	Dealing with 4D Data.	7
B.2	Counterexample for Lemma 1	7
B.3	Counterexamples for Previous SMP Approaches.	8
C	On Permutation Equivariance	9
C.1	Mathematical Proof of Equivariance Fulfilling Lemma 1	9
C.2	Mathematical Proof of Equivariance for EquiFusion	11
C.3	Equivariance in a Generative Diffusion Model	13
C.4	Evidence that Previous SHMP Approaches are not Equivariant . .	14
D	More Details on EquiFusion	16
D.1	Training Losses	16
D.2	Inference on Partial Skeletons	17
D.3	Parametrizing Motion as Bone Directions	17
D.4	Architecture Details	18
E	Implementation Details	20
F	Metrics Definition	20
F.1	Unified Metrics: Comparing among Datasets	21
G	Experimental Settings	21
G.1	Baselines	21
G.2	Adapting Nymeria to HMP	22
G.3	Inference on FPS Different From Training Time	23
H	Additional Experiments	25
H.1	Computational Efficiency and Inference Time	25
H.2	Zero-Shot Kinematics on AMASS	25
H.3	Analysis on Retargeting	29
H.4	Single-Kinematics: AMASS, Nymeria, H36M	30
H.5	Extended Tables from Main and not Unified Metrics	32
I	Ablations and Validations on EquiFusion	33
J	Qualitative Examples	35

A Extended Discussion on Related Works and Positioning

A.1 Motion Retargeting

In the following, we discuss retargeting approaches in detail and highlight why methods addressing isomorphic and homeomorphic kinematics do not apply to our SHMP case.

Isomorphic Kinematics. If two kinematics differ only in the length of their bones and are otherwise identical, we are dealing with isomorphic graphs. This case represents for us different human subjects of the same HMP dataset, and in our notation, both skeletons belong to the same kinematics $\mathcal{K}$ (Sec. 3.1. In the case of isomorphic graphs, motion retargeting between humanoid characters for digital animations has been vastly investigated, moving first from expensive optimization pipelines [19, 30, 37, 56, 109] to learned approaches with [27, 48] or without [3, 66, 116, 117] paired GT data, involving not only kinematics but also RGB [3], skinning information [66, 86, 87], or both [116]. While this line of approaches achieves high precision, it is not relevant for us, because we are interested in transferring motion across datasets, not within.

Homeomorphic Kinematics. Other approaches [2, 44, 137] deal with homeomorphic graphs, i.e. kinematics that share the same end-effectors and can be translated into each other by subdivision or merging of edges. Such case is not present among existing SHMP kinematics, as it suffices to day that end-effectors varies (H36M does have feet, FreeMan [122] has ears, etc.), but it would theoretically correspond to translating any kinematics $\mathcal{K}$ to a primal coarse skeleton with a very low number of keypoints (probably 5) and consequently vast information loss. Considering existing homeomorphic approaches, they often rely on information not available in SHMP, such as end-effector rotations [44] or rigged skeleton meshes such as SMPL.

Non-homeomorphic Kinematics. Instead, our work considers kinematics conversion between non-homeomorphic skeletons. Here we follow the retargeting approach from Holden *et al.* [43], already established in SHMP by HumanMAC [18]. While more recent approaches for non-homeomorphic retargeting exist from the domain of robotics and character animation, they are not applicable to our case. These require joint rotations of end-effectors and reference T-poses [57], skinning [65], meshes [121], or both skinning and meshes [49] - even without the skeleton itself [65, 121] - which is information not always available in SHMP. Recent robotics approaches for non-homeomorphic humanoids achieve [85] around 100mm or ca 90mm [17] reconstruction error for a whole sequence, but their code is not publicly available. Other approaches focus on semantic motion transfer through common latent spaces between humanoid and four-legged animals, where, due to lack of GT, the error can only be measured in terms of fidelity [51, 63]. Overall, we see our line of work as orthogonal: we do

Table 6: EquiFusion supports any input kinematics, including missing joints. Current SHMP Lock acquired approaches cannot perform inference on unseen or partial skeletons out-of-the-box, and need to be piped with retargeting or completion pipelines that increase complexity and lower performance.

Method	joint perm	new $\mathcal{K}$	occluded limbs	limbs gen	motion repr
Motron	✗	✗	✗	✗	Quat
DLow	✗	✗	✗	✗	Eucl
BeLFusion	✗	✗	✗	✗	Eucl
CoMusion	✗	✗	✗	✗	DFT
SkelDiff	✗	✗	✗	✗	Eucl
+retargeting	✓	—	✗	✗	//
+completion	✗	✓	—	✗	//
EquiFusion	✓	✓	✓	✓	bone dirs

not seek to benchmark retargeting approaches and find the most suitable one for SHMP, we aim to solve SHMP end-to-end.

Learned Human Priors. Since the very successful human parametrization SMPL, a wide line of works has developed, estimating SMPL parameters from images, single or multiview videos. This line of works also gave birth to methods that learn pose priors from data [53, 92, 97, 104] and allow reprojection of noisy motion to a more realistic space. While such prior does not solve skeleton retargeting, it could, in theory, further refine the output of retargeting. However, these spaces are only designed to suit the SMPL parametrization, and often require direct parametrization over the SMPL parameters [72], which may amount to several minutes (!) for a single motion sequence. Additionally, not all priors consider the temporal dimension of a motion, increasing yes the realism of human poses for a single timestep, but possibly decreasing temporal coherence between frames of the same motion. Furthermore, SMPL is only compatible with the AMASS kinematics, and not with others. For these computational and applicability reasons we do not further consider pose or motions priors among our retargeting possibilities.

A.2 Stochastic 3D Human Motion Prediction (SHMP)

Human Motion Prediction (HMP) aims to predict a future motion given a past observation motion. HMP methods have used various deep learning architectures, such as recurrent networks [32, 39, 47, 70, 81, 93], temporal convolutions [58, 83], and more recently transformers [6, 14, 82] and graph neural networks (GCN) [5, 25, 61, 62, 77, 79]. While some previous works tackled the problem in a deterministic manner [22, 60, 62] forecasting a single future, probabilistic or stochastic HMP (SHMP) aims to predict multiple futures. The reason behind

this distinction is also that the different tasks tend to consider different prediction time horizons: deterministic HMP deals with rather short-term forecasting, while stochastic SHMP makes predictions up to 2 s by observing 0.5 s. The longer the prediction timespan, the higher the possibilities of different semantic actions to take place and thus the necessity to model the problem as probabilistic. Our paper aims at SHMP, and the deterministic case is treated in detail in the next subsection Sec. A.3.

Limitations. So far, SHMP methods are trained and evaluated on each kinematics $\mathcal{K}^i$ independently. This yields multiple trained instances per method where training hyperparameters are carefully adapted to each dataset manually, which is a laborious procedure for method applications [18]. On top of these limitations, existing approaches cannot be tested out-of-the-box on novel kinematics $\mathcal{K}^j$ with $j \neq i$ and need to be retrained from scratch. This case also includes motions where limbs are occluded or not present in the subject i.e. partial kinematics. In this case, the occluded limbs need to be completed first via heuristics before being fed to current approaches (Tab. 6). Our work fills this gap. While approaches that naturally support multiple skeleton graphs have already been proposed for other computer vision tasks [24, 34, 45, 57, 68, 104] and are discussed in section Sec. A.4, no such approach exists for SHMP yet. The reason is that all previous SHMP architectures learn kinematics-specific weights to extract stronger prior on the training data. We elaborate on this mathematical aspect in Sec. C.4.

Closest Competitor Baselines. We will now specifically comment in detail on the most recent state-of-the-art SHMP approaches, which base the generative modeling on denoising diffusion models: BeLFusion [9], CoMusion [108], and SkelDiff [23]. CoMusion employs a diffusion transformer in input space to generate coarse predictions and refines them with a GCN [52]. CoMusion is an exponent of the line of HMP works that employ the Discrete Cosine Transform (DCT) on the temporal dimension of the input and treat the body joints as features. This line of work draws on many exponents from the field of deterministic HMP. Instead, BeLFusion and SkelDiff are latent diffusion models that down-sample the inputs in a lower-dimensional latent space via recurrent GCN-based architectures. While BeLFusion requires three training stages and an additional network to embed the observation, SkelDiff uses the same encoder, based on Typed-Graph Convolutions [102], trained to handle flexible input lengths for both the GT and the observation, and thus requiring only two training stages. We follow the same insight that temporal compression should happen similarly for both future and past, and reduce the number of required networks by employing the same encoder to embed both conditioning past and training GT in latent space. We further design our autoencoder as a masked transformer autoencoder where tokens are the joint dimensions, instead as a recurrent GRU [9, 108]. It is interesting that although transformers and GCN are meant to handle input with a flexible structure, none of the methods above support different skeleton kinematics.

Our Kinematics-Agnostic Solution. With this motivation, we propose to broaden the horizon of SHMP and investigate realistic scenarios of zero-shot inference on novel kinematics in SHMP, a capacity particularly relevant when dealing with motions extracted from videos [71, 90, 95, 99]. We present EquiFusion, the first SHMP method that is kinematics-agnostic by design. EquiFusion can also handle occluded or missing joints out-of-the-box without preprocessing (see overview in Tab. 6). This property is achieved by treating the motion kinematics as input and defining the model’s weights independently of the body joints. We discover that such condition holds naturally if a model is permutation equivariant (PEQ) with respect to the joint order. Despite including permutation equivariant components such as GCN layers and transformer layers, none of the existing SHMP baselines is end-to-end permutation equivariant w.r.t joint ordering and we prove it in Sec. C.4. In other applications [126, 140], PEQ has already been shown to outperform other networks when data is scarce [107, 142]. We thus fill the gap by implementing EquiFusion as a permutation equivariant latent diffusion model.

A.3 Deterministic Approaches to 3D Human Motion Prediction

Deterministic HMP addresses a time horizon significantly shorter than ours (usually 0.4s instead of 2s) and does so without modeling a distribution over possible futures but by regressing a single deterministic future. Occluded motions and training on multiple kinematics have already been investigated by deterministic HMP, but to the best of our knowledge, only explicitly. Meaning that training directly targets occlusions and completion via auxiliary tasks, losses, and networks [21, 127] and does not involve considerations about equivariance, while in our case we deal with occlusion without having seen any occluded motion at training time. Other works [103], do yes train on multiple skeletons, but do not support skeletons that were not seen during training at inference time. Instead, we do support novel kinematics at inference even when relying on just a single one during training.

A.4 Beyond our Task: Human Pose Estimation, Text-to-Motion, Unconditional Generation, and more

A vast number of tasks in computer vision deal with the human body, but from quite different viewpoints. In this section, we contextualize some of these tasks to our task of Human Motion Prediction (HMP) and comment on the techniques employed to address multiple kinematics.

Human Pose Estimation. One of the oldest fields dealing with humans is what we can consider the upstream pipeline to HMP [71, 90, 95, 99]: human pose estimation from images. This task takes as input an image, and delivers 2D or 3D locations of human body joints or keypoints. Such a set of keypoints for a single timestep is referred to as *pose*. When human pose estimation is applied to a series of consecutive images i.e. a video, the obtained 3D keypoint sequence is a *motion* i.e. a sequence of poses. This motion sequence is the input of HMP models.

Naturally, by definition, this task does not usually consider aspects that are at the core of HMP: 1) the time dimension, and 2) consequentiality and causality between past and future. However, as images naturally include occlusions of human body parts, this field has long been interested in dealing with occlusions or multiple keypoint configurations. We comment here on the most relevant approaches to us: GFPose [20], PUMPS [84], and 3D-LFM [24].

GFPose [20] solves human pose estimation for multiple applications - including pose completion, denoising, and lifting - but handles occlusions only through ad-hoc, explicit training (random masking) and does not handle multiple full-body skeleton kinematics. 3D-LFM [24] is a foundation model for 2D-to-3D lifting of keypoints, is agnostic to input categories but does not mention equivariance. Very recently, the Arxiv work PUMPS [84] investigates skeleton-agnosticism in the scope of human motion completion by preprocessing motions and transforming them as pointcloud. They do yes address motion denoising and 2d-to-3d motion lifting, but they finetune for it instead of doing it zero-shot.

Text-to-Motion Generation. Another task involving human motion is text-to-motion generation. The domain strongly differs from ours, as the goal is to generate motion with high fidelity to the input text, regardless of the input motion observation. For example, MotionDiffuse [138], which trains on SMPL and thus considers only the AMASS kinematics. Additionally, semantic action labels are part of the input in this task, where this is generally not the case for HMP. Some works attempt to separate motion from the skeleton. For example, the approach of Liu *et al.* [68] that generates motions from text for any skeleton with a token-based VQ-VAE, while instead we compress deterministically over the whole time dimension in a unique global latent code. They do not rely on equivariance and do not discuss partialities or occlusions. Another approach, AnyTop [34], generates realistic animal motion from an input skeleton relying on joint classes and text descriptors. Occlusions are also not discussed, but if removing end-effector is possible, it would at least necessitate ad-hoc modeling of the input. While these approaches are related, they cannot be translated to HMP in a straightforward manner. We do not want to apply a motion to a new skeleton (where there is a clear separation), but rather observe the past of a previously unseen skeleton and predict its future in the same format. Our model must also extract semantic information from that past in a format that is both unseen and non-textual.

Motion Search and Tokenization In the attempt to categorize or semantically represent motions to allow grouping, motion transfer, or motion search, we see some works discussing topology-agnosticism. A very recent Arxiv work [129] deals with different animal or character skeletons by leveraging text, meshes, and delivering a unique representation token for the whole motion regardless of the skeleton format. Precisely, their representation is kinematics invariant, not equivariant, a consequence of the differences in problem statement. Another work [57] addresses motion search and classification with a disentangled latent space for retargeting and character animation with a GCN-based architecture.

However, it requires training pairs that showcase the same motion but with different kinematics, which would be a significant drawback in our case since such pairs are not directly available.

B About Lemma 1: Weights Dependent on the Kinematics Cardinality

As we observe in the main paper body with Lemma 1, the weights $\mathbf{W}$ learned by a model cannot depend on the number of joints seen at training, if the model wants to generalize to arbitrary kinematic chains $\mathcal{K}$. Before we define the notation to prove this equation, we stress here a significant challenge of the SHMP that influenced the property of equivariance w.r.t. joint dimension in previous works: dealing with 4D data.

B.1 Dealing with 4D Data.

The input $\mathbf{X} \in \mathbb{R}^{T_P \times J \times 3}$ to SHMP is multidimensional by definition, as it includes temporal and spatial information (4D). In addition to these two dimensions, the joint dimension J represents an additional degree. Together with the XYZ 3D dimensions, it delivers spatial information and is thus often merged together [18, 26, 108, 119, 136], but semantically it can represent an additional orthogonal dimension. Choosing how to deal with these dimensions is a challenge intrinsic to SHMP. Extracting meaningful features for SHMP means combining successfully temporal, spatial and joint information. In the process, the motion may be reshaped and the joint dimension fused with others implicitly fixing the joint set. Any operation of this kind effectively treats the joints as features [9, 18, 26, 78, 108, 119, 136]. In terms of choosing a suitable architecture for the problem, this translates to, for example, having multiple dimensions available as token dimension for a transformer architecture. If the time dimension is chosen as the token dimension, joints are consequently treated as features and thus not in an equivariant manner. This is the case for all previous transformer approaches [18, 108], where joints are treated as tokens just alternately in subcomponents [23, 108] or at intermediate stages [26, 78, 124].

B.2 Counterexample for Lemma 1

For the sake of discussion, let us reshape an input motion $\mathbf{M} \in \mathbb{R}^{T \times J \times 3}$ to $\mathbf{X} \in \mathbb{R}^{J \times F}$. Here $F = T \cdot 3$, but in the following, for simplicity, we omit the XYZ dimension, and continue our discussion with $F = T$ without loss of generalization. For such a two-dimensional input, network operations for feature extraction can be expressed in terms of a basic matrix multiplication $o()$ as follows:

$$o(\mathbf{X}) = \mathbf{W}\mathbf{X}\mathbf{G}, \quad (6)$$

where $\mathbf{W} \in \mathbb{R}^{m_1 \times J}$, and $\mathbf{G} \in \mathbb{R}^{F \times m_2}$, with m_1, m_2 arbitrary dimensions. If $\mathbf{W}$ is learned from data, we are then training with a kinematics $\mathcal{K} \in \mathbb{K}_{\text{train}}$ of cardinality $J = |\mathcal{K}|$ or with multiple kinematics $\mathcal{K}^1, \dots, \mathcal{K}^B \in \mathbb{K}_{\text{train}}$ where the kinematics

with the largest number of joints has cardinality $J = \max_{i \in \{1, \dots, B\}} |\mathcal{K}^i|$. When performing inference on a motion $\mathbf{X}'$ having kinematics $\mathcal{K}' \notin \mathbb{K}_{\text{train}}$

$$o(\mathbf{X}') = \mathbf{W}\mathbf{X}'\mathbf{G} \text{ is undefined if } |\mathcal{K}'| \neq J. \quad (7)$$

To be precise, in our mathematical setting, multiplication with learned matrices can only take place *from the right* i.e. joint-independent feature extraction, and not from the left. We can indeed easily see that, in our case, right multiplications $r(\mathbf{X})$ are permutation equivariant

$$\mathbf{P}r(\mathbf{X}) = \mathbf{P}(\mathbf{X}\mathbf{G}) = (\mathbf{P}\mathbf{X})\mathbf{G} = r(\mathbf{P}\mathbf{X}), \quad (8)$$

due to the associative property of matrix multiplication. In contrast, left multiplications $l(\mathbf{X})$ are not permutation equivariant

$$\mathbf{P}l(\mathbf{X}) = \mathbf{P}(\mathbf{W}\mathbf{X}) \neq \mathbf{W}(\mathbf{P}\mathbf{X}) = l(\mathbf{P}\mathbf{X}), \quad (9)$$

because matrix multiplication is in general a non-commutative operation.

B.3 Counterexamples for Previous SMP Approaches.

In the following, we lead back mathematical operations employed by previous approaches to the previous equation Eq. (6), which proves Lemma 1 by counterexample. We categorize these operations by the architecture they stem from. Overall, we can categorize these approaches in two groups: Approaches that treat joints as features (1,3,4), and approaches that rely on graph convolutions but learn the aggregation (2,5).

1. **MLP or Linear Layer.** Employed by [9, 26]. In our notation, this equals treating the joints as features i.e. multiplication from the left:

$$o_{MLP}(\mathbf{X}') = \mathbf{W}\mathbf{X}' \text{ is undefined if } |\mathcal{K}'| \neq J. \quad (10)$$

2. **Graph Convolutions with Learned Aggregation Matrix.** Employed by [26, 78, 108, 128], where $\mathbf{W}$ is learned. In our notation, this is coincident to our main example, with an additional sum of joint-independent features $\tilde{\mathbf{F}} \in \mathbb{R}^{F \times m_2}$.

$$o_{GC_{\text{learned}}}(\mathbf{X}') = \mathbf{W}\mathbf{X}'\mathbf{G} + \tilde{\mathbf{F}} \text{ is undefined if } |\mathcal{K}'| \neq J. \quad (11)$$

3. **Recurrent Units (GRU, LSTM).** Employed to extract temporal features in [9, 119, 136]. Any gate of a recurrent network (i.e. input gate, forget gate, output gate, etc.) is applied to a single pose $\mathbf{X}_t \in \mathbb{R}^{J \times 1}$ i.e. to each timestep t of sequence, and relies eventually on a hidden state $\mathbf{H}_t \in \mathbb{R}^{m_1 \times 1}$. In our notation, with the hidden state weight $\mathbf{G}_{RNN} \in \mathbb{R}^{m_1 \times m_1}$ it is expressed as

$$o_{RNN}(\mathbf{X}'_t) = \mathbf{W}\mathbf{X}'_t + \mathbf{G}_{RNN}\mathbf{H}_{t-1} \text{ is undefined if } |\mathcal{K}'| \neq J. \quad (12)$$

4. **Transformers with Time as Token Dimension and Joints as Features.** Employed by [18, 108]. Since the query, key, and value matrices are computed via linear layers i.e. multiplication from the left (see case 1) with learned weight matrices $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V \in \mathbb{R}^{m_1 \times J}$, this case also does not comply with the Lemma.

$$o_{Transf}(\mathbf{X}') = \text{softmax} \left(\frac{\sigma(\mathbf{W}_Q \mathbf{X}') \sigma(\mathbf{W}_K \mathbf{X}')^\top}{\sqrt{F_o}} \right) \sigma(\mathbf{W}_V \mathbf{X}') \quad (13)$$

is undefined if $|\mathcal{K}'| \neq J$.

Here $\sigma()$ represents the non-linear activation function of choice.

5. **Typed-Graph Convolutions.** A popular choice in SHMP is learning typed weights $\mathbf{G}_{\tau(j)} \in \mathbb{R}^{F \times m_2}$, where each joint is assigned to a type $\tau(j) \in \mathbb{T}$ (shoulder, hip, elbow, etc.) [23, 102] regardless of the body side (left or right). Here, the motion of a single joint is expressed as $\mathbf{X}_j \in \mathbb{R}^F$, and $\mathbb{T}_{train}$ is the set of joint types seen at training.

$$o_{GC_{typed}}(\mathbf{X}') = \mathbf{W} \begin{bmatrix} \mathbf{X}'_{\tau(0)} \mathbf{G}_0 \\ \mathbf{X}'_{\tau(1)} \mathbf{G}_1 \\ \vdots \end{bmatrix} \quad \text{is undefined if } \tau(j') \notin \mathbb{T}_{train}. \quad (14)$$

Regardless of whether the weights $\mathbf{W}$ are learned, this paradigm does naturally not generalize to joint whose types have not been seen during training (*e.g.* as when training on H36M, that has no feet, and testing on kinematics that have feet Tab. 11).

As an interesting side note for future works, the widely employed Discrete Cosine Transform (DCT) [18, 108], does not contradict Lemma 1 per se, as it acts only on the temporal domain. However, up to date, it is usually paired only with approaches that treat the joints as features.

C On Permutation Equivariance

C.1 Mathematical Proof of Equivariance Fulfilling Lemma 1

Given an input $\mathbf{X} \in \mathbb{R}^{J \times F}$ and a permutation matrix $\mathbf{P} \in \mathbb{R}^{J \times J}$ that reorders the joints, an operation $o(\mathbf{X}) = \mathbf{W}\mathbf{X}\mathbf{G}$ as described in Sec. B is permutation equivariant if:

$$\mathbf{P}o(\mathbf{X}) = o(\mathbf{P}\mathbf{X}) \quad (15)$$

Here we show that an operation $o()$ that is permutation equivariant w.r.t. $\mathbf{P}$ naturally fulfils Lemma 1.

Let us substitute the operation $o()$ into the equivariance definition:

$$\begin{aligned} \mathbf{P}(\mathbf{W}\mathbf{X}\mathbf{G}) &= \mathbf{W}(\mathbf{P}\mathbf{X})\mathbf{G} \\ \mathbf{P}\mathbf{W}\mathbf{X}\mathbf{G} &= \mathbf{W}\mathbf{P}\mathbf{X}\mathbf{G} \\ \mathbf{P}\mathbf{W} &= \mathbf{W}\mathbf{P} \end{aligned} \quad (16)$$

Thus, equivariance is fulfilled if it holds $\mathbf{PW} = \mathbf{WP}$. Let us consider a general case where the permutation matrix $\mathbf{P}$ swaps joints i and j , but not joint j i.e.

$$\mathbf{P}_{kl} = \begin{cases} 1 & \text{if } (k, l) \in \{(i, j), (j, i)\} \\ 1 & \text{if } k = l \text{ and } k \notin \{i, j\} \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

We now consider different entries of the equation $\mathbf{PW} = \mathbf{WP}$ in dependence of the indices i, j, k .

1. **Swapped Indices (i, j) .** We first consider the entry (i, j) where joints have been swapped.

$$\begin{aligned} (\mathbf{PW})_{ij} &= (\mathbf{WP})_{ij} \\ \sum_m^J \mathbf{P}_{im} \mathbf{W}_{mj} &= \sum_m^J \mathbf{W}_{im} \mathbf{P}_{mj} \\ \mathbf{P}_{ij} \mathbf{W}_{jj} + \sum_{m \neq j}^J \mathbf{P}_{im} \mathbf{W}_{mj} &= \mathbf{W}_{ii} \mathbf{P}_{ij} + \sum_{m \neq i}^J \mathbf{W}_{im} \mathbf{P}_{mj} \\ 1 \cdot \mathbf{W}_{jj} + 0 &= \mathbf{W}_{ii} \cdot 1 + 0 \\ \mathbf{W}_{jj} &= \mathbf{W}_{ii} \end{aligned} \quad (18)$$

2. **Unswapped Indices (i, k) .** We now consider the entry (i, k) where joints have not been swapped.

$$\begin{aligned} (\mathbf{PW})_{ik} &= (\mathbf{WP})_{ik} \\ \sum_m^J \mathbf{P}_{im} \mathbf{W}_{mk} &= \sum_m^J \mathbf{W}_{im} \mathbf{P}_{mk} \\ \mathbf{P}_{ij} \mathbf{W}_{jk} + \sum_{m \neq j}^J \mathbf{P}_{im} \mathbf{W}_{mk} &= \mathbf{W}_{ik} \mathbf{P}_{kk} + \sum_{m \neq k}^J \mathbf{W}_{im} \mathbf{P}_{mk} \\ \mathbf{W}_{jk} &= \mathbf{W}_{ik} \end{aligned} \quad (19)$$

3. **Unswapped Indices (j, k) .** We now consider the entry (j, k) where joints have not been swapped.

$$\begin{aligned} (\mathbf{PW})_{jk} &= (\mathbf{WP})_{jk} \\ \sum_m^J \mathbf{P}_{jm} \mathbf{W}_{mk} &= \sum_m^J \mathbf{W}_{jm} \mathbf{P}_{mk} \\ \mathbf{P}_{ji} \mathbf{W}_{ik} + \sum_{m \neq i}^J \mathbf{P}_{jm} \mathbf{W}_{mk} &= \mathbf{W}_{jk} \mathbf{P}_{kk} + \sum_{m \neq k}^J \mathbf{W}_{jm} \mathbf{P}_{mk} \\ \mathbf{W}_{ik} &= \mathbf{W}_{jk} \end{aligned} \quad (20)$$

From these equations we just obtained conditions for the weight matrix. According to the first equality Eq. (18), it follows that all diagonal entries must have the same value. For the off-diagonal elements, from the second equality Eq. (19) it follows that all elements of a column must be equal, and from the second equality Eq. (20) it follows that all elements of a row must be equal. This means that all off-diagonal entries have the same value. Therefore, every diagonal entry is represented by a scalar α and each off-diagonal entry by a scalar β :

$$\begin{aligned} \mathbf{W}_{ij} &= \alpha \delta_{ij} + \beta(1 - \delta_{ij}), \quad \text{or} \\ \mathbf{W} &= \alpha \mathbb{I} + \beta \mathbf{1}\mathbf{1}^\top, \end{aligned} \quad (21)$$

where δ is the Kronecker delta. With such learned weights, we formulate the permutation equivariant operation $o_{PEQ}()$ as

$$o_{PEQ}(\mathbf{X})_j = \alpha \mathbf{X}_j + \beta \cdot \sum_{m \neq j}^J \mathbf{X}_m \quad (22)$$

Let us thus show that such a permutation equivariant operation by design fulfills the Lemma 1: the learned weights are not dependent on the number of joints. First, the summation in Eq. (22) is invariant to the number of joints, which is a property of any sum operation. Second, the set of learnable parameters of this operation is

$$\Theta = \{\alpha, \beta\} \quad \text{and} \quad |\Theta| = 2 \quad (23)$$

Here we see that $|\Theta| = 2$ is not dependent of J . Indeed,

$$\frac{d|\Theta|}{dJ} = \frac{d(2)}{dJ} = 0 \implies |\Theta| \in O(1) \quad (24)$$

Thus the number of parameters does not scale with J , but is constant:

$$\forall J \in \mathbb{N}, \quad \dim(\Theta) = \text{const.} \quad (25)$$

The Lemma is thus naturally satisfied: equivariance is a sufficient condition, even if not a necessary one. Instead, any learned matrix $\mathbf{W}$ that does not fulfill the equivariance constraint, has the set of learned parameters $\Theta_{non-PEQ} = \{w_{ij} \mid i, j \in \{1, \dots, J\}\}$, implying $|\Theta_{non-PEQ}| = J^2$ and that the number of learned parameters grows with the size of the kinematics $|\Theta_{non-PEQ}| \in O(J^2)$, which contradicts the Lemma.

C.2 Mathematical Proof of Equivariance for EquiFusion

A model is permutation equivariant with respect to a permutation matrix $\mathbf{P}$ if all its operations are permutation equivariant. Here we prove mathematically that our graph convolution Eq. (3) and self-attention Eq. (4) operations are equivariant.

Graph Convolution. For simplicity, we report again here Eq. (3):

$$c(\mathbf{Z}, \mathbf{A}) = \mathbf{D}^{-1} \mathbf{A} \mathbf{Z} \mathbf{G}_1 + \mathbf{Z} \mathbf{G}_0 + \mathbf{b} \quad \text{with} \quad \mathbf{D} = \text{diag}(\mathbf{A} \mathbf{1}) \quad (26)$$

We want to prove

$$c(\mathbf{PZ}, \mathbf{PAP}^\top) = \mathbf{P} c(\mathbf{Z}, \mathbf{A}) \quad (27)$$

and start from the left side:

$$\begin{aligned} c(\mathbf{PZ}, \mathbf{PAP}^\top) &= \mathbf{D}_P^{-1} \mathbf{P} \mathbf{A} \underbrace{\mathbf{P}^\top \mathbf{P}}_{\mathbb{I}} \mathbf{Z} \mathbf{G}_1 + \mathbf{PZ} \mathbf{G}_0 + \mathbf{b} \\ &= (\mathbf{P} \mathbf{D} \mathbf{P}^\top)^{-1} \mathbf{P} \mathbf{A} \mathbf{Z} \mathbf{G}_1 + \mathbf{PZ} \mathbf{G}_0 + \mathbf{b} \\ &= (\mathbf{P}^\top)^{-1} \mathbf{D}^{-1} \mathbf{P}^{-1} \mathbf{P} \mathbf{A} \mathbf{Z} \mathbf{G}_1 + \mathbf{PZ} \mathbf{G}_0 + \mathbf{b} \\ &= \mathbf{P} \mathbf{D}^{-1} \underbrace{\mathbf{P}^\top \mathbf{P}}_{\mathbb{I}} \mathbf{A} \mathbf{Z} \mathbf{G}_1 + \mathbf{PZ} \mathbf{G}_0 + \mathbf{b} \\ &= \mathbf{P} \mathbf{D}^{-1} \mathbf{A} \mathbf{Z} \mathbf{G}_1 + \mathbf{PZ} \mathbf{G}_0 + \mathbf{b} \\ &= \mathbf{P} (\mathbf{D}^{-1} \mathbf{A} \mathbf{Z} \mathbf{G}_1 + \mathbf{Z} \mathbf{G}_0 + \mathbf{b}) \\ &= \mathbf{P} c(\mathbf{Z}, \mathbf{A}) \end{aligned} \quad (28)$$

where we used $\mathbf{D}_P = \text{diag}(\mathbf{PAP}^\top \mathbf{1}) = \mathbf{P} \mathbf{D} \mathbf{P}^\top$ in the second equality and $\mathbf{Pb} = \mathbf{b}$ (since $\mathbf{b} \in \mathbb{R}^{1 \times F_0}$ and $\mathbf{P} \in \mathbb{R}^{J \times J}$) in the last, and $\mathbf{P}^{-1} = \mathbf{P}^\top$ in general.

Self-Attention Here we report the self-attention operation of the main paper body Eq. (4)

$$\begin{aligned} \text{Att}(\mathbf{Z}, \mathbf{A}) &= \text{softmax} \left(c_Q(\mathbf{Z}, \mathbf{A}) c_K(\mathbf{Z}, \mathbf{A})^\top / \sqrt{F_o} \right) c_V(\mathbf{Z}, \mathbf{A}) \\ &= \text{softmax} \left(\mathbf{Q} \mathbf{K}^\top / \sqrt{F_o} \right) \mathbf{V}, \end{aligned} \quad (29)$$

where the query, key and value matrices are defined as $\mathbf{Q} = c_Q(\mathbf{Z}, \mathbf{A})$, $\mathbf{K} = c_K(\mathbf{Z}, \mathbf{A})$, and $\mathbf{V} = c_V(\mathbf{Z}, \mathbf{A})$ respectively. We want to prove:

$$\text{Att}(\mathbf{PZ}, \mathbf{PAP}^\top) = \mathbf{P} \text{Att}(\mathbf{Z}, \mathbf{A}) \quad (30)$$

We already proved with Eq. (28) that the convolution $c(\cdot)$ is equivariant as defined in Eq. (26) and thus we can employ Eq. (26). From the left

$$\begin{aligned}
 & \text{Att}(\mathbf{PZ}, \mathbf{PAP}^\top) \\
 &= \text{softmax} \left(\frac{c_Q(\mathbf{PZ}, \mathbf{PAP}^\top) c_K(\mathbf{PZ}, \mathbf{PAP}^\top)^\top}{\sqrt{F_o}} \right) c_V(\mathbf{PZ}, \mathbf{PAP}^\top) \\
 &= \text{softmax} \left(\frac{\mathbf{PQ}(\mathbf{PK})^\top}{\sqrt{F_o}} \right) \mathbf{PV} \\
 &= \text{softmax} \left(\frac{\mathbf{PQK}^\top \mathbf{P}^\top}{\sqrt{F_o}} \right) \mathbf{PV} \\
 &= \mathbf{P} \text{softmax} \left(\frac{\mathbf{QK}^\top}{\sqrt{F_o}} \right) \underbrace{\mathbf{P}^\top \mathbf{P}}_{\mathbf{I}} \mathbf{V} \\
 &= \mathbf{P} \left(\text{softmax} \left(\frac{\mathbf{QK}^\top}{\sqrt{F_o}} \right) \mathbf{V} \right) \\
 &= \mathbf{P} \text{Att}(\mathbf{Z}, \mathbf{A})
 \end{aligned} \tag{31}$$

where we considered that the softmax operation is equivariant to permutation, because its denominator (the sum of exponentials) is a commutative operation that remains constant regardless of the order of the elements.

End-to-End Equivariance. Since our network consists of these two equivariant operations and residual layers, as can be seen in Fig. 6 and is discussed in Sec. D.4, our architecture is end-to-end permutation equivariant.

C.3 Equivariance in a Generative Diffusion Model

Let's discuss the implications of a generative formulation such as denoising diffusion models [42] in the context of permutation equivariance. In a diffusion model, a sample is generated from random noise $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ sampled from a univariate Gaussian distribution. Since the architecture of our denoiser is end-to-end permutation equivariant with respect to the permutation matrix $\mathbf{P}$, it naturally holds:

$$f(\mathbf{PX}, \mathbf{PAP}^\top, \mathbf{P}\varepsilon) = \mathbf{P}f(\mathbf{X}, \mathbf{A}, \varepsilon). \tag{32}$$

We refer to this equation Eq. (32) as equivariance under "fixed noise", and prove this numerically as *Ours on $\mathbf{P} + \mathbf{P}\varepsilon$* in Tab. 4. In this case, equivariance holds strictly, as fixing the noise results in a deterministic mapping. Instead, when the noise is not fixed, but randomly sampled as in every generative model, equivariance is said to hold on a distributional level and not on a sample level [67, 112, 118]. We elaborate on this concept starting from a straightforward example. We sample two different noise variables $\varepsilon_1, \varepsilon_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and obtain two different samples, according to the definition of a generative model:

$$f(\mathbf{X}, \mathbf{A}, \varepsilon_1) \neq f(\mathbf{X}, \mathbf{A}, \varepsilon_2). \tag{33}$$

Indeed, this core aspect is what allows us to generate novel latents from different noise samples. Of course, this holds also in the case of permutation

$$f(\mathbf{P}\mathbf{X}, \mathbf{P}\mathbf{A}\mathbf{P}^\top, \varepsilon_1) \neq \mathbf{P}f(\mathbf{X}, \mathbf{A}, \varepsilon_2). \quad (34)$$

However, when considering not only a single sample but the whole test set, the output distributions of $f(\mathbf{P}\mathbf{X}, \mathbf{P}\mathbf{A}\mathbf{P}^\top)$ and $f(\mathbf{X}, \mathbf{A})$ have identical shape, up to the transformation $\mathbf{P}$ that reorders the joint dimensions

$$p_{f(\mathbf{P}\mathbf{X}, \mathbf{P}\mathbf{A}\mathbf{P}^\top)} \simeq \mathbf{P}p_{f(\mathbf{X}, \mathbf{A})}. \quad (35)$$

We refer to this property as *distributional* equivariance or equivariance *under stochastic sampling* [67, 112, 118]

$$f(\mathbf{P}\mathbf{X}, \mathbf{P}\mathbf{A}\mathbf{P}^\top) \stackrel{d}{=} \mathbf{P}f(\mathbf{X}, \mathbf{A}). \quad (36)$$

and prove it numerically as *Ours on $\mathbf{P}$* in Tab. 4.

Another interesting aspect here is the following. Since a univariate Gaussian distribution is invariant to permutation, applying the permutation $\mathbf{P}$ to a fresh random sample delivers another sample from the same distribution $\mathbf{P}\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. In our setting, this means that computing the evaluation metrics for $f(\mathbf{P}\mathbf{X}, \mathbf{P}\mathbf{A}\mathbf{P}^\top)$ is coincident to computing the metrics of $f(\mathbf{X}, \mathbf{A})$ for a different initialization seed.

C.4 Evidence that Previous SHMP Approaches are not Equivariant

Previous SHMP approaches are not equivariant because they either learn the aggregation matrix for a specific kinematics, or because they treat the joints as features overfitting to the training kinematics. We provide experimental quantitative results in Tab. 7 and mathematical proof following the paradigm of Sec. B.

Empirical Evidence via Quantitative Experiments (Tab. 7). In this section, we highlight the results in Tab. 7, showing that current state-of-the-art HMP models are by far not permutation equivariant (PEQ). We consider models trained on a single Kinematics and its corresponding dataset, AMASS [76], the largest and most diverse dataset in SHMP. When testing on the AMASS test split, which has the same kinematics as at training time, we apply a random permutation of the body joints in the input sequences. We see that previous SHMP approaches fail dramatically, since the metrics strongly differ between conventional evaluation (as reported in Tab. 17) and the evaluation under permutation (*On $\mathbf{P}$*). This behavior is to be expected, as previous approaches were never trained for permutations and for each kinematics the input joints are expected in a predefined order, matching the one seen at training time. Modifying any of these methods to an equivariant architecture is not straightforward (see Sec. B). Instead, our method has the only equivariant architecture and thus it is equivariant by design. The metrics for EquiFusion exhibit very low variation: as expected, our results are close but not identical under random permutation at inference time. The reason is that any generative model is permutation equivariant on a distribution level, as discussed in Sec. C.3. The two evaluations can thus be interpreted as evaluations under different initial random seeds.

Table 7: Evaluation with random permutation at inference time. For each method, we report 1) conventional evaluation metrics on AMASS dataset [76] as in Tab. 17, 2) Evaluation on the same data but with body joints randomly permuted (*On P*). Previous HMP approaches fail dramatically, as our method has the only permutation equivariant architecture. No model has seen permutations during training, and all models are trained on the same kinematics and dataset, AMASS. The most constant values under permutation are highlighted in **bold**, second-best are underlined.

Method	Precision ↓			Div ↑	Real ↓	B Real ↓	
	ADE	FDE	MAE	APD	CMD	str	jit
DLow [136]	0.590	0.612	8.510	13.170	15.185	8.41	0.40
on P	2.389	2.491	65.179	13.137	44.113	1.87	0.03
DivSamp [26]	0.564	0.647	8.027	24.724	50.239	11.17	0.82
on P	4.167	1.896	53.504	37.297	10171.783	2.73	0.96
BeLFusion [9]	0.513	0.560	7.125	9.376	16.995	7.19	0.34
on P	1.530	1.719	54.826	11.902	40.985	1.85	0.04
CoMusion [108]	0.494	0.547	6.715	10.848	9.636	4.04	0.25
on P	2.282	2.456	64.120	15.229	23.136	1.88	0.04
SkelDiff [23]	0.480	0.545	6.124	9.456	11.417	3.15	0.20
on P	2.041	2.440	64.339	3.371	29.083	153.52	2.57
EquiFusion (A)	0.496	0.560	6.214	8.241	13.097	0.00	0.00
on P	0.497	0.562	6.221	8.242	13.089	0.00	0.00

Mathematical Evidence and Discussion . In the following, we provide mathematical proofs of why current methods are not equivariant.

1. **MLP or Linear Layer.** Since the weights $\mathbf{W}$ are learned for specific joint indices, swapping rows in the input does not swap the corresponding rows in the output.

$$o_{MLP}(\mathbf{P}\mathbf{X}') = \mathbf{W}(\mathbf{P}\mathbf{X}') \neq \mathbf{P}(\mathbf{W}\mathbf{X}') \quad \text{because} \quad \mathbf{W}\mathbf{P} \neq \mathbf{P}\mathbf{W} \quad (\text{in general}). \quad (37)$$

2. **Graph Convolutions with Learned Aggregation Matrix.** Even with a spatial aggregation matrix $\mathbf{G}$, the left-multiplication by learned weights $\mathbf{W}$ ties features to specific indices.

$$o_{G\text{learned}}(\mathbf{P}\mathbf{X}') = \mathbf{W}(\mathbf{P}\mathbf{X}')\mathbf{G} + \tilde{\mathbf{F}} \neq \mathbf{P}(\mathbf{W}\mathbf{X}'\mathbf{G} + \tilde{\mathbf{F}}). \quad (38)$$

3. **Recurrent Units (GRU, LSTM).** Because the hidden state $\mathbf{H}_{t-1}$ and the input transformation $\mathbf{W}$ are fixed to a specific joint ordering, permuting the input vector $\mathbf{X}'_t$ breaks the alignment with the learned parameters.

$$o_{RNN}(\mathbf{P}\mathbf{X}'_t) = \mathbf{W}(\mathbf{P}\mathbf{X}'_t) + \mathbf{G}_{RNN}\mathbf{H}_{t-1} \neq \mathbf{P}(\mathbf{W}\mathbf{X}'_t + \mathbf{G}_{RNN}\mathbf{H}_{t-1}). \quad (39)$$

4. **Transformers with Time as Token Dimension and Joints as Features.** Since the projections to Query, Key, and Value spaces are linear layers acting on the joint dimension (multiplication from the left), the attention map becomes corrupted under permutation.

$$o_{Tranf}(\mathbf{P}\mathbf{X}') \propto \text{attn}(\mathbf{W}_Q\mathbf{P}\mathbf{X}', \mathbf{W}_K\mathbf{P}\mathbf{X}')(\mathbf{W}_V\mathbf{P}\mathbf{X}') \neq \mathbf{P}o_{Tranf}(\mathbf{X}'). \quad (40)$$

Additionally, current transformer-based approaches employ positional encoding, which is by definition not equivariant as it is designed to contain order information. When employing attention, the token dimension is typically associated with time [18, 108], and just alternately in subcomponents [23, 108] or at intermediate stages [26, 78, 124] with joints.

5. **Typed-Graph Convolutions.** Because each joint index j is mapped to a specific weight $\mathbf{G}_j$ based on its type $\tau(j)$, permuting the joints $\mathbf{P}\mathbf{X}$ moves a joint of one type (e.g., "hip") into a slot evaluated by a weight for another type (e.g., "shoulder").

$$o_{GC_{typed}}(\mathbf{P}\mathbf{X}') = \mathbf{W} \begin{bmatrix} (\mathbf{P}\mathbf{X}')_0 \mathbf{G}_0 \\ (\mathbf{P}\mathbf{X}')_1 \mathbf{G}_1 \\ \vdots \end{bmatrix} \neq \mathbf{P}o_{GC_{typed}}(\mathbf{X}'). \quad (41)$$

These architectures cannot be straightforwardly rendered PEQ by removing or adapting components, and ultimately rely on a fixed joint number and ordering.

D More Details on EquiFusion

D.1 Training Losses

We follow the training paradigm of latent diffusion models [98]: we first train an autoencoder to learn a temporally-compressed latent space and then a denoiser to denoise true latent variables in that latent space.

Autoencoder: Reconstruction Loss. Similar to other approaches [9, 23], the autoencoder learns a latent space by reconstructing complete motion sequences from their latent representations. Given a motion $\mathbf{M} \in \mathbb{R}^{T \times N \times 3}$, the encoder compresses it into a latent vector $\mathbf{z} \in \mathbb{R}^{N \times L}$, and the decoder reconstructs the motion $\tilde{\mathbf{M}}$ from this latent code. The model is optimized following the reconstruction loss

$$\mathcal{L}_{\text{rec}}(\mathbf{M}, \tilde{\mathbf{M}}) := \|\mathbf{M} - \tilde{\mathbf{M}}\|_1. \quad (42)$$

Denoiser: Diffusion Loss. In the diffusion training process $q(\mathbf{z}_t | \mathbf{z}_{t-1})$, a clean latent sample $\mathbf{z}_0$ is progressively perturbed over timesteps $t = 1, \dots, T$ by adding noise of magnitude proportional to t according to a cosine noise scheduler, yielding a Gaussian distribution at $t = T$. In the generative denoiser learns an approximation of the reverse process $p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t)$, which iteratively denoises $\mathbf{z}_T$

back towards the data distribution. For each diffusion timestep t , the denoiser regresses directly the denoised latent $\mathbf{z}_\theta$ [9, 23, 96, 108], rather than the added noise [42, 98]. To avoid penalizing samples that are different from the ground truth yet realistic, we relax the diffusion objective [40] by sampling $k = 50$ times and backpropagating the loss to the closest sample [9, 23]:

$$\mathcal{L}_{\text{diff}}(\mathbf{z}_\theta^k, \mathbf{z}) = \mathbb{E}_{\mathbf{Y}, \mathbf{X}, t} \arg \min_k (\bar{\alpha}_t \|\mathbf{z}_\theta^k - \mathbf{z}\|). \quad (43)$$

As this relaxation achieves higher diversity in the predictions but extends training time, we thus present some of our ablations without relaxation $k = 1$ [9, 23].

D.2 Inference on Partial Skeletons

Our method has never seen partial skeletons with missing joints during training. While partiality is highly relevant for real-world applications, data collection is not straightforward, and there are no specific SHMP datasets to date. Partial motions are thus usually investigated by masking limbs of motions parametrized with existing full-body kinematics. We follow this procedure and mask the input (both motion and adjacency) by randomly picking limb IDs.

EquiFusion has never been trained for the generation of missing limbs either, but we observe this capability as a side effect. We generate missing limbs with a simple heuristic as a proof of concept. For an input observation $\mathbf{X}$ with missing limbs, we obtain its latent embedding as $\mathbf{z}_{\text{past}}$ with the corresponding partial adjacency matrix $\mathbf{A}$. In the latent embedding, joints and limbs that were not present in the input are also not present. Since we know the adjacency matrix of the full-body skeleton, we can employ it in the diffusion process and in the decoder. During diffusion, we can effectively generate the missing parts by sampling noise also for the missing limbs. But to do this, since the diffusion process uses the observation latent as conditioning, we need to find an initialization value for the missing limbs in the latent representation of the past motion. To increase realism in the generated body part, we initialize the conditioning past latent through the values of the opposite limbs. In the qualitatives, we see that this does not result in symmetric motions for the generated output. We believe more ad-hoc or sophisticated inference heuristics, or rather training strategies, can be applied to target this issue, and leave it as an object of future work.

D.3 Parametrizing Motion as Bone Directions

Here, we provide more details about our motion parametrization as bone directions. An overview is provided in Fig. 5, displaying how this parameterization can be applied to any HMP model, not only EquiFusion. We feed the limb direction vectors to the encoder, allowing the network to access both orientation and implicit bone length information, since the directions are not normalized. For decoding, the model predicts unnormalized limb directions, which are then normalized and rescaled by the bone lengths of the input skeleton. This effectively corresponds to predicting pure limb directions while enforcing constant

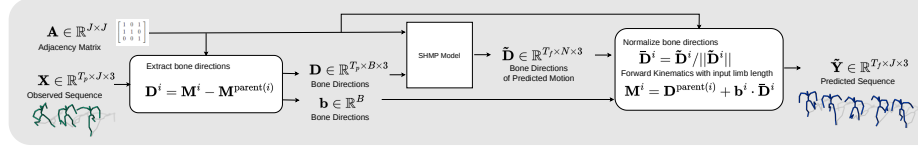


Fig. 5: Pipeline overview of parametrizing motions as bone directions. We also depict the masking procedure of EquiFusion. The parametrization can be applied to any SHMP model without additional modification. 3D keypoints are computed from bone directions via inverse kinematics.

bone lengths, thereby removing length jitter trivially and ensuring geometric consistency in the reconstructed motions. During our preliminary studies we discovered that both feeding and computing the loss on the unnormalized bone directions instead of the normalized version improves training stability and delivers better performance. This representation remains numerically stable during training [11] and does not suffer from singularities [35, 102].

D.4 Architecture Details

Overview In this section, we provide a detailed description of the input-output flow of our model and the architecture of the autoencoder and denoiser. A visualization is given in Fig. 6. We remark here that we will make our code public.

Single Networks

Encoder The challenge of a fully PEQ architecture for HMP consists in extracting meaningful temporal and spatial information without breaking the PEQ property. One could extract such information in parallel for time and joint dimensions (inspired by Google Inception architecture), which requires a higher amount of resources and slower forward passes. In preliminary experiments, we observed that this approach yielded no benefits. This provided inspiration for parallel branches, which resulted in slightly improved reconstruction and faster convergence during autoencoder training. While the first parallel branch halves the initial feature dimension of $3T$, the residual message-passing block before the second parallel branch reduces the feature dimension to L . To allow for the processing of sequences of arbitrary length, we employ zero-padding along the time dimension of the encoder’s input. This allows us to take the training stage II and at inference, use the past motion as input, following previous works [23] and save resources for training a network solely for the past [9]. The autoencoder is never trained with the gradient from the past observation.

Decoder For the decoder, we stack two transformer blocks, followed by two residual message passing blocks each. An initial residual message passing block is used to increase the feature dimension from L to $3T$.

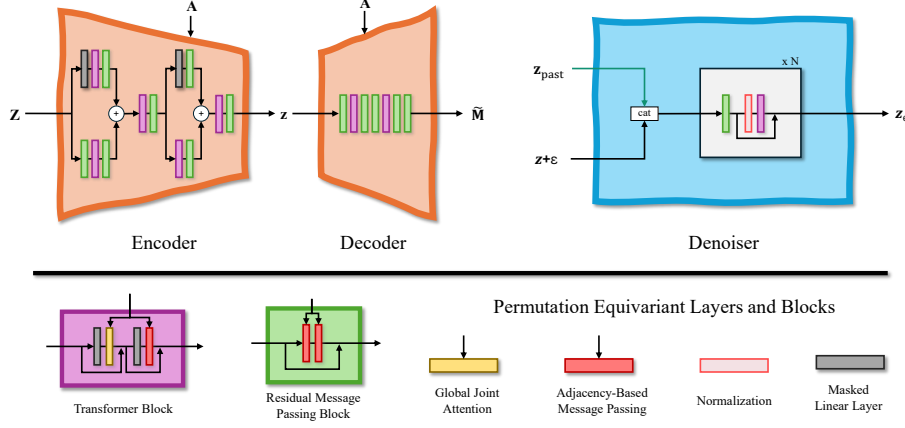


Fig. 6: Overview of the architecture layers of our model. We provide a more detailed view on the permutation equivariant architecture of our model, including layers and connections described in Sec. D.4. Our code will be publicly available.

Denoiser In the denoiser, we concatenate the conditioning latent vector of the past motion z_{past} with the current latent vector z before feeding it into the blocks. Each block consists of a message passing block and a transformer block with a skip connection, with Root Mean Square Layer normalization as commonly paired with transformers.

Layers

Adjacency-Based Message Passing This layer performs the graph convolution operation described in Eq. (3).

Transformer Block For the transformer Block, we use the attention mechanism described in Eq. (4) in the main paper, where the query, key, and value weights are extracted from the input features via graph convolutions. We add a residual connection around both blocks and nonlinearities.

Residual Message Passing Block We group pairs of two of the adjacency-based message passing layers with a hyperbolic tangent nonlinearity in between and a skip connection around both layers, following the successful fashion of conventional residual blocks.

Masked Linear Layer To ensure that our architecture remains PEQ despite masking, we employ linear layers only on the feature dimension, never on the joint dimension, ensuring that masked joints always have zeroed features.

E Implementation Details

We employ the same training hyperparameters for any dataset or dataset combination. The autoencoder is trained for 300 epochs, while the denoiser for 375 epochs with a learning rate of 0.005 and $T = 10$ diffusion steps, following a cosine noise scheduler [89]. At inference we draw from a DDPM sampler [42]. Both networks are trained with Adam on PyTorch. Our model is **always** trained for 7.9M parameters, at least twice as compact as the smallest diffusion competitor and even more compact than VAE baselines. Numbers reported with inference time in Tab. 10. We train the autoencoder on an RTX5000 and the diffusion model on an NVIDIA A40. The longest training does not take more than 5 days, shorter than the closest competitor SkelDiff, which also needs to be trained anew for every new dataset. Following previous works [23, 102], we chose a latent dimension of $L = 96$, achieving a 4x compression of the input space. When training with kinematics that have a different number of joints (i.e. AMASS and Nymeria), we zero-pad the inputs to the highest joint number, which has no implication for our equations as the weight matrices are independent on the number of joints. As the HMP task is defined for a past of 0.5 seconds and a future of 2 seconds, it results in a different number of input and output frames depending on the FPS (Hz) of each dataset. We train our model on the maximum FPS (60 Hz, as in AMASS and Nymeria), and deal with lower FPS by frame interpolation (see Sec. G.3). When training the diffusion model, to avoid spurious correlations between the noise and the adjacency matrix (of which the model sees only one (AMASS) or two instances (AMASS + Nymeria)), we include a random permutation for the joints of 0.5.

For our experiments without the bone direction parametrization, we train our model with 3D keypoints as input, employing the same rescaling approach of SkelDiff [23].

F Metrics Definition

Overview We report the precision metrics of the Average Distance Error (ADE), the Final Distance Error (FDE), and the Mean Angle Error (MAE), in degrees, together with their multimodal equivalents (MMADE, MMFDE). For diversity, we report the Average Pairwise Distance (APD) between predictions. The Average Pairwise Distance Error (APDE), relates the APD with the multimodal GT. Realism metrics are comprised of the Cumulative Motion Distribution (CMD), which penalizes deviations from the expected average displacement of the dataset and body realism in the form of limb stretching (str) and jittering (jit). Their respective mean and Root Mean Squared Error (RMSE) are reported in percentage. In the main paper body, for space reasons in most tables we report a subset of all metrics, selecting the most relevant ones. The others behave analogously, as can be seen in the corresponding extended version of each table in the App..

On AMASS, FID is conventionally not computed. The FID computation requires features from a classifier, and labels necessary to train this model for

AMASS do not exist. On H36M, we do not compute FID for the ZeroVelocity baseline as it does not output a distribution. For the detailed metrics equations, we refer to the appendix of Curreli *et al.* [23] and the main body of Barquero *et al.* [9]. For computing APDE with the different retargeting procedures, we do not perform retargeting on the reference values but keep them in the original space.

F.1 Unified Metrics: Comparing among Datasets

We realize that conventional SHMP metrics are unsuitable to compare model performance across datasets. Let's take as an example the Average Distance Error (ADE). For a set of k predictions $\tilde{\mathbf{Y}} \in \mathbb{R}^{F \times J \times 3}$, the ADE is defined as

$$\text{ADE}(\tilde{\mathbf{Y}}, \mathbf{Y}) = \min_k \frac{1}{F} \sum_{t=0}^F \sqrt{\sum_{j=1}^{J*3} ({}^k\tilde{\mathbf{Y}}_t^j - \mathbf{Y}_t^j)^2}. \quad (44)$$

We note here that the joint dimension is included in the Euclidean Distance computation, thus not averaging over the number of joints. Summing instead of averaging over the number of joints does not allow to compare ADE scores on topologies or datasets that exhibit a different number of joints.

We simply employ a ADE version *unified* among datasets, by averaging over the joint dimension:

$$u\text{ADE}(\tilde{\mathbf{Y}}, \mathbf{Y}) = \min_k \frac{1}{F} \frac{1}{J} \sum_{t=0}^F \sum_{j=1}^J \sqrt{\sum_{d=1}^3 ({}^k\tilde{\mathbf{Y}}_t^{j,d} - \mathbf{Y}_t^{j,d})^2}. \quad (45)$$

Analogously, we define $u\text{FDE}$, $u\text{MMADE}$, and $u\text{MMFDE}$ measured in decimeters; and $u\text{APD}$ measured in meters.

G Experimental Settings

Following the HMP task definition of previous works, in our experiments, we consider a past of 0.5 seconds and a future of 2 seconds. While the AMASS topology is designed for 22 joints, H36M has 17, and Nymeria 23 (including the hip joint).

G.1 Baselines

Details. Since we are the first to adapt Nymeria for the task of HMP, we also need to train previous methods on this dataset. When the code of the latest baseline was available, we trained them on Nymeria employing their configurations for AMASS. We chose this setting because the two datasets are comparable in size, and the number of keypoints differs only by one (while H36M is much smaller and has fewer joints). For HumanMAC, the checkpoint on AMASS or

its configurations is not available. Thus we increase the number of layers and the training time compared to the H36M configuration. To obtain the $uADE$, $uFDE$, and $uAPD$ metrics (Fig. 3) when the AMASS checkpoint is not available, we compute them through interpolation from their ADE, FDE, APD.

Adapting a Baseline to Multidataset Training. In Tab. 4 we presented version of SkelDiff [23] adapted to multidataset training, showing that data alone does not lead to generalization in a zero-shot kinematics setting. To adapt the baseline to multidataset we first removed all components that were dependent on specific joint *types*: 1) the weights of the typed-graph convolutions, which used to be learned independently per joint type (*e.g.* shoulders, legs, hands, etc.), were substituted by a single weight matrix learned for all joint types simultaneously, 2) the anisotropy was removed from the diffusion training, as it relied on fixed joint positions given by the adjacent matrix. Then we trained the model with joint permutation as data augmentation. This lead us to a baseline that employs similarly to us attention and graph convolutions, but in the graph convolution as described in (3) learns the aggregation matrix W from data instead of using the adjacency matrix. To support kinematics chains of different size, we set the size of this matrix equal to the largest cardinality of the kinematics in our training set. We note that this approach by definition cannot generalize at inference to kinematics chains of size larger than the ones seen at training. Beyond this specific scenario, the resulting baseline can potentially fulfill Lemma 1 in dependence of the training data.

Retargeting. In the main paper, we discuss the retargeting from AMASS to H36M. AMASS has 22 joints, while H36M has 17, so joints that do not have any correspondence in H36M are simply discarded. Converting the output back from H36M to AMASS to evaluate in the input topology would imply creating limbs and joints that are not present in H36M. To this purpose, we completed the retargeting algorithm of Holden *et al.* [43] employed by HumanMAC [18] with suitable positioning of the collarbones on the shoulder line and feet perpendicular to the leg bones, making use of the foot lengths in the observation.

G.2 Adapting Nymeria to HMP

The Nymeria dataset is a very large collection of motion data originated from egocentric motion in-the-wild through body tracking in connection with Project Aria. It is recorded for 23 joints, including the root hip joint. It contains 300 hours on a total of 1200 sequences with 264 participants, recorded in 50 indoor and outdoor locations. We find the quality of the recorded motion to be sub-optimal for motion prediction, as it often occurs in real-life situations: a large percentage of recordings display moments of stillness. To derive a dynamic level more similar to AMASS, we prune the data, removing sequence parts that have a mean joint velocity between consecutive frames inferior to 6.913 cm, for a total of 51M removed frames. We thus retain 1094 of the original sequences and split

44 of them into suparts. This leaves us with a total of 10M frames, a size comparable to AMASS. The overall data remains relatively static, as evident from the evaluation scores of the ZeroVelocity baseline on AMASS and Nymeria: the ADE on Nymeria is significantly lower, despite having one additional joint.

G.3 Inference on FPS Different From Training Time

As the HMP task is defined for a past of 0.5 seconds and a future of 2 seconds, it results in a different number of input and output frames depending on the FPS (Hz) with which each dataset was recorded. We train our model on the maximum FPS (60 Hz, as in AMASS and Nymeria), and deal with lower FPS by frame interpolation. For example, for cross-retargeting on H36M, collected at 50 FPS, we upsample the 25 input frames to 30 frames by duplicating some selected ones. The 120 output frames are donwsampled to 100 frames analogously, to match the FPS of the GT. So, when evaluating at a different FPS than the training data, we evaluate at the frequency of the evaluation dataset (the original FPS of the GT). We follow the same procedure to allow baselines to perform cross-topology (through retargeting) at different FPS than the one seen at train time (Tab. 19, Tab. 1, Tab. 11, Tab. 12). To show that such procedure does not give us any advantage in comparison with other state-of-the-art baselines, we conduct two experiments.

Table 8: Quantitative results on AMASS for methods fed an input downsampled to 30 FPS and upsampled to 60 FPS again. The dataset is recorded at 60 FPS, and baselines trained at the same resolution. Results and rankings are consistent with Tab. 17, showing that methods are robust to such small temporal distortion in the input data.

Method	Precision ↓			Div ↑	Real ↓	Body Real ↓			
	ADE	FDE	MAE	APD	CMD	mean ↓		RMSE ↓	
						str	jit	str	jit
DLow	0.589	0.615	0.148	<u>13.166</u>	<u>15.849</u>	8.40	0.39	11.05	0.55
DivSamp	0.564	0.649	0.140	24.719	50.230	11.18	0.82	16.73	1.07
BeLFusion	<u>0.507</u>	<u>0.570</u>	<u>0.124</u>	7.461*	19.627	<u>7.21</u>	<u>0.22</u>	<u>8.74</u>	<u>0.29</u>
SkelDiff	0.480	0.548	0.107	9.453	11.418	3.15	0.20	4.44	0.26

Table 8 . First, we show that current methods are robust to FPS interpolation in Tab. 8. We downsample the input to 30FPS and upsample it again before feeding it to methods trained with 60 FPS on AMASS. By comparing the evaluation numbers with the standard evaluation scores of Tab. 17, we see that methods are robust to such light temporal distortion in the input. The ranking remains unchanged and we observe just negligible variations in the score. These may be

Table 9: Quantitative results on AMASS for methods evaluated at 30FPS instead of 60 FPS. Dataset is recorded at 60 FPS, and baselines trained at the same resolution. The GT and the method’s output are downsampled to 30 FPS. Results are consistent with Tab. 17, up to range changes introduced by averaging on a smaller number of frames (APD).

Method	Precision ↓			Div ↑	Real ↓	Body Real ↓			
	ADE	FDE	MAE	APD	CMD	mean ↓		RMSE ↓	
						str	jit	str	jit
DLow	0.586	0.611	0.148	<u>9.289</u>	15.849	8.37	0.78	11.03	1.10
DivSamp	0.561	0.645	0.139	17.451	50.230	11.13	1.65	16.68	2.13
BeLFusion	0.505	0.565	0.124	5.261	19.627	7.18	<u>0.44</u>	8.73	<u>0.59</u>
CoMusion	<u>0.491</u>	<u>0.546</u>	<u>0.117</u>	7.644	9.661	<u>4.03</u>	0.65	<u>5.61</u>	0.85
SkelDiff	0.477	0.544	0.106	6.665	<u>11.418</u>	3.14	0.39	4.44	0.51

Table 10: Model footprint for a single H36M inference (RTX 6000). Our model does not require multiple instances or more parameters to generalize to additional kinematics or datasets, while this does not hold for all other models (see Fig. 3).

	Memory↓	NumParams↓	Time↓
DLow [136]	31 MB	8.1 M	111 ms
DivSamp [26]	88 MB	23.1 M	8 ms
BeLFusion [9]	53 MB	17.8 M	10 341 ms
HumanMAC [18]	114 MB	28.7 M	7 438 ms
CoMusion [108]	87 MB	19 M	153 ms
SkelDiff [23]	106 MB	26.5 M	412 ms
EquiFusion	31MB	7.9M*	192 ms

dependent on the fact that generative methods are sensitive to random generator states, which are initialized differently between different GPU architectures despite the same random seed. We conduct both experiments on a RTX6000.

Table 9. Second, we show that evaluating methods by changing the FPS of the GT does not change the ranking (Tab. 9). We conduct this experiment on AMASS by downsampling both output and GT to 30 FPS. While the ranking is unchanged, we see that the range of some metrics has changed (APD) compared to the reference table (Tab. 17). The reason behind this range shift is that the metrics are computed by averaging over a smaller number of frames (60 instead of 120). Hence, to ensure a fair comparison with other methods and with other tables of conventional HMP evaluation, we decide to upsample the output to the original dataset FPS.

H Additional Experiments

H.1 Computational Efficiency and Inference Time

Footprint. The state-of-the-art performance of our method is also accompanied by efficiency, both in terms of memory usage and computational complexity. In Tab. 10, we compare our model’s computational footprint to that of other competitors for a single dataset, H36M. Our method is the smallest in terms of parameters, being from two to three times smaller compared to the closest competitor (SkelDiff). At the same time, it is twice as fast at inference. Methods with a similar time and memory footprint, such as DLow, are VAE-based and produce significantly worse results in all metrics (see, for example, Tab. 16). On a single dataset, training takes around half of the time of the closest competitor, SkelDiff. Additionally, we do not need to retrain a new model for each dataset, unlike other approaches. This leads to the scalability advantages described in the next paragraph.

Scalability. As proven mathematically in Sec. C.1, we do not scale with the number of kinematics or joints, while previous approaches do. We scale constantly, as shown in Fig. 3. Hence, a single model, trained once, in less time than others, is enough for all datasets. When considering one dataset, we are 75% more compact than the most competitive baseline SkelDiff. When considering three datasets (AMASS, Nymeria, H36M), we are 90% more compact than SkelDiff: we still require a single model, while others require three instances. Our efficiency becomes particularly advantageous for applications that require zero-shot kinematics reasoning. To train and natively process different skeleton kinematics, previous methods require retraining and storing a dedicated network for each. Hence, their memory usage grows linearly with the number of skeletal structures considered. Instead, our method naturally operates across kinematics and performs both training and inference with a single model.

H.2 Zero-Shot Kinematics on AMASS

In the main paper, we reported the results for methods trained on AMASS and tested on H36M (Tab. 1). For completeness, we also report the inverse, where models are trained on H36M and tested on AMASS. We highlight that such a case is particularly challenging for method generalization, as 1) H36M is a significantly smaller dataset, 2) the kinematics of H36M $\mathcal{K}^H$ has only 17 joints, while $\mathcal{K}^A$ of AMASS has 22 joints. In general, methods trained on H36M may lead to overfitting, also due to the low number of subjects. We report results in Tab. 11 with analogous results as when investigate the opposite direction (AMASS $\leftrightarrow$ H36M in (Tab. 1)). When trained only on H36M, EquiFusion performs in line with the state of the art. However, an advantage of our method is the possibility to experiment with other data priors without any modification. We observe that training on Nymeria yields improved results, suggesting that this dataset is a better fit to the target distribution. For other methods, this would require

defining specific retargeting techniques for every pair of training and test distributions. The retargeting method used between AMASS and H36M [43] cannot be applied directly to Nymeria, since the spine and hips have distinct structures, and so we can't compare with other methods trained in same conditions.

Table 11: Evaluation of Zero-Shot Kinematics on AMASS($\mathcal{K}^A$) [76]. Baselines are trained on H36M [46] (H, kinematics $\mathcal{K}^H$). This is a challenging retargeting case, the inverse direction of the case presented in the main body Tab. 1: the inference kinematics $\mathcal{K}^A$ has more joints than the training one $\mathcal{K}^H$ (22 vs 17 joints). We are the only existing method supporting inference on novel kinematics out-of-the-box, while previous approaches require additional kinematics conversion [18, 43]. We are the first method to support multiple kinematics natively and thus present a model trained additionally on Nymeria [75] (N, $\mathcal{K}^N$). The best results are highlighted in **bold**, second-best are underlined. Conventional metrics rank identically, see Tab. 12.

Units Method	Precision ↓			Multimodal GT ↓			Div ↑	Real ↓	Body Real ↓			
									mean ↓		RMSE ↓	
	cm <i>u</i> ADE	cm <i>u</i> FDE	deg° MAE	cm <i>u</i> MMA	cm <i>u</i> MMF	– APDE	m <i>u</i> APD	– CMD	% str	% jit	% str	% jit
ZeroVel	1.234	1.629	7.779	1.360	1.694	9.292	0.000	39.34	0.00	0.00	0.00	0.00
ZeroVel+ [43]	2.829	2.862	14.969	2.847	2.872	9.292	0.000	39.34	12.92	0.00	12.92	0.00
TPK [119]+ [43]	14.24	15.32	19.87	14.52	15.26	2.321	1.465	22.66	30.88	0.32	32.83	0.49
DLow [136]+ [43]	13.85	14.83	19.86	14.19	14.81	4.941	2.593	21.03	31.24	0.35	33.44	0.53
GSPS [78]+ [43]	11.17	12.73	13.33	11.88	12.94	7.578	2.915	23.60	22.37	0.24	23.89	0.35
DivSamp [26]+ [43]	11.20	12.84	13.37	11.91	12.95	8.611	3.289	21.05	21.45	0.26	23.49	0.37
BeLFusion [9]+ [43]	10.94	12.59	12.77	11.69	12.77	3.233	1.159	24.60	19.08	0.19	20.65	0.27
CoMusion [108]+ [43]	12.37	13.59	18.94	13.12	13.76	<u>2.065</u>	<u>1.992</u>	18.00	30.70	0.48	32.25	0.70
SkelDiff [23]+ [43]	11.88	13.75	19.72	12.71	13.97	2.080	1.643	<u>15.66</u>	26.50	0.29	28.16	0.42
EquiFusion (H)	9.71	11.39	7.88	10.81	11.86	3.700	1.070	23.39	0.00	0.00	0.00	0.00
EquiFusion (N)	<u>9.27</u>	<u>10.72</u>	<u>7.52</u>	<u>10.66</u>	<u>11.39</u>	1.981	1.675	13.19	0.00	0.00	0.00	0.00
EquiFusion (H+N)	8.76	10.07	7.26	10.15	10.76	2.841	1.333	17.63	0.00	0.00	0.00	0.00

Table 12: Version of Tab. 11 with conventional metrics instead of unified metrics.

Method	Precision ↓			Multimodal GT ↓			Div ↑	Real ↓	Body Realism ↓			
									mean ↓		RMSE ↓	
	ADE	FDE	MAE	MMA	MMF	APDE	APD	CMD	str	jit	str	jit
ZeroVel	0.755	0.992	7.779	0.814	1.015	9.299	0.000	39.338	0.00	0.00	0.00	0.00
TPK [119]+ [43]	0.770	0.820	19.867	0.781	0.816	2.321	7.896	22.659	30.88	0.32	32.83	0.49
DLow [136]+ [43]	0.749	0.792	19.859	0.764	0.790	4.941	13.713	21.031	31.24	0.35	33.44	0.53
GSPS [78]+ [43]	0.660	0.738	13.334	0.688	0.744	7.578	16.198	23.596	22.37	0.24	23.89	0.35
DivSamp [26]+ [43]	0.663	0.742	13.371	0.692	0.744	8.611	17.745	21.046	21.45	0.26	23.49	0.37
BeLFusion [9]+ [43]	0.649	0.734	12.770	0.680	0.739	3.233	6.512	24.604	19.08	0.19	20.65	0.27
CoMusion [108]+ [43]	0.683	0.739	18.938	0.717	0.744	2.065	10.856	17.999	30.70	0.48	32.25	0.70
SkelDiff [23]+ [43]	0.668	0.752	19.722	0.701	0.760	2.080	8.956	15.660	26.50	0.29	28.16	0.42
EquiFusion (H)	0.588	0.688	7.883	0.644	0.708	3.700	6.370	23.393	0.00	0.00	0.00	0.00
EquiFusion (N)	0.564	0.653	7.516	0.635	0.682	1.981	9.418	13.185	0.00	0.00	0.00	0.00
EquiFusion (H+N)	0.536	0.617	7.259	0.607	0.646	2.841	7.776	17.630	0.00	0.00	0.00	0.00

Table 13: Evaluation of Zero-Shot Kinematics on H36M($\mathcal{K}^H$) [46] with additional retargeting scenario. Additionally to the scenarios $RT_{I/O}$ presented as default in Tab. 1, we investigate two additional retargeting scenarios, $RT_{I/GT}$ and RT_{GT^2} . See Sec. H.3 for their description.

Method	RT	Precision ↓			Multimodal GT ↓			Div ↑	Realism ↓		Body Realism ↓			
		uADE	uFDE	MAE	uMMA	uMMF	APDE	uAPD	CMD	FID	mean ↓		RMSE ↓	
											str	jit	str	jit
ZeroVel	-	11.77	17.88	6.753	13.74	18.56	8.085	0.000	22.822	-	0.00	0.00	0.00	0.00
	RT _{I/O}	11.77	17.88	6.362	19.88	23.72	8.085	0.000	22.822	-	0.05	0.00	0.05	0.00
	RT _{I/GT}	11.39	17.09	6.467	19.46	22.96	8.085	0.000	22.822	-	0.52	0.00	0.52	0.00
	RT _{I/GT} ²	11.77	17.88	6.362	19.88	23.72	8.085	0.000	22.822	-	0.05	0.00	0.05	0.00
TPK [119] + [43]	RT _{I/O}	13.81	16.13	22.276	14.60	16.22	<u>1.968</u>	1.469	10.051	3.773	19.55	0.46	22.21	0.73
	RT _{I/GT}	13.60	15.85	22.324	18.20	19.07	1.914	1.410	10.362	-	23.11	0.58	26.85	0.91
	RT _{I/GT} ²	13.80	16.12	22.259	18.34	19.29	<u>1.968</u>	1.469	10.051	3.662	19.73	0.46	22.39	0.73
DLow [136] + [43]	RT _{I/O}	12.71	14.80	21.887	13.60	14.97	2.060	2.060	9.204	2.875	20.26	0.53	23.47	0.82
	RT _{I/GT}	12.55	14.63	22.207	17.59	18.18	2.880	2.014	9.326	-	23.83	0.65	28.08	1.01
	RT _{I/GT} ²	12.70	14.80	22.011	17.71	18.37	2.060	2.060	9.204	2.620	20.38	0.53	23.60	0.82
GSPS [78] + [43]	RT _{I/O}	9.29	11.91	8.107	10.85	12.35	2.373	2.069	7.409	1.735	11.51	0.38	14.00	0.49
	RT _{I/GT}	9.06	11.66	7.889	16.23	16.84	3.034	1.966	<u>7.329</u>	-	14.98	0.48	18.20	0.63
	RT _{I/GT} ²	9.22	11.86	7.008	16.50	17.15	2.386	2.070	7.403	1.620	11.33	0.38	13.84	0.49
DivSamp [26] + [43]	RT _{I/O}	9.27	12.61	8.374	11.42	13.27	10.510	4.210	47.783	5.629	18.47	1.01	24.51	1.39
	RT _{I/GT}	9.08	12.40	8.322	17.07	18.04	13.525	4.267	48.441	-	21.30	1.19	28.65	1.65
	RT _{I/GT} ²	9.20	12.59	7.572	17.34	18.34	10.519	<u>4.213</u>	47.852	5.082	18.33	1.02	24.42	1.40
BeLFusion [9] + [43]	RT _{I/O}	9.24	11.62	8.200	10.93	12.16	2.284	1.305	8.031	1.195	9.81	0.34	12.16	0.46
	RT _{I/GT}	9.05	11.54	9.565	16.52	17.07	2.104	1.252	8.013	-	12.20	0.43	15.34	0.58
	RT _{I/GT} ²	9.17	11.60	7.273	16.76	17.26	2.284	1.305	8.031	1.093	9.68	0.34	12.05	0.46
CoMusion [108] + [43]	RT _{I/O}	10.07	12.15	21.066	12.49	12.96	2.370	2.070	8.587	1.426	15.98	0.51	17.57	0.68
	RT _{I/GT}	9.95	12.00	21.730	17.20	17.11	3.354	2.029	8.664	-	21.07	0.68	23.31	0.90
	RT _{I/GT} ²	10.08	12.16	21.186	17.39	17.29	2.370	2.070	8.587	1.175	15.85	0.51	17.45	0.68
SkelDiff [23] + [43]	RT _{I/O}	10.81	14.99	14.947	12.75	15.42	2.995	0.992	7.616	5.252	11.25	0.28	12.86	0.39
	RT _{I/GT}	10.45	14.48	14.146	17.78	19.58	2.805	0.923	8.139	-	15.14	0.32	16.89	0.46
	RT _{I/GT} ²	10.73	14.93	14.722	18.14	20.12	2.995	0.992	7.616	4.906	11.08	0.28	12.69	0.39
EquiFusion	-	<u>7.86</u>	<u>10.47</u>	<u>5.861</u>	<u>10.66</u>	<u>11.61</u>	2.248	1.973	7.061	0.691	0.00	0.00	0.00	0.00
EquiFusion	-	7.71	10.21	5.683	10.58	11.36	2.597	1.797	7.349	0.504	0.00	0.00	0.00	0.00

Table 14: Evaluation of Zero-Shot Kinematics on AMASS($\mathcal{K}^A$) [76] with additional retargeting scenario. Additionally to the scenarios $\text{RT}_{I/O}$ presented as default in Tab. 11, we investigate two additional retargeting scenarios, $\text{RT}_{I/GT}$ and RT_{GT^2} . See Sec. H.3 for their description.

Method		Precision ↓			Multimodal GT ↓			Div ↑	Real ↓	Body Real ↓			
		$u\text{ADE}$	$u\text{FDE}$	MAE	$u\text{MMA}$	$u\text{MMF}$	APDE	$u\text{APD}$	CMD	mean ↓ str	jit	RMSE ↓ str	jit
ZeroVel	-	1.234	1.629	7.779	1.360	1.694	9.292	0.000	39.338	0.00	0.00	0.00	0.00
	$\text{RT}_{I/O}$	2.829	2.862	14.969	2.847	2.872	9.292	0.000	39.338	12.92	0.00	12.92	0.00
	$\text{RT}_{I/GT}$	1.288	1.713	8.634	1.951	2.262	9.292	0.000	39.338	0.38	0.00	0.38	0.00
	RT_{I/GT^2}	1.232	1.626	<u>7.548</u>	1.859	2.143	9.292	0.000	39.338	0.98	0.00	0.98	0.00
TPK [119] + [43]	$\text{RT}_{I/O}$	14.24	15.32	19.87	14.52	15.26	2.321	1.465	22.66	30.88	0.32	32.83	0.49
	$\text{RT}_{I/GT}$	14.13	15.42	18.25	15.58	15.85	2.581	1.531	22.68	23.99	0.35	26.51	0.53
	RT_{I/GT^2}	13.65	14.82	18.84	14.98	15.35	2.321	1.465	22.66	22.76	0.33	25.08	0.50
DLow [136] + [43]	$\text{RT}_{I/O}$	13.85	14.83	19.86	14.19	14.81	4.941	2.593	21.03	31.24	0.35	33.44	0.53
	$\text{RT}_{I/GT}$	13.67	14.82	18.27	15.35	15.35	4.069	2.707	21.05	24.49	0.38	27.36	0.58
	RT_{I/GT^2}	13.22	14.27	18.76	14.79	14.95	4.941	2.593	21.03	23.17	0.35	25.78	0.54
GSPS [78] + [43]	$\text{RT}_{I/O}$	11.17	12.73	13.33	11.88	12.94	7.578	2.915	23.60	22.37	0.24	23.89	0.35
	$\text{RT}_{I/GT}$	10.65	12.51	10.64	17.07	17.85	6.430	3.073	24.22	14.62	0.28	16.59	0.40
	RT_{I/GT^2}	10.17	11.92	8.66	16.17	16.86	7.578	2.915	23.60	13.11	0.25	14.90	0.36
DivSamp [26] + [43]	$\text{RT}_{I/O}$	11.20	12.84	13.37	11.91	12.95	8.611	<u>3.289</u>	21.05	21.45	0.26	23.49	0.37
	$\text{RT}_{I/GT}$	10.53	12.44	11.35	17.30	18.16	7.234	3.457	21.56	11.82	0.28	14.37	0.40
	RT_{I/GT^2}	10.15	11.99	9.03	16.57	17.44	8.611	<u>3.289</u>	21.05	11.53	0.26	13.98	0.37
BeLFusion [9] + [43]	$\text{RT}_{I/O}$	10.94	12.59	12.77	11.69	12.77	3.233	1.159	24.60	19.08	0.19	20.65	0.27
	$\text{RT}_{I/GT}$	10.35	12.31	9.84	16.37	17.15	3.615	1.229	24.49	10.07	0.22	12.10	0.31
	RT_{I/GT^2}	9.90	11.72	8.26	15.53	16.21	3.233	1.159	24.60	9.91	0.20	11.88	0.29
CoMusion [108] + [43]	$\text{RT}_{I/O}$	12.37	13.59	18.94	13.12	13.76	2.065	1.992	18.00	30.70	0.48	32.25	0.70
	$\text{RT}_{I/GT}$	12.15	13.53	17.61	15.11	15.09	1.840	2.089	18.02	22.46	0.54	24.57	0.79
	RT_{I/GT^2}	11.67	13.00	17.51	14.70	14.77	2.065	1.992	18.00	23.04	0.50	24.82	0.73
SkelDiff [23] + [43]	$\text{RT}_{I/O}$	11.88	13.75	19.72	12.71	13.97	2.080	1.643	15.66	26.50	0.29	28.16	0.42
	$\text{RT}_{I/GT}$	11.40	13.52	18.06	15.13	15.43	2.190	1.745	<u>15.23</u>	21.27	0.34	23.32	0.49
	RT_{I/GT^2}	11.10	13.15	18.48	14.55	14.75	2.080	1.643	15.66	21.29	0.30	23.06	0.43
EquiFusion (H)	-	9.71	11.39	7.88	10.81	11.86	3.700	1.070	23.39	0.00	0.00	0.00	0.00
EquiFusion (N)	-	<u>9.27</u>	<u>10.72</u>	<u>7.52</u>	<u>10.66</u>	<u>11.39</u>	<u>1.981</u>	1.675	13.19	0.00	0.00	0.00	0.00
EquiFusion (H+N)	-	8.76	10.07	7.26	10.15	10.76	2.841	1.333	17.63	0.00	0.00	0.00	0.00

H.3 Analysis on Retargeting

An Upper Bound for Baseline’s zero-shot Performance. Isolating the re-targeting error from the baseline error completely is not possible, but via triangle inequality we estimate an upper bound: $\mathcal{E}(H | A) \leq \Delta_{\text{RT}}(\mathcal{K}^H \rightarrow \mathcal{K}^A) + \mathcal{E}(A | A)$. In other words, the total error must be lower than the GT retargeting error summed with the SHMP baseline error. We compute it for SkelDiff and obtain $\mathcal{E}(H | A) \leq \Delta_{\text{RT}}(\mathcal{K}^H \rightarrow \mathcal{K}^A) + \mathcal{E}(A | A) = 11.77 + 10$ where the first number comes from the ADE of the retargeted GT in Tab. 15 and the second from SkelDiff evaluated on the train kinematics $\mathcal{K}^A$ (Tab. 17. Here $\Delta_{\text{RT}}(\mathcal{K}^H \rightarrow \mathcal{K}^A)$ is computed by applying a full retargeting cycle to all GT sequences of the test split as $\text{RT}_{A \rightarrow H}(\text{RT}_{H \rightarrow A}(\mathcal{K}^H))$ and measuring their reconstruction error.

Method	uADE	uFDE	MAE	uAPD	CMD	FID	str	jit
Our variance over 3 seeds (σ^2)	$4e^{-4}$	$7.95e^{-4}$	0.008	0.012	0.011	0.003	0.000	0.000
GT Error $\Delta_{\text{RT}}(\mathcal{K}^H \rightarrow \mathcal{K}^A)$	11.77	17.88	6.362	0.0	22.82	0.606	5.22	0.0

Table 15: We report the variance of our main model of Tab. 1 for zero-shot kinematics on H36M and the reconstruction error of the GT for the retargeting procedure on the same scenario.

Additional Retargeting scenarios Additionally to the most straightforward retargeting scenario discussed in the main paper body, we investigate two additional ones.

Additional Retargeting Scenarios. To facilitate the following discussion, we refer to $\mathcal{K}^{\text{GT}}$ as the skeleton kinematics of the experiment dataset and to $\mathcal{K}^{\text{NN}}$ as the one actually adopted by the network during training (i.e. fixed for current approaches). When the two do not agree, the most correct approach to simulate a real-life scenario is to first convert the input to the $\mathcal{K}^{\text{NN}}$ topology, pass it through the network, and then convert it back the output to $\mathcal{K}^{\text{GT}}$, such that it can be used to compute our metrics. This is the approach we follow in the main body, and we refer to this approach as $\text{RT}_{\text{I/O}}$. It simulates an actual applicative scenario, where the $\mathcal{K}^{\text{GT}}$ specifies both the input and the target domain. However, the network’s output topology $\mathcal{K}^{\text{NN}}$ can be sufficient for some downstream applications, regardless of $\mathcal{K}^{\text{GT}}$. Hence, we propose $\text{RT}_{\text{I/GT}}$, where the network’s output is stored in $\mathcal{K}^{\text{NN}}$, and the ground-truth future is instead retargeted to compute the metrics. Finally, we also consider that the retargeting function is not bijective and projects skeletons into a subspace. To isolate this effect from evaluation, we propose RT_{GT^2} : additionally to applying $\text{RT}_{\text{I/O}}$, we retarget the ground truth twice (from $\mathcal{K}^{\text{GT}}$ to $\mathcal{K}^{\text{NN}}$ and back to $\mathcal{K}^{\text{GT}}$). This way, both prediction and GT undergo the same retargeting procedure and belong to the same representation space.

Table 16: Comparison on Human3.6M [46]. Bold and underlined results correspond to the best and second-best results among the diffusion based models (DM), respectively.

Method	$\mathcal{K}_{\text{new}}$	Precision ↓			MM GT ↓		Div ↑	Real ↓		Body Realism ↓			
		ADE	FDE	MAE	MMA	MMF	APD	CMD	FID	mean ↓ str	jit	RMSE ↓ str	jit
Alg ZeroVelocity	✓	0.597	0.884	6.753	0.683	0.909	0.000	22.812	0.606	0.00	0.00	0.00	0.00
VAE	TPK [119]	✗	0.461	0.560	8.056	0.522	0.569	6.723	6.326	0.538	6.69	0.24	8.37
	DLow [136]	✗	0.425	0.518	6.856	0.495	0.531	11.741	4.927	1.255	7.67	0.28	9.71
	GSPS [78]	✗	0.389	0.496	7.171	0.476	0.525	14.757	10.758	2.103	4.83	0.19	6.17
	DivSamp [26]	✗	0.370	0.485	6.257	0.475	0.516	15.310	11.692	2.083	6.16	0.23	7.85
DM	HumanMAC [18]	✗	0.369	0.480	6.167	0.509	0.545	6.301	-	-	<u>4.01</u>	0.46	6.04
	BeLFusion [9]	✗	0.372	0.474	6.107	0.473	<u>0.507</u>	7.602	5.988	0.209	5.39	<u>0.17</u>	6.63
	CoMusion [108]	✗	0.350	0.458	5.904	0.494	0.506	<u>7.632</u>	3.202	0.102	4.61	0.41	5.97
	SkelDiff [23]	✗	<u>0.344</u>	0.450	5.556	<u>0.487</u>	0.512	7.249	<u>4.178</u>	0.123	3.90	0.16	4.96
DM	EquiFusion(A+N)	✓	0.395	0.522	5.683	0.533	0.574	8.492	7.349	0.504	0.00	0.00	0.00
	EquiFusion(A+N+H)	✓	0.347	0.456	<u>5.121</u>	0.493	0.519	7.086	7.355	<u>0.105</u>	0.00	0.00	0.00
	EquiFusion(H)	✓	0.351	<u>0.451</u>	5.051	0.491	0.515	6.501	7.730	0.158	0.00	0.00	0.00

Evaluation on Zero-Shot Kinematics on H36M. We present here in Tab. 13 the same experiment of Tab. 1 but with additional retargeting scenarios. Here $\text{RT}_{\text{I/O}}$ is coincident with Tab. 1. We see that the scenario $\mathcal{K}^{\text{GT}}$ consistently delivers the lowest precision error, this is thus the most favourable setup for the model. We are evaluating in the output space of the model with a GT converted from H36M to AMASS: the model projects the degraded input to a rather stable distribution - the one learned at train time - and the output is not further converted or degraded. The degradation resulting from the preprocessing retargeting (before feeding the input to the network) is not reflected in the output linearly, as shown by the dissimilarity of ca. 10cm to the degraded GT. Overall, in our experiments, it is not possible to decouple the error of the SHMP model and the retargeting error, as a GT in the desired kinematics does not exist. The other case, RT_{GT^2} , performs similarly to $\text{RT}_{\text{I/O}}$, which is expected: the conversion error in this direction for a GT sequence amounts to 2.27mm (since feet for AMASS are inserted and then removed).

Evaluation on Zero-Shot Kinematics on AMASS. We present here in Tab. 14 the same experiment of Tab. 11 but with additional retargeting scenarios. Here $\text{RT}_{\text{I/O}}$ is coincident with Tab. 11. In this experiment setting, the retargeting error is easier on the networks: the input kinematics AMASS is strongly cropped to fit the number of joints in H36M, thus the input is more similar to the distribution seen by the network at train time. When retargeting both the model output and the GT to AMASS, newly added joints as the feet exhibit similar behavior in the two cases, thus RT_{GT^2} delivers the lowest error.

H.4 Single-Kinematics: AMASS, Nymeria, H36M

Here we report results of methods trained and tested on the same kinematics (i.e. dataset), as in prior works [9, 18, 23, 26, 108, 136].

H36M . In Tab. 16, we report evaluation results on the H36M dataset [46]. It is remarkable that, while the main focus of our work is on enabling zero-shot kinematics processing, our method achieves very competitive results. We also observe that incorporating further datasets in this case is less beneficial in terms of precision, as network capacity is used to represent different distributions. Instead, it still provides improvements in the diversity of the generated movements. This demonstrates that our network is capable of exploiting the combination of different data priors to generate other realistic hypotheses. Particularly, it can leverage the very diverse prior of AMASS to novel kinematics distributions with high realism and precision. We believe this fact may be significant for further experiments investigating the effect of different training distributions.

AMASS . Following previous works, we also employ the AMASS cross-dataset evaluation protocol [9, 18, 23, 108]. In Tab. 17, We achieve competitive results across all metrics. Interestingly, this is the only dataset where leveraging more data or multiple data priors does not improve quantitative evaluation. We believe this is an indicator of the very high motion diversity present in the AMASS distribution compared to other datasets.

Table 17: Quantitative results for AMASS dataset [76]. Not all metrics are available for HumanMAC(see Sec. G.1). The best results are highlighted in **bold**, second-best are underlined. The symbol ‘-’ indicates that the results are not reported in the baseline work.

Type Method		κ_{new}	Precision ↓			MM GT ↓			Div ↑	Real ↓	Body Realism ↓			
			ADE	FDE	MAE	MMA	MMF	APDE	APD	CMD	mean ↓ str	↓ jit	RMSE ↓ str	↓ jit
Alg	ZeroVelocity	✓	0.755	0.992	7.779	0.814	1.015	-	0.000	39.262	0.00	0.00	0.00	0.00
VAE	TPK [119]	✗	0.656	0.675	10.191	0.658	0.674	2.265	9.283	17.127	7.34	0.34	9.69	0.48
	DLow [136]	✗	0.590	0.612	8.510	0.618	0.617	4.243	13.170	15.185	8.41	0.40	11.06	0.58
	GSPS [78]	✗	0.563	0.613	9.045	0.609	0.633	4.678	12.465	18.404	6.65	0.29	8.98	0.37
	DivSamp [26]	✗	0.564	0.647	8.027	0.623	0.667	15.837	24.724	50.239	11.17	0.82	16.71	1.0
DM	HumanMAC [18]	✗	0.511	0.554	-	0.593	0.591	-	9.321	-	-	-	-	-
	BeLFusion [9]	✗	0.513	0.560	7.125	0.569	0.585	1.977	9.376	16.995	7.19	0.34	9.03	0.34
	CoFusion [108]	✗	<u>0.494</u>	<u>0.547</u>	6.715	0.469	0.466	2.328	<u>10.848</u>	9.636	4.04	0.25	5.63	0.52
	SkelDiff [23]	✗	0.480	0.545	6.124	<u>0.561</u>	<u>0.580</u>	<u>2.067</u>	9.456	<u>11.417</u>	3.15	0.20	4.45	0.26
DM	EquiFusion(A+N)	✓	0.504	0.573	6.314	0.582	0.608	2.568	8.055	15.450	0.00	0.00	0.00	0.00
	EquiFusion(A+N+H)	✓	0.508	0.574	6.337	0.585	0.609	2.485	8.272	14.926	0.00	0.00	0.00	0.00
	EquiFusion(A)	✓	0.496	0.560	6.214	0.576	0.596	2.518	8.241	13.097	0.00	0.00	0.00	0.00

Nymeria . We train and evaluate latest diffusion baselines whose code was available on the Nymeria dataset in Tab. 18. Our method performs on par with previous works and achieves significantly better realism. Comparing the range of ADE among AMASS and Nymeria, for example, on the ZeroVelocity algorithmic baseline, it is evident that the motion quality of Nymeria is overall more static.

Table 18: Quantitative results for Nymeria [75] dataset. We trained the latest diffusion baselines with code available following their configuration for AMASS, as the datasets are comparable in size. See Sec. G.1 for details. As our method has no limb stretching by definition of the motion parametrization, 0.12% corresponds to the stretching present in the GT data due to minor sensor inaccuracies.

Type	Method	$\mathcal{K}_{\text{new}}$	Precision ↓			Div ↑	Real ↓	Body Realism ↓			
			ADE	FDE	MAE	APD	CMD	mean ↓		RMSE ↓	
								str	jit	str	jit
Alg	ZeroVelocity	✓	0.519	0.698	4.608	0.0	23.344	0.14	0.0	0.14	0.0
DM	Belfusion [9]	✗	0.343	0.419	4.318	5.280	-	4.76	0.13	5.69	0.16
	HumanMAC [18]	✗	0.318	0.403	4.547	5.689	-	2.37	0.50	4.59	0.62
	SkelDiff [23]	✗	0.278	0.359	3.227	6.450	4.267	1.78	0.09	2.34	0.12
DM	EquiFusion (A+N)	✓	0.299	0.375	3.373	6.016	4.475	0.12	0.00	0.12	0.00
	EquiFusion (A+N+H)	✓	0.302	0.376	3.409	6.399	4.087	0.12	0.00	0.12	0.00
	EquiFusion (N)	✓	<u>0.293</u>	<u>0.369</u>	<u>3.305</u>	<u>6.358</u>	3.818	0.12	0.00	0.12	0.00

Table 19: Zero-shot kinematics on H36M($\mathcal{K}^H$) [46] without unified metrics. Version of Tab. 1 with conventional metrics instead of unified metrics.

Method	Precision ↓		Multimodal GT ↓			Div ↑	Realism ↓		Body Realism ↓				
	ADE	FDE	MAE	MMA	MMF	APDE	APD	CMD	FID	mean ↓		RMSE ↓	
										str	jit	str	jit
ZeroVel	0.597	0.884	6.753	0.683	0.909	8.085	0.000	22.812	0.606	0.00	0.00	0.00	0.00
TPK [119]+ [43]	1.154	0.983	22.686	1.155	0.987	1.968	7.221	10.051	7.522	19.44	0.46	22.10	0.73
DLow [136]+ [43]	1.094	0.948	22.272	1.096	0.951	2.060	9.683	9.204	6.192	20.12	0.53	23.34	0.82
GSPS [78]+ [43]	1.193	1.051	11.439	1.194	1.053	2.373	9.985	7.409	5.325	11.71	0.38	14.23	0.49
DivSamp [26]+ [43]	1.285	1.120	11.854	1.282	1.123	10.510	18.576	47.783	7.749	18.63	1.01	24.68	1.39
BeLFusion [9]+ [43]	1.226	1.029	10.957	1.225	1.033	2.284	6.483	8.031	6.579	10.52	0.34	12.84	0.46
CoMusion [108]+ [43]	1.221	1.028	22.521	1.218	1.032	2.370	9.926	8.587	5.249	15.97	0.51	17.55	0.68
SkelDiff [23]+ [43]	1.372	1.193	17.232	1.371	1.194	2.995	5.420	7.616	7.146	11.25	0.28	12.86	0.39
EquiFusion(A)	0.403	0.533	5.861	0.536	0.585	2.248	9.320	7.061	0.691	0.00	0.00	0.00	0.00
EquiFusion(A+N)	0.395	0.522	5.683	0.533	0.574	2.597	8.492	7.349	0.504	0.00	0.00	0.00	0.00

Table 20: Quantitative results for zero-shot on MoYo for models trained on AMASS. For completeness and future works, we include unified metrics (uADE, uFDE, uAPD). Full metric evaluation of Tab. 2 in main.

Method	Precision ↓			Multimodal GT ↓			Div ↑	Real ↓	Body Realism ↓			
	ADE	FDE	MAE	uADE	uFDE	uAPD	APD	CMD	mean ↓		RMSE ↓	
									str	jit	str	jit
ZeroVel	0.709	1.187	7.954	1.188	2.015	0.000	0.000	20.333	0.00	0.00	0.00	0.00
SkelDiff	0.567	0.892	8.048	0.952	1.524	13.304	2.454	15.710	7.25	0.29	9.44	0.41
EquiFusion(A+N)	0.492	<u>0.786</u>	6.596	<u>0.813</u>	<u>1.299</u>	12.467	2.177	7.095	0.00	0.00	0.00	0.00
EquiFusion(A)	<u>0.501</u>	0.780	<u>6.982</u>	0.834	1.302	<u>13.153</u>	<u>2.333</u>	<u>11.961</u>	0.00	0.00	0.00	0.00

H.5 Extended Tables from Main and not Unified Metrics

In this section, we report for transparency and future works the same tables as in the main paper body, but with additional metrics. Since SHMP has a wide

Table 21: Ablations for the motion parametrization as bone directions on AMASS. For both methods, our parametrization improves realism and body realism metrics by at least 10%. Full metric evaluation of Tab. 5 in main.

Method	Mot	Precision ↓			Multimodal GT ↓			Div ↑	Real ↓	Body Realism ↓			
	Mot									mean ↓		RMSE ↓	
		ADE	FDE	MAE	MMA	MMF	APDE	APD	CMD	str	jit	str	jit
SkelDiff [23]	M	0.480	0.545	6.124	0.562	0.579	2.067	9.456	11.418	3.15	0.20	4.45	0.26
SkelDiff [23]	L	0.496	0.546	6.193	0.575	0.581	1.900	9.960	9.143	0.00	0.00	0.00	0.00
EquiFusion	L	0.501	0.561	6.551	0.577	0.595	2.397	8.348	13.963	3.58	0.27	5.04	0.34
EquiFusion	M	0.498	0.559	6.173	0.577	0.596	2.489	8.413	12.530	0.00	0.00	0.00	0.00

Table 22: Occlusion of a random limb (leg or arm) at inference on AMASS. CMD metric does not apply as it is related only to the motion distribution of the full joint skeleton. FID is not available for missing joints. When joints are missing, Multimodal GT becomes loosely related and is hence discarded. Extended version of main Tab. 3.

Method	Precision ↓					Diversity ↑		Body Realism ↓			
								mean ↓		RMSE ↓	
	ADE	FDE	uADE	uFDE	MAE	APD	uAPD	str	jit	str	jit
SkelDiff [23]	-	-	-	-	-	-	-	-	-	-	-
SkelDiff [23]+rp	0.574	0.727	9.050	1.105	6.996	8.890	1.486	8.15	0.27	9.95	0.39
SkelDiff [23]+sl	0.567	0.683	0.926	1.103	<u>7.162</u>	<u>9.274</u>	<u>1.556</u>	5.64	0.23	7.11	0.31
EquiFusion(A+N)	0.499	0.553	0.765	0.853	8.777	9.099	1.413	0.00	0.00	0.00	0.00
EquiFusion(A)	<u>0.553</u>	<u>0.618</u>	<u>0.868</u>	<u>0.978</u>	10.190	10.152	1.635	<u>0.00</u>	<u>0.00</u>	<u>0.00</u>	<u>0.00</u>

spectrum of metrics, many of which correlate, not all metrics were presented in the main paper body due to redundancy and space reasons.

1. Conventional Metrics for Tab. 1, without unified metrics. Here in Tab. 19 we see that ranking is maintained between conventional and unified metrics.
2. Full metric evaluation for MoYoga of Tab. 2 can be found in Tab. 20.
3. Full metric evaluation for the ablation on the motion parametrization as bone direction in Tab. 5 can be found in Tab. 21.
4. Full metrics evaluation for occlusion of random limbs on AMASS in Tab. 3 can be found in Tab. 22.

I Ablations and Validations on EquiFusion

We validate our model through extensive experiments, investigating the training methodology and the cross-topology application. In Tab. 23 (A), we present ablations for an early stage of our model, trained on AMASS and Nymeria (A+N) with a relaxation of the diffusion objective of $k=50$ (as our final model) and tested for cross-topology on H36M.

Component	Top	Precision ↓			Div ↑	Eff ↓
	eval	uADE	uFDE	MAE	uAPD	#par
drop3joint30		0.834	0.948	6.395	1.355	8M
$L = 256$		0.847	0.960	6.522	<u>1.431</u>	17M
CondTop	A	0.825	0.933	6.389	1.443	8M
CondTop+ $L = 256$		0.846	0.959	<u>6.396</u>	1.301	17M
<u>Ours</u>		<u>0.832</u>	<u>0.940</u>	6.401	1.350	8M
drop3joint30		0.787	1.044	5.859	<u>2.060</u>	8M
$L = 256$		0.769	1.032	5.650	1.826	17M
CondTop	H	0.841	1.100	6.319	2.704	8M
CondTop+ $L = 256$		<u>0.783</u>	<u>1.040</u>	<u>5.841</u>	1.751	17M
<u>Ours</u>		0.801	1.055	5.844	1.888	8M

(a) Ablations with k=50.

Component	Precision ↓			Div ↑	Real ↓
	ADE	FDE	MAE	APD	CMD
+CondBoneLength	<u>0.519</u>	<u>0.610</u>	6.580	<u>5.591</u>	18.093
+posEmbedAdd	0.559	0.661	7.257	4.642	20.636
+posEmbedConcat	0.528	0.627	6.843	5.026	19.347
<u>Ours</u>	0.519	0.608	<u>6.639</u>	5.610	<u>18.188</u>

(b) Ablations with k=1.

Table 23: Ablations for early stages of our model trained on AMASS and Nymeria (A+N) . (A): with k=50. (B): with k=1.

Random Topology Augmentation. We first investigate in Tab. 23 (drop3joints30) whether randomly removing up to 3 joints with a probability of 30% during training increases the cross-topology performance. While the improvement is present, we consider it as minor and not worth the additional component.

Diffusion Conditioning on Topology. As our model is designed to be topology-agnostic and topology information derives only from the input adjacency matrix, we investigate whether a stronger, explicit conditioning on learned graph topology features strengthens performance in both same- and cross-topology settings. We remark here that such feature computation must be permutation equivariant with respect to both the input motion and adjacency matrix, a not straightforward challenge. Interestingly, we see in Tab. 23 (CondTop) that this improves the same-dataset performance, but strongly affects the cross-dataset performance negatively. Further attempts to let the learned conditioning generalize to unseen graphs via data augmentation have not led to significant improvements.

Latent size 96 vs 256. We increase the latent size from 96 to 256 ($L = 256$), for a total of 17M parameters against the previous 8M. This additional capacity translates into worse precision on the seen dataset AMASS, but better precision on generation on H36M. It seems the additional capability results in an over-fitting phenomenon with respect to seen "data", but not seen "topologies". In combination with conditioning the diffusion model on permutation equivariance topology features (CondTop+ $L = 256$), the increased capacity strongly increases the cross-generalization precision compared to conditioning with less parameters. At equal number of parameters, the conditioning still performs worse.

AE vs VAE. In the early stages of our training, we also attempted a variational autoencoder instead of an autoencoder. The results for generation were rather poor, and we discarded the option. Training autoencoder and latent diffusion models together can be challenging, as a good latent space for reconstruction is not directly a good latent space for generation [133]. Yao *et al.* mention indeed that VAE have among the worst generation quality when paired with a latent diffusion model.

Anisotropic vs isotropic diffusion. We decided against the recent anisotropic diffusion paradigm [23], in contrast to the conventional isotropic training we employed: exploiting correlation in the noise and aiming for permutation equivariance are at two opposite spectra.

Permutation Equivariant Positional Embeddings. In Tab. 23 (b), we ablate against permutation equivariant formulations of positional embeddings [74] in encoder and decoder, in the variants of addition and concatenation (posEmbedAdd, posEmbedConcat). We implement an equivariant version of Graphormer’s attention bias [134] for the attention layers, a non-trivial procedure as the equivariance constraint must be fulfilled for a matrix and not a vector in this case. We implement a learned and not learned variant, but find that in both cases it does not lead to improved performance and hence do not include it in further experiments.

J Qualitative Examples

In Figs. 7 to 13 we report qualitative examples for our experiments. Following previous works [9, 23, 108, 136], we report out of 50 predictions, the sample closest to the GT, and the two predictions that maximize diversity when paired with the closest to GT sample. For the most challenging settings (missing limbs, out-of-distribution data), we are only interested in the sample closest to GT.

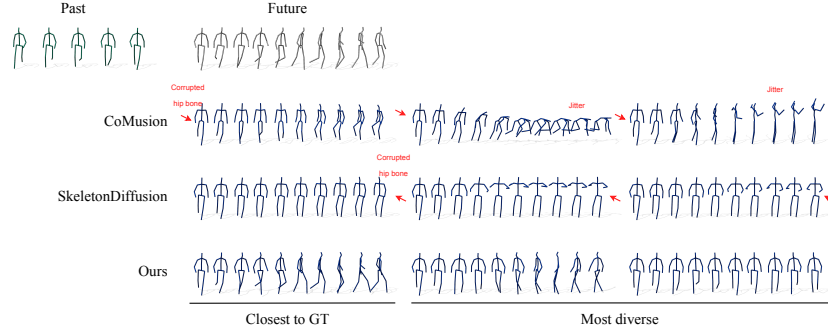


Fig. 7: Qualitative Results for the cross-topology experiment on H36M of Tab. 1. We report out of 50 predictions, the sample closest to the GT, and the two predictions that maximize diversity when paired with the closest to GT sample. Segment n. 605.

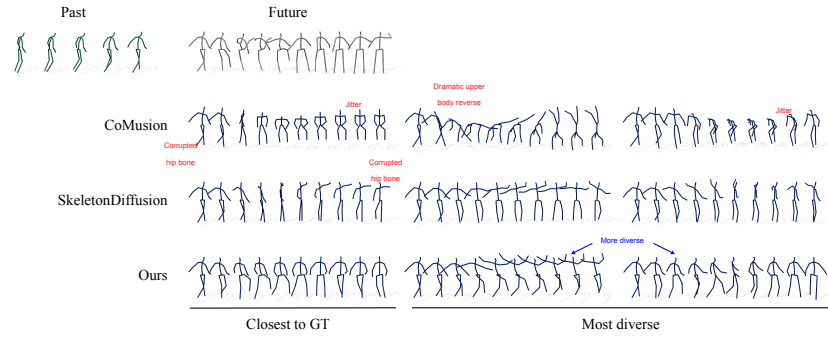


Fig. 8: Qualitative Results for the cross-topology experiment on H36M of Tab. 1. Segment n. 1774.

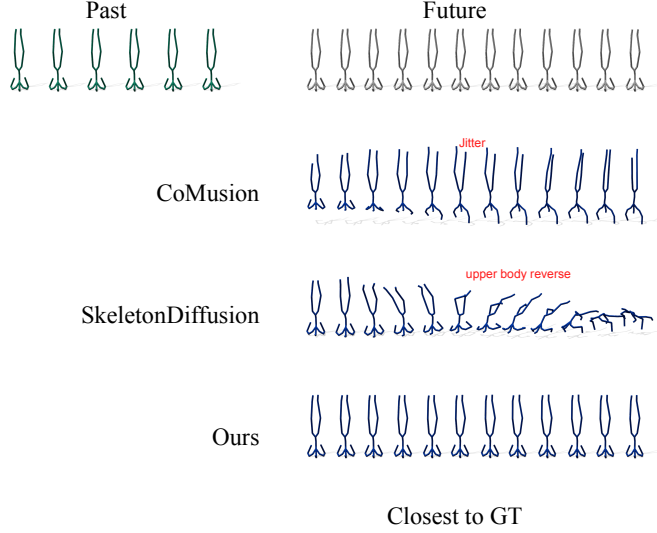


Fig. 9: Qualitative Results for the out-of-distribution testing on the MoCap Yoga dataset Tab. 2. Segment n. 4651. As the setting is quite challenging, we report only the example closest to GT.

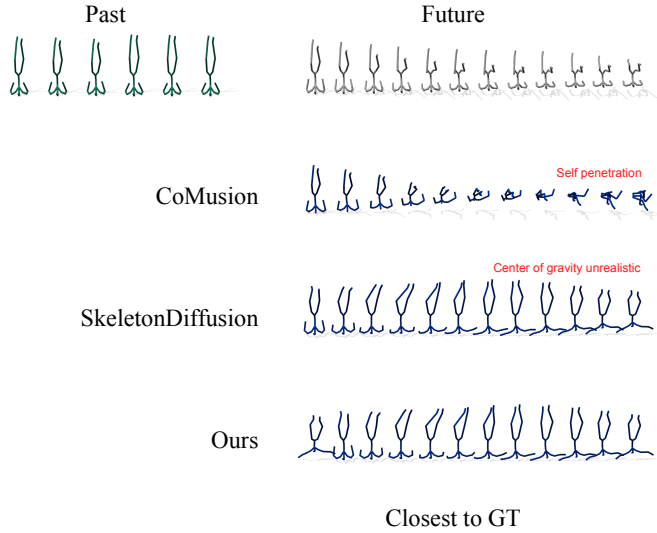


Fig. 10: Qualitative Results for the out-of-distribution testing on the MoCap Yoga dataset Tab. 2. Segment n. 4656.

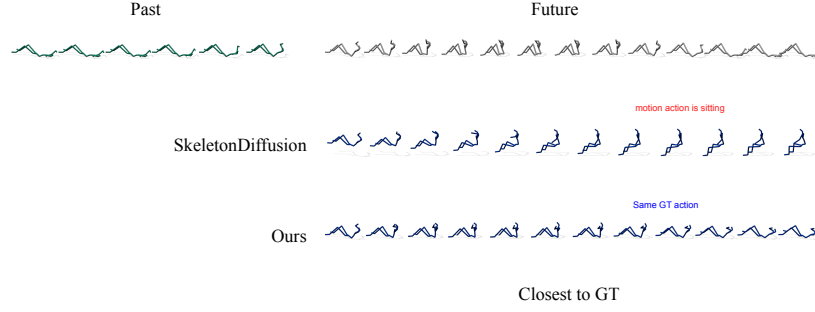


Fig. 11: Qualitative Example of missing left arm in the observation. SkelDiff has been paired with the symmetric limb pipeline for input completion. Test on AMASS Segment n. 12324.

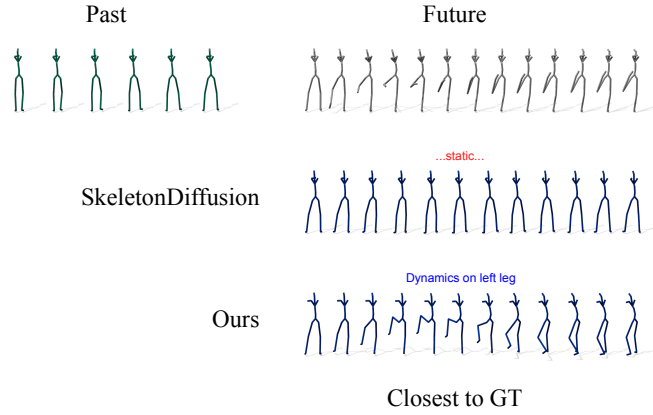


Fig. 12: Qualitative Example of missing both arms in the observation. SkelDiff can only be paired with the restpose approach to complete the input before further processing. Test on AMASS Segment n. 11100.

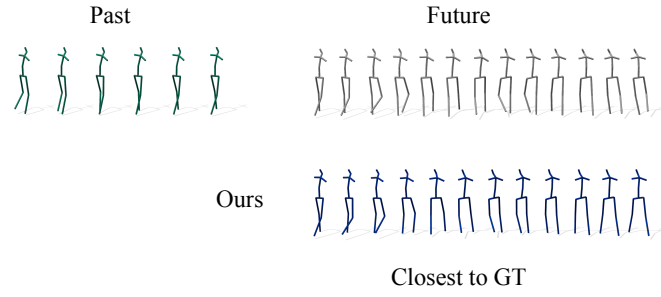


Fig. 13: Qualitative Example of missing both arms in the observation in a cross-topology setting. We do not compare with other methods, as they would require being extended with both retargeting and completion and be exposed to too high degradation. Test on H36M Segment n. 200.