Pose and Skeleton-aware Neural IK for Pose and Motion Editing

Dhruv Agrawal
ETH Zürich
Zürich, Switzerland
DisneyResearch|Studios
Zürich, Switzerland
dhruv.agrawal@inf.ethz.ch

Martin Guay
DisneyResearch|Studios
Zürich, Switzerland
martin.guay@disneyresearch.com

Jakob Buhmann
DisneyResearch|Studios
Zürich, Switzerland
jakob.buhmann@disneyresearch.com

Dominik Borer
DisneyResearch|Studios
Zürich, Switzerland
dominik.borer@disneyresearch.com

Robert W. Sumner
DisneyResearch|Studios
Zürich, Switzerland
ETH Zürich
Zürich, Switzerland
sumner@disneyresearch.com

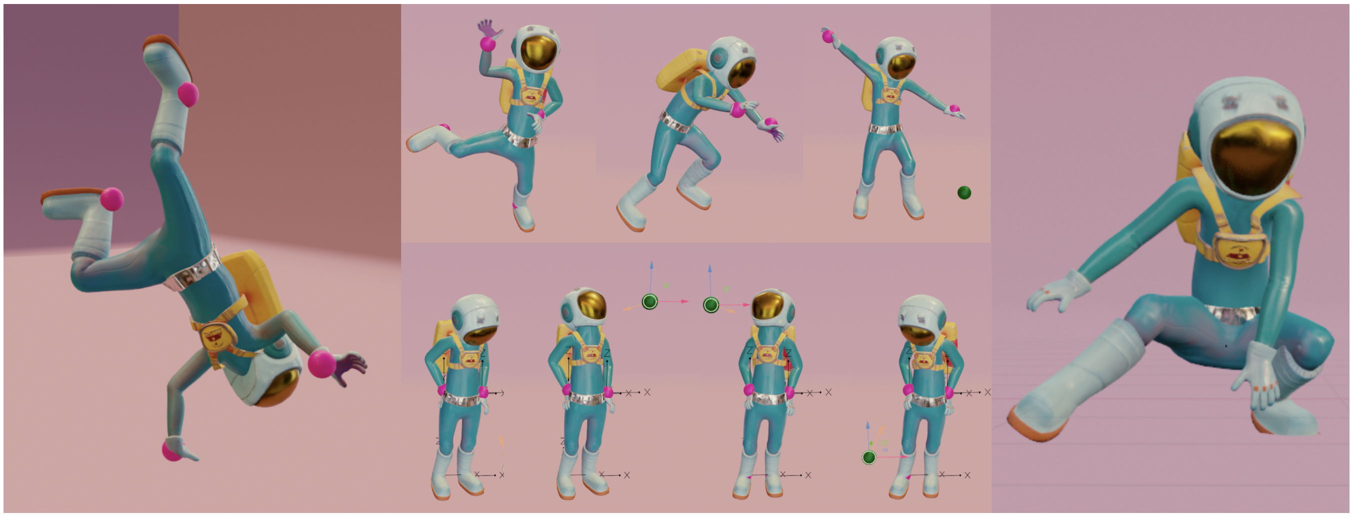


Figure 1: Examples of Poses created using our method. Pink spheres represent positional constraint on the end-effectors and green spheres represents look-at constraint for the *Neck* joint. Our model is able to satisfy the given constraints tightly and also allows quick posing and editing.

ABSTRACT

Posing a 3D character for film or game is an iterative and laborious process where many control handles (e.g. joints) need to be manipulated to achieve a compelling result. Neural Inverse Kinematics (IK) is a new type of IK that enables sparse control over a 3D character pose, and leverages full body correlations to complete the un-manipulated joints of the body. While neural IK is promising, current methods are not designed to preserve previous edits in posing workflows. Current models generate a single pose from the handles only—regardless of what was there previously—making it difficult to preserve any variations and hindering tasks such as pose and motion editing.

In this paper, we introduce SKEL-IK, a novel architecture and training scheme that is conditioned on a base pose, and designed to flow information directly onto the skeletal graph structure, such that hard constraints can be enforced by blocking information flows at certain joints. As a result, we are able to satisfy both hard and soft constraints, as well as preserve un-manipulated parts of the body when desired. Finally, by controlling the base pose in different ways, we demonstrate the ability of our model to perform tasks such as generating variations and quickly editing poses and motions; with less erosion of the base poses compared to the current state-of-the-art.

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0315-7/23/12…$15.00
https://doi.org/10.1145/3610548.3618217

CCS CONCEPTS

• Computing methodologies → Animation; **• Human-centered computing →** *Interaction design.*

KEYWORDS

skeletal networks, pose authoring, learned inverse kinematics, 3D animation

ACM Reference Format:

Dhruv Agrawal, Martin Guay, Jakob Buhmann, Dominik Borer, and Robert W. Sumner. 2023. Pose and Skeleton-aware Neural IK for Pose and Motion Editing. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3610548.3618217

1 INTRODUCTION

Posing a character can at times be repetitive, and deep learning applied to pose and motion autocompletion has recently become a popular area of research. In fact, the recent ProtoRes IK model [Oreshkin et al. 2022] showed that it was possible to train a single model on a large dataset to predict quality full body poses, from different configurations of sparse handles.

Building upon their progress, we identified certain workflows where neural IK is still limited and could not be used. First, animators want the ability to impose hard constraints *strictly*. Current methods however, which generate a pose from a global latent code, can cause changes due to other user-controlled constraints. Secondly, animators may want to use the sparse handles to edit an existing motion such as captured motion (mocap). However, encoding the handles and decoding them erodes the motions; hindering the use of neural IK for this task. And finally, animators may want to generate variations or use a different tool to alter a pose, which unfortunately are lost after moving the handles, as can be seen in our video.

In this paper we propose SKEL-IK, a neural IK network conditioned both on sparse handles *and* a base pose–generating poses that match the constraints, while preserving the base pose as much as possible. Our network operates directly on the skeletal graph structure, thus enabling both hard and soft constraints by blocking information flow at specific joints in the case of hard constraints.

The challenge with conditioning on the base pose is that we do not have labelled data of the posing process, but rather final movements from motion capture. To overcome this challenge, we introduce a pose-preservation parameter in our network, together with losses and a training scheme, providing the ability to preserve either the shape of a pose, or the global positioning; enabling more flexibility to animators to achieve their vision. Finally, we evaluate our model over different benchmarks, and show results of our approach for creating poses, generating variations and editing mocap by manipulating sparse handles.

2 RELATED WORK

Inverse kinematics as an optimization problem has been studied extensively both in graphics and robotics [Balestrino et al. 1984; Buss and Kim 2005; Girard and Maciejewski 1985; Yamane and Nakamura 2003]. One limitation of these methods is that local minima might not look natural. Data-driven methods based on different parametric models such as linear, or gaussian processes, have been used as a prior in the optimization process, or a projection to avoid unnatural local minima [Csiszar et al. 2017; D'Souza et al. 2001; Ren and Ben-Tzvi 2020]. The main limitation of these approaches was their specialization to the amount of constraints and specialization to a small motion space or action. Training a single model for an entire dataset and variable amount of constraints was not feasible. Recently, deep learning has proven able to learn such a representation [Oreshkin et al. 2022].

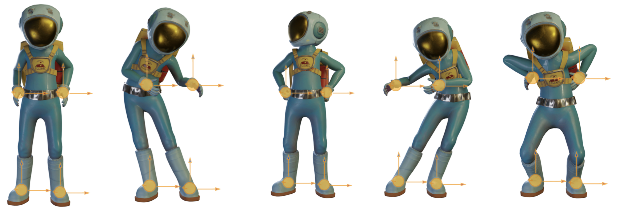


Figure 2: Different possible poses for the same set of positional constraints (shown as golden circles). Decoding a pose from only the constraints would result in an average pose and lose variability present in the data.

Neural IK

Protores [Oreshkin et al. 2022] is designed to generate a full pose from the handles. The model builds upon a prototypical function that compresses the variable number of handles into a generalized pose and improves it over multiple residual blocks. Since, [Voleti et al. 2022] extends ProtoRes to be conditioned on SMPL [Loper et al. 2015] shape parameters and gender, in order to generate fully meshed characters instead of skeletons. Transforming the handles into a global pose vector means that these methods generate or decode a pose that might be influenced from other handles. Also, there is no mechanism to preserve the previous state of the pose the user is editing. Two issues we address in this paper with a skeletal graph neural network conditioned on the previous base pose.

Motion reconstruction from sparse IMUs is another problem closely related to sparse pose reconstruction. These methods infer poses leveraging temporal coherence, and train on motion sequences. For example, [Huang et al. 2018] used data from 6 IMUs to learn a bi-directional RNN, which was since then extended with Transformers [Jiang et al. 2022] and physically correct motions [Yi et al. 2022]. Recently, [Castillo et al. 2023] used a diffusion model to recover motion from only head and wrist mounted IMUs. As this work focuses on pose reconstruction, we do not train for temporal coherency. However, the skeletal structure of our model can be intuitively extended to address temporal motion completion.

Skeletal Networks

Graph Neural Networks (GNNs) are natural model candidates for processing skeletal structures. Compared to traditional Fully Connect Networks, GNNs process information on a per node or a per edge level. Aberman et al. [Aberman et al. 2020] use skeletal convolutions and pooling in order to reduce different skeletons into a

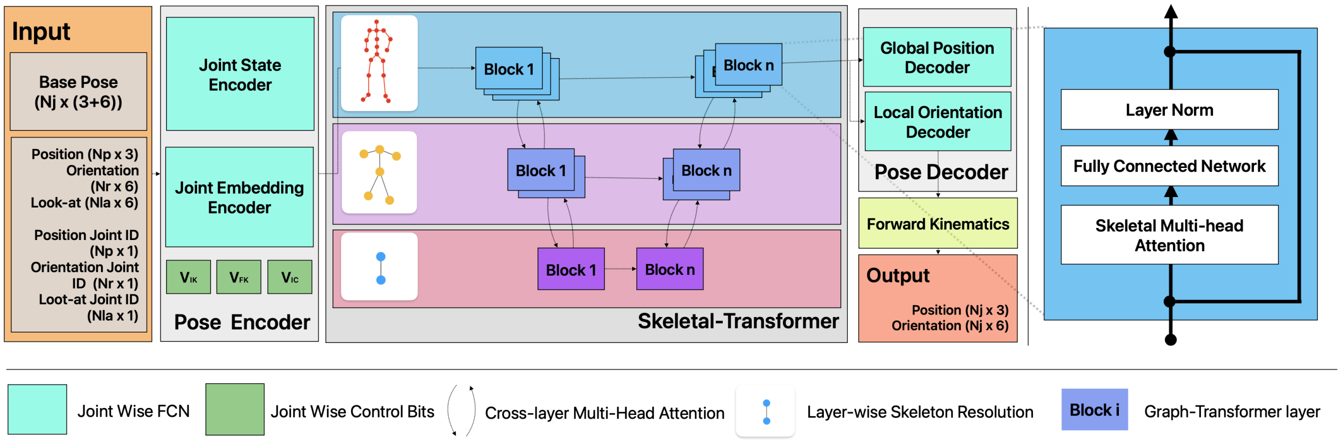


Figure 3: Given a base pose and a set of constraints, we first initialize the graph using the Joint State Encoder. The Skeletal-Transformer then exchanges information between the joints at different resolution. The final encodings of the joint-level joints are decoded into positions and orientations independently. Finally, we use traditional forward kinematics to recover the final global pose.

reduced common *primal skeleton* for transferring motions across different topologies. Then through *unpooling*, they can reconstruct a higher resolution retargeted skeleton. For pose estimation from video, Li et al. [Li et al. 2021] build a hierarchical VAE using skeletal convolutions and predefined-pooling to learn a generic motion prior. They then showcase the use of these motion priors for various tasks including Human Pose Estimation from videos, Motion Completion, and Motion Interpolation. [Grigorev et al. 2023] and [Pfaff et al. 2021] maintain edge and vertex level features at different resolution.

Aggregation Mechanisms. Many of the works on graph neural networks involve aggregating neighboring edges. In [Gilmer et al. 2017], *Summation* is proposed, in [Bruna et al. 2014] Spectral Convolutions, and recently Multi-Head Attention for graphs in [Veličković et al. 2018]. More recently, [Dwivedi and Bresson 2021] also use the attention mechanism on graphs and introduce an adapted version of a traditional Transformer model [Vaswani et al. 2017] for use on graphs. [Wang et al. 2022] allow the system to discover the pools for the pooling operation via a node clustering assignment problem. Because our pools are semantically defined, they do not need to be rediscovered. Instead we focus on facilitating exchange between different skeleton resolutions by introducing a cross-layer attention.

3 SKEL-IK

We introduce a new neural network conditioned on both sparse handles and a base pose that operates directly on the skeletal graph structure, passing information from the constrained joints to neighboring joints over multiple message passing steps (Section 3.3). This allows us to learn local joint-level features resulting in both better preservation of the base pose, as well as to impose hard constraints by blocking information flow at specific joints.

The challenge with conditioning on the base pose is that we do not have datasets of animators animating, but rather final motions and poses. To circumvent this limitation, we devised a training scheme and loss (Section 4) that samples random poses to expose our network to all kinds of configurations the user may be in.

3.1 Skeletal Neural Network

We represent each joint (e.g. hip, right elbow, left ankle, . . .) as a node in our graph and the bones (femur, humerus, . . .) as edges. Each node maintains two vectors, the joint embedding J^i_{emb} and the joint state J^i_{state}.

J^i_{emb} is a linear embedding of one-hot encoded joint ID with additional control bits. We first generate an intermediate embedding, $\widehat{J^i_{emb}}$, derived from the joint ID:

$$\widehat{J^i_{emb}} = \mathbf{W}_{\mathbf{emb}}\, J^i_{one-hot}\,, \tag{1}$$

where $i \in \{\text{hip, head, right wrist}, \ldots\}$ and $\mathbf{W}_{\mathbf{emb}}$ is a learned linear transformation.

For the unconstrained joints, our approach can be viewed as mixing between the base pose and the set of constraints. The additional control bits give control to the artist on the nature and the ratio of mixing between the two.

Preserve-orient $\left(\mathbf{c}^i_{IK}\right)$ is a real value in the range [0,1] informing our network to preserve the shape of the pose (local orientation).

Preserve-pos $\left(\mathbf{c}^i_{FK}\right)$ is a parameter in the range [0,1] informing our network to preserve the spatial aspect of the pose (global positions).

Is Constrained $\left(\mathbf{c}^i_{IC}\right)$ is a Boolean valued bit that differentiates between base and constrained joints, and is set to **True** only if the joint has a positional, orientation, or look-at constraint.

Hence, J_{emb} is defined as:

$$J_{emb} = \widehat{J_{emb}} \oplus \mathrm{C}_{FK} \oplus \mathrm{C}_{IK} \oplus \mathrm{C}_{IC}\,, \tag{2}$$

where $\mathrm{C}_* = \left[\mathbf{c}^i_*\right]_i$ and $\oplus$ is the concatenation operation.

3.2 Pose & Constraints Encoder

The second per-joint vector, J^i_{state}, encodes the joint transformation. It is independent of joint identity, J^i_{emb}. We use a fully connected neural network with one hidden layer to encode the position and orientation of each joint into its respective joint state.

During a forward pass, we first initialize the nodes in the skeletal graph using the Joint State Encoder, as shown in Fig. 3, with a sample from our training dataset. Next, we overwrite J^i_{state} for the constrained joints with desired position or orientation states.

3.3 Skeletal Transformer

The main architecture of our model is the Skeletal-Transformer, as shown in Fig. 3. The Transformer is built as a series of Graph-Transformer layers similar to [Dwivedi and Bresson 2021]. Inside each layer, information is exchanged amongst neighboring joints via the attention mechanism. Using attention in place of other aggregation functions allow the joints to dynamically attend to their neighbors.

We use J_{emb} as the keys $\mathbf{K}$ and queries $\mathbf{Q}$, and J_{state} as the values $\mathbf{V}$. The attention weights are assigned based on joint identity, J^i_{emb}. As, the linear embedding $\widehat{J^i_{emb}}$ is fixed during inference, the attention weights become a function of only the control bits.

The fully connected network and layer normalization then give the updated joint states. Since the fully connected network operates in parallel over the joints, no inter-joint information occurs in these steps. We only update the states J_{state} for the unconstrained joints. The constrained joints are derived purely from the Pose Encoder. In order to enable preserving states, we use a residual connection in order to easily learn the identity function.

The evolution of the joint states from the Skeletal-Transformer can hence be summarised as:

$$\widehat{J^t_{state}} = \text{MHA}\big(\mathbf{K} = J_{emb}, \mathbf{V} = J^{t-1}_{state}, \mathbf{Q} = J_{emb}\big) \tag{3}$$

$$J^t_{state} = \overline{\mathbf{C}_{\text{IC}}} \otimes \text{Layer-Norm}\Big(FCN\big(\widehat{J^t_{state}}\big)\Big) + J^{t-1}_{state}\ , \tag{4}$$

where J^t_{state} is the joint state after step t, $\overline{\mathbf{C}_{\text{IC}}}$ is the inverse of $\mathbf{C}_{IC}$ and $\otimes$ is the Hadamard product.

3.3.1 Layered Skeletal Structure. While the number of nodes in our graph is not large, the length of the longest path in our graph can be comparatively long. For example, a path from the left foot to the right wrist involves over half of the joints in our graph. These long kinematic chains can slow down the rate of information flow through the graph. Therefore, to add shorter paths, we adopt a layered architecture for our skeletal network. We perform message passing at different resolutions simultaneously, similar to [Grigorev et al. 2023].

We first down-sample our full resolution skeleton (Fig. 4a) to a limb-level skeleton (Fig. 4b) with one node for each limb, in addition to nodes for the spine and the hip joints. We further down-sample to a body-level skeleton (Fig. 4c) with nodes corresponding to upper and lower bodies respectively.

This reduces the number of message passing steps required to converge to the final pose. We found that using 6 steps at full resolution and 4 steps on limb-level and 2 steps on body-level skeletons to give the best results. The order of the steps is shown in Fig. 3.

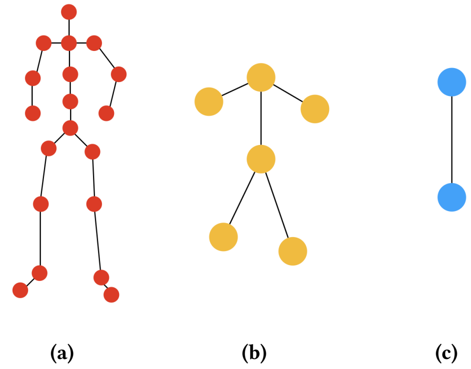


Figure 4: Working at different resolutions allows our model to learn local and global features. (a) Joint-level skeleton with one graph node for each joint. (b) Limb-level skeleton pools joints into one node for hip, spine and each of the four limbs. (c) Body-level skeleton further reduces to one node each for the upper and the lower body.

Lastly, we need to define appropriate pooling and unpooling functions between the layers of our Skeletal-Transformer. Previous works have used summation or a predefined linear combination of the components. We use masked inter-level Multi-Head Attention blocks to mix information between the two layers. We designed the mask such that a node can only attend to itself and appropriate nodes from the other resolution. This allows the network to dynamically assign weights to information from different nodes across layers.

3.3.2 Confidence Mask. Our network operates on the skeleton directly and is thus exposed to the information of the base pose. At the start of our neighborhood aggregation process ($t = 0$), only the constrained joints hold information that needs to be propagated through our skeleton. Therefore, we define a joint-level mask M^l_t to indicate which joints hold new information in layer l after block t. At the start of the stage, $M^{Joint}_{t=0}$ is the same as *Is Constrained* vector, $\mathbf{C}_{IC}$. The limb-level and body-level masks are defined as:

$$M^l_{t=0}[j] = \begin{cases} 1 & \text{if } \exists \text{ Joint } j_0,\ \text{s.t. } j_0 \in j \text{ and } \mathbf{v}^{j_0}_{IC} = 1 \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

At the end of every Skeletal-Transformer block, the mask for layer $l \in \{\text{joint}, \text{limb}, \text{body}\}$ is updated:

$$M^l_t = \mathbf{A}^l M^l_{t-1}\ , \tag{6}$$

where $\mathbf{A}^l$ is the adjacency matrix corresponding to the skeleton of layer l. Each entry in the mask has an upper bound set to 1, which means full neighbor influence, and prevents the messages from increasing for nodes with degree greater than 1.

Model	Constraints		Pose Preservation		Num Parameters
	Position ↓	Orientation ↓	Position ↓	Orientation ↓	
FCNN	0.1076	0.0637	0.2789	0.2876	**28.4M**
ProtoRes	**0.0304**	**0.0508**	—	—	39.7M
ProtoResWithPrev	0.0625	0.0526	0.274	0.2644	39.7M
SKEL-IK (Ours)	0.0388	0.0517	**0.2148**	**0.2581**	29.6M
Local mode	0.1899	0.0596	0.2591	0.2932	
Global mode	0.0388	0.0522	0.3777	0.3425	
Pose Freeze mode	0.0480	0.0517	0.3606	**0.2581**	
Place Freeze mode	0.1672	0.0549	**0.2148**	0.3538	

Table 1: Comparing the performance of our model against the baselines described in Section 5.2. We use $\mathcal{L}^2$ error for the positional constraints and the geodesic error for the orientations. We compare the performance on satisfying the constraints as well as preserving an initial pose. Our model can match the performance of the current state-of-the-art model at sparse posing while also allows strict hard constraints and local pose editing. We list the performance of SKEL-IK as the minimum across its modes. All pose aware models were trained with $\lambda_{pp} = 1.0$.

3.4 Pose Decoder

In the last stage of our model, the Pose Decoder decodes the final joint encodings (concatenated J_{emb} and J_{state}) into final positions and orientations. We implement the Pose Decoder as two fully connected networks with one hidden layer shown as Global Position and Local Orientation Decoders in Fig. 3. Similar to the Pose Encoder, the final pose is decoded independently for each joint. Therefore, there is no information exchange among the joints during this stage.

The final global positions and orientations are derived using traditional forward kinematics from the predicted local orientations and a fixed rest pose. Note that the predicted global positions are not used for the final output but training with them empirically helped the overall performance of our model.

By enforcing information sharing only through a graph network, we have direct control over the influence of each joint. This design enables hard constraints on joints by disabling information sharing for the concerned joints.

4 POSE-PRESERVATION LOSS

To allow position, orientation and look-at constraints, we adopt the same loss functions as [Oreshkin et al. 2022] and defer details to the supplementary material.

In addition to their losses, we learn to preserve characteristics from the base pose and introduce a Pose-Preservation Loss, $\mathcal{L}_{pp}$. We add position $\mathcal{L}^2$ and orientation geodesic losses between the output pose and the base pose:

$$\begin{aligned}\mathcal{L}_{pp}(\mathbf{y},\mathbf{y}',\mathbf{R},\mathbf{R}') = &\, ||\mathbf{C}_{FK} \otimes \overline{M^{joint}_{t=0}} \otimes (\mathbf{y}-\mathbf{y}')||_2^2 \\ &+ \arccos\left[\left(tr\left(\mathbf{C}_{IK}\otimes\overline{M^{joint}_{t=0}}\otimes\mathbf{R}'^{T}\mathbf{R}\right)-1\right)/2\right] ,\end{aligned} \tag{7}$$

where $\mathbf{y}$ and $\mathbf{R}$ are the base position and orientation and $\mathbf{y}'$ and $\mathbf{R}'$ are the respective predictions. Finally, $\overline{M^{joint}_{t=0}}$, inverse of the binary mask introduced in Eq. (5), is used to not penalize the constrained joints. Since $\mathbf{C}_{*K} \in [0,1]^{N_{joints}}$, Eq. (7) weights $\mathcal{L}_{pp}$ according to the amount of base pose that should be preserved in the final output.

Hence, our total loss is comprised of four terms:

$$\mathcal{L} = \mathcal{L}_{pos} + \mathcal{L}_{orient} + \mathcal{L}_{look-at} + \lambda_{pp}\mathcal{L}_{pp} \, , \tag{8}$$

with $\mathcal{L}_{pos}$, $\mathcal{L}_{orient}$, and $\mathcal{L}_{look-at}$ as in [Oreshkin et al. 2022] and λ_{pp} controls the relative weight of $\mathcal{L}_{pp}$.

5 EVALUATION

In this section, we evaluate our model against state-of-the-art Neural IK models and validate our model design through an ablation study.

5.1 Dataset and Implementation details

All the models are trained on our own motion capture dataset and on MiniUnity and MiniMixamo datasets provided by [Oreshkin et al. 2022]. Our dataset includes recording from two subjects retargeted to a common 23-joint skeleton. It contains a total of 95,000 frames of locomotion, idling, stretching, lying down and sitting motion.

Secondly, we set the dimensions of J_{state} and J_{emb} to 512. Similarly, the hidden layers of the Pose Encoder and Decoder are also of 512 dimensional. Lastly, the Multi-Head Attention blocks in each Graph-Transformer layer use attention heads.

This results in a model that has 29.6M parameters. Our model converges on our dataset in 36 hours when training with an AMD 5800X CPU and an NVIDIA RTX 3090 GPU. We list the remaining hyperparameters in the supplementary materials.

By design, our model has very different properties based on the configuration of $\mathbf{C}_{FK}$ and $\mathbf{C}_{IK}$. Therefore, for ease of analysis, we evaluate our model in the following modes:

- Global mode: $\mathbf{C}_{FK} = \mathbf{0}$; $\mathbf{C}_{IK} = \mathbf{0}$
- Local mode: $\mathbf{C}_{FK} = \mathbf{1}$; $\mathbf{C}_{IK} = \mathbf{1}$
- Place Freeze mode: $\mathbf{C}_{FK} = \mathbf{1}$; $\mathbf{C}_{IK} = \mathbf{0}$
- Pose Freeze mode: $\mathbf{C}_{FK} = \mathbf{0}$; $\mathbf{C}_{IK} = \mathbf{1}$

5.2 Baselines

We compare our approach against the ProtoRes model by [Oreshkin et al. 2022]. Since, it does not allow conditioning on a base pose,

Model	MiniUnity				MiniMixamo			
	Constraints		Pose Preservation		Constraints		Pose Preservation	
	Position ↓	Orientation ↓	Position ↓	Orientation ↓	Position ↓	Orientation ↓	Position ↓	Orientation ↓
ProtoRes	0.0153	0.0583	–	–	**0.0262**	**0.0695**	–	–
ProtoResWithPrev	0.0234	0.0607	0.4472	0.3050	0.0549	0.0920	0.5336	0.3815
SKEL-IK (Ours)	**0.0148**	**0.0557**	**0.3358**	**0.2529**	0.0360	0.0889	**0.3735**	**0.2674**

Table 2: Comparing the performance of different models on the publicly available MiniUnity and MiniMixamo datasets provided by [Oreshkin et al. 2022]. The constraints are set on Hips and both feet and the base pose is selected randomly. All models were trained with $\lambda_{pp} = 0.1$.

ProtoRes can diverge from the original pose for very sparse number of handles. According to our knowledge, currently there exists no neural IK model that can be conditioned on another pose. Therefore, we derive two other models as baselines.

FCNN is a fully connected autoencoder with a three layer fully connected encoder and a decoder that is identical to ProtoRes. We input base positions and orientations for the unconstrained joints and the desired positions and orientations for the constrained joint simultaneously to the model.

Next, we modify the original ProtoRes model, ProtoResWithPrev, to allow conditioning on a separate pose. We introduce two new handles types, *Preserve-pos* and *Preserve-orient*, that encode joint-wise base transforms. The input to the model contains $2 \times N_{joints}$ additional handles defining the base pose to be preserved.

5.3 Quantitative Analysis

In this section, we compare the performance of our model against the various baselines described previously. We compare different models, using the $\mathcal{L}^2$ reconstruction error for the positions and the geodesic error for the orientations, for the constraints as well as for the base pose. The summary of our results can be seen in Table 1 for our own dataset and in Table 2 for MiniUnity and MiniMixamo datasets provided by [Oreshkin et al. 2022].

The constraints specific errors reported in Table 1 and Table 2 are the average error on only the constrained joints. Subsequently, the Pose preservation metrics are only measured on the unconstrained joints. Hence, we do not penalize the models for moving the constrained joints from their base transforms.

To illustrate the added flexibility of our model, we detail the performance of each different mode in addition to overall best metrics in Table 1. Amongst these modes, we see that Global mode best satisfies the constraints at the expense of worse pose-preservation error. This characteristic is similar to the base ProtoRes model where the global correlations are encouraged.

In contrast, Local mode, in Table 1, better preserves the base pose at the expense of higher constraints error. We see that Pose Freeze mode best preserves the base pose orientations at the expense of the positions. The Place Freeze mode has the opposite effect with the best performance at preserving base positions.

Without any task-specific architecture, the FCNN baseline constantly performs worse than both ProtoRes models and our model. Across the three datasets, we see that the original ProtoRes is the best at satisfying constraints. This is expected since satisfying constraints and the preserving an initial pose are partially adversarial problems. Even with identical architecture to ProtoRes, we see that ProtoResWithPrev model has worse performance on satisfying the constraints. However, our model in Global mode performs similar to ProtoRes while also providing conditioning on a pose.

In Table 2, we see SKEL-IK performs similar to ProtoRes at satisfying constraints and beats ProtoResWithPrev at preserving the base pose. These results are also consistent with results on our dataset on MiniMixamo and MiniUnity datasets. For brevity, we do not list the performance of each mode individually.

5.4 Ablation Study

We perform an ablation study on the various design changes leading to our final model. A list of all these developments can be found in Table 3. We start with a basic Graph Attention network as described by [Veličković et al. 2018] and additively introduce new features to reach our final design. The initial model only works on a joint-level skeleton and is not trained to preserve any base pose. As we can see in Table 3, the performance of this model is very limited. Adding the confidence mask provides some improvements due to better flow of information. This suggests that information from unconstrained joints can be noisy at the start of the message passing process.

We observe that adding a Pose-preservation loss has a negative effect on the model's ability to satisfy the constraints. But training with this loss allows us to preserve characteristics of a base pose. Next, we introduce working on skeletons at different resolutions. This also increases the total number of parameters in our model. We use average pooling and broadcast unpooling at this stage. This increment gives us the single biggest improvement in both, satisfying the constraints and preserving the base pose.

We next introduce the control bits to have control over the mixing of the base pose and the constraints. This has small impact on the metrics while providing more control over the solution. Finally, we use Graph-Transformer layer in place of simple Multi-Head Attention blocks. This further increases the number of parameters in the model and results in substantial improvements across all metrics.

5.5 Joint Embedding Control Bits

We introduced control bits in Section 3.1. Here, we qualitatively analyze their effect on the output of our model.

Model	Constraints		Pose Preservation	
	Position ↓	Orientation ↓	Position ↓	Orientation ↓
Base-GAT	0.8439	0.2884	–	–
+ Confidence Mask	0.7977	0.2253	–	–
+ Pose-preservation Loss	0.8100	0.2345	0.4451	0.3970
+ Multi-Res Architecture	0.2284	0.1168	0.2342	0.3808
+ Control Bits	0.2268	0.0972	0.2938	0.3514
+ Cross-layer attention	0.1482	0.0672	0.2878	0.3050
+ Graph-Transformer Layer	**0.0388**	**0.0517**	**0.2148**	**0.2581**

Table 3: Ablation Study. We additively introduce a new feature in each row and measure the performance as $\mathcal{L}^2$ reconstruction and geodesic errors on the constraints and a base pose. We see that each of our additions either improves our performance (Multi-Res Architecture and Graph-Transformer layer) or provides a new control to the user (Pose-Preservation loss and Control Bits).

In each row of Fig. 6, we solve from identical base pose and set of constraints but in different solve modes of our model. In Global mode, the model is not penalized for modifying the base pose. Therefore, it behaves as a Global IK Solver, not dissimilar to ProtoRes. We see this global effect in column 6 of Fig. 6. In the first two rows, while the other solutions maintain the position of the unconstrained right arm, we see global correlations between the two arms in the last column. Similarly, in rows 4 & 5, the legs change their positions significantly compared to the base pose. This demonstrates that while our model excels at performing local edits, we can still learn global correlations present in the data.

The last two rows in Fig. 6 show the same set of constraints but from two different angles. We constrain the positions of the hip joint along with both feet. Additionally, we add an orientation constraint on the hip joint. While the four outputs appear similar in Row 5, one can see a clear difference from the second angle. Local and Place Freeze modes (columns 3 & 4) preserve the global base position of the wrists. This causes the wrist to move relative to the rest of the body. In contrast, in Pose Freeze mode (column 5), we see that the local orientations of the shoulders are preserved. This causes arms to rotate along with the rest of the body. Lastly, in Global mode (column 6), we see that the pose of the hands changes completely.

6 RESULTS

SKEL-IK is designed to be local, state aware, and controllable. This enables it to create poses with hard constraints, to generate variations and preserve them in the pose crafting process, as well as to quickly modify mocap sequences. These applications are best appreciated in our accompanying video.

6.1 Hard Constraints

We can see in our video a comparison between [Oreshkin et al. 2022] and our SKEL-IK method for establishing hard constraints. While a global neural inverse kinematics method will always have some correlations between distant joints, SKEL-IK can completely eliminate these correlations by disconnecting the skeletal graph itself. Hence, when the constraints are feasible, we still see joints such as the knees wobbling around when manipulating an arm with ProtoRes. In our experiments, this behaviour persists even in over-constrained scenario with up to 20 constraints. In contrast, our approach can strictly keep the joints still with fewer constraints.

However, in cases where the constraints configuration is infeasible, as when the two wrists are further apart than the wingspan of the skeleton, the hard constraints cannot be satisfied perfectly. In these cases, the parent joint will be preferentially satisfied due to the analytical forward kinematics step in our method. One could also engineer rules into the user interface to prevent constraints from getting into such impossible configurations.

6.2 Variability

Completing full pose from sparse handles is an under-constrained problem. As such, there can be a number of feasible poses for a specific set of constraints (for example, Fig. 2). If the final latent representation of the complete pose is only derived from the specified constraints, all of these variations are lost. Our model allows rediscovering these variations by conditioning on different base poses. By conditioning on random poses from a pose library, we can quickly generate different poses satisfying the same set of constraints. The base poses can even be selected more intelligently with Nearest Neighbor search or by the artist from a pose library.

As we see in Fig. 5, given a set of sparse handles, we can generate a number of natural poses that satisfy the defined set of handles. These suggestions can be used as part of a adaptive pose suggestion system, as starting points for an animation or as background characters that cannot feasibly be animated by hand (for example, crowd simulation). By specifying sparse constraints, one could easily generate a number of different animations to fill the scene.

6.3 Motion Editing

Motion editing refers to making changes to a base animation. With motion capture becoming more and more prevalent in film making, cleaning or modifying motion capture is becoming a staple of many modern film production. Animators usually manipulate rig handles comprised of FK and IK handles along keyframe interpolation to fix movements. Consider a situation where the size or location of a prop has to adjusted after the motion capture from the actors has already been recorded. Editing with traditional animation tools or rerecording the sequence are both expensive options.

As shown in our video, with SKEL-IK we can use the original motion capture as our base animation and only add a new constraint for the edited kick. In this example, we additvely edited the original motion using only three keyframes specifying the offsets of the foot constraint. While using such few constraints leads to an erosion of the poses using previous neural IK methods, we are able to better preserve the base animation and can alter motions more quickly.

7 DISCUSSION AND LIMITATIONS

During training the control bits, $\mathbf{C}_{FK}$ and $\mathbf{C}_{IK}$, are randomly set in the range $[0, 1]$. From our experiments, we see that smoothly moving through the range results in a smooth interpolation between the local and global modes during inference. We include an illustration of these results in our accompanying video.

Our method can result in qualitatively non-natural poses when the base pose is not natural or far from the handles configuration, such as when a 180 degrees turn is required to reach the handles. When such cases arise, the user can simply regenerate in global mode to remove the large bias on the base pose.

From our experience, the model fails for cases that are not represented in the dataset, as is common for data driven systems. These cases can occur when animators exaggerate a pose for stylistic or dramatic effects. For example, when trying to bend the spine backwards, we have seen that the model is not able to do so. However using IK optimization as a post process can help in such cases. Alternatively, fine-tuning the model on a curated stylised dataset can improve its performance in such exaggerated cases.

8 CONCLUSION AND FUTURE WORK

In this work, we introduced SKEL-IK, a new neural IK based on a skeletal graph structure that is able to preserve characteristics of the previous (base) pose in posing workflows. This enables neural IK to be applied in new ways such as for generating variations and for motion editing using sparse handles. We explored biasing our network towards a base pose and showing different biases such as preserving position, or shape (orientation), which opens the door for even exploration. For example, one could train for a symmetry parameter to bias towards symmetric poses given sparse handles.

One of the benefits of our skeletal graph structure is to impose hard constraints strictly. And a remaining challenge with neural IK bridging over to movies and games production are the different character topologies and proportions animators use. In future work, we could explore transferring knowledge across skeletal representations similarly to [Aberman et al. 2020], in order to provide neural IK tools for different characters and proportions.

Lastly, while our work focused on pose editing by conditioning on a base pose, our model can be easily extended for applications such as motion completion using a temporal graph structure or to incorporate additional constraints such as a interpenetration, ground penetration or producing physically valid poses.

REFERENCES

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 62–1.

Aldo Balestrino, Giuseppe De Maria, and Lorenzo Sciavicco. 1984. Robust control of robotic manipulators. *IFAC Proceedings Volumes* 17, 2 (1984), 2435–2440.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*.

Samuel R Buss and Jin-Su Kim. 2005. Selectively damped least squares for inverse kinematics. *Journal of Graphics tools* 10, 3 (2005), 37–49.

Angela Castillo, Maria Escobar, Guillaume Jeanneret, Albert Pumarola, Pablo Arbeláez, Ali Thabet, and Artsiom Sanakoyeu. 2023. BoDiffusion: Diffusing Sparse Observations for Full-Body Human Motion Synthesis. *arXiv preprint arXiv:2304.11118* (2023).

Akos Csiszar, Jan Eilers, and Alexander Verl. 2017. On solving the inverse kinematics problem using neural networks. In *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE, 1–6.

Aaron D'Souza, Sethu Vijayakumar, and Stefan Schaal. 2001. Learning inverse kinematics. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, Vol. 1. IEEE, 298–303.

Vijay Prakash Dwivedi and Xavier Bresson. 2021. A Generalization of Transformer Networks to Graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications* (2021).

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*. PMLR, 1263–1272.

Michael Girard and Anthony A Maciejewski. 1985. Computational modeling for the computer animation of legged figures. *ACM SIGGRAPH Computer Graphics* 19, 3 (1985), 263–270.

Artur Grigorev, Bernhard Thomaszewski, Michael J Black, and Otmar Hilliges. 2023. HOOD: Hierarchical Graphs for Generalized Modelling of Clothing Dynamics. *Computer Vision and Pattern Recognition (CVPR)*.

Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. 2018. Deep Inertial Poser Learning to Reconstruct Human Pose from SparseInertial Measurements in Real Time. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 37, 6 (Nov. 2018), 185:1–185:15.

Yifeng Jiang, Yuting Ye, Deepak Gopinath, Jungdam Won, Alexander W Winkler, and C Karen Liu. 2022. Transformer Inertial Poser: Real-time Human Motion Reconstruction from Sparse IMUs with Simultaneous Terrain Generation. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.

Jiaman Li, Ruben Villegas, Duygu Ceylan, Jimei Yang, Zhengfei Kuang, Hao Li, and Yajie Zhao. 2021. Task-generic hierarchical human motion prior using vaes. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 771–781.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.

Boris N. Oreshkin, Florent Bocquelet, Felix G. Harvey, Bay Raitt, and Dominic Laflamme. 2022. ProtoRes: Proto-Residual Network for Pose Authoring via Learned Inverse Kinematics. In *International Conference on Learning Representations*.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. 2021. Learning Mesh-Based Simulation with Graph Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=roNqYL0_XP

Hailin Ren and Pinhas Ben-Tzvi. 2020. Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks. *Robotics and Autonomous Systems* 124 (2020), 103386.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

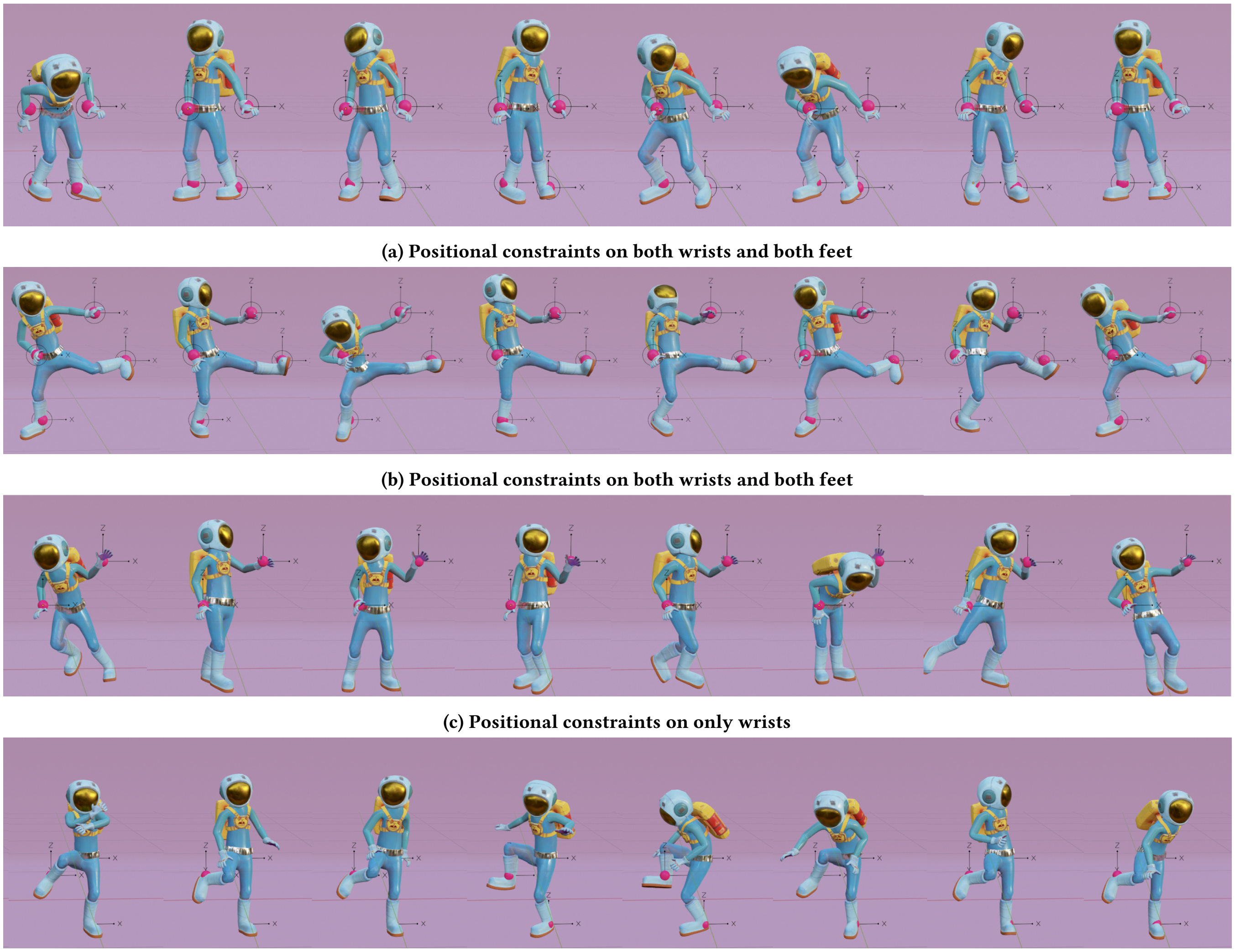
Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJXMpikCZ

Vikram Voleti, Boris Oreshkin, Florent Bocquelet, Félix Harvey, Louis-Simon Ménard, and Christopher Pal. 2022. SMPL-IK: Learned Morphology-Aware Inverse Kinematics for AI Driven Artistic Workflows. In *SIGGRAPH Asia 2022 Technical Communications*. 1–7.

Yu Wang, Liang Hu, Yang Wu, and Wanfu Gao. 2022. Graph Multihead Attention Pooling with Self-Supervised Learning. *Entropy* 24, 12 (2022), 1745.

Katsu Yamane and Yoshihiko Nakamura. 2003. Natural motion animation through constraining and deconstraining at will. *IEEE Transactions on visualization and computer graphics* 9, 3 (2003), 352–360.

Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. 2022. Physical inertial poser (pip): Physics-aware real-time human motion tracking from sparse inertial sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13167–13178.



(a) Positional constraints on both wrists and both feet

(b) Positional constraints on both wrists and both feet

(c) Positional constraints on only wrists

(d) Positional constraints on both feet and the hip

Figure 5: By initializing our model with random sampled base poses, we generate a number of varying poses for the same set of handles. Even when all the end effectors are constrained, a large amount of variation can be seen in (a) neutral pose and (b) extended left leg. We see even more variation with fewer constraints as seen in constraining (c) wrists and (d) feet and hip.

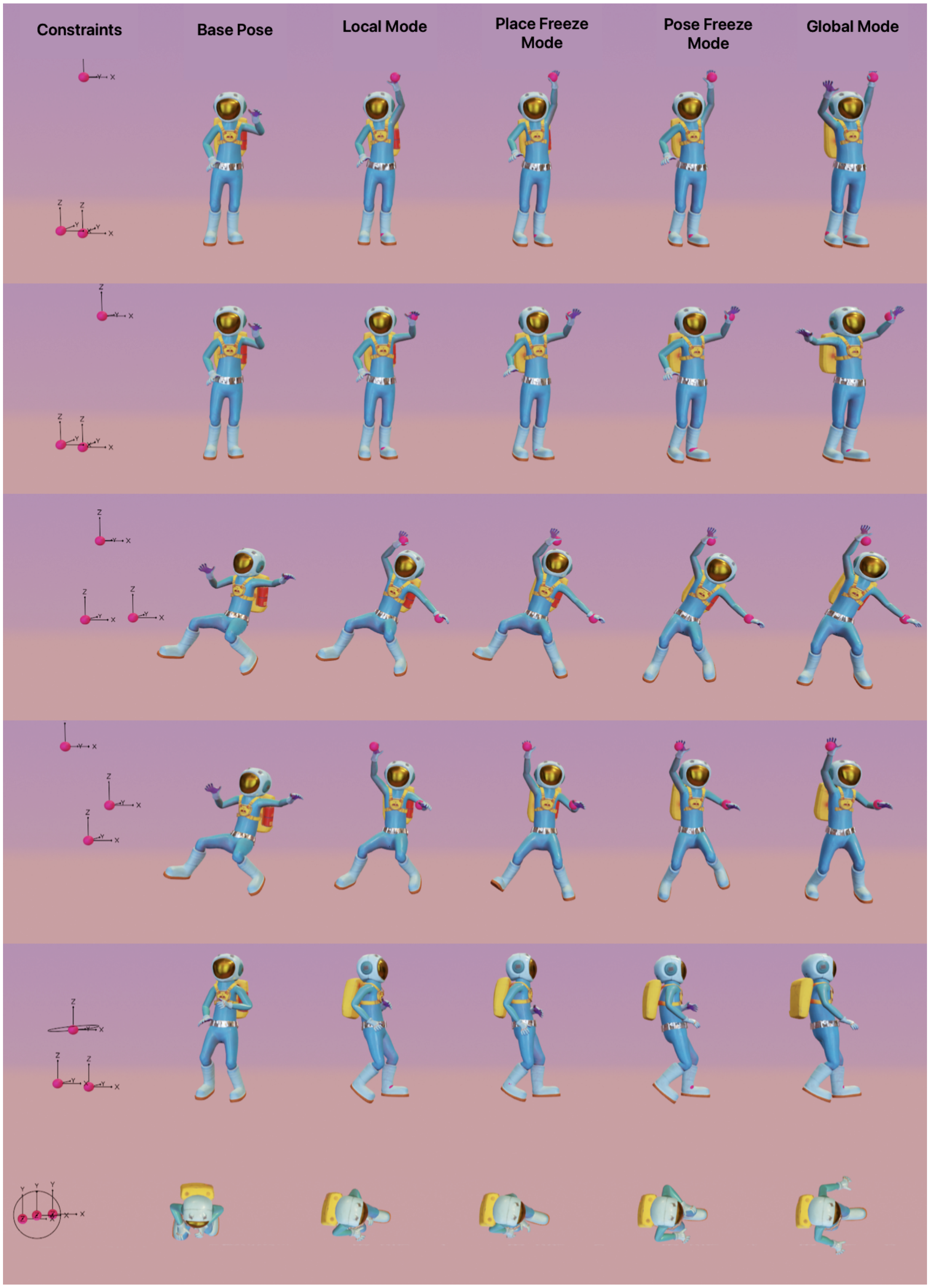


Figure 6: Comparing the characteristics of our model in different execution modes. Column 1 shows the active positional and orientation constraints. Column 2 shows the base pose used for initialization. We see maximum global correlation in global mode (Column 6). The unconstrained body parts move to best satisfy the constraints. Local (Column 3), Place Freeze (Column 4) and Pose Freeze (Column 5) modes retain different characteristics of the base pose. We see this difference the best when an orientation constraint (Row 5 and 6) is applied. The hands behave differently compared to the remaining pose across different modes.