Progressive Guessing to Fixed Point: Rethinking Human Motion Prediction with Deep Equilibrium Models
——Supplementary Material——

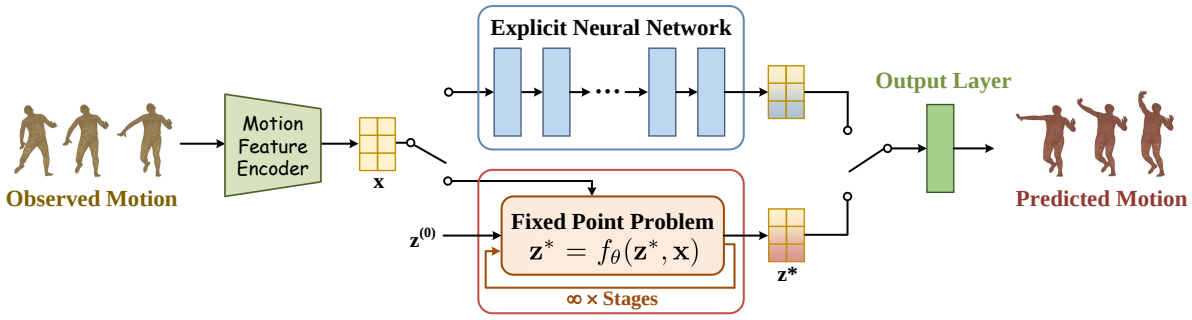

Figure 1. Comparison between conventional explicit neural networks (upper) and our DEQ method (lower) for human motion prediction.

In the following, we first present the related work about error correction in Section A, and provide the illustrations about explicit motion prediction baselines and our implicit DEQ-based method B. We also illustrate different prediction settings in Section C and the detailed fixed-point reusing in Section D. We then provide the theoretical proofs in Section E, and Pytorch-style pseudo-code in Section F. Next, we provide more details about datasets and implementation in Section G. In Section H, we present and analyze further experimental results. Finally, we discuss the limitations of our approach and directions for future work in Section I.

A. Related Work

A.1. Error Correction

Recent studies have explored error correction to enhance prediction accuracy of handling streaming data, such as time series forecasting [9], traffic forecasting [8], and human motion prediction [15, 16]. These approaches incorporate feedback from newly observed data to continuously guide the ongoing predictions. For instance, [15] models the deviation between consecutive motion segments and corrects the base prediction using MLP-based and GRU-based architectures. We draw inspiration from this idea, and integrate an error correction mechanism into DEQ-based models via a well-designed lightweight equivariant correction adapter, without breaking equivariance.

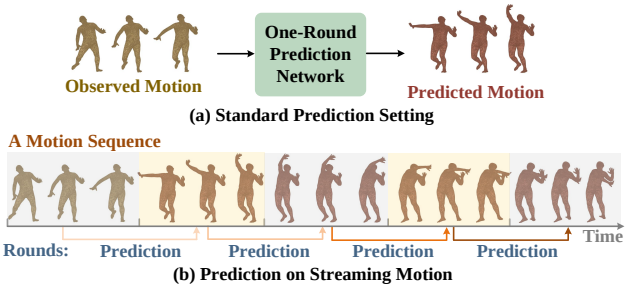

Figure 2. Different prediction settings.

B. Illustrated Comparison

Figure 1 illustrates the comparison between conventional explicit neural networks and our DEQ method for human motion prediction. Given an observed motion sequence, a motion feature encoder first extracts compact representations. The upper branch employs an explicit neural network to predict motion features through a finite stack of layers. In contrast, the lower branch formulates prediction as a fixed-point problem, where the latent state $\mathbf{z}^*$ is obtained by solving $\mathbf{z}^* = f_\theta(\mathbf{z}^*, \mathbf{x})$ through implicit equilibrium iterations. Both branches employ an output layer for final prediction.

C. Different Prediction Settings

Standard human motion prediction (*i.e.,* Sec. 4.2 in main paper) performs a one-shot prediction of a T_f-frame future sequence given a T_p-frame observation, as shown in Figure 2(upper). In contrast, prediction on streaming motion data (*i.e.,* Sec. 4.3 in main paper) operates in a round-by-round

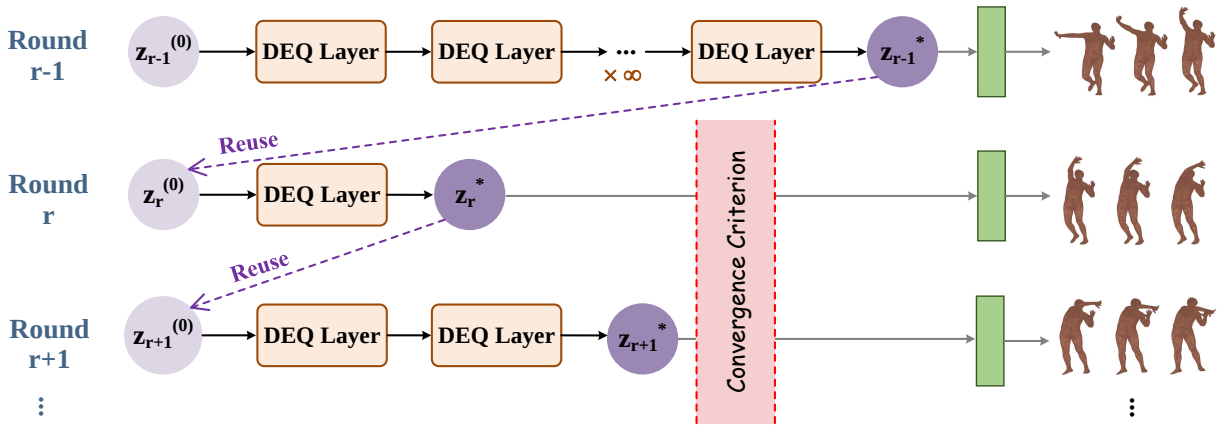

Figure 3. Fixed-point reusing on streaming motion data. We adaptively determine the number of DEQ iterations via convergence criterion.

manner, as shown in Figure 2(lower).

D. Illustrated Fixed-Point Reusing

As shown in Figure 3, our framework reuses the fixed-point from the previous round as a warm initialization for the current DEQ solve. Instead of enforcing a fixed number of iterations, we allow the DEQ solver to adaptively decide how many updates are needed by monitoring a convergence criterion. The iteration process automatically terminates once the fixed-point updates fall below a tolerance, enabling each round to converge with only necessary number of steps.

E. Theoretical Proofs

In this section, we prove Theorem 1 in our paper which shows the overall equivariance of our network. Moreover, we provide more details about Implicit Function Theorem (IFT) [2].

E.1. Proof of Theorem 1

We first review the detailed definitions of equivariance and invariance as follows:

Definition 1 *Let* $\mathbf{X}$ *be an input,* $\mathcal{F}(\cdot)$ *be an operation and* $\mathbf{Y} = \mathcal{F}(\mathbf{X})$ *be the corresponding output. Operation* $\mathcal{F}(\cdot)$ *is called equivariant under Euclidean transformation if*

$$\mathbf{Z}\mathbf{R} + \mathbf{t} = \mathcal{F}(\mathbf{X}\mathbf{R} + \mathbf{t}) \quad \forall \mathbf{R} \in \mathrm{SO}(3), \forall \mathbf{t} \in \mathbb{R}^3.$$

Definition 2 *Let* $\mathbf{X}$ *be an input,* $\mathcal{F}(\cdot)$ *be an operation and* $\mathbf{Y} = \mathcal{F}(\mathbf{X})$ *be the corresponding output. Operation* $\mathcal{F}(\cdot)$ *is called invariant under Euclidean transformation if*

$$\mathbf{Y} = \mathcal{F}(\mathbf{X}\mathbf{R} + \mathbf{t}) \quad \forall \mathbf{R} \in \mathrm{SO}(3), \forall \mathbf{t} \in \mathbb{R}^3.$$

We then give the following proof for Theorem 1.

1. For the initialization $\mathcal{F}_{\mathrm{DL}}(\cdot)$ in Eq.(4) of main paper, the decomposed initial geometric guess $\mathbf{G}^{(0)}$ is equivariant, and the decomposed initial pattern guess $\mathbf{H}^{(0)}$ is invariant:

$$\mathbf{G}^{(0)}\mathbf{R} + \mathbf{t},\ \mathbf{H}^{(0)} = \mathcal{F}_{\mathrm{DL}}(\mathbf{X}_{1:T_p}\mathbf{R} + \mathbf{t}).$$

Proof: When transforming the past motion, we have:

$$\mathbf{X}_{1:T_p}\mathbf{R} + \mathbf{t} \ \Rightarrow\ \mathbf{X}_{rep}\mathbf{R} + \mathbf{t} \ \Rightarrow\ \mathbf{G}^{(0)}\mathbf{R} + \mathbf{t},$$

which shows $\mathbf{G}^{(0)}$ is equivariant to the input motion. For the i-th joint (denoted as $\mathbf{X}_i$) of $\mathbf{X}_{rep}$, we also have:

$$\begin{aligned}
&\triangle(\mathbf{X}_i\mathbf{R} + t) = \triangle(\mathbf{X}_i)\mathbf{R} = \mathbf{V}_i\mathbf{R}, \\
&\rho_i^t = \|\mathbf{V}_i^t\mathbf{R}\|_2^2 = \mathbf{V}_i^t\mathbf{R}\mathbf{R}^T{\mathbf{V}_i^t}^T = \mathbf{V}_i^t{\mathbf{V}_i^t}^T = \|\mathbf{V}_i^t\|_2^2, \\
&\delta_i^t = \mathrm{angle}(\mathbf{V}_i^t\mathbf{R}, \mathbf{V}_i^{t-1}\mathbf{R}) = \frac{\mathbf{V}_i^t\mathbf{R}(\mathbf{V}_i^{t-1}\mathbf{R})^T}{\|\mathbf{V}_i^t\mathbf{R}\|_2\|\mathbf{V}_i^{t-1}\mathbf{R}\|_2} \\
&\quad = \frac{\mathbf{V}_i^t{\mathbf{V}_i^{t-1}}^T}{\|\mathbf{V}_i^t\|_2\|\mathbf{V}_i^{t-1}\|_2} = \mathrm{angle}(\mathbf{V}_i^t, \mathbf{V}_i^{t-1}).
\end{aligned}$$

Therefore, $\mathbf{H}^{(0)} = [\rho; \delta]$ is invariant to the input motion under Euclidean transformation.

2. For the ℓ-th DEQ update layer in Eq.(5) of main paper, geometric feature learning $\mathcal{F}_{\mathrm{G}} = \{\mathcal{F}_{\mathrm{EGI}}, \mathcal{F}_{\mathrm{EGFL}}\}$ is equivariant, and pattern feature learning $\mathcal{F}_{\mathrm{H}} = \{\mathcal{F}_{\mathrm{IPI}}, \mathcal{F}_{\mathrm{IPFL}}\}$ is invariant:

$$\begin{aligned}
\mathbf{G}^{(\ell+1)}\mathbf{R} + \mathbf{t} &= \mathcal{F}_{\mathrm{G}}(\mathbf{G}^{(\ell)}\mathbf{R} + \mathbf{t}, \mathbf{X}_{\mathrm{G}}\mathbf{R} + \mathbf{t}, \mathbf{H}^{(\ell)}, \mathbf{X}_{\mathrm{C}}) \\
\mathbf{H}^{(\ell+1)} &= \mathcal{F}_{\mathrm{H}}(\mathbf{G}^{(\ell)}\mathbf{R} + \mathbf{t}, \mathbf{H}^{(\ell)}, \mathbf{X}_{\mathrm{H}}).
\end{aligned}$$

Proof: We first show the geometric input injection $\mathcal{F}_{\mathrm{EGI}}(\cdot)$ is equivariant. When transforming the observed geometric feature $\mathbf{X}_{\mathrm{G}}$ and the input geometric guess $\mathbf{G}^{(\ell)}$, for every i-th joint $(i = 1, 2, \cdots, J)$,

$$\begin{aligned}
&\phi_{g_1}(\mathbf{G}^{(\ell)}\mathbf{R} + \mathbf{t} - (\overline{\mathbf{G}}^{(\ell)}\mathbf{R} + \mathbf{t})) + \overline{\mathbf{G}}^{(\ell)}\mathbf{R} + \mathbf{t} \\
&= \mathbf{W}_{g_1}(\mathbf{G}^{(\ell)}\mathbf{R} - \overline{\mathbf{G}}^{(\ell)}\mathbf{R}) + \overline{\mathbf{G}}^{(\ell)}\mathbf{R} + \mathbf{t} \\
&= (\mathbf{W}_{g_1}(\mathbf{G}^{(\ell)} - \overline{\mathbf{G}}^{(\ell)}) + \overline{\mathbf{G}}^{(\ell)})\mathbf{R} + \mathbf{t} \\
&= \mathbf{g}\mathbf{R} + \mathbf{t},
\end{aligned}$$

indicating that $\mathbf{g}$ in Eq.(6) is equivariant.

$$\begin{aligned}
&\phi_{g_2}(\mathbf{g}\mathbf{R}+\mathbf{t}+\mathbf{X}_{\mathrm{G}}\mathbf{R}+\mathbf{t}-(\overline{\mathbf{g}}\mathbf{R}+\mathbf{t}) \\
&\quad -(\overline{\mathbf{X}}_{\mathrm{G}}\mathbf{R}+\mathbf{t}))+\overline{\mathbf{X}}_{\mathrm{G}}\mathbf{R}+\mathbf{t} \\
&=\mathbf{W}_{g_2}(\mathbf{g}\mathbf{R}+\mathbf{X}_{\mathrm{G}}\mathbf{R}-\overline{\mathbf{g}}\mathbf{R}-\overline{\mathbf{X}}_{\mathrm{G}}\mathbf{R})+\overline{\mathbf{X}}_{\mathrm{G}}\mathbf{R}+\mathbf{t} \\
&=(\mathbf{W}_{g_2}(\mathbf{g}+\mathbf{X}_{\mathrm{G}}-\overline{\mathbf{g}}-\overline{\mathbf{X}}_{\mathrm{G}})+\overline{\mathbf{X}}_{\mathrm{G}})\mathbf{R}+\mathbf{t} \\
&=\mathbf{P}\mathbf{R}+\mathbf{t}.
\end{aligned}$$

Therefore, $\mathbf{P}$ in Eq.(6) is equivariant, and $\mathcal{F}_{\mathrm{EGI}}(\cdot)$ is equivariant to the input motion under Euclidean transformation.

We then show the pattern input inject $\mathcal{F}_{\mathrm{IPI}}(\cdot)$ is invariant. Since the input pattern guess $\mathbf{H}^{(\ell)}$ and the observed pattern feature $\mathbf{X}_{\mathrm{H}}$ are invariant, when transforming the input motion, we have

$$\begin{aligned}
&\phi_{h_2}(\phi_{h_1}(\mathcal{F}_{\mathrm{DIL}}(\mathbf{X}\mathbf{R}+\mathbf{t}))+\mathcal{F}_{\mathrm{IL}}(\mathbf{X}\mathbf{R}+\mathbf{t})) \\
&=\phi_{h_2}(\phi_{h_1}(\mathbf{H}^{\ell})+\mathbf{X}_{\mathrm{H}}) \\
&=\mathbf{Q}
\end{aligned}$$

Thus, $\mathbf{Q}$ in Eq.(6) is invariant, and $\mathcal{F}_{\mathrm{IPI}}$ is invariant to the input motion under Euclidean transformation.

Combined with the property that $\mathcal{F}_{\mathrm{EGFL}}$ is equivariant and $\mathcal{F}_{\mathrm{IPFL}}$ is invariant (which has been proven in EqMotion [19]), we have:

$$\begin{aligned}
&\mathcal{F}_{\mathrm{G}}(\mathbf{G}^{(\ell)}\mathbf{R}+\mathbf{t},\mathbf{X}_{\mathrm{G}}\mathbf{R}+\mathbf{t},\mathbf{H}^{(\ell)},\mathbf{X}_{\mathrm{C}}) \\
&=\mathcal{F}_{\mathrm{EGFL}}(\mathcal{F}_{\mathrm{EGI}}(\mathbf{G}^{(\ell)}\mathbf{R}+\mathbf{t},\mathbf{X}_{\mathrm{G}}\mathbf{R}+\mathbf{t}),\mathcal{F}_{\mathrm{IPI}}(\mathbf{H}^{(\ell)},\mathbf{X}_{\mathrm{H}}),\mathbf{X}_{\mathrm{C}}), \\
&=\mathcal{F}_{\mathrm{EGFL}}(\mathbf{P}\mathbf{R}+\mathbf{t},\mathbf{Q},\mathbf{X}_{\mathrm{C}}) \\
&=\mathbf{G}^{(\ell+1)}\mathbf{R}+\mathbf{t},
\end{aligned}$$

and

$$\begin{aligned}
&\mathcal{F}_{\mathrm{H}}(\mathbf{G}^{(\ell)}\mathbf{R}+\mathbf{t},\mathbf{H}^{(\ell)},\mathbf{X}_{\mathrm{H}}) \\
&=\mathcal{F}_{\mathrm{IPFL}}(\mathcal{F}_{\mathrm{EGI}}(\mathbf{G}^{(\ell)}\mathbf{R}+\mathbf{t},\mathbf{X}_{\mathrm{G}}\mathbf{R}+\mathbf{t}),\mathcal{F}_{\mathrm{IPI}}(\mathbf{H}^{(\ell)},\mathbf{X}_{\mathrm{H}}),\mathbf{X}_{\mathrm{H}}) \\
&=\mathcal{F}_{\mathrm{IPFL}}(\mathbf{P}\mathbf{R}+\mathbf{t},\mathbf{Q},\mathbf{X}_{\mathrm{H}}) \\
&=\mathbf{H}^{(\ell+1)}.
\end{aligned}$$

This shows the geometric learning $\mathcal{F}_{\mathrm{G}}=\{\mathcal{F}_{\mathrm{EGI}},\mathcal{F}_{\mathrm{EGFL}}\}$ is equivariant, and pattern learning $\mathcal{F}_{\mathrm{H}}=\{\mathcal{F}_{\mathrm{IPI}},\mathcal{F}_{\mathrm{IPFL}}\}$ is invariant in the l-th DEQ update layer.

3. The adapter $\mathcal{F}_{\mathrm{ECA}}(\cdot)$ in Eq.(12) of main paper is equivariant:

$$\widehat{\mathbf{Y}}_{r+1}\mathbf{R}+\mathbf{t}=\mathcal{F}_{\mathrm{ECA}}(\mathbf{G}^{*}_{r+1}\mathbf{R}+\mathbf{t},\mathbf{H}^{*}_{r+1},\mathbf{H}').$$

Proof: We adopt a one-layer EGNN [10, 14] to incorporate the deviation information. For notational simplicity, we denote $\mathbf{G}^{*}_{r+1}$ as $\mathbf{G}$. The whole process of our equivariant correction adapter is formally expressed as:

$$\begin{aligned}
&\mathbf{H}'=\mathcal{F}_{\mathrm{DL}}(\widehat{\mathbf{Y}}_r-\mathbf{X}_{r+1}),\ \mathbf{H}=\mathbf{H}^{*}_{r+1}+\mathsf{MLP}(\mathbf{H}'), \\
&\mathbf{m}_{ij}=\psi_{\mathrm{m}}([\mathbf{h}_i;\mathbf{h}_j;\|\mathbf{G}_i-\mathbf{G}_j\|_{2,\mathrm{col}}]), \\
&\mathbf{G}'_i=\mathbf{G}_i+\frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\psi_x(\mathbf{m}_{ij})\cdot(\mathbf{G}_i-\mathbf{G}_j), \\
&\widehat{\mathbf{Y}}_{r+1}=\mathcal{F}_{\mathrm{EOL}}(\mathbf{G}'),
\end{aligned}$$

where ψ_m and ψ_x represent MLPs, $\mathcal{N}(i)$ denotes the set of neighboring joints associated with i-th joint, $\mathbf{m}_{ij}$ refers to an $\mathbb{E}(3)$-invariant message transmitted from joint j to joint i, which is used to aggregate weights.

As proven previously, $\mathbf{H}'$ obtained from $\mathcal{F}_{\mathrm{DL}}(\cdot)$ is invariant, and thus $\mathbf{H}$ is invariant. Besides, for the c-th column ($c=1,2,\cdots,C$), we have:

$$\begin{aligned}
&\|\mathbf{G}_i\mathbf{R}+\mathbf{t}-(\mathbf{G}_j\mathbf{R}+\mathbf{t})\|_2 \\
&=\|\mathbf{G}_i\mathbf{R}-\mathbf{G}_j\mathbf{R}\|_2 \\
&=\|\mathbf{G}_i-\mathbf{G}_j\|_2.
\end{aligned}$$

Therefore, the column-wise ℓ_2-distance of geometric feature is invariant. Since the pattern feature $\mathbf{H}$ is invariant, the edge feature $\mathbf{m}_{ij}$ is invariant. We then have the following equivariance:

$$\begin{aligned}
&\mathbf{G}_i\mathbf{R}+\mathbf{t}+\frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\psi_x(\mathbf{m}_{ij})(\mathbf{G}_i\mathbf{R}+\mathbf{t}-(\mathbf{G}_j\mathbf{R}+\mathbf{t})) \\
&=\mathbf{G}_i\mathbf{R}+\mathbf{t}+\frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\psi_x(\mathbf{m}_{ij})\cdot(\mathbf{G}_i-\mathbf{G}_j)\mathbf{R} \\
&=(\mathbf{G}_i+\frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\psi_x(\mathbf{m}_{ij})\cdot(\mathbf{G}_i-\mathbf{G}_j))\mathbf{R}+\mathbf{t} \\
&=\mathbf{G}'_i\mathbf{R}+\mathbf{t}.
\end{aligned}$$

Thus, $\mathbf{G}'_i$ is equivariant. As noted in EqMotion, $\mathcal{F}_{\mathrm{EOL}}(\cdot)$ is equivariant, *i.e.,* $\widehat{\mathbf{Y}}_{r+1}\mathbf{R}+\mathbf{t}=\mathcal{F}_{\mathrm{EOL}}(\mathbf{G}'\mathbf{R}+\mathbf{t})$. Combining these equations, we show the equivariance of our adapter $\mathcal{F}_{\mathrm{ECA}}(\cdot)$.

4. The forward pass in Eq.(7) of main paper preserves original equivariance/invariance under Anderson solver.

Proof: We adopt the Anderson iterative method [1, 18] to update the state $\mathbf{z}^{(\ell)}$ by leveraging the previous m_ℓ results. The update is computed as follows:

$$\mathbf{z}^{(\ell+1)}=\beta\sum_{i=0}^{m_\ell}\alpha_i^{\ell}\mathcal{F}(\mathbf{z}^{(\ell-m_\ell+i)})+(1-\beta)\sum_{i=0}^{m_\ell}\alpha_i^{\ell}\mathbf{z}^{(\ell-m_\ell+i)},$$

where $\mathcal{F}=\{\mathcal{F}_{\mathrm{G}},\mathcal{F}_{\mathrm{H}}\}$, and α_i^{ℓ} is the weight for the i-th previous step at the ℓ-th iteration, subject to $\sum_{i=0}^{m_\ell}\alpha_i^{\ell}=1$, and β is a hyper-parameter. More details are in [18].

When transforming input geometric guess $\mathbf{G}$, we have:

$$\begin{aligned}
&\beta\sum_{i=0}^{m_\ell}\alpha_i^{\ell}\mathcal{F}(\mathbf{G}^{(\ell-m_\ell+i)}\mathbf{R}+\mathbf{t})+(1-\beta)\sum_{i=0}^{m_\ell}\alpha_i^{\ell}(\mathbf{G}^{(\ell-m_\ell+i)}\mathbf{R}+\mathbf{t}) \\
&=\beta\sum_{i=0}^{m_\ell}\alpha_i^{\ell}\mathcal{F}(\mathbf{G}^{(\ell-m_\ell+i)})\mathbf{R}+\beta\sum_{i=0}^{m_\ell}\alpha_i^{\ell}\mathbf{t} \\
&\quad+(1-\beta)\sum_{i=0}^{m_\ell}\alpha_i^{\ell}(\mathbf{G}^{(\ell-m_\ell+i)})\mathbf{R}+(1-\beta)\sum_{i=0}^{m_\ell}\alpha_i^{\ell}\mathbf{t} \\
&=\left(\beta\sum_{i=0}^{m_\ell}\alpha_i^{\ell}\mathcal{F}(\mathbf{G}^{(\ell-m_\ell+i)})+(1-\beta)\sum_{i=0}^{m_\ell}\alpha_i^{\ell}\mathbf{G}^{(\ell-m_\ell+i)}\right)\mathbf{R} \\
&\quad+\sum_{i=0}^{m_\ell}\alpha_i^{\ell}\mathbf{t} \\
&=\mathbf{G}^{(\ell+1)}\mathbf{R}+\mathbf{t}.
\end{aligned}$$

Similarly, we also have the invariance for pattern guess $\mathbf{H}$:

$$\mathbf{H}^{(\ell+1)} = \beta \sum_{i=0}^{m_\ell} \alpha_i^\ell \mathcal{F}(\mathbf{H}^{(\ell-m_\ell+i)}) + (1-\beta) \sum_{i=0}^{m_\ell} \alpha_i^\ell \mathbf{H}^{(\ell-m_\ell+i)}.$$

This shows that the forward pass of DEQs preserves original equivariance/invariance under Anderson iterative solver.

5. The fixed-point reusing $\mathbf{z}_{r+1}^{(0)} = \mathbf{z}_r^*$ does not affect the network's equivariance and invariance.

Proof: For the first round, $\mathbf{G}_1^{(0)}, \mathbf{H}_1^{(0)} = \mathcal{F}_{\text{DL}}(\mathbf{X}_{1:T_p})$ are equivariant and invariant, respectively. And the final fixed point for the first round $\mathbf{G}_1^*$ and $\mathbf{H}_1^*$ retain the same properties, as proven before. In the second round, we reuse these fixed points as the initialization, *i.e.,* $\mathbf{G}_2^{(0)} = \mathbf{G}_1^*$ and $\mathbf{H}_2^{(0)} = \mathbf{H}_1^*$, which naturally preserves the equivariance and invariance. Therefore, fixed-point reuse does not alter these properties.

Combining all components, we conclude that the entire MotionDEQ architecture remains equivariant. □

E.2. Implicit Function Theorem

Theorem 2 *Assume that the training loss is $\mathcal{L}_{\text{MSE}}(\cdot)$ and the network is parameterized by θ. Given a fixed-point motion representation $\mathbf{z}^* = \{\mathbf{G}^*, \mathbf{H}^*\}$ and the input $\mathbf{x} = \{\mathbf{X}_G, \mathbf{X}_H, \mathbf{X}_C\}$, the gradient of DEQ motion is given by*

$$\frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \theta} = \frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \mathbf{z}^*} \left(\mathbf{I} - \frac{\partial f(\mathbf{z}^*, \mathbf{x}|\theta)}{\partial \mathbf{z}^*} \right)^{-1} \frac{\partial f(\mathbf{z}^*, \mathbf{x}|\theta)}{\partial \theta}.$$

The proof of implicit function theorem (IFT) can be found in [2]. However, we have to compute the inverse of the Jacobian matrix $\mathcal{J} = (\mathbf{I} - \frac{\partial f(\mathbf{z}^*, \mathbf{x}|\theta)}{\partial \mathbf{z}^*})^{-1}$, which requires $\mathcal{O}(N^3)$ computational complexity. Recent efforts employ a linear system to approximate the inverse-Jacobian matrix:

$$\mathcal{J}^T = \mathcal{J}^T \frac{\partial f(\mathbf{z}^*, \mathbf{x}|\theta)}{\partial \mathbf{z}^*} + \frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \mathbf{z}^*},$$

where T represents the transpose of the matrix. However, this approach still relies on an external optimization solver to iteratively compute the solution. As noted in [4], we can approximate $\mathcal{J}$ with its Neumann series expansion to obtain gradients:

$$\frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \theta} \approx \lim_{N \to \infty} \frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \mathbf{z}^*} \sum_{n=0}^{N} \left(\frac{\partial f(\mathbf{z}^*, \mathbf{x}|\theta)}{\partial \mathbf{z}^*} \right)^n \frac{\partial f(\mathbf{z}^*, \mathbf{x}|\theta)}{\partial \theta}.$$

Many efforts set $N = 0$ to derive Jacobian-Free Backpropagation (JFB) for approximating the gradients. While this inexact gradient simplifies the backward pass to a single-step computation $\frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \theta} \approx \frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \mathbf{z}^*} \frac{\partial f(\mathbf{z}^*, \mathbf{x}|\theta)}{\partial \theta}$, it yields suboptimal prediction performance. Instead, we employ $N = 1$ to obtain truncated gradients to better approximate $\mathcal{J}$:

$$\frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \theta} \approx \frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \mathbf{z}^*} \left(\mathbf{I} + \frac{\partial f(\mathbf{z}^*, \mathbf{x}|\theta)}{\partial \mathbf{z}^*} \right) \frac{\partial f(\mathbf{z}^*, \mathbf{x}|\theta)}{\partial \theta}.$$

F. Pseudo Code

We provide a Pytorch-style pseudo-code for the DEQ motion prediction in Algorithm 1.

Algorithm 1 DEQ motion training

```
def train(
    M, # iteration forward times
    encoder, # past motion extractor
    decoder, # output layer
    solver, # fixed-point solver, e.g., Anderson
    func, # one DEQ update layer
    MSE, # loss function
    x, # observed motion (B, Tx, J*3)
    y, # ground truth future motion (B, Ty, J*3)
    z, # fixed-point (B, J, C*3+D)
    prev_z, # fixed point of previous round
    freq, # number of correction losses
    gamma # weights of correction
):
    """
    B: batch size
    J: joint number
    C: geometric feature dimension
    D: pattern feature dimension
    Tx/Ty: past/future motion length
    """
    cond = encoder(x)
    with torch.no_grad():
        # Fixed-Point Reusing
        z, zm = solver(func, cond, freq, z0=prev_z)

    if training:
        loss = MSE(y, decoder(z))
        if sparse_correction:
            for i in range(freq):
                zm_i = decoder(zm[i])
                loss += gamma[i] * MSE(y, zm_i)
        return z, loss
    return z
```

G. Experimental Details

G.1. Dataset Description

Human3.6M [7] includes motion recordings from seven subjects (S1, S5, S6, S7, S8, S9, and S11) performing 15 categories of daily actions, each represented by 22 skeletal joints. We follow the standard protocol by downsampling the sequences from 50 fps to 25 fps and converting them into 3D joint coordinates after removing global rotation and translation. Subjects S5 and S11 are designated for testing and validation, respectively, while the remaining subjects serve as the training set.

CMU-MoCap contains a diverse set of motions spanning eight action categories and originally annotated with 38 joints in 3D coordinates after eliminating global transformation. Following prior work [3, 12], we keep 25 commonly used joints and split the data into independent training and testing subsets.

Table 1. The hyper-parameters of MotionDEQ.

Hyperparameters	Notation	Value
The observed motion	$\mathbf{X}_{1:T_p}$	
The predicted motion	$\widehat{\mathbf{Y}}_{T_p+1:T_p+T_f}$	
The ground truth future motion	$\mathbf{Y}$	
The observed geometric feature	$\mathbf{X}_{\mathrm{G}}$	
The observed pattern feature	$\mathbf{X}_{\mathrm{H}}$	
The observed interaction graph	$\mathbf{X}_{\mathrm{C}}$	
The ℓ-th geometric guess	$\mathbf{G}^{(\ell)}$	
The ℓ-th pattern guess	$\mathbf{H}^{(\ell)}$	
The ℓ-th guess	$\mathbf{z}^{(\ell)}$	
The final fixed point	$\mathbf{z}^*$	
The r-prediction round fixed point	$\mathbf{z}_r^*$	
The network parameters	θ	
The initialized decomposed layer	$\mathcal{F}_{\mathrm{DL}}(\cdot)$	
The input geometric injection layer	$\mathcal{F}_{\mathrm{EGI}}(\cdot)$	
The input pattern injection layer	$\mathcal{F}_{\mathrm{IPI}}(\cdot)$	
The equivariant correction adapter	$\mathcal{F}_{\mathrm{ECA}}(\cdot)$	
The fixed point iteration steps for training	T_{train}	20
The fixed point iteration steps for inference	T_{infer}	30
The number of sparse correction losses		3
The weight of sparse correction losses	γ	0.8
The geometric feature dimension for short-term prediction	C	72
The geometric feature dimension for long-term prediction	C	96
The pattern feature dimension	D	64
The early stopping tolerance	ϵ	0.001
The learning rate		0.0005
The learning rate decay		0.98
The weight normalization		True
The DCT using		True
The batch size		100
The number of epochs		100

3DPW [17] provides challenged in-the-wild human motion sequences captured across both indoor and outdoor environments. Each pose is represented in 3D with 23 joints, and the framerate of each motion is recorded at 30 fps.

G.2. Implementation Details

To prevent potential ambiguity in the notation of MotionDEQ, we summarize all symbols in Table 1. For the sake of reproducibility, we further include the hyperparameters used during training.

H. Additional Experiments

H.1. Maximum Refinement Iterative Steps

We investigate the impact of different fixed-point iteration steps (T_{train}) during the training of our MotionDEQ model. As summarized in Table 2, setting $T_{\text{train}} = 20$ yields the best average performance with an MPJPE of 65.95. In general, increasing the number of refinement steps during training improves predictive accuracy. However, performance does not further improve beyond $T_{\text{train}} = 20$, which may be attributed to the model's already sufficient representational capacity, making additional refinement steps redundant.

Table 2. Different fixed-point iteration steps during training on 3DPW dataset.

	T_{train}	200ms	400ms	800ms	average
MotionDEQ	5	32.97	66.84	103.58	69.45
	10	31.28	63.06	99.62	67.10
	20	**30.63**	**61.38**	98.11	**65.95**
	30	30.69	61.54	97.85	65.97

H.2. Long-term Motion Prediction

Table 3 reports the long-term prediction accuracy (more than 400ms but less than 1000ms) on the Human3.6M

Table 3. Comparisons of long-term prediction on Human3.6M. Results at 560ms and 1000ms in the future are shown. '†' denotes the average results over 5 runs from our reimplementation using released code due to its high sensitivity to network initialization. The best and second-best results are marked in **bold** and underline, respectively.

scenarios	walking		eating		smoking		discussion		directions		greeting		phoning		posing	
millisecond	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms
Res-sup. [13]	81.7	100.7	79.9	100.2	94.8	137.4	121.3	161.7	110.1	152.5	156.1	166.5	141.2	131.5	194.7	240.2
LTD [12]	54.1	59.8	53.4	77.8	50.7	72.6	91.6	121.5	71.0	101.8	115.4	148.8	69.2	103.1	114.5	173.0
MSR-GCN [3]	52.7	63.0	52.5	77.1	49.5	71.6	88.6	117.6	71.2	100.6	116.3	147.2	68.3	104.4	116.3	174.3
PGBIG [11]	48.1	56.4	51.1	76.0	46.5	69.5	87.1	118.2	69.3	100.4	110.2	143.5	65.9	102.7	106.1	164.8
EqMotion† [19]	43.1	53.6	47.8	73.0	**41.0**	**63.4**	**76.4**	**106.8**	70.0	100.9	109.0	143.3	64.6	101.5	**88.9**	**145.0**
SiMLPe [5]	46.8	55.7	49.6	74.5	47.2	69.3	85.7	116.3	73.1	106.7	**99.8**	**137.5**	66.3	103.3	103.4	168.7
LuKAN [6]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ours	**42.7**	**53.1**	**47.0**	**72.5**	41.8	64.8	80.6	112.7	**69.1**	**99.8**	104.1	141.4	**63.8**	**100.2**	93.6	152.5

scenarios	purchases		sitting		sittingdown		takingphoto		waiting		walkingdog		walkingtogether		average	
millisecond	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms
Res-sup. [13]	122.7	160.3	167.4	201.5	205.3	277.6	117.0	143.2	146.2	196.2	191.3	209.0	107.6	131.1	97.6	130.5
LTD [12]	102.0	143.5	78.3	119.7	100.0	150.2	77.4	119.8	79.4	108.1	111.9	148.9	55.0	65.6	81.6	114.3
MSR-GCN [3]	101.6	139.2	78.2	120.0	102.8	155.5	77.9	121.9	76.3	106.3	111.9	148.2	52.9	65.9	81.1	114.2
PGBIG [11]	95.3	133.3	**74.4**	116.1	96.7	147.8	74.3	118.6	72.2	103.4	104.7	139.8	51.9	64.3	76.9	110.3
EqMotion† [19]	97.2	137.9	75.2	117.3	97.2	149.1	76.4	121.3	72.0	105.4	105.8	141.5	**46.0**	**58.4**	74.0	107.9
SiMLPe [5]	**93.8**	**132.5**	75.4	**114.1**	95.7	142.4	71.0	112.8	71.6	104.6	105.6	141.2	50.8	61.5	75.7	109.4
LuKAN [6]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	75.7	109.3
Ours	94.5	134.5	74.7	114.8	**94.6**	**141.6**	**70.9**	**112.0**	**70.4**	**102.1**	**104.5**	**139.4**	50.2	61.0	**73.5**	**106.8**

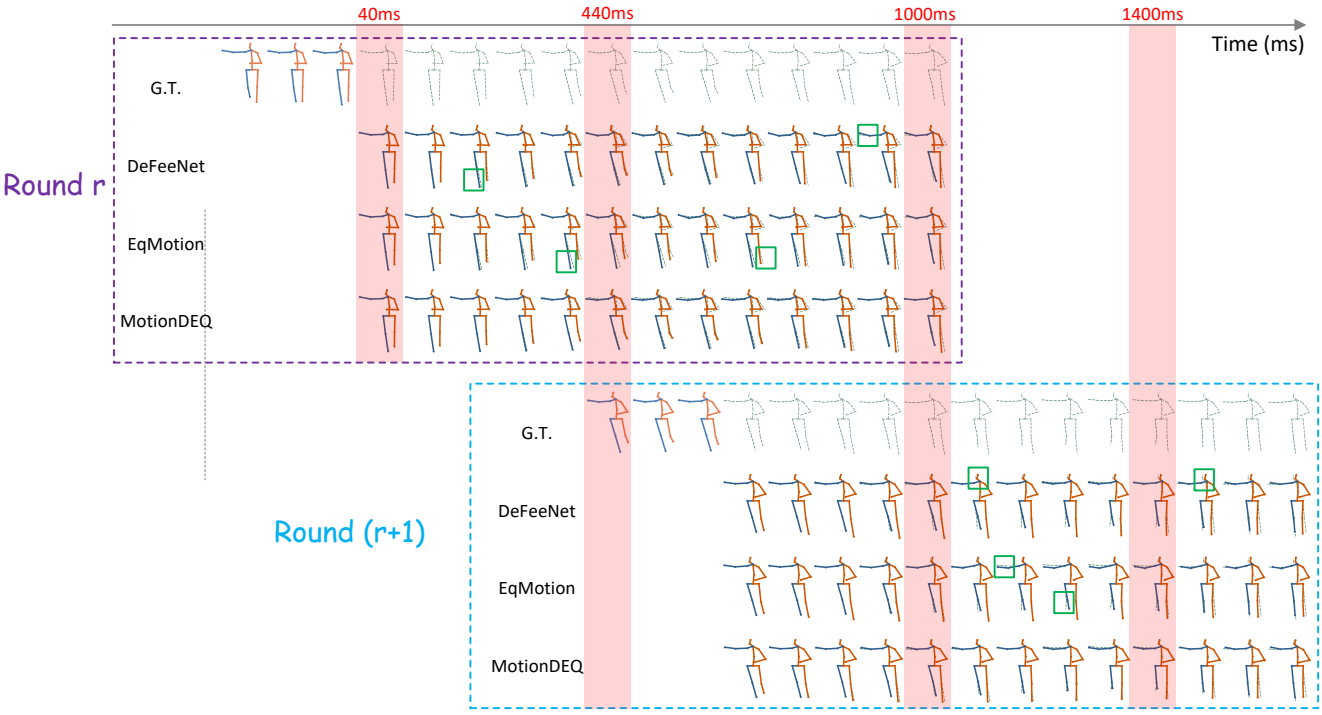

Figure 4. Visualizations on streaming motion data of our MotionDEQ, compared to other baselines.

dataset. Our MotionDEQ attains the highest accuracy on the majority of actions and achieves the second-best performance on the others. Furthermore, MotionDEQ outperforms EqMotion in long-term prediction, further demonstrating the powerful representational capability of DEQs.

H.3. Visualization Results of Streaming Prediction

In Figure 4, we present additional qualitative results of motion prediction on streaming data, comparing our method with recent baselines, including DeFeeNet [15] and EqMotion [19]. The green dotted line indicates the ground truth future motion. As shown, our predictions consistently align more closely with the ground truth across two consecutive prediction rounds, demonstrating the ability of our method to effectively handle streaming human motion data and generate more accurate motion predictions.

I. Limitation & Future Work

While our MotionDEQ demonstrates promising results in human motion prediction, the training cost per epoch is generally higher than that of explicit models, primarily due to the iterative fixed-point solving process in the forward pass. Inference time remains acceptable in practice, especially under our warm initial guess strategy, which significantly reduces the number of iterations required in streaming settings, as validated in Figure. 6(b) of the main paper. Future work can be made to to develop more efficient and stable DEQ training strategies. Another promising avenue is to explore DEQs in multi-agent and interactive scenarios, such as molecule dynamics and pedestrian trajectory prediction.

References

[1] Donald G Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM*, 12(4):547–560, 1965. 3

[2] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *NeurIPS*, 32, 2019. 2, 4

[3] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In *ICCV*, pages 11467–11476, 2021. 4, 6

[4] Zhengyang Geng, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin. On training implicit models. *NeurIPS*, 34:24247–24260, 2021. 4

[5] Wen Guo, Yuming Du, Xi Shen, Vincent Lepetit, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Back to mlp: A simple baseline for human motion prediction. In *WACV*, pages 4809–4819, 2023. 6

[6] Md Zahidul Hasan, Abdessamad Ben Hamza, and Nizar Bouguila. Lukan: A kolmogorov-arnold network framework for 3d human motion prediction. In *BMVC*, 2025. 6

[7] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(7):1325–1339, 2013. 4

[8] Daejin Kim, Youngin Cho, Dongmin Kim, Cheonbok Park, and Jaegul Choo. Residual correction in real-time traffic forecasting. In *CIKM*, pages 962–971, 2022. 1

[9] Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, Baolin Sun, and Mingyuan Zhou. Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In *ICLR*, 2024. 1

[10] Zongzhao Li, Jiacheng Cen, Bing Su, Tingyang Xu, Yu Rong, Deli Zhao, and Wenbing Huang. Large language-geometry model: When llm meets equivariance. In *ICML*, 2025. 3

[11] Tiezheng Ma, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Progressively generating better initial guesses towards next stages for high-quality human motion prediction. In *CVPR*, pages 6437–6446, 2022. 6

[12] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *ICCV*, pages 9489–9497, 2019. 4, 6

[13] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *CVPR*, pages 2891–2900, 2017. 6

[14] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *ICML*, pages 9323–9332, 2021. 3

[15] Xiaoning Sun, Huaijiang Sun, Bin Li, Dong Wei, Weiqing Li, and Jianfeng Lu. Defeenet: Consecutive 3d human motion prediction with deviation feedback. In *CVPR*, pages 5527–5536, 2023. 1, 6

[16] Xiaoning Sun, Huaijiang Sun, Bin Li, Dong Wei, Weiqing Li, and Jianfeng Lu. Moml: Online meta adaptation for 3d human motion prediction. In *CVPR*, pages 1042–1051, 2024. 1

[17] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, pages 601–617, 2018. 5

[18] Homer F Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011. 3

[19] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *CVPR*, pages 1410–1420, 2023. 3, 6