Test-time Personalizable Forecasting of 3D Human Poses

Qiongjie Cui[1*] Huaijiang Sun[1†]
Jianfeng Lu[1] Weiqing Li[1] Bin Li [2] Hongwei Yi[3], Haofan Wang[4]
[1]Nanjing University of Science and Technology, Nanjing, China
[2]Tianjin AiForward Science and Technology Co., Ltd., China
[3]Max Planck Institute for Intelligent Systems, Tubingen, Germany [4]Xiaohongshu Inc
cuiqiongjie@njust.edu.cn sunhuaijiang@njust.edu.cn

Abstract

Current motion forecasting approaches typically train a deep end-to-end model from the source domain data, and then apply it directly to target subjects. Despite promising results, they remain non-optimal, due to privacy considerations, the test person and his/her natural properties (e.g., behavioral trait) are typically unseen in training. In this case, the source pre-trained model has a low ability to adapt to these out-of-source characteristics, resulting in an unreliable prediction. To tackle this issue, we propose a novel helper-predictor test-time personalization approach (H/P-TTP), which allows for a generalizable representation of out-of-source subjects to gain more realistic predictions. Concretely, the helper is preceded by explicit and implicit augmenters, where the former yields noisy sequences to improve robustness, while the latter is to generate novel-domain data with an adversarial learning paradigm. Then, the domain-generalizable learning is achieved where the helper can extract cross-subject invariant-knowledge to update the predictor. At test time, given a new person, the predictor is able to be further optimized to empower personalized capabilities to the specific properties. Extensive experiments show that with H/P-TTP, the existing models are significantly improved for various unseen subjects. The project page will be available at https://sites.google.com/view/hp-ttp.

1. Introduction

Forecasting the high-fidelity future human poses, conditioned on a given historical sequence, has attracted increasing attention in recent years. In a variety of 3D vision-based applications, such as autonomous driving and robot navigation, it offers tremendous potential, especially for tasks that call for seamless interaction with humans [11, 9, 29].

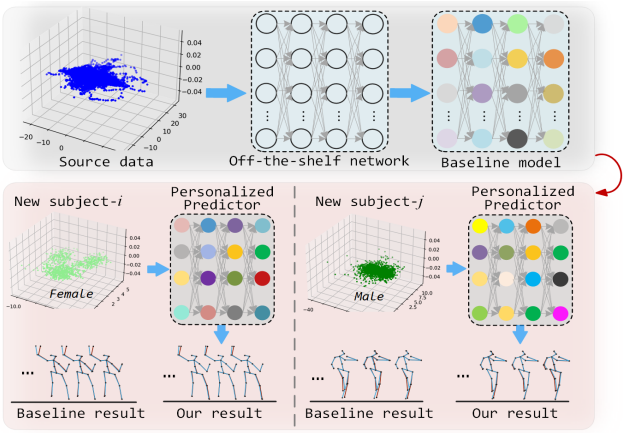

Figure 1. Starting from off-the-shelf networks, we first train a baseline model [26] on the source data, which is further optimized and personalized to the target person during the test phase. Here, we present an illustration of the pose sequence of various subjects, in which each 3D feature embedding is created by t-SNE. We notice that the motion patterns of test subjects fail to match the source one. Moreover, our prediction skeleton (blue) is closer to the ground truth (red) against the vanilla baseline [26].

Recent years have witnessed a proliferation in the exploration of this emerging field [16, 5, 2]. In particular, ongoing efforts are being devoted to deep learning approaches, which typically train a generic model from large-scale datasets, and then apply it to new test subjects and samples during the deployment phase. We notice that these methods have become mainstream and are now rapidly evolving [49, 58, 24, 32, 3, 33].

Despite notable successes, the existing approaches are sub-optimal, because the generic model cannot be adapted to unseen personalized subjects [48, 21, 52]. To be precise, due to data limitations or privacy concerns, unseen test persons, as well as their individual properties (*e.g.*, age, gender, or behavioral trait), are typically disjointed with the training. It may lead to a large distribution gap between the target and source domain [19, 42, 50], as shown in Figure 1.

*Work at Xiaohongshu Inc
†Corresponding author

For example, for the H3.6M dataset [22], a widely-adopted data splitting strategy takes subject-5 (S5) as the test and the rest as the training [10, 34, 25]. Here, the inherent properties (height, and body proportion) of S5 are significantly different from the other subjects; hence, the pre-trained parameters may not be specific or personalized for the S5. We also note that, out-of-source persons are inevitable in the deployment, which poses a major bottleneck to current predictive algorithms [3].

To solve it, a novel helper-predictor test-time personalization framework (H/P-TTP) is proposed. Starting with ready-to-use networks, our approach can attain tailored parameters according to the characteristics of various unseen subjects. Concretely, the H/P-TTP advances the recent success of knowledge distillation into test-time adaptation by augmenting source instances to out-of-source distributions, enabling domain-generalizable learning. The helper involves an identical architecture with the predictor, except for the explicit and implicit augmenters. To improve the robustness, by adding noise to the samples, the explicit augmenter extracts task-relevant information and ignores irrelevant ones. In contrast, the implicit augmenter is a trainable motion-style transformer to achieve the augmentations with new properties, which follows the adversarial training paradigm, with the goal of maximizing the discrepancy with the source subject, while ensuring semantic proximity. With the *max-min* optimization, the helper would be able to observe and cope with the novel-styled samples to gain the cross-subject invariant context, and distill the knowledge to update the predictor.

During testing, the model can be further updated to adapt to an unseen test person. We also observe that due to psychological or environmental factors, even for the same person, his/her motion characteristics may vary. More intuitively, for test data with large domain gaps, the predictor needs greater updates, and conversely, a mild personalization is expected. For this purpose, an adaptive learning rate strategy is proposed to dynamically consider the demand for test-time personalization. Owing to the adaptive test-time learning rate, our T/P-TTP can adapt to the intrinsic attributes of an unseen test subject, as well as to his/her specific motion dynamics, as shown in Figure 2.

Our contributions are as follows: 1) We notice the issue: the existing approaches are ill-adapted to the out-of-source subjects, and propose a novel H/P-TTP model to address it. 2) We also propose an adaptive learning rate strategy to further consider the extent to which the model needs to be personalized. 3) Our H/P-TTP is integrated into many existing motion forecasting approaches, and substantially boosts the performance on unseen subjects.

2. Related Work

Human Pose Forecasting. Deep end-to-end networks are highly sought-after for solving this issue [5, 41]. In particular, due to the intrinsic temporal nature of 3D human motion, it is typically considered a sequence-to-sequence problem, where various variants of RNNs have been developed [35, 36, 15, 17]. Despite the promising development, RNNs suffer from error accumulation, convergence to an undesired static pose [24, 27, 25].

To tackle these drawbacks, feed-forward networks have been recently explored as a potential alternative [25, 39, 34, 33]. Particularly, with better interpretability, GCNs are introduced to extract the local semantic connectivity of the 3D human skeleton [34, 10, 8, 11, 24, 27, 36, 26, 54]. Transformers have also been introduced to capture the long-range dependencies of motion sequences [32].

We have noticed that, current deep learning-based approaches have emerged as the prevailing solution with promising results. However, an obvious challenge still exists, namely, the source pre-trained model cannot be personalized for the specific properties of unseen target persons.

Domain Generalization (DG). It aims to generalize the learned model from source domain datasets to unseen target domains. Recent advances reveal that data augmentation is an attractive strategy for DG, with the increase in the diversity of training and test data [20, 1, 56, 59]. For this purpose, a variety of works are proposed, *e.g.*, adversarial attack [57, 60], image perturbation [60], and domain randomization [46, 47], where the main heuristic is to learn cross-domain invariant feature. Particularly, [55] introduces a novel-style augmentation in teacher-student frameworks, in which novel synthesized data allows for domain-invariant context to be transferred to the student network. In contrast to the standard DG, which trains a more generalizable model, our work considers going to improve the off-the-shelf network to adapt to the specific characteristics of unseen new subjects and their motion sequence.

Test-time Adaptation (TTA). Our test-time personalization is relevant to the TTA, which is expected to improve the generalization ability of out-of-distribution test samples. TTA falls into source-free adaptation, and before making decisions, allows for fine-tuning the pre-trained model to adapt to the specific test samples during inference, without accessing the source domain data. For example, according to the test data, prediction-time batch normalization (PTBN) [38] can recalibrate the BN statistics. [51] utilizes test-time entropy minimization (TENT) to update the trainable parameters in BN layers. Test-time training (TTT) [44] adjusts the feature extractor in the test phase via the update of the self-supervised auxiliary task of image rotation prediction. TTT++ [30] exploits contrastive learning to align the first- and second-order statistics of the test samples with the source domain data, improving the TTT further. [7] introduces an auxiliary image reconstruction to the primary deblurring task, and then resorts to meta-learning to fine-tune the primary branch to adapt to test samples. Consid-

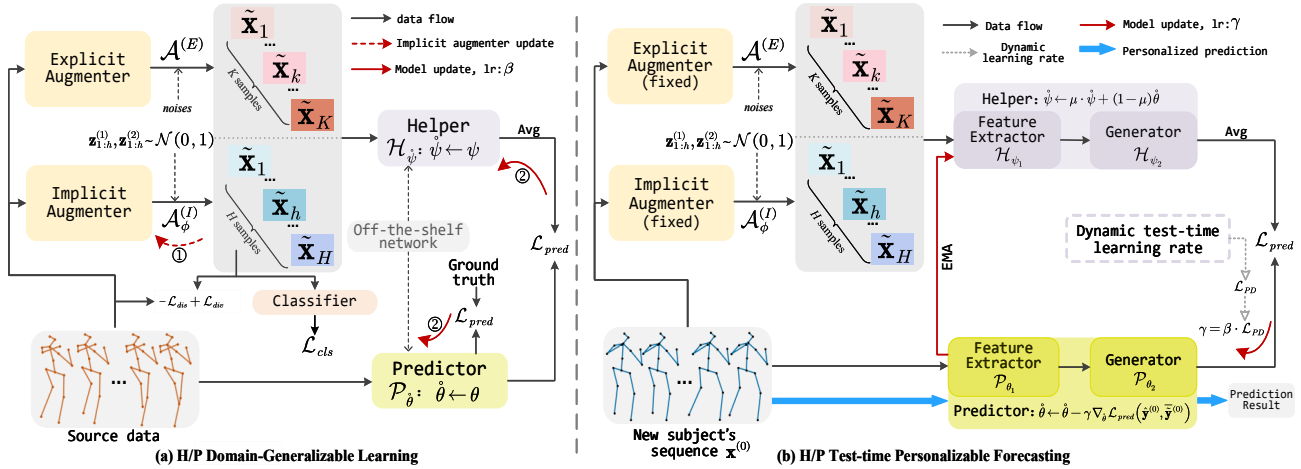

Figure 2. **Illustration of our approach.** It mainly includes a parameter-free explicit augmenter $\mathcal{A}^{(E)}$ and a learnable implicit augmenter $\mathcal{A}_\phi^{(I)}$, as well as the helper $\mathcal{H}_\psi$ and predictor $\mathcal{P}_\theta$. $\mathcal{A}^{(E)}$ is to attain the noisy samples to improve the robustness and assist $\mathcal{A}_\phi^{(I)}$ to generate the novel-domain samples. **(a) H/P domain-generalizable learning** involves two steps: The implicit augmenter is first trained with the adversarial learning paradigm, to make the diverse novel-domain samples, and reserve the semantics; $\mathcal{P}_\theta$ is trained to minimize the prediction loss $\mathcal{L}_{pred}$, while $\mathcal{H}_\psi$ is expected to achieve the consistent output with the predictor to distill the domain-invariant knowledge. We set the learning rate to β to obtain the base model $\{\mathring{\psi}, \mathring{\theta}\}$. **(b) H/P Test-time Personalizable Forecasting** is to further optimize the base helper/predictor to attain the personalized parameters for a new subject. We also analyze the demand of test-time personalization of a specific sequence, which is achieved by dynamically adjusting the test learning rate.

ering the changing application environment, [53] integrates the teacher-student framework into continue test-time adaptation (CoTTA) to distill domain-generalizable knowledge and improve the robustness of the unseen domain.

Our inspiration comes partially from the above literature, with the following important distinctions: (1) It can be integrated with any existing deep end-to-end model. (2) Due to the meaningful novel-domain augmentation, high interpretability is provided. (3) It is possible to identify the demand to which the test sample needs to be adapted.

3. Proposed Approach

Given an observed sequence $\mathbf{x}_{1:T} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T]$ of a person across a time horizon T, our goal is to predict his/her future actions $\mathbf{y}_{1:\Delta T} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_{\Delta T}]$, in which each frame records the 3D coordinates of N joints. Typically, current approaches [35, 34, 11, 10] learn a generic mapping $f : \mathbf{x}_{1:T} \rightarrow \mathbf{y}_{1:\Delta T}$. As an extension, our H/P-TTP intends to boost their generalization capability to make personalizable predictions of unseen test subjects/motions.

3.1. Novel Data Augmentation

Our H/P-TTP consists of a predictor and a helper with an identical structure, parameterized by $\mathcal{P}_\theta$ and $\mathcal{H}_\psi$, derived from any existing networks or newly-designed as well. In contrast to the predictor, the helper, outfitted with the *implicit and explicit augmenters*, w.r.t. $\mathcal{A}_\phi^{(I)}$ and $\mathcal{A}^{(E)}$, is dedicated to generating the augmented data.

Implicit Augmenter. It is introduced to attain a generalizable feature, which falls into the max-min adversarial learning scope. Intuitively, the implicit augmenter $\mathcal{A}_\phi^{(I)}$ is expected to generate diverse novel-domain augmentations with a variety of attributes (*e.g.*, ages, genders, statures) for the source sequence. After the adversarial learning is finished, the predictor allows for the distillation of domain-generalized representations from the helper, thus improving its personalization ability for unseen subject sequences.

We note that, $\mathcal{A}_\phi^{(I)}$ can be interpreted as a specially designed motion-style migration network, which is used to implicitly diversify the sequence into its younger/older or taller/lower views. It is associated with the literature on diverse human motion generation [3, 2, 12]. Therefore, we set up $\mathcal{A}_\phi^{(I)}$ with a similar network as [33], where the significant extraordinary is that it is trained with the max-min optimization and to yield novel-styled counterparts [55].

Given an observed sequence $\mathbf{x}$, the purpose of $\mathcal{A}_\phi^{(I)}$ is to produce stylized augmentations $\tilde{\mathbf{x}}^{(I)} = \mathcal{A}_\phi^{(I)}(\mathbf{x})$ yet retaining its semantic (action label). To stimulate the augmentations with sufficiently distinct appearances, we propose to maximize the discrepancy $\mathcal{L}_{dis}$ between the original observed motion and H augmented ones $\{\tilde{\mathbf{x}}_h^{(I)}\}_{h=1}^H$:

$$\max_{\mathcal{A}^{(I)}} \mathcal{L}_{dis}(\tilde{\mathbf{x}}^{(I)}, \mathbf{x}) = \sum_{h=1}^{H} \left\| \frac{\tilde{\mathbf{x}}_h^{(I)}}{\|\tilde{\mathbf{x}}_h^{(I)}\|_2} - \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \right\|_2. \quad (1)$$

To motivate diversity across the implicit augmentations, consistent with [58, 33], the diversity-promoting prior loss $\mathcal{L}_{div}$ with a normalizing factor $\delta = 100$ is adopted:

$$\mathcal{L}_{div} = \frac{2}{H(H-1)} \sum_{j=1}^{H} \sum_{h=j+1}^{H} \exp\left(-\frac{\left\| \tilde{\mathbf{x}}_j^{(I)} - \tilde{\mathbf{x}}_h^{(I)} \right\|_1}{\delta} \right). \quad (2)$$

In addition, to preserve the semantic proximity, we also introduce an action classifier to make the category label of the novel-styled sequence consistent with the observed one. Given the correct label p in the C-sized label space of a sequence and the resulting one $\tilde{p}_h^{(I)}$ of h-th implicit augmentation, it is trained with cross-entropy loss $\mathcal{L}_{cls}$:

$$\mathcal{L}_{cls} = -\frac{1}{CH}\sum_{h=1}^{H} p \log \tilde{p}_h^{(I)}. \tag{3}$$

Then, the implicit augmenter $\mathcal{A}_\phi^{(I)}$ is optimized to maximize the discrepancy, and minimize the diversity and classification error, which is expected to generate diverse augmentations as much as possible, while maintaining the semantic information with the original one [24]. After observing the new-domain augmentations, the helper can extract the domain-generalizable representation to optimize the predictor to personalize according to input samples.

Explicit Augmenter. To incorporate the additional information into the implicit augmenter, we introduce a non-trainable explicit one. In instance learning, each image is regarded as an instance, and we wish to train the network so that the representations of different augmented views of the same instance are as close as possible to each other. It provides more rich data for training, and helps learn transformation invariant representations [37, 40]. For natural images, random rotation, flipping, and noise injection are the most common augmentation strategies [14, 23, 42].

Besides, [8] develops a multi-task framework, in which the auxiliary branch is designed as motion reconstruction to provide meaningful cues for human pose prediction.

Motivated by the above-mentioned work, we, therefore, design the explicit augmenter $\mathcal{A}^{(E)}$ to produce the noisy counterparts. Formally, given a distribution sampled from the observed sequences, $\mathbf{x} \sim \boldsymbol{q}(\mathbf{x})$, it is injected with several Gaussian noises with a series of variances of a fixed interval, producing K augmentations with different noise levels:

$$\boldsymbol{q}(\tilde{\mathbf{x}}_k^{(E)}|\mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}}_k^{(E)}; \mathbf{x}, \sigma^2\mathbf{I}), \tag{4}$$

where $\{\tilde{\mathbf{x}}_k^{(E)}\}_{k=1}^K$ is the resulting noisy counterparts, which, together with the novel-domain augmentation $\{\tilde{\mathbf{x}}_h^{(I)}\}_{h=1}^H$ from the learnable $\mathcal{A}_\phi^{(I)}$, is used to train the helper.

Here, we note that, the non-parametric explicit augmenter provides the following advantages: (1) The noise counterparts are supplied, facilitating the helper to explore new patterns in the novel-domain augmented data. (2) It eliminates an overly optimistic assumption: the trainable implicit augmenter will necessarily yield novel-domain samples. (3) More robustness to noise is also included.

3.2. H/P Domain-Generalizable Learning

The objective of our helper is to improve the generalization of the predictor for unseen domain samples by observing the new-domain augmentations and noise counterparts,

Algorithm 1 H/P Domain-Generalizable Learning

Require: implicit augmenter $\mathcal{A}_\phi^{(I)}$; helper $\mathcal{H}_\psi$ and predictor $\mathcal{P}_\theta$ with the identical structure; learning rates α, β; batch size B; hyperparameters λ_{dis}, λ_{div}, λ_{cls}
Ouput: the learned parameter ψ, θ, ϕ;
1: randomly initialize ψ, θ and ϕ;
2: **while** *not converge* **do**
3: sample a training mini-batch $\{\mathbf{x}_b, \mathbf{y}_b\}_{b=1}^B$;
4: **for** *each b* **do** $\triangleright$*update implicit augmenter*
5: $\phi \leftarrow \phi - \alpha\nabla_\phi(-\lambda_{dis}\mathcal{L}_{dis} + \lambda_{div}\mathcal{L}_{div} + \lambda_{cls}\mathcal{L}_{cls})$;
6: **end for**
7: **end while**
8: **while** *not converge* **do**
9: sample a training mini-batch $\{\mathbf{x}_b, \mathbf{y}_b\}_{b=1}^B$;
$\triangleright$ *update the helper and predictor*
10: $\psi \leftarrow \psi - \sum_{b=1}^B \beta\nabla_\psi\mathcal{L}_{pred}(\tilde{\bar{\mathbf{y}}}_b^{(\mathcal{H})}, \hat{\mathbf{y}}_b^{(\mathcal{P})})$.;
11: $\theta \leftarrow \theta - \sum_{b=1}^B \beta\nabla_\theta\mathcal{L}_{pred}(\hat{\mathbf{y}}_b^{(\mathcal{P})}, \mathbf{y}_b)$
12: **end while**

as shown in Figure 2 (left). The $\mathcal{A}^{(E)}$ is parameter-free, while $\mathcal{A}_\phi^{(I)}$ is trained to solve the following problem:

$$\arg\min_{\phi\in\Phi}\max\ \lambda_{dis}\mathcal{L}_{dis} + \lambda_{div}\mathcal{L}_{div} + \lambda_{cls}\mathcal{L}_{cls}, \tag{5}$$

where $\lambda_{dis} = 0.5, \lambda_{div} = 0.3$ and $\lambda_{cls} = 0.2$ are the trade-off parameters. With a learning rate $\alpha = 0.001$, the implicit augmenter $\mathcal{A}_\phi^{(I)}$ is optimized:

$$\phi \leftarrow \phi - \alpha\nabla_\phi(-\lambda_{dis}\mathcal{L}_{dis} + \lambda_{div}\mathcal{L}_{div} + \lambda_{cls}\mathcal{L}_{cls}). \tag{6}$$

We then train both the helper $\mathcal{H}_\psi$ and predictor $\mathcal{P}_\theta$ to achieve the H/P domain-generalizable learning. The optimization of $\mathcal{H}_\psi$ is achieved using the gradient descent:

$$\psi \leftarrow \psi - \beta\nabla_\psi\mathcal{L}_{pred}(\tilde{\bar{\mathbf{y}}}^{(\mathcal{H})}, \hat{\mathbf{y}}^{(\mathcal{P})}), \tag{7}$$

where $\hat{\mathbf{y}}^{(\mathcal{P})}$ is the predictor's result, $\beta = 0.001$ is the learning rate. The average outcome $\tilde{\bar{\mathbf{y}}}^{(\mathcal{H})}$ is yielded over a total of $H + K$ augmentations $\{\tilde{\mathbf{x}}_i\}_{i=1}^{H+K}$ fed into the helper $\mathcal{H}_\psi$. Following [10, 8], the prediction loss $\mathcal{L}_{pred}$ is denoted as L_2 distance. The update of the predictor $\mathcal{P}_\theta$ can be achieved:

$$\theta \leftarrow \theta - \beta\nabla_\theta\mathcal{L}_{pred}(\hat{\mathbf{y}}^{(\mathcal{P})}, \mathbf{y}), \tag{8}$$

where $\mathbf{y}$ is the future ground-truth (gt) action.

The optimization of Eq. 7 and Eq. 8 ensures that the helper and predictor have consistent outcomes that are as close to the actual future sequence as possible. The domain-generalizable knowledge from the helper is progressively transferred to the predictor. The overall H/P domain-generalizable learning is summarized in Algorithm 1. Then, the base model, w.r.t $\{\mathring{\psi}, \mathring{\theta}\}$, is achieved.

3.3. H/P Test-time Personalizable Forecasting

During inference, for a new person, we consider further fine-tuning the base model to adapt to his/her specific properties. However, in practical applications, even for the same subject, his/her motion patterns may differ due to external

factors, such as environmental changes. This requires varying degrees of test-time personalization (TTP) to adapt.

For this purpose, we propose to adjust the learning rate of test-time personalization. Our intuition is that, for the current step, its previous samples and their correct future sequences are observed deterministically. We are able to know how the association of the helper/predictor feature leads to the difference between the predicted and true values. Therefore, for the next sample, at test-time, calculating the distinction between the features of the helper and predictor, we can also gain insight into the relation of their expected outputs; if it is strong, then the helper can guide the predictor well, and a lower learning rate should be assigned to update the predictor, otherwise, a larger one.

Memory queue. For simplicity, the base helper $\mathcal{H}_{\mathring{\psi}}$ and predictor $\mathcal{P}_{\mathring{\theta}}$ are decomposed: *feature extractor* and *generator*, w.r.t $\mathring{\psi} = \{\mathring{\psi}_1, \mathring{\psi}_2\}$ and $\mathring{\theta} = \{\mathring{\theta}_1, \mathring{\theta}_2\}$. We denote h and p as the helper and predictor. The outcome of the feature extractor and generator are denoted as $\mathbf{f}$ and $\mathbf{o}$. For a historical sample $\mathbf{x}^{(m)}$, $\mathbf{f}^{(m)}_{\{h\leftrightarrow p\}}$ and $\mathbf{o}^{(m)}_{\{h\leftrightarrow p\}}$ are the feature and outcome differences of h/p branches, calculated by a distance metric $\mathcal{D}$ (*i.e.*, euclidean distance), informally expressed $\leftrightarrow$. $\mathbf{o}^{(m)}_{\{p\leftrightarrow gt\}}$ calculates that the difference between the predictor's result and ground truth, indicating, from real history views, whether the predictor is well enough.

We build a M-elements memory queue $\mathbb{M} = \{(\mathbf{f}^{(m)}_{\{h\leftrightarrow p\}}, \mathbf{o}^{(m)}_{\{p\leftrightarrow gt\}})\}^{M}_{m=1}$, denoted as:

$$\rightarrow \underbrace{(\mathbf{f}^{(1)}_{\{h\leftrightarrow p\}}, \mathbf{o}^{(1)}_{\{p\leftrightarrow gt\}}), ..., (\mathbf{f}^{(M)}_{\{h\leftrightarrow p\}}, \mathbf{o}^{(M)}_{\{p\leftrightarrow gt\}})}_{\mathbb{M}}, \rightarrow \tag{9}$$

It records, in terms of the real history, the relation of the h/p features and the difference between the predicted results and the ground truth (GT). In other words, it measures whether the helper is able to distill domain-generalizable information to the predictor for previous samples (or whether the model has been calibrated well). We note that $\mathbb{M}$ is updated by First In First Out (FIFO).

Given a new arriving sample $\mathbf{x}^{(0)}$, we calculate the feature difference $\mathbf{f}^{(0)}_{\{h\leftrightarrow p\}}$ and the outcome difference $\mathbf{o}^{(0)}_{\{h\leftrightarrow p\}}$. To measure its demand to be personalized, $\mathbf{f}^{(0)}_{\{h\leftrightarrow p\}}$ is taken as the query, and retrieve D most similar ones $\{\mathbf{f}^{(d)}_{\{h\leftrightarrow p\}}\}^{D}_{d=1}$ from $\mathbb{M}$, with their corresponding $\{\mathbf{o}^{(d)}_{\{p\leftrightarrow gt\}}\}^{D}_{d=1}$ as the support set $\mathbb{D}$:

$$\underbrace{\mathbf{o}^{(1)}_{\{p\leftrightarrow gt\}}, ...\mathbf{o}^{(d)}_{\{p\leftrightarrow gt\}}, ...\mathbf{o}^{(D)}_{\{p\leftrightarrow gt\}}}_{\mathbb{D}} \tag{10}$$

This retrieval process is implemented by D-Nearest Neighbor algorithm.

Dynamic learning rate. We then average the values in $\mathbb{D}$ as the reference $\mathbf{r}$. The prediction discrepancy (PD) of $\mathbf{x}^{(0)}$ is derived:

$$\mathcal{L}_{PD} = \frac{1}{2}\left(\mathcal{L}_{KL}\left(\mathbf{r} \| \mathbf{o}^{(0)}_{\{h\leftrightarrow p\}}\right) + \mathcal{L}_{KL}\left(\mathbf{o}^{(0)}_{\{h\leftrightarrow p\}} \| \mathbf{r}\right)\right), \tag{11}$$

Algorithm 2 H/P Test-time Personalizable Forecasting

Require: test sample $\mathbf{x}^{(0)}$; base predictor $\mathring{\theta} = \{\mathring{\theta}_1, \mathring{\theta}_2\}$; base helper $\mathring{\psi} = \{\mathring{\psi}_1, \mathring{\psi}_2\}$; memory queue $\mathbb{M}$; learning rate β; momentum μ; number of nearest neighbors D;
Output: final prediction $\hat{\mathbf{y}}^*$;
1: augment $\mathbf{x}^{(0)}$ to $H+K$ augmentations $\{\tilde{\mathbf{x}}^{(0)}_i\}^{H+K}_{i=1}$ and make $\bar{\tilde{\mathbf{x}}}^{(0)}$ equal to their average;
2: calculate $\mathbf{f}^{(0)}_{h\leftrightarrow p} = \mathcal{D}(\mathcal{P}_{\theta_1}(\mathbf{x}^{(0)}), \mathcal{H}_{\psi_1}(\bar{\tilde{\mathbf{x}}}^{(0)}))$, $\mathbf{o}^{(0)}_{h\leftrightarrow p} = \mathcal{D}(\mathcal{P}_{\theta}(\mathbf{x}^{(0)}), \mathcal{H}_{\psi}(\bar{\tilde{\mathbf{x}}}^{(0)}))$;
3: retrieve the support set $\mathbb{D}$ from $\mathbb{M}$ using D-neighbors nearest with the query $\mathbf{f}^{(0)}_{h\leftrightarrow p}$;
4: ensemble the values in $\mathbb{D}$ as the reference $\mathbf{r}$;
5: calculate the prediction discrepancy $\mathcal{L}_{PD}$ using Eq. 11;
6: obtain the adjusted learning rate γ using Eq. 12;
7: update the predictor $\mathcal{P}_{\theta} : \theta^* \leftarrow \mathring{\theta}$, using Eq. 13;
8: update the helper $\mathcal{H}_{\psi} : \psi^* \leftarrow \mathring{\psi}$, using Eq. 14;
9: return $\hat{\mathbf{y}}^* = \mathcal{P}_{\theta^*}(\mathbf{x})$;

For the next sample in continuous flow, we have known the actual future sequence $\mathbf{y}^{(0)}$ of the previous sample $\mathbf{x}^{(0)}$.
1: $\mathbf{o}^{(0)}_{p\leftrightarrow gt} = \mathcal{D}(\mathcal{P}_{\theta}(\mathbf{x}^{(0)}), \mathbf{y}^{(0)})$; $\triangleright$*update memory queue*
2: $\mathbb{M}$.pop() && $\mathbb{M}$.push($\mathbf{f}^{(0)}_{h\leftrightarrow p}, \mathbf{o}^{(0)}_{p\leftrightarrow gt}$);

where $\mathcal{L}_{KL}$ is the KL divergence of two distributions.

If the value of $\mathcal{L}_{PD}$ is small, the predictor's outcome coincides with the real history (the helper can guide the predictor well); otherwise, it is not consistent (a major update is needed). Based on this consideration, we adjust the learning rate to make a dynamic TTP. At test time, we obtain the adjusted learning rate γ as:

$$\gamma = \beta \cdot \mathcal{L}_{PD}. \tag{12}$$

Our TTP begins with the base model, $\mathcal{P}_{\mathring{\theta}}$ and $\mathcal{H}_{\mathring{\psi}}$, obtained from the H/P novel-domain generalizable training stage. Then, we attain the predictor's personalized parameter to boost the generalization ability for the specific subject $\mathbf{x}^{(0)}$ using a single gradient descent step:

$$\mathring{\theta} \leftarrow \mathring{\theta} - \gamma \cdot \nabla_{\mathring{\theta}} \mathcal{L}_{pred}(\hat{\mathbf{y}}^{(0)}, \bar{\tilde{\mathbf{y}}}^{(0)}), \tag{13}$$

where $\hat{\mathbf{y}}^{(0)} = \mathcal{P}_{\mathring{\theta}}(\mathbf{x})$ is the immediate result, and the pseudo label $\bar{\tilde{\mathbf{y}}}^{(0)}$ is the average prediction of the base helper over $H+K$ augmentations. Our model can be regarded as a special teacher-student framework, and therefore, after the predictor's update, we use exponential moving average (EMA) [53] with $\mu = 0.95$ momentum to optimize the helper:

$$\mathring{\psi} \leftarrow \mu \cdot \mathring{\psi} + (1-\mu) \cdot \mathring{\theta}. \tag{14}$$

When the TTP is completed, we re-write the personalized parameters $\{\mathring{\theta}, \mathring{\psi}\}$ as $\{\theta^*, \psi^*\}$, and perform a forward pass to yield the final prediction $\hat{\mathbf{y}}^* = \mathcal{P}_{\theta^*}(\mathbf{x})$. Note that, after $P = 72$ test-time personalizations, the personalized parameters $\{\theta^*, \psi^*\}$ are rolled back to the base model $\{\mathring{\theta}, \mathring{\psi}\}$ to avoid catastrophic forgetting. We summarize the H/P test-time personalizable forecasting in Algorithm 2.

4. Experiments

4.1. Implementation Details

As illustrated in Figure 2, our H/P-TTP begins with an off-the-shelf network, and includes two major phases: (1) H/P novel-domain generalizable training, (2) test-time personalizable forecasting. For the former, we first train the implicit augmenter $\mathcal{A}_\phi^{(I)}$ using Eq. 6. Both implicit and explicit augmenters yield $H{=}K{=}8$ augmentations. The variances of Gaussian noise injected into the explicit augmenter are set to $\sigma^2 = [10, 15, 20, 25, 30, 35, 40, 45]$ in millimeter. Moreover, the classifier is set to a 3-layers MLP with 512 hidden units to ensure a similar action label between the augmentations and the original sample. Once $\mathcal{A}_\phi^{(I)}$ is trained, it is fixed in the following process. We exploit Eq. 7 and Eq. 8 to train the predictor $\mathcal{P}_\theta$ and helper $\mathcal{H}_\psi$. The predictor/helper are optimized using AdamW optimizer with the weight decay factor (1e$-$2) on mini-batches of size 32, to achieve the base model. The size of the memory queue $\mathbb{M}$ is set to $M = 36$, and the support set $\mathbb{D}$ is $D = 12$. Specifically, we divide the baseline networks by the penultimate layer, where the previous layers are treated as the feature extractor, and it and its following layers are the generator.

4.2. Datasets

Our H/P-TTP is evaluated on 3 benchmark datasets.

Dataset-1: H3.6M [22] is the most widely-used dataset for pose prediction, containing $\approx$ 3.6 million frames of 15 action categories performed by 7 human subjects. Consistent with [34, 6], all sequences are downsampled to 25 fps, and represented as a $N = 17$ joint skeleton.

Dataset-2: GRAB [45] is a newly-introduced benchmark, including $\approx$ 1.6 million poses of 29 actions from 10 human subjects. Compared with H3.6M, the pose sequences in GRAB are more diverse and involve interaction with the physical world, which is, therefore, more complex and challenging. We down-sample the sequences to 30 fps with 25 joints in each skeleton [13].

Dataset-3: HumanEva-I [43] consists of 3 human subjects, performing 6 pre-defined actions. All sequence is down-sampled to 30 fps, represented by a 15-joint skeleton. The results are reported in the supplementary material.

We note that, for H3.6M [22], the length of the observed and predicted sequence is 25 frames, while for both GRAB [45] and HumanEva-I [43], it is 30.

4.3. Experimental Setups

We use 2 main setups to analyze our model (1.x and 2.x as stated in Table 1). As in [34, 32, 10, 31, 13], the former is to evaluate the general prediction ability on common data splitting. The latter is newly-designed to evaluate the personalizable ability of unseen subjects, where S_x is the test subjects, and the disjointed ones S_x^- is the training.

Purpose	general predictive ability (1.x)		predictive ability for unseen subjects (2.x)		
Setup	(1.1)	(1.2)	(2.1)	(2.2)	(2.3)
Dataset	H3.6M [22]	GRAB[45]	H3.6M[22]	GRAB[45]	HumanEva-I [43]
Test	S_5	S_{10}	S_x	S_x	S_x
Training	S_5^-	S_{10}^-	S_x^-	S_x^-	S_x^-

Table 1. Experimental setups.

4.4. Baselines

We exploit 6 approaches, emerged in recent years as our baselines, including ResSup [35], LTD [34], HisRep [32], MSR [11], PGBIG [31], and SPGSN [26]. They are separately integrated into our H/P-TTP framework to activate the prediction performance for unseen targets. [35] is RNN-based, [32] is attention-based, and the rest are based on GCNs. For the general predictive ability, we use the common data splitting, as the setup-1.x in Table 1, and the setup-2.x in Table 1 is used to endow the baselines with the personalizable ability for unseen subjects. We note that our ultimate goal is to equip the H/P-TTP with each baseline to improve the prediction ability, which is therefore renamed, *e.g.*, PGBIG [31]+H/P-TTP, SPGSN [26]+H/P-TTP.

4.5. Protocols

To comprehensively investigate the performance of our H/P-TTP, the following 4 protocols are exploited.

Protocol-1: The widely-applied mean per-joint position error (MPJPE) [22, 34, 6, 31] is first used:

$$E_{\text{MPJPE}} = \frac{1}{N}\sum\nolimits_{n=1}^{N} \|\boldsymbol{p}_n - \hat{\boldsymbol{p}}_n^*\|_2, \tag{15}$$

where $\hat{\boldsymbol{p}}_n^*$ is the 3D position of n-th joint in a final predicted frame $\hat{\boldsymbol{y}}_t^*$, $\boldsymbol{p}_n$ is the gt. N is the number of joints.

Protocol-2: We observe that compared with the GT, some predicted pose coordinates have a slight global offset. To eliminate the error independent of poses, we supplement the Procrustes aligned MPJPE (P-MPJPE), which calculates the MPJPE after aligning the predicted pose to the gt pose by a rigid transformation called Procrustes Analysis (PA) [28]. For a frame t in a skeleton layout S, it is defined as:

$$E_{\text{P-MPJPE}} = \frac{1}{N}\sum\nolimits_{n=1}^{N} \|m_{\text{PA,S}}^t(n) - m_{\text{gt,S}}^t(n)\|_2, \tag{16}$$

where $m_{\text{PA,S}}^t(n)$ is the coordinate of n-th joint after alignment using PA, $m_{\text{gt,S}}^t(n)$ is the corresponding gt.

Protocol-3: Both MPJPE and P-MPJPE are hard to quantify the performance bound. Therefore, Percentage of Correct 3D Keypoint (PCK) is also introduced, which statistics the proportion of predicted joints with MPJPE errors smaller than a pre-defined threshold. Motivated by the current 3D human pose estimation [4, 18], we use the threshold of 150mm, re-written as PCK@150mm.

Protocol-4: In addition, we also use the mean per-bone length error (MPBLE) to evaluate the bone-length difference. The MPBLE is defined as:

$$E_{\text{MPBLE}} = \frac{1}{N-1}\sum\nolimits_{n=1}^{N-1} \|l_n - \hat{l}_n^*\|_1, \tag{17}$$

Method	ResSup		ResSup +H/P-TTP		LTD		LTD +H/P-TTP		HisRep		HisRep +H/P-TTP		MSR [11]		MSR +H/P-TTP		PGBIG		PGBIG +H/P-TTP		SPGSN		SPGSN +H/P-TTP	
Time (ms)	400	1000	400	1000	400	1000	400	1000	400	1000	400	1000	400	1000	400	1000	400	1000	400	1000	400	1000	400	1000
MPJPE [mm] ↓	115.2	130.5	107.4	125.6	63.5	114.3	59.6	109.8	58.3	112.1	56.8	109.4	62.9	114.2	60.8	108.6	58.5	110.3	56.3	105.1	58.3	109.6	**55.6**	**103.7**
P-MPJPE [mm] ↓	96.6	111.0	89.2	107.1	53.1	101.5	47.3	94.6	52.6	100.1	46.2	94.8	55.7	101.3	51.2	93.7	51.2	97.2	49.7	95.6	51.8	96.7	**49.4**	**90.2**
PCK@150mm [%]↑	57.5	50.3	59.2	53.7	70.4	66.0	72.5	69.7	71.2	65.5	73.7	69.4	75.5	70.1	80.3	72.0	77.3	69.6	82.4	71.5	80.1	71.2	**85.4**	**74.6**
MPBLE [mm] ↓	33.6	45.7	30.5	43.6	25.3	34.0	23.5	32.6	26.4	32.4	22.0	31.1	27.2	35.2	24.7	33.2	23.0	29.3	19.4	27.3	19.3	25.5	**16.3**	**23.8**

Table 2. **Average short-term (400ms) & long-term (1000ms) prediction performance** (on H3.6M [22]) using the common *setup-1.1*. We observe that, in general, with the H/P-TTP, the baselines achieve better results than the trivial ones. It reveals that S_5 includes the different motion properties from the other subjects, while our H/T-TTP can adapt them with the personalized parameters.

Method	MSR		MSR +H/P-TTP		PGBIG		PGBIG +H/P-TTP		SPGSN		SPGSN +H/P-TTP	
Time (ms)	500	1000	500	1000	500	1000	500	1000	500	1000	500	1000
MPJPE ↓	96.3	178.6	88.2	147.3	92.2	157.2	**84.4**	141.2	91.3	144.5	85.5	**136.4**
P-MPJPE ↓	82.3	131.3	78.8	122.8	79.8	127.5	66.2	**114.2**	77.2	129.0	**64.9**	117.3
PCK@150mm↑	75.3	65.6	78.3	67.6	75.8	66.4	79.0	69.1	77.0	67.8	**81.1**	**70.4**
MPBLE ↓	21.3	28.5	20.2	27.0	20.3	28.0	19.5	26.2	19.2	26.6	**18.0**	**24.7**

Table 3. **Average short- (500ms) & long-term (1000ms) performance** (on GRAB [45]) using the *setup-1.2*.

where l_n and $\hat{l}_n^*$ are the bone length of n-th bone in the gt and predicted pose, respectively. $N-1$ is the total number of bones in a skeleton. Smaller MPBLEs tend to show a more stable visualization of the predicted frames.

4.6. Results Analysis

General prediction ability is first evaluated under the condition of common data splitting [10, 8] with all 6 baselines, as the *setups-1*.x in Table 1. Specifically, for the H3.6M, Table 2 compares the average result of the 4 protocols over the samples of S_5 with the experimental *setup-1.1*. Moreover, under the *setup-1.2*, Table 3 reports the comparisons on the GRAB dataset [45]. From the results, we observe that, for the used 2 datasets, once equipped with the H/P-TTP, all baselines coincidentally achieve better performance. The reason is that for the common data splitting of H3.6M and GRAB, the test subject (subject-5 or subject-10) is disjointed from the training ones. Stated in a different way, the motion properties of the target subject-5 (S5) are unknown in training (out-of-source), which is not considered by the trivial baselines, thus making the prediction task more challenging. By contrast, with the domain-generalizable learning and test-time personalization, our H/P-TTP can attain the personalized parameters to adapt to the dynamic properties of the new test subject (S5).

Prediction ability for unseen subjects is then analyzed using the newly-designed data splitting. Because of data limitations and privacy concerns, the statures, ages, or genders of the target subjects S_x in the deployment environment are usually evident in the training stage, resulting in the distribution gap and out-of-source motion properties (*e.g.*, motion style, rhythm). To further evaluate the personalization ability of our H/P-TTP for unseen subjects, we design the *setup-2.1&2.2* in Table 1 for H3.6M and GRAB datasets, respectively. Due to their better performance, only both PGBIG and SPGSN are selected as the competitors. The comparison results are reported in Table 4 and Table 5, in which each subject S_x is tested separately and averaged over all its samples. From the results, we observe that, with H/P-TTP, the PGBIG [31]+H/P-TTP and SPGSN [26]+H/P-TTP are calibrated under almost all target subjects. It evidences that the motion properties of out-of-source unseen subjects are indeed adapted.

We also present the qualitative comparison of the SoTA SPGSN [26], and SPGSN [26]+H/P-TTP in Figure 3, under *camera-takepicture-1* of unseen S_7 from the GRAB dataset.

4.7. Ablation Studies

Next, we investigate the impact of several key components. The ablation experiments report the average (at 1000ms) under the *setup-2.1* in Table 1 on H3.6M dataset.

Intuitively, the **(1) test-time personalization (TTP) forecasting process** can further optimize the base model to adapt to unseen target subjects. It is confirmed in Table 6. We also evaluate the impact of the **(2) implicit and explicit augmenter** ($\mathcal{A}^{(E)}$ and $\mathcal{A}_{\phi}^{(I)}$), and **(3) dynamic learning rate**. As shown in Table 6, both $\mathcal{A}^{(E)}$ and $\mathcal{A}_{\phi}^{(I)}$ provide the positive effect, while $\mathcal{A}_{\phi}^{(I)}$ is greater. Moreover, TTP is indeed helpful, and especially with the dynamic learning rate, the performance is further improved.

	$\mathcal{A}^{(E)}$	$\mathcal{A}_{\psi}^{(I)}$	TTP	dynamic learning rate	MPJPE [mm] ↓
SPGSN [26]	✓		✓	✓	109.5
		✓	✓	✓	104.2
	✓	✓			114.3
	✓	✓	✓		107.4
	✓	✓	✓	✓	102.8

Table 6. Impact of both augmenters ($\mathcal{A}^{(E)}$ and $\mathcal{A}^{(I)}$), TTP and dynamic learning rate on the SPGSN [26] performance.

(4) Impact of the initial learning rate β is also investigated in Table 7(left), which is essential to H/P domain-generalizable learning and TTP. We see that β is insensitive to the performance. We suggest that it is because it can be calibrated by the dynamic learning rate at test time.

(5) Impact of the number of augmentations of the implicit and explicit augmenter is shown in Table 7(left), and $H = K = 8$ achieves the better prediction.

(6) Number ($u = [0, 1, 2, 3, 4]$) **of gradient descent of TTP** is investigated in Table 7(right). We observe that, with

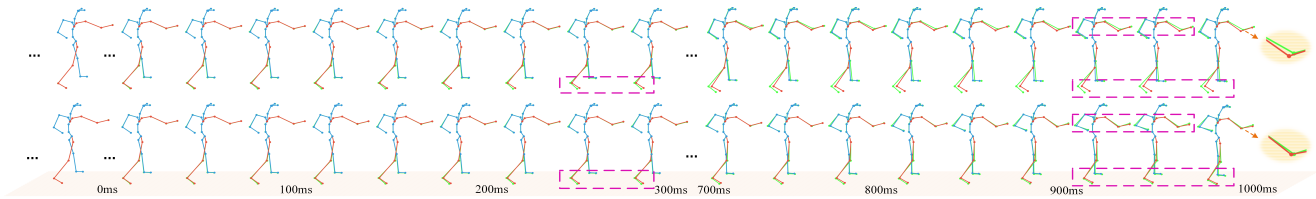

Figure 3. **Predicted pose visualization**, where the top is vanilla SPGSN [26] and the bottom is SPGSN [26]+H/P-TTP. The underlying green denotes the gt, and the red/blue is the prediction. The contrasting segments are highlighted in the purple box. We also enlarge some important details in the yellow ellipse. It is clear that with our H/P-TTP, the predicted poses are more accurate.

	MPJPE [mm] ↓				PMPJPE [mm] ↓				PCK@150mm [%] ↑				MPBLE [mm] ↓			
unseen subjects	PGBIG	PGBIG +H/P-TTP	SPGSN	SPGSN +H/P-TTP	PGBIG	PGBIG +H/P-TTP	SPGSN	SPGSN +H/P-TTP	PGBIG	PGBIG +H/P-TTP	SPGSN	SPGSN +H/P-TTP	PGBIG	PGBIG +H/P-TTP	SPGSN	SPGSN +H/P-TTP
S_1	116.6	**109.3**	111.4	109.6	97.7	94.5	95.4	**91.9**	67.2	69.3	67.9	**72.4**	29.6	25.8	26.0	**25.2**
S_6	103.6	98.2	98.6	**95.2**	84.9	80.9	81.8	**78.5**	61.8	63.9	64.5	**67.0**	26.2	22.9	23.1	**22.3**
S_7	107.0	**102.1**	110.4	103.8	88.9	**84.5**	92.9	86.3	64.2	66.3	64.9	**69.4**	27.4	23.9	24.1	**23.3**
S_8	96.5	**92.1**	100.3	95.4	78.8	77.8	83.4	**74.2**	67.8	71.9	68.5	**73.0**	24.0	20.9	21.1	**20.3**
S_9	105.3	101.2	104.5	**100.1**	87.0	**83.5**	90.3	86.8	63.2	65.3	63.9	**68.4**	26.6	23.3	23.5	**22.7**
S_{11}	120.8	115.5	117.0	**112.8**	101.5	97.5	99.0	**95.5**	70.2	72.3	70.9	**75.4**	30.0	26.7	26.9	**26.1**
Average	108.3	103.1	107.0	**102.8**	89.8	86.5	90.5	**85.5**	65.7	68.2	66.4	**70.9**	27.3	23.9	24.1	**23.3**

Table 4. **Average performance comparison** (of both SoTA PGBIG [31] and SPGSN [26], on H3.6M dataset [22]) of the end predicted pose (1000ms) over samples of each unseen subject S_x with $x = [1, 6, 7, 8, 9, 11]$, and the corresponding average over all unseen subjects.

	MPJPE [mm] ↓				PMPJPE [mm] ↓				PCK@150mm [%] ↑				MPBLE [mm] ↓			
unseen subjects	PGBIG	PGBIG +H/P-TTP	SPGSN	SPGSN +H/P-TTP	PGBIG	PGBIG +H/P-TTP	SPGSN	SPGSN +H/P-TTP	PGBIG	PGBIG +H/P-TTP	SPGSN	SPGSN +H/P-TTP	PGBIG	PGBIG +H/P-TTP	SPGSN	SPGSN +H/P-TTP
S_1	177.5	152.0	176.3	**151.5**	146.8	**136.3**	149.6	137.0	63.8	65.3	64.1	**68.6**	28.4	25.3	27.1	**25.0**
S_2	182.5	164.3	179.3	**156.0**	154.3	150.4	150.8	**146.4**	58.0	60.6	60.4	**63.7**	31.5	28.6	29.1	**26.6**
S_3	162.3	**143.2**	157.2	143.6	144.1	**134.2**	140.7	136.7	64.2	66.0	66.3	**70.4**	26.8	25.2	25.7	**23.8**
S_4	167.0	149.1	160.3	**145.7**	148.5	133.9	138.7	**128.0**	63.7	**69.3**	67.2	68.9	27.7	25.4	26.0	**24.1**
S_5	142.1	**126.0**	142.8	127.5	121.4	109.5	125.6	**104.9**	65.0	67.3	70.3	**72.7**	22.2	**20.3**	22.7	20.4
S_6	167.2	150.4	163.7	**144.2**	147.7	139.3	139.1	**126.8**	62.0	**68.3**	65.3	67.6	27.8	**24.3**	26.2	24.9
S_7	140.4	122.8	138.7	**116.5**	123.4	111.2	115.2	**107.0**	64.8	67.2	68.3	**71.0**	22.4	20.5	20.9	**18.6**
S_8	133.9	112.7	130.4	**111.9**	118.5	**106.5**	110.7	106.2	66.9	70.2	70.1	**73.2**	20.8	18.8	19.2	**17.2**
S_9	153.6	**120.0**	154.6	121.3	135.7	107.1	134.8	**105.4**	62.2	63.0	66.3	**70.2**	25.1	**22.8**	25.2	23.1
Average	158.6	137.7	155.9	**135.5**	137.8	125.4	133.9	**124.1**	63.4	66.4	66.5	**69.6**	25.9	23.5	24.7	**22.6**

Table 5. **Average performance comparison** (of both SoTA PGBIG [31] and SPGSN [26], on GRAB dataset [45]) of the end predicted pose (1000ms) over samples of each unseen subject S_x with $x = [1, 2, 3, 4, 5, 6, 7, 8, 9]$, and the average over all unseen subjects.

	β	H	K	MPJPE [mm] ↓
SPGSN [26] +H/P-TTP	0.0005	**8**	**8**	103.4
	0.001	6	6	109.2
		8	**8**	102.8
		12	12	107.4
		16	16	105.5
	0.0015	**8**	**8**	103.1

	u	MPJPE [mm] ↓
SPGSN [26] +H/P-TTP	**1**	102.8
	2	103.4
	3	104.8

Table 7. Impact of the initial learning rate β, augmentation numbers (H and K) of both $\mathcal{A}_{\psi}^{(I)}$ and $\mathcal{A}^{(E)}$, and the update numbers u of the TTP process on MPJPE of the SPGSN [26]+H/P-TTP.

	σ^2 [mm]	MPJPE [mm] ↓
SPGSN [26] +H/P-TTP	[5,10,15,20,25,30,35,40]	103.1
	[10,15,20,25,30,35,40,45]	102.8
	[15,20,25,30,35,40,45,50]	103.5

M	D	MPJPE [mm] ↓
18	6	112.4
	9	109.5
36	9	104.5
	12	102.8
48	12	107.4
	18	106.8

Table 8. Impact of the variance levels of noise injected into $\mathcal{A}^{(E)}$, the size of the memory queue and support set.

the dynamic learning rate, our TTP only needs to perform a single update. More updates bring no further benefits.

(7) Impact of the variances of Gaussian noise injection to the explicit augmenter is shown in Table 8(left), where the middle one attains the best performance.

(8) Impact of the sizes of $\mathbb{M}$ and $\mathbb{D}$. As reported in Table 8(right), the better result is balanced on the condition of $M = 36$ and $D = 12$. Too small leads to catastrophic forgetting, while too large increases the memory burden.

5. Conclusion

In this work, we propose a novel helper-predictor test-time personalization (H/P-TTP) framework for 3D human pose forecasting with unseen target subjects. It can be easily integrated into existing deep end-to-end approaches to enable the personalizability of specific target test samples. With the dynamic test-time learning rate, the H/P-TTP further considers the degree to which a specific person's motion pattern needs to be personalized. On several benchmarks, experiments demonstrate that the proposed H/P-TTP significantly boosts their prediction performance. Therefore, we reasonably conclude that our model is more practical for unseen test subjects in real-world applications.

References

[1] Peshal Agarwal, Danda Pani Paudel, Jan-Nico Zaech, and Luc Van Gool. Unsupervised Robust Domain Adaptation without Source Data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2009–2018, 2022. 2

[2] Sadegh Aliakbarian, Fatemeh Saleh, Lars Petersson, Stephen Gould, and Mathieu Salzmann. Contextually plausible and diverse 3d human motion prediction. In *ICCV*, pages 11333–11342, 2021. 1, 3

[3] Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Lars Petersson, and Stephen Gould. A Stochastic Conditioning Scheme for Diverse Human Motion Prediction. In *CVPR*, pages 5223–5232, 2020. 1, 2, 3

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d Human Pose Estimation: New Benchmark and State of The Art Analysis. In *CVPR*, pages 3686–3693, 2014. 6

[5] Emad Barsoum, John Kender, and Zicheng Liu. HP-GAN: Probabilistic 3D Human Motion Prediction via GAN. In *CVPR*, pages 1418–1427, 2018. 1, 2

[6] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, et al. Learning Progressive Joint Propagation for Human Motion Prediction. In *ECCV*, pages 226–242. Springer, 2020. 6

[7] Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. Test-Time Fast Adaptation for Dynamic Scene Deblurring via Meta-Auxiliary Learning. In *CVPR*, pages 9137–9146, 2021. 2

[8] Qiongjie Cui and Huaijiang Sun. Towards Accurate 3D Human Motion Prediction From Incomplete Observations. In *CVPR*, pages 4801–4810, June 2021. 2, 4, 7

[9] Qiongjie Cui, Huaijiang Sun, Yupeng Li, and Yue Kong. Efficient human motion recovery using bidirectional attention network. *Neural Computing and Applications*, 32:10127–10142, 2020. 1

[10] Qiongjie Cui, Huaijiang Sun, and Fei Yang. Learning Dynamic Relationships for 3D Human Motion Prediction. In *CVPR*, pages 6519–6527, 2020. 2, 3, 4, 6, 7

[11] 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. 1, 2, 3, 6, 7

[12] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Diverse Human Motion Prediction via Gumbel-Softmax Sampling from an Auxiliary Space. In *ACM MM*, pages 5162–5171, 2022. 3

[13] Christian Diller, Thomas A. Funkhouser, and Angela Dai. Forecasting Characteristic 3D Poses of Human Actions. *CVPR*, pages 15893–15902, 2022. 6

[14] Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised Representation Learning by Rotation Feature Decoupling. In *CVPR*, pages 10364–10374, 2019. 4

[15] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent Network Models for Human Dynamics. In *ICCV*, pages 4346–4354, 2015. 2

[16] Anand Gopalakrishnan, Ankur Mali, Dan Kifer, C. Lee Giles, and Alexander Ororbia. A Neural Temporal Model for Human Motion Prediction. In *CVPR*, 2019. 1

[17] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José M. F. Moura. Adversarial Geometry-Aware Human Motion Prediction. In *ECCV*, pages 786–803, 2018. 2

[18] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt. In The Wild Human Pose Estimation using Explicit 2D Features and Intermediate 3D Representations. In *CVPR*, pages 10905–10914, 2019. 6

[19] Miao Hao, Yizhuo Li, Zonglin Di, Nitesh B Gundavarapu, and Xiaolong Wang. Test-Time Personalization with a Transformer for Human Pose Estimation. In *NeurIPS*, 2021. 1

[20] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The Many Faces of Robustness: A Critical Analysis of Out-of-distribution Generalization. In *ICCV*, pages 8340–8349, 2021. 2

[21] Minhao Hu, Tao Song, Yujun Gu, Xiangde Luo, Jieneng Chen, Yinan Chen, Ya Zhang, and Shaoting Zhang. Fully Test-Time Adaptation for Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 251–260. Springer, 2021. 1

[22] 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)*, 36:1325–1339, 2014. 2, 6, 7, 8

[23] Nikos Komodakis and Spyros Gidaris. Unsupervised Representation Learning by Predicting Image Rotations. In *ICLR*, 2018. 4

[24] Bin Li, Jian Tian, Zhongfei Zhang, Hailin Feng, and Xi Li. Multitask Non-Autoregressive Model for Human Motion Prediction. *IEEE Transactions on Image Processing*, 2020. 1, 2, 4

[25] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional Sequence to Sequence Model for Human Dynamics. In *CVPR*, pages 5226–5234, 2018. 2

[26] Maosen Li, Siheng Chen, Zijing Zhang, Lingxi Xie, Qi Tian, and Ya Zhang. Skeleton-Parted Graph Scattering Networks for 3D Human Motion Prediction. *arXiv preprint arXiv:2208.00368*, 2022. 1, 2, 6, 7, 8

[27] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic Multiscale Graph Neural Networks for 3D Skeleton Based Human Motion Prediction. In *CVPR*, pages 214–223, 2020. 2

[28] Yang Li, Kan Li, Shuai Jiang, Ziyue Zhang, Congzhentao Huang, and Richard Yi Da Xu. Geometry-driven self-supervised method for 3d human pose estimation. In *AAAI*, volume 34, pages 11442–11449, 2020. 6

[29] Mengyuan Liu, Fanyang Meng, and Yongsheng Liang. Generalized pose decoupled network for unsupervised 3d skeleton sequence-based action representation learning. *Cyborg and Bionic Systems*, 2022:0002, 2022. 1

[30] Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. TTT++:

When Does Self-supervised Test-time Training Fail or Thrive? *NeurIPS*, 34:21808–21820, 2021. 2

[31] 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, 7, 8

[32] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History Repeats Itself: Human Motion Prediction via Motion Attention. In *ECCV*, pages 474–489, 2020. 1, 2, 6

[33] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. Generating Smooth Pose Sequences for Diverse Human Motion Prediction. In *ICCV*, 2021. 1, 2, 3

[34] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning Trajectory Dependencies for Human Motion Prediction. In *ICCV*, pages 9489–9497, 2019. 2, 3, 6

[35] Julieta Martinez, Michael J Black, and Javier Romero. On Human Motion Prediction using Recurrent Neural Networks. In *CVPR*, pages 2891–2900, 2017. 2, 3, 6

[36] Angel Martínez-González, Michael Villamizar, and Jean-Marc Odobez. Pose Transformers (POTR): Human Motion Prediction with Non-Autoregressive Transformers. In *ICCV*, pages 2276–2284, 2021. 2

[37] Ishan Misra and Laurens van der Maaten. Self-supervised Learning of Pretext-invariant Representations. In *CVPR*, pages 6707–6717, 2020. 4

[38] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating Prediction-time Batch Normalization for Robustness under Covariate Shift. *arXiv preprint arXiv:2006.10963*, 2020. 2

[39] Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. Quaternet: A Quaternion-based Recurrent Model for Human Motion. *International Journal of Computer Vision*, (128):855–872, 2020. 2

[40] Senthil Purushwalkam and Abhinav Gupta. Demystifying Contrastive Self-supervised Learning: Invariances, Augmentations and Dataset Biases. *NeurIPS*, 33:3407–3418, 2020. 4

[41] Alejandro Hernandez Ruiz, Juergen Gall, and Francesc Moreno-Noguer. Human Motion Prediction via Spatio-Temporal Inpainting. In *CVPR*, pages 7134–7143, 2018. 2

[42] Chajin Shin, Taeoh Kim, Sangjin Lee, and Sangyoun Leey. Test-Time Adaptation for Out-Of-Distributed Image Inpainting. In *ICIP*, 2021. 1, 4

[43] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1):4–27, 2010. 6

[44] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time Training with Self-supervision for Generalization under Distribution Shifts. In *ICML*, pages 9229–9248. PMLR, 2020. 2

[45] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A Dataset of Whole-Body Human Grasping of Objects. In *ECCV*, 2020. 6, 7, 8

[46] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to The Real World. In *IROS*, pages 23–30, 2017. 2

[47] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *CVPRW*, pages 969–977, 2018. 2

[48] Thomas Varsavsky, Mauricio Orbes-Arteaga, Carole H Sudre, Mark S Graham, Parashkev Nachev, and M Jorge Cardoso. Test-time Unsupervised Domain Adaptation. In *MICCAI*, pages 428–436. Springer, 2020. 1

[49] Borui Wang, Ehsan Adeli, Hsu-kuang Chiu, De-An Huang, and Juan Carlos Niebles. Imitation Learning for Human Pose Prediction. In *ICCV*, pages 7124–7133, 2019. 1

[50] Dequan Wang, Shaoteng Liu, Sayna Ebrahimi, Evan Shelhamer, and Trevor Darrell. On-target adaptation. *arXiv preprint arXiv:2109.01087*, 2021. 1

[51] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully Test-time Adaptation By Entropy Minimization. *arXiv preprint arXiv:2006.10726*, 2020. 2

[52] Qin Wang, Dengxin Dai, Lukas Hoyer, Luc Van Gool, and Olga Fink. Domain adaptive semantic segmentation with self-supervised depth estimation. In *ICCV*, pages 8515–8525, 2021. 1

[53] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual Test-time Domain Adaptation. In *CVPR*, pages 7201–7211, 2022. 3, 5

[54] Sirui Xu, Y-X Wang, and L-Y Gui. Diverse Human Motion Prediction Guided by Multi-level Spatial-temporal Anchors. In *ECCV*, 2022. 2

[55] Fu-En Yang, Yuan-Chia Cheng, Zu-Yun Shiau, and Yu-Chiang Frank Wang. Adversarial Teacher-student Representation Learning for Domain Generalization. *NeurIPS*, 34:19448–19460, 2021. 2, 3

[56] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A Fourier Perspective on Model Robustness in Computer Vision. *NeurIPS*, 32, 2019. 2

[57] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019. 2

[58] Ye Yuan and Kris Kitani. Dlow: Diversifying Latent Flows for Diverse Human Motion Prediction. In *ECCV*, pages 346–364, 2020. 1, 3

[59] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test Time Robustness via Adaptation and Augmentation. *arXiv preprint arXiv:2110.09506*, 2021. 2

[60] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep Domain-adversarial Image Generation for Domain Generalisation. In *AAAI*, volume 34, pages 13025–13032, 2020. 2