Few-Shot Human Motion Prediction via Meta-Learning

Liang-Yan Gui, Yu-Xiong Wang, Deva Ramanan, and José M. F. Moura

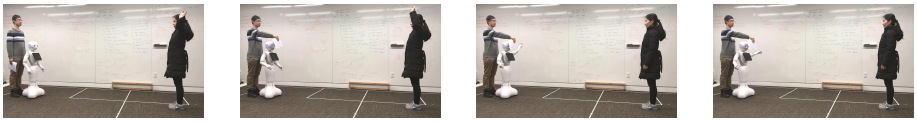
Carnegie Mellon University
{lgui,yuxiongw,deva,moura}@andrew.cmu.edu

Abstract. Human motion prediction, forecasting human motion in a few milliseconds conditioning on a historical 3D skeleton sequence, is a long-standing problem in computer vision and robotic vision. Existing forecasting algorithms rely on extensive annotated motion capture data and are brittle to novel actions. This paper addresses the problem of *few-shot* human motion prediction, in the spirit of the recent progress on few-shot learning and meta-learning. More precisely, our approach is based on the insight that having a good generalization from few examples relies on both a generic initial model and an effective strategy for adapting this model to novel tasks. To accomplish this, we propose *proactive and adaptive meta-learning (PAML)* that introduces a novel combination of model-agnostic meta-learning and model regression networks and unifies them into an *integrated, end-to-end* framework. By doing so, our meta-learner produces a generic initial model through aggregating contextual information from a variety of prediction tasks, while effectively adapting this model for use as a task-specific one by leveraging *learning-to-learn* knowledge about how to transform few-shot model parameters to many-shot model parameters. The resulting PAML predictor model significantly improves the prediction performance on the heavily benchmarked H3.6M dataset in the small-sample size regime.

Keywords: Human motion prediction · Few-shot learning · Meta-learning

1 Introduction

One of the hallmarks of human intelligence is the ability to predict the future based on past observations. Through perceiving and forecasting how the environment evolves and how a fellow human acts, a human learns to interact with the world [60]. Remarkably, humans acquire such a prediction ability from just a few experiences, which is yet generalizable across different scenarios [50]. Similarly, to allow natural and effective interaction with humans, artificial agents (*e.g.*, robots) should be able to do the same, *i.e.*, forecasting how a human moves or acts in the near future conditioning on a series of historical movements [29]. As a more concrete example illustrated in Figure 1, when deployed in natural environments, robots are supposed to predict unfamiliar actions after seeing only a few examples [27, 20]. While human motion prediction has attracted increasing



(a) (b) (c) (d)

Fig. 1. Illustration of the importance of few-shot human motion prediction as a first step towards seamless human-robot interaction and collaboration. In real-world scenarios, the prediction typically happens *in an on-line, streaming manner* with limited training data. Specifically, a robot has acquired a general-purpose prediction ability, *e.g.*, through learning on several known action classes using our meta-learning approach. The robot is then deployed in a natural environment. Now a person performs certain *never-before-seen* action, *e.g.*, greeting, while the robot is watching (Fig. (a)). The person then stops, and the robot has no sensory inputs, which is illustrated by blinding its eyes with a sheet of paper (Fig. (b)). The robot adapts the generic initial model for use as a task-specific predictor model, predicts the future motion of the person, and performs or demonstrates it in a human-like, realistic way (Figs. (c) and (d)).

attention [16, 26, 32, 9, 19], the existing approaches rely on extensive annotated motion capture (mocap) data and are brittle to novel actions.

We believe that the significant gap between human and machine prediction arises from two issues. First, motion dynamics are difficult to model because they entangle physical constraints with goal-directed behaviors [32]. Beyond some action classes (*e.g.*, walking) [8, 22], it is challenging to generate sophisticated physical models for general types of motion [42]. Second, there exists a lack of large-scale, annotated motion data. Current mocap datasets are constructed with dedicated sensored environments and so are *not scalable*. This motivates the exploration of motion models learned from limited training data. Unfortunately, a substantial amount of annotated data is required for the state-of-the-art deep recurrent encoder-decoder network based models [16, 26, 32, 18, 4, 19] to learn the desired motion dynamics. One stark evidence of this is that a constant pose predictor [32], as a naïve approach that does not produce interesting motion, sometimes achieves the best performance. An attractive solution is learning a "basis" of underlying knowledge that is shared across a wide variety of action classes, including never-before-seen actions. This can be in principle achieved by transfer learning [38, 3, 44, 68] in a way that fine-tunes a pre-trained network from another task which has more labeled data; nevertheless, the benefit of pre-training decreases as the source task diverges from the target task [70].

Here we make the first attempt towards *few-shot human motion prediction*. Inspired by the recent progress on few-shot learning and meta-learning [58, 47, 61, 66, 14], we propose a general meta-learning framework — *proactive and adaptive meta-learning (PAML)*, which can be applied to human motion prediction. *Our key insight* is that having a good generalization from few examples relies on both a generic initial model and an effective strategy for adapting this model to novel tasks. We then introduce a novel combination of the state-of-the-art model-agnostic meta-learning (MAML) [14] and model regression networks (MRN) [66,

69], and unify them into an *integrated, end-to-end* framework. MAML enables the meta-learner to aggregate contextual information from various prediction tasks and thus produces a generic model initialization, while MRN allows the meta-learner to adapt a few-shot model and thus improves its generalization.

More concretely, a beneficial common initialization would serve as a good point to start training for a novel action being considered. This can be accomplished by explicitly learning the initial parameters of a predictor model in a way that the model has maximal performance on a new task after the parameters have been updated with a few training examples from that new task. Hence, we make use of MAML [14], which initializes the weights of a network such that standard stochastic gradient descent (SGD) can make rapid progress on a new task. We learn this initialization through a meta-learning procedure that learns from a large set of motion prediction tasks with small amounts of data. After obtaining the pre-trained model, MAML uses one or few SGD updates to adapt it to a novel task. Although the initial model is somewhat generic, plain SGD updates can only slightly modify its parameters [68] especially in the small-sample size regime; otherwise, it would lead to severe over-fitting to the new data [23]. This is still far from satisfactory, because the obtained task-specific model is different from the one that would be learned from a large set of samples.

To address this limitation, we consider meta-learning approaches that learn an update function or learning rule. Specifically, we leverage MRN [66, 69] as the adaptation strategy, which describes a method for learning from small datasets through estimating a generic model transformation. That is, MRN learns a meta-level network that operates on the space of model parameters, which is trained to regress many-shot model parameters (trained on large datasets) from few-shot model parameters (trained on small datasets). While MRN was developed in the context of convolutional neural networks, we extend it to recurrent neural networks. By unifying MAML with MRN, our resulting PAML model is not only directly initialized to produce the desired parameters that are useful for later adaptation, but it can also be effectively adapted to novel actions through exploiting the structure of model parameters shared across action classes.

Our contributions are three-fold. (1) To the best of our knowledge, this is the first time the few-shot learning problem for human motion prediction has been explored. We show how meta-learning can be operationalized for such a task. (2) We present a novel meta-learning approach, combining MAML with MRN, that jointly learns a generic model initialization and an effective model adaptation strategy. Our approach is general and can be applied to different tasks. (3) We show how our approach significantly facilitates the prediction of novel actions from few examples on the challenging mocap H3.6M dataset [25].

2 Related Work

Human motion prediction has great application potential in computer vision and robotic vision, including human-robot interaction and collaboration [29], motion generation for computer graphics [30], action anticipation [28, 24], and proactive

decision-making in autonomous driving systems [37]. It is typically addressed by state-space equations and latent-variable models. Traditional approaches focus on hidden Markov models [7], linear dynamic models [41], Gaussian process latent variable models [62, 59], bilinear spatio-temporal basis models [1], and restricted Boltzmann machines [54, 53, 52, 55]. In the deep learning era, recurrent neural networks (RNNs) based approaches have attracted more attention and significantly pushed the state of the art [16, 26, 18, 32, 19].

Flagship techniques include LSTM-3LR and ERD [16], SRNNs [26], and residual sup. [32]. LSTM-3LR (3-layer long short-term memory network) learns pose representation and temporal dynamics simultaneously via curriculum learning [16]. In additional to the concatenated LSTM units as in LSTM-3LR, ERD (encoder-recurrent-decoder) further introduces non-linear space encoders for data pre-processing [16]. SRNNs (structural RNNs) model human activity with a hand-designed spatio-temporal graph and introduce the encoded semantic knowledge into recurrent networks [26]. These approaches fail to consider the shared knowledge across action classes and they thus learn action-specific models and restrict the training process on the corresponding subsets of the mocap dataset. Residual sup. is a simple sequence-to-sequence architecture with a residual connection, which incorporates the action class information via one-hot vectors [32]. Despite their promise, these existing methods directly learn on the target task with large amounts of training data and cannot generalize well from a few examples or to novel action classes. There has been little work on few-shot motion prediction as ours, which is crucial for robot learning in practice. Our task is also significantly different from few-shot imitation learning: while this line of work aims to learn and mimic human motion from demonstration [39, 15, 11, 71], our goal is to *predict unseen future* motion based on historical observations.

Few-shot or low-shot learning has long stood as one of the unsolved fundamental problems and been addressed from different perspectives [56, 13, 63, 45, 64, 66, 21, 65, 61, 14, 17, 67]. Our approach falls more into a classic yet recently renovated class of approaches, termed as meta-learning that frames few-shot learning itself as a "*learning-to-learn*" problem [57, 58, 47]. The idea is to use the common knowledge captured among a set of few-shot learning tasks during meta-training for a novel few-shot learning problem, in a way that (1) accumulates statistics over the training set using RNNs [61], memory-augmented networks [45], or multilayer perceptrons [12], (2) produces a generic network initialization [14, 36, 65], (3) embeds examples into a universal feature space [51], (4) estimates the model parameters that would be learned from a large dataset using a few novel class examples [6] or from a small dataset model [66, 69], (5) modifies the weights of one network using another [48, 46, 49], and (6) learns to optimize through a learned update rule instead of hand-designed SGD [2, 31, 43].

Often, these prior approaches are developed with image classification in mind, and cannot be easily re-purposed to handle different model architectures or readily applicable to other domains such as human motion prediction. Moreover, they aim to either obtain a better model initialization [14, 36, 65] or learn an update function or learning rule [48, 5, 2, 43, 66], *but not both*. By contrast, we present a

unified view by taking these two aspects into consideration and show how they complement each other in an end-to-end meta-learning framework. Our approach is also general and can be applied to other tasks as well.

3 Proactive and Adaptive Meta-Learning

We now present our meta-learning framework for few-shot human motion prediction. The predictor (*i.e.*, learner) is a recurrent encoder-decoder network, which frames motion prediction as a sequence-to-sequence problem. To enable the predictor to rapidly produce satisfactory prediction from just a few training sequences for a novel task (*i.e.*, action class), we introduce proactive and adaptive meta-learning (PAML). Through meta-learning from a large collection of few-shot prediction tasks on known action classes, PAML jointly learns a generic model initialization and an effective model adaptation strategy.

3.1 Meta-Learning Setup for Human Motion Prediction

Human motion is typically represented as sequential data. Given a historical motion sequence, we predict possible motion in the short-term or long-term future. In *few-shot motion prediction*, we aim to train a predictor model that can quickly adapt to a new task using only a few training sequences. To achieve this, we introduce a *meta-learning mechanism* that treats entire prediction tasks as training examples. During meta-learning, the predictor is trained on a set of prediction tasks guided by a *high-level meta-learner*, such that the trained predictor can accomplish the desired few-shot adaptation ability.

The predictor (*i.e.*, learner), represented by a parametrized function $\mathcal{P}_\theta$ with parameters θ, maps an input historical sequence $\mathbf{X}$ to an output future sequence $\widehat{\mathbf{Y}}$. We denote the input motion sequence of length n as $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n\}$, where $\mathbf{x}^i \in \mathbb{R}^d, i = 1, \ldots, n$ is a mocap vector consisting of a set of 3D body joint angles [35], and d is the number of joint angles. The learner predicts the future sequence $\widehat{\mathbf{Y}} = \{\widehat{\mathbf{x}}^{n+1}, \widehat{\mathbf{x}}^{n+2}, \ldots, \widehat{\mathbf{x}}^{n+m}\}$ in the next m timesteps, where $\widehat{\mathbf{x}}^j \in \mathbb{R}^d, j = n+1, \ldots, n+m$ is the predicted mocap vector at the j-th timestep. The groundtruth of the future sequence is denoted as $\mathbf{Y}^{gt} = \{\mathbf{x}^{n+1}, \mathbf{x}^{n+2}, \ldots, \mathbf{x}^{n+m}\}$.

During meta-learning, we are interested in training a *learning procedure* (*i.e.*, the meta-learner) that enables the predictor model to adapt to a large number of prediction tasks. For the k-shot prediction task, each task $\mathcal{T} = \{\mathcal{L}, \mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}\}$ aims to predict a certain action from a few (k) examples. It consists of a loss function $\mathcal{L}$, a small training set $\mathcal{D}_{\text{train}} = \left\{\left(\mathbf{X}_u, \mathbf{Y}_u^{gt}\right)\right\}, u = 1, \ldots, k$ with k action-specific past and future sequence pairs, and a test set $\mathcal{D}_{\text{test}}$ that has a set number of past and future sequence pairs for evaluation. A frame-wise Euclidean distance is commonly used as the loss function $\mathcal{L}$ for motion prediction. For each task, the meta-learner takes $\mathcal{D}_{\text{train}}$ as input and produces a predictor (*i.e.*, learner) that achieves high average prediction performance on its corresponding $\mathcal{D}_{\text{test}}$.

More precisely, we consider a distribution $p(\mathcal{T})$ over prediction tasks that we want our predictor to be able to adapt to. Meta-learning algorithms have two

phases: meta-training and meta-test. During meta-training, a prediction task $\mathcal{T}_i$ is sampled from $p(\mathcal{T})$, and the predictor $\mathcal{P}$ is trained on its corresponding small training set $\mathcal{D}_{\text{train}}$ with the loss $\mathcal{L}_{\mathcal{T}_i}$ from $\mathcal{T}_i$. The predictor is then improved by considering how the test error on the corresponding test set $\mathcal{D}_{\text{test}}$ changes with respect to the parameters. This test error serves as the *training error* of the meta-learning process. During meta-test, a held-out set of prediction tasks drawn from $p(\mathcal{T})$ (*i.e.*, novel action classes), each with its own small training set $\mathcal{D}_{\text{train}}$ and test set $\mathcal{D}_{\text{test}}$, is used to evaluate the performance of the predictor.

3.2 Learner: Encoder-Decoder Architecture

We use the state-of-the-art recurrent encoder-decoder network based motion predictor in [32] as our learner $\mathcal{P}$. The encoder and decoder consist of GRU (gated recurrent unit) [10] cells as building blocks. The input sequence is passed through the encoder to infer a latent representation. This latent representation and a seed motion frame are then fed into the decoder to output the first timestep prediction. The decoder takes its own output as the next timestep input and generates further prediction sequentially. Different from [32], to deal with novel action classes, we do not use one-hot vectors to indicate the action class.

3.3 Proactive Meta-Learner: Generic Model Initialization

Intuitively, if we have a *universal predictor* that is broadly applicable to a variety of tasks in $p(\mathcal{T})$ instead of a specific task, it would serve as a good point to start training for a novel target task. We explicitly learn such a general-purpose initial model by using model-agnostic meta-learning (MAML) [14]. MAML is developed for gradient-based learning rules (*e.g.*, SGD) and aims to learn a model in a way that a few SGD updates can make rapid progress on a new task.

Concretely, when adapting to a new task $\mathcal{T}_i$, the initial parameters θ of the predictor become θ_i'. In MAML, this is computed using one or more SGD updates on $\mathcal{D}_{\text{train}}$ of task $\mathcal{T}_i$. For the sake of simplicity and without loss of generality, we consider one SGD update:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}\left(\mathcal{P}_\theta\right), \tag{1}$$

where α is the learning rate hyper-parameter. We optimize θ such that the updated θ_i' will produce maximal performance on $\mathcal{D}_{\text{test}}$ of task $\mathcal{T}_i$. When averaged across the tasks sampled from $p(\mathcal{T})$, we have the meta-objective function:

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(\mathcal{P}_{\theta_i'}\right) = \min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(\mathcal{P}_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\mathcal{P}_\theta)}\right). \tag{2}$$

Note that the meta-optimization is performed over the predictor parameters θ, whereas the objective is computed using the updated parameters θ'. This meta-optimization across tasks is performed via SGD in the form of

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(\mathcal{P}_{\theta_i'}\right), \tag{3}$$

where β is the meta-learning rate hyper-parameter. During each iteration, we sample task mini-batch from $p(\mathcal{T})$ and perform the corresponding learner update in Eqn. (1) and meta-learner update in Eqn. (3).

3.4 Adaptive Meta-Learner: Model Adaptation Strategy

In MAML, the model parameters θ_i' of a new task $\mathcal{T}_i$ are obtained by performing a few plain SGD updates on top of the initial θ using its small training set $\mathcal{D}_{\text{train}}$. After meta-training, θ tend to be generic. However, with limited training data from $\mathcal{D}_{\text{train}}$, SGD updates can only modify θ slightly, which is still far from the desired θ_i^* that would be learned from a large set of target samples. Higher-level knowledge is thus necessary to guide the model adaptation to novel tasks.

In fact, during meta-training, for each of the *known action classes*, we have a large training set of annotated sequences, and we sample from this original large set to generate few-shot training sequences. Note that for the novel classes during meta-test, there are no large annotated training sets. Such a setup — meta-learners are trained by sampling small training sets from a large universe of annotated examples — is common in few-shot image classification through meta-learning [61, 66, 14, 21]. While the previous approaches (*e.g.*, MAML) only use this original large set for sampling few-shot training sets, we *explicitly* leverage it and learn the corresponding many-shot model θ_i^* for $\mathcal{T}_i$. During sampling, if some tasks are sampled from the same action class, while they have their own few-shot training sequences, these tasks correspond to the same θ_i^* of that action class. We then use model regression networks (MRN) [66, 69] as the adaptation strategy. MRN is developed in image classification scenarios and obtains learning-to-learn knowledge about a generic transformation from few-shot to many-shot models.

Let θ_i^0 denote the model parameters learned from $\mathcal{D}_{\text{train}}$ by using SGD (*i.e.*, θ_i' in Eqn. (1)). Let θ_i^* denote the *underlying* model parameters learned from a large set of annotated samples. We aim to make the updated θ_i' as close as to the desired θ_i^*. MRN assumes that there exists a generic non-linear transformation, represented by a regression function $\mathcal{H}_\phi$ parameterized by ϕ in the model parameter space, such that $\theta_i^* \approx \mathcal{H}_\phi\left(\theta_i^0\right)$ for a broad range of tasks $\mathcal{T}_i$. The square of the Euclidean distance is used as the regression loss. We then estimate $\mathcal{H}_\phi$ based on a large set of known tasks $\mathcal{T}_i$ drawn from $p(\mathcal{T})$ during meta-training:

$$\min_{\phi} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \left\| \mathcal{H}_\phi\left(\theta_i^0\right) - \theta_i^* \right\|_2^2 . \tag{4}$$

Consistent with [66], we use multilayer feed-forward networks as $\mathcal{H}$.

3.5 An Integrated Framework

We introduce the adaptation strategy in both the meta-training and meta-test phases. For task $\mathcal{T}_i$, after performing a few SGD updates on small training set $\mathcal{D}_{\text{train}}$, we then apply the transformation $\mathcal{H}$ to obtain θ_i'. Eqn. (1) is modified as

$$\theta_i' = \mathcal{H}_\phi\left(\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}\left(\mathcal{P}_\theta\right)\right). \tag{5}$$

Algorithm 1: PAML Meta-Training for k-Shot Human Motion Prediction

Require: Learner: motion predictor model $\mathcal{P}_\theta$ with parameters θ;
MRN adaptation meta-network: $\mathcal{H}_\phi$ with parameters ϕ
Require: $p(\mathcal{T})$: distribution over prediction tasks
Require: α, β, γ: learning or meta-learning rate hyper-parameters;
λ: trade-off hyper-parameter

Randomly initialize θ and ϕ
while *not done* **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for *all* $\mathcal{T}_i$ **do**
    Learn (or retrieve) θ_i^* from the original large set of annotated past and future sequence pairs of the corresponding action class
    Sample k action-specific past and future sequence pairs $\mathcal{D}_{\text{train}} = \left\{\left(\mathbf{X}_u, \mathbf{Y}_u^{gt}\right)\right\}, u = 1, \ldots, k$ from $\mathcal{T}_i$
    Evaluate $\mathcal{L}_{\mathcal{T}_i}\left(\mathcal{P}_\theta\right)$ on $\mathcal{D}_{\text{train}}$
    Compute adapted parameters using Eqn. (5), *i.e.*, performing SGD updates then applying adaptation $\mathcal{H}$: $\theta_i' = \mathcal{H}_\phi\left(\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}\left(\mathcal{P}_\theta\right)\right)$
    Sample $\mathcal{D}_{\text{test}} = \left\{\left(\mathbf{X}_v, \mathbf{Y}_v^{gt}\right)\right\}$ from $\mathcal{T}_i$ for the meta-update
    Evaluate $\widetilde{\mathcal{L}}_{\mathcal{T}_i}\left(\mathcal{P}_{\theta_i'}\right) = \mathcal{L}_{\mathcal{T}_i}\left(\mathcal{P}_{\theta_i'}\right) + \frac{1}{2}\lambda \left\|\theta_i' - \theta_i^*\right\|_2^2$ on $\mathcal{D}_{\text{test}}$ using Eqn. (6)
  end for
  Update θ and ϕ by performing SGD:
  $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \widetilde{\mathcal{L}}_{\mathcal{T}_i}\left(\mathcal{P}_{\theta_i'}\right)$, $\phi \leftarrow \phi - \gamma \nabla_\phi \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \widetilde{\mathcal{L}}_{\mathcal{T}_i}\left(\mathcal{P}_{\theta_i'}\right)$
end while

During *meta-training*, for task $\mathcal{T}_i$, we also have the underlying parameters θ_i^*, which are obtained by performing SGD updates on the corresponding large sample set. Now, the meta-objective in Eqn. (2) becomes

$$\min_{\theta,\phi} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \widetilde{\mathcal{L}}_{\mathcal{T}_i}\left(\mathcal{P}_{\theta_i'}\right) = \min_{\theta,\phi} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(\mathcal{P}_{\theta_i'}\right) + \frac{1}{2}\lambda \left\|\theta_i' - \theta_i^*\right\|_2^2, \tag{6}$$

where λ is the trade-off hyper-parameter. This is a *joint optimization* with respect to both θ and ϕ, and we perform the meta-optimization across tasks using SGD, as shown in Algorithm 1. Hence, we integrate both model initialization and adaptation into an end-to-end meta-learning framework. The model is initialized to produce the parameters that are optimal for its adaptation; meanwhile, the model is adapted by leveraging "learning-to-learn" knowledge about the relationship between few-shot and many-shot models. During *meta-test*, for a novel prediction task, with the learned generic model initialization θ and model adaptation $\mathcal{H}_\phi$, we use Eqn. (5) to obtain the task-specific predictor model.

4 Experimental Evaluation

In this section, we explore the use of our proactive and adaptive meta-learning (PAML). PAML is general and can be in principle applied to a broad range

Table 1. Performance sanity check of our approach by comparing with some state-of-the-art meta-learning approaches to few-shot image classification on the widely used mini-ImageNet dataset. Our PAML outperforms these baselines, showing its general effectiveness for few-shot learning

Method	5-Way Accuracy	
	1-shot	5-shot
Matching Networks [61]	$43.56\% \pm 0.84\%$	$55.31\% \pm 0.73\%$
MAML [14]	$48.70\% \pm 1.84\%$	$63.11\% \pm 0.92\%$
Meta-Learner LSTM [43]	$43.44\% \pm 0.77\%$	$60.60\% \pm 0.71\%$
Prototypical Networks [51]	$46.61\% \pm 0.78\%$	$65.77\% \pm 0.70\%$
Meta Networks [34]	$49.21\% \pm 0.96\%$	$--$
PAML (Ours)	$\mathbf{53.26}\% \pm \mathbf{0.52}\%$	$\mathbf{68.19}\% \pm \mathbf{0.61}\%$

of few-shot learning tasks. For performance calibration, we begin with a sanity check of our approach on a standard few-shot image classification task and compare with existing meta-learning approaches. We then focus on our main task of human motion prediction. Through comparing with the state-of-the-art motion prediction approaches, we show that PAML significantly improves the prediction performance in the small-sample size regime.

4.1 Sanity Check on Few-Shot Image Classification

The majority of the existing few-shot learning and meta-learning approaches are developed in the scenario of classification tasks. As a sanity check, the first question is how our meta-learning approach compares with these prior techniques. For a fair comparison, we evaluate on the standard few-shot image classification task. The most common setup is an N-way, k-shot classification that aims to classify data into N classes when we only have a small number (k) of labeled instances per class for training. The loss function is the cross-entropy error between the predicted and true labels. Following [61, 43, 51, 14, 34, 33], we evaluate on the most widely used mini-ImageNet benchmark. It consists of 64 meta-training and 24 meta-test classes, with 600 images of size 84×84 per class.

During meta-training, each task is sampled as an N-way, k-shot classification problem: we first randomly sample N classes from the meta-training classes; for each class, we randomly sample k and 1 examples to form the training and test set, respectively. During meta-test, we report performance on the unseen classes from the meta-test classes. We use the convolutional network in [14] as the classifier (*i.e.*, learner). Our model adaptation meta-network is a 2-layer fully-connected network with Leaky ReLU nonlinearity.

Table 1 summarizes the performance comparisons in the standard 5-way, 1-/5-shot setting. Our PAML consistently outperforms all the baselines. In particular, there is a notable 5% performance improvement compared with MAML, showing the complementary benefits of our model adaptation strategy. This sanity check verifies the effectiveness of our meta-learning framework. Moreover, some of these existing methods, such as matching networks [61] and prototypi-

cal networks [51], are designed with few-shot classification in mind, and are not readily applicable to other domains such as human motion prediction.

4.2 Few-Shot Human Motion Prediction

We now focus on using our meta-learning approach for human motion prediction. To the best of our knowledge, we are the first ones that explore the few-shot learning problem for human motion prediction. Due to the lack of published protocols, we propose our evaluation protocol for this task.

Dataset. We evaluate on Human 3.6M (H3.6M) [25], a heavily benchmarked, large-scale mocap dataset that has been widely used in human motion analysis. H3.6M contains seven actors performing 15 varied actions. Following the standard experimental setup in [16, 26, 32], we down-sample the dataset by two, train on six subjects, and test on subject five. Each action contains hours of video from these actors performing such activity. Sequence clips are randomly taken from the training and test videos to construct the corresponding training and test sequences [26]. Given the past 50 mocap frames (2 seconds in total), we forecast the future 10 frames (400ms in total) in short-term prediction and the future 25 frames (1 second in total) in long-term prediction.

Few-shot learning task and meta-learning setup. We use 11 action classes for meta-training: directions, greeting, phoning, posing, purchases, sitting, sitting down, taking photo, waiting, walking dog, and walking together. And we use the remaining 4 action classes for meta-test: walking, eating, smoking, and discussion. These four actions are commonly used to evaluate motion prediction algorithms [16, 26, 32]. The *k-shot motion prediction task* which we address is: for a certain action, given a small collection of k *action-specific* past and future sequence pairs, we aim to learn a predictor model so that it is able to predict the possible future motion for a new past sequence from that action. Accordingly, the setup of k-shot prediction tasks in meta-learning is as follows. During meta-training, for each task, we randomly select one action out of 11, and we sample k action-specific sequence pairs as $\mathcal{D}_{\text{train}}$. During meta-test, for each of the 4 novel actions, we sample k sequence pairs from its training set to produce the small set $\mathcal{D}_{\text{train}}$. We then adapt our meta-learned predictor for use as the target action-specific predictor. We evaluate it on the corresponding test set. We run five trials for each action and report the average performance.

Implementation details. In our experiments, the predictor is residual sup., the state-of-the-art encoder-decoder network for motion prediction [32]. For the encoder and decoder, we use a single GRU cell [10] with hidden size 1,024, respectively. Following [32], we use tied weights between the encoder and decoder. We use fully-connected networks with Leaky ReLU nonlinearity as our model adaptation meta-networks. In most cases, k is set as 5 and we also evaluate how performance changes when k varies. By cross-validation, the trade-off hyper-parameter λ is set as 0.1, the learning rate α is set as 0.05, and the meta-learning rates β and γ are set as 0.0005. For the predictor, we clip the gradient to a maximum ℓ_2-norm of 5. We run $10,000$ iterations during meta-training. We use PyTorch [40] to train our model.

Table 2. Mean angle error comparisons between our PAML and variants of the state-of-the-art residual sup. [32] on the 4 novel actions of H3.6M for $k = 5$-shot human motion prediction. Our PAML consistently and significantly outperforms all the baselines. In particular, it is superior to the multi-task learning and transfer learning baselines on all the actions across different time horizons

		Walking						Eating					
milliseconds		80	160	320	400	560	1000	80	160	320	400	560	1000
residual sup. [32] w/ (Baselines)	$\text{Scratch}_{\text{spec}}$	1.90	1.95	2.16	2.18	1.99	2.00	2.33	2.31	2.30	2.30	2.31	2.34
	$\text{Scratch}_{\text{agn}}$	1.78	1.89	2.20	2.23	2.02	2.05	2.27	2.16	2.18	2.27	2.25	2.31
	$\text{Transfer}_{\text{ots}}$	0.60	0.75	0.88	0.93	1.03	1.26	0.57	0.70	0.91	1.04	1.19	1.58
	Multi-task	0.57	0.71	0.79	0.85	0.96	1.12	0.59	0.68	0.83	0.93	1.12	1.33
	$\text{Transfer}_{\text{ft}}$	0.44	0.55	0.85	0.95	0.74	1.03	0.61	0.65	0.74	0.78	0.86	1.19
Meta-learning (Ours)	PAML	**0.35**	**0.47**	**0.70**	**0.82**	**0.80**	**0.83**	**0.36**	**0.52**	**0.65**	**0.70**	**0.71**	**0.79**
		Smoking						Discussion					
milliseconds		80	160	320	400	560	1000	80	160	320	400	560	1000
residual sup. [32] w/ (Baselines)	$\text{Scratch}_{\text{spec}}$	2.88	2.86	2.85	2.83	2.80	2.99	3.01	3.13	3.12	2.95	2.62	2.99
	$\text{Scratch}_{\text{agn}}$	2.53	2.61	2.67	2.65	2.71	2.73	2.77	2.79	2.82	2.73	2.82	2.76
	$\text{Transfer}_{\text{ots}}$	0.70	0.84	1.18	1.23	1.38	2.02	0.58	0.86	1.12	1.18	1.54	2.02
	Multi-task	0.71	0.79	1.09	1.20	1.25	1.23	0.53	0.82	1.02	1.17	1.33	1.97
	$\text{Transfer}_{\text{ft}}$	0.87	1.02	1.25	1.30	1.45	2.06	0.57	0.82	1.11	1.11	1.37	2.08
Meta-learning (Ours)	PAML	**0.39**	**0.66**	**0.81**	**1.01**	**1.03**	**1.01**	**0.41**	**0.71**	**1.01**	**1.02**	**1.09**	**1.12**

Baselines. For a fair comparison, we compare with residual sup. [32], which is the same predictor as ours *but is not meta-learned*. In particular, we evaluate its variants in the small-sample size regime and consider learning both action-specific and action-agnostic models in the following scenarios.

- **Action-specific training from scratch**: for each of the 4 target actions, we learn an action-specific predictor from its k training sequences pairs.
- **Action-agnostic training from scratch**: we learn a single predictor for the 4 target actions from all their training sequence pairs.
- **Off-the-shelf transfer**: we learn a single predictor for the 11 meta-training actions from their large amounts of training sequence pairs, and directly use this predictor for the 4 target actions without modification.
- **Multi-task learning**: we learn a single predictor for all the 15 actions from large amounts of training sequence pairs of the 11 meta-training actions and k sequence pairs per action of the 4 target actions.
- **Fine-tuning transfer**: after learning a single predictor for the 11 meta-training actions from their large amounts of training sequence pairs, we fine-tune it to be an action-specific predictor for each of the 4 target actions, respectively, using its k training sequence pairs.

Evaluation metrics. We evaluate our approach both quantitatively and qualitatively. For the quantitative evaluation, we use the standard metric — mean square error between the predicted motion and the groundtruth motion in the angle space [16, 26, 32]. Following the preprocessing in [54, 32], we exclude the translation and rotation of the whole body. We also qualitatively visualize the prediction frame by frame.

Comparison with the state-of-the-art approaches. Table 2 shows the quantitative comparisons between our PAML and a variety of variants of residual sup. While residual sup. has achieved impressive performance with a large amount of annotated mocap sequences [32], its prediction significantly degrades in the small-sample size regime. As expected, directly training the predictor from a few examples leads to poor performance (*i.e.*, with the angle error in range $2 \sim 3$), due to severe over-fitting. In such scenarios of training from scratch, learning an action-agnostic model is slightly better than learning an action-specific one (*e.g.*, decreasing the angle error by 0.1 at 80ms for walking), since the former allows the predictor to exploit some common motion regularities from multiple actions. By transferring knowledge from relevant actions with large sets of samples in a more principled manner, the prediction performance is slightly improved. This is achieved by multi-task learning, *e.g.*, training an action-agnostic predictor using both the 11 source and 4 target actions, or transfer learning, *e.g.*, first training an action-agnostic predictor using the source actions, and then using it either in an off-the-shelf manner or through fine-tuning.

However, modeling multiple actions is more challenging than modeling each action separately, due to the significant diversity of different actions. The performance improvement of these multi-task learning and transfer learning baselines is limited and their performance is also comparably low. This thus demonstrates the general difficulty of our few-shot motion prediction task. By contrast, our PAML *consistently and significantly* outperforms all the baselines on all the actions across different time horizons, showing the effectiveness of our meta-learning mechanism. There is even a noticeable performance boost for the complicated motion (*e.g.*, decreasing the angle error by 0.3 at 80ms for smoking). By explicitly learning from a large number of few-shot prediction tasks during meta-training, PAML is able to extract and leverage knowledge shared *both across different actions and across multiple few-shot prediction tasks*, thus improving the prediction of novel actions from a few examples by a large margin.

Moreover, as mentioned before, most of the current meta-learning approaches, such as matching networks [61] and prototypical networks [51], are developed for the simple tasks like image classification with task-specific model architectures (*e.g.*, learning an embedding space that is useful for nearest neighbor or prototype classifiers), which are not readily applicable to our problem. Unlike them, our approach is general and can be effectively used across a broad range of tasks, as shown in Table 1 and Table 2. Figure 2 further visualizes our prediction and compares with one of the top performing baselines. From Figure 2, we can see that our PAML generates lower-error, more smooth, and realistic prediction.

Ablation studies. In Table 3 and Table 4 we evaluate the contributions of different factors in our approach to the results.

Model initialization vs. model adaptation. Our meta-learning approach consists of two components: a generic model initialization and an effective model adaptation meta-network. In Table 3, we can see that each component by itself is superior to the baselines reported in Table 2 in almost all the scenarios. This shows that meta-learning, in general, by leveraging shared knowledge across

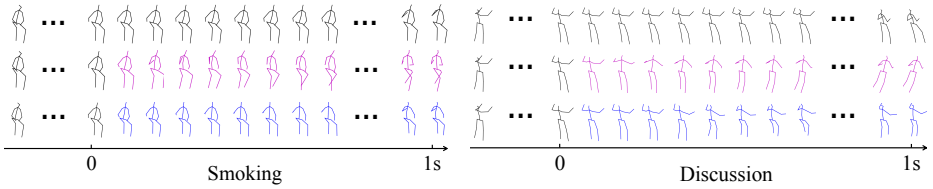


Fig. 2. Visualizations for $k = 5$-shot motion prediction on smoking and discussion. Top: the input sequence and the groundtruth of the prediction sequence. Middle: multi-task learning of residual sup. [32], one of the top performing baselines. Bottom: our prediction results. The groundtruth and the input sequences are shown in black, and the predictions are shown in color. Our PAML produces more smooth and human-like prediction. **Best viewed in color with zoom.**

Table 3. Ablation on model initialization vs. adaptation. Each component by itself outperforms the fine-tuning baseline. Our full model achieves the best performance

		Walking						Eating					
milliseconds		80	160	320	400	560	1000	80	160	320	400	560	1000
Top baseline	$\text{Transfer}_{\text{ft}}$	0.44	0.55	0.85	0.95	0.74	1.03	0.61	0.65	0.74	0.78	0.86	1.19
Meta-learning (Ours)	PAML w/ init	0.40	0.51	0.76	0.86	0.89	0.92	0.49	0.55	0.68	0.74	0.77	0.94
	PAML w/ adapt	0.39	0.52	0.73	0.86	0.90	0.93	0.50	0.59	0.73	0.76	0.81	0.92
	full PAML	**0.35**	**0.47**	**0.70**	**0.82**	**0.80**	**0.83**	**0.36**	**0.52**	**0.65**	**0.70**	**0.71**	**0.79**
		Smoking						Discussion					
milliseconds		80	160	320	400	560	1000	80	160	320	400	560	1000
Top baseline	$\text{Transfer}_{\text{ft}}$	0.87	1.02	1.25	1.30	1.45	2.06	0.57	0.82	1.11	1.11	1.37	2.08
Meta-learning (Ours)	PAML w/ init	0.53	0.72	0.95	1.07	1.11	1.18	0.54	0.77	1.02	1.07	1.36	1.55
	PAML w/ adapt	0.58	0.79	0.86	1.03	1.09	1.12	0.47	0.79	1.12	1.15	1.16	1.26
	full PAML	**0.39**	**0.66**	**0.81**	**1.01**	**1.03**	**1.01**	**0.41**	**0.71**	**1.01**	**1.02**	**1.09**	**1.12**

relevant tasks, enables us to deal with a novel task in a sample-efficient way. Moreover, our full PAML model consistently outperforms its variants, showing the complementarity of each component. This verifies the importance of simultaneously learning a generic initial model and an effective adaptation strategy.

Structure of $\mathcal{H}$. In Table 4 we compare different implementations of the model adaptation meta-network $\mathcal{H}$: as a simple affine transformation, or as networks with $2 \sim 4$ layers. Since Leaky ReLU is used in [66], we try both ReLU and Leaky ReLU as activation function in the hidden layers. The results show that 3-layer fully-connected networks with Leaky ReLU achieve the best performance.

Impact of training sample sizes. In the previous experiments, we focused on a fixed $k = 5$-shot motion prediction task. To test how our meta-learning approach benefits from more training sequences, we evaluate the performance change with respect to the sample size k. Figure 3 summarizes the comparisons with fine-tuning transfer, one of the top performing baselines reported in Table 2, when k varies from 1 to 100 at 80ms. As a reference, we also include the *oracle* performance, which is the residual sup. baseline trained on the entire training set of the target action (*i.e.*, with thousands of annotated sequence pairs). Figure 3 shows that our approach consistently outperforms fine-tuning and improves its

Table 4. Ablation on the structure of $\mathcal{H}$. We vary the number of fully-connected layers and try ReLU and Leaky ReLU as activation function. The results show that "3-layer, Leaky ReLU" works best, but in general $\mathcal{H}$ is robust to specific implementation choices

	Walking					
milliseconds	80	160	320	400	560	1000
PAML w/ 1-layer, None	0.39	0.54	0.73	0.86	0.85	0.91
PAML w/ 2-layer, ReLU	0.39	0.51	0.75	0.85	0.86	0.92
PAML w/ 2-layer, Leaky ReLU	0.38	0.48	0.74	0.83	0.88	0.91
PAML w/ 3-layer, ReLU	0.37	0.50	0.71	**0.82**	0.83	0.88
PAML w/ 3-layer, Leaky ReLU	**0.35**	**0.47**	**0.70**	**0.82**	**0.80**	**0.83**
PAML w/ 4-layer, ReLU	0.37	0.51	0.72	0.86	0.83	0.90
PAML w/ 4-layer, Leaky ReLU	0.36	0.49	0.73	0.83	0.83	0.89

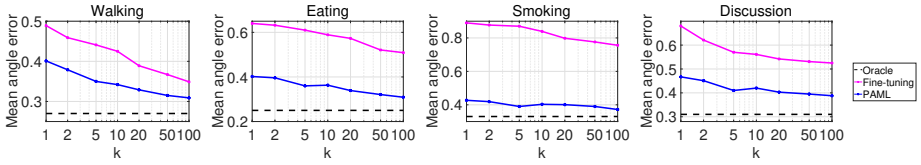


Fig. 3. Impact of the training sample size k for k-shot motion prediction. We compare our PAML with fine-tuning transfer of residual sup. [32], one of the top performing baselines. As a reference, we also include the oracle performance, which is residual sup. trained *from thousands of annotated sequence pairs*. X-axis: number of training sequence pairs k per task. Y-axis: mean angle error. Ours consistently outperforms fine-tuning and *with only* 100 *sequences*, it achieves the performance close to the oracle.

performance with more and more training sequences. Interestingly, through our meta-learning mechanism, *with only* 100 *sequences*, we achieve the performance that is close to the oracle trained from thousands of sequences.

5 Conclusions

In this work we have formulated a novel problem of few-shot human motion prediction and proposed a conceptually simple but powerful approach to address this problem. Our key insight is to jointly learn a generic model initialization and an effective model adaptation strategy through meta-learning. To do so, we utilize a novel combination of model-agnostic meta-learning and model regression networks, two meta-learning approaches that have complementary strengths, and unify them into an integrated, end-to-end framework. As a sanity check, we demonstrate that our approach significantly outperforms existing techniques on the most widely benchmarked few-shot image classification task. We then present the state-of-the-art results on few-shot human motion prediction.

Acknowledgments. The motivating example with a Pepper robot in the introduction section was done in the Laboratory of Professor Manuela Veloso at CMU. We thank Professor Veloso and Mr. Kevin Zhang for all their help.

References

1. Akhter, I., Simon, T., Khan, S., Matthews, I., Sheikh, Y.: Bilinear spatiotemporal basis models. ACM Transactions on Graphics (TOG) **31**(2), 17:1–17:12 (April 2012)
2. Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B., De Freitas, N.: Learning to learn by gradient descent by gradient descent. In: Advances in Neural Information Processing Systems (NIPS). pp. 3981–3989. Barcelona, Spain (December 2016)
3. Azizpour, H., Sharif Razavian, A., Sullivan, J., Maki, A., Carlsson, S.: From generic to specific deep representations for visual recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 36–45. Boston, MA, USA (June 2015)
4. Barsoum, E., Kender, J., Liu, Z.: HP-GAN: Probabilistic 3D human motion prediction via GAN. arXiv preprint arXiv:1711.09561 (November 2017)
5. Bengio, S., Bengio, Y., Cloutier, J., Gecsei, J.: On the optimization of a synaptic learning rule. In: Conference on Optimality in Biological and Artificial Networks. Dallas, TX, USA (February 1992)
6. Bertinetto, L., Henriques, J.F., Valmadre, J., Torr, P., Vedaldi, A.: Learning feed-forward one-shot learners. In: Advances in Neural Information Processing Systems (NIPS). pp. 523–531. Barcelona, Spain (December 2016)
7. Brand, M., Hertzmann, A.: Style machines. In: ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH). pp. 183–192. New Orleans, LA, USA (July 2000)
8. Brubaker, M.A., Fleet, D.J., Hertzmann, A.: Physics-based person tracking using the anthropomorphic walker. International Journal of Computer Vision (IJCV) **87**(1-2), 140 (March 2010)
9. Bütepage, J., Black, M.J., Kragic, D., Kjellström, H.: Deep representation learning for human motion prediction and classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1591–1599. Honolulu, HI, USA (July 2017)
10. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. In: Syntax, Semantics and Structure in Statistical Translation (SSST). pp. 103–111. Doha, Qatar (October 2014)
11. Duan, Y., Andrychowicz, M., Stadie, B., Ho, O.J., Schneider, J., Sutskever, I., Abbeel, P., Zaremba, W.: One-shot imitation learning. In: Advances in Neural Information Processing Systems (NIPS). pp. 1087–1098. Long Beach, CA, USA (December 2017)
12. Edwards, H., Storkey, A.: Towards a neural statistician. In: International Conference on Learning Representations (ICLR). Toulon, France (April 2017)
13. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **28**(4), 594–611 (2006)
14. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning (ICML). pp. 1126–1135. Sydney, Australia (August 2017)
15. Finn, C., Yu, T., Zhang, T., Abbeel, P., Levine, S.: One-shot visual imitation learning via meta-learning. In: Conference on Robot Learning (CoRL). Mountain View, CA, USA (November 2017)

16. Fragkiadaki, K., Levine, S., Felsen, P., Malik, J.: Recurrent network models for human dynamics. In: IEEE International Conference on Computer Vision (ICCV). pp. 4346–4354. Las Condes, Chile (December 2015)
17. Fu, Y., Xiang, T., Jiang, Y.G., Xue, X., Sigal, L., Gong, S.: Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content. IEEE Signal Processing Magazine (SPM) **35**(1), 112–125 (January 2018)
18. Ghosh, P., Song, J., Aksan, E., Hilliges, O.: Learning human motion models for long-term predictions. In: International Conference on 3D Vision (3DV). pp. 458–466. Qingdao, China (October 2017)
19. Gui, L.Y., Wang, Y.X., Liang, X., Moura, J.M.F.: Adversarial geometry-aware human motion prediction. In: European Conference on Computer Vision (ECCV). Munich, Germany (September 2018)
20. Gui, L.Y., Zhang, K., Wang, Y.X., Liang, X., Moura, J.M.F., Veloso, M.M.: Teaching robots to predict human motion. In: IEEE/RSJ International Conference on Intelligent Robots (IROS). Madrid, Spain (October 2018)
21. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: IEEE International Conference on Computer Vision (ICCV). pp. 3037–3046. Venice, Italy (October 2017)
22. Hauberg, S., Sommer, S., Pedersen, K.S.: Gaussian-like spatial priors for articulated tracking. In: European Conference on Computer Vision (ECCV). pp. 425–437. Crete, Greece (September 2010)
23. Held, D., Thrun, S., Savarese, S.: Robust single-view instance recognition. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 2152–2159. Stockholm, Sweden (May 2016)
24. Huang, D.A., Kitani, K.M.: Action-reaction: Forecasting the dynamics of human interaction. In: European Conference on Computer Vision (ECCV). pp. 489–504. Zurich (September 2014)
25. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: 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 (July 2014)
26. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-RNN: Deep learning on spatio-temporal graphs. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5308–5317. Las Vegas, NV, USA (June-July 2016)
27. Jong, M.d., Zhang, K., Rhodes, T., Schmucker, R., Zhou, C., Ferreira, S., Cartucho, J., Veloso, M., Roth, A.: Towards a robust interactive and learning social robot. In: International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Stockholm, Sweden (July 2018)
28. Koppula, H., Saxena, A.: Learning spatio-temporal structure from RGB-D videos for human activity detection and anticipation. In: International Conference on Machine Learning (ICML). pp. 792–800. Atlanta, GA, USA (June 2013)
29. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **38**(1), 14–29 (May 2016)
30. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. In: ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH). pp. 473–482. San Antonio, TX, USA (July 2002)
31. Li, K., Malik, J.: Learning to optimize. In: International Conference on Learning Representations (ICLR). Toulon, France (April 2017)

32. Martinez, J., Black, M.J., Romero, J.: On human motion prediction using recurrent neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4674–4683. Honolulu, HI, USA (July 2017)
33. Mishra, N., Rohaninejad, M., Chen, X., Abbeel, P.: A simple neural attentive meta-learner. In: International Conference on Learning Representations (ICLR). Vancouver, Canada (April-May 2018)
34. Munkhdalai, T., Yu, H.: Meta networks. In: International Conference on Machine Learning (ICML). pp. 2554–2563. Sydney, Australia (August 2017)
35. Murray, R.M., Li, Z., Sastry, S.S., Sastry, S.S.: A mathematical introduction to robotic manipulation. CRC Press, 1 edition (March 1994)
36. Nichol, A., Schulman, J.: Reptile: A scalable metalearning algorithm. arXiv preprint arXiv:1803.02999 (March 2018)
37. Paden, B., Čáp, M., Yong, S.Z., Yershov, D., Frazzoli, E.: A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Transactions on Intelligent Vehicles (T-IV) **1**(1), 33–55 (March 2016)
38. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering (TKDE) **22**(10), 1345–1359 (October 2010)
39. Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 763–768. Kobe, Japan (May 2009)
40. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: Advances in Neural Information Processing Systems (NIPS) Autodiff Workshop. Long Beach, CA, USA (December 2017)
41. Pavlovic, V., Rehg, J.M., MacCormick, J.: Learning switching linear models of human motion. In: Advances in Neural Information Processing Systems (NIPS). pp. 981–987. Vancouver, Canada (December 2001)
42. Poppe, R.: Vision-based human motion analysis: An overview. Computer Vision and Image Understanding (CVIU) **108**(1-2), 4–18 (October 2007)
43. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: International Conference on Learning Representations (ICLR). Toulon, France (April 2017)
44. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: An astounding baseline for recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 806–813. Columbus, OH, USA (June 2014)
45. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: International Conference on Machine Learning (ICML). pp. 1842–1850. New York City, NY, USA (June 2016)
46. Schmidhuber, J.: Learning to control fast-weight memories: An alternative to dynamic recurrent networks. Neural Computation **4**(1), 131–139 (1992)
47. Schmidhuber, J., Zhao, J., Wiering, M.: Shifting inductive bias with success-story algorithm, adaptive Levin search, and incremental self-improvement. Machine Learning **28**(1), 105–130 (1997)
48. Schmidhuber, J.: Evolutionary principles in self-referential learning. (On learning now to Learn: The Meta-Meta-Meta...-Hook). Diploma thesis, Technische Universitat Munchen, Munich, Germany (May 1987)
49. Schmidhuber, J.: A neural network that embeds its own meta-levels. In: IEEE International Conference on Neural Networks (ICNN). pp. 407–412. San Francisco, CA, USA (March-April 1993)

50. Schmidt, L.A.: Meaning and compositionality as statistical induction of categories and constraints. Ph.D. thesis, Massachusetts Institute of Technology, Boston, MA, USA (September 2009)
51. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems (NIPS). pp. 4077–4087. Long Beach, CA, USA (December 2017)
52. Sutskever, I., Hinton, G.E., Taylor, G.W.: The recurrent temporal restricted Boltzmann machine. In: Advances in Neural Information Processing Systems (NIPS). pp. 1601–1608. Whistler, Canada (December 2009)
53. Taylor, G.W., Hinton, G.E.: Factored conditional restricted Boltzmann machines for modeling motion style. In: International Conference on Machine Learning (ICML). pp. 1025–1032. Montreal, Canada (June 2009)
54. Taylor, G.W., Hinton, G.E., Roweis, S.T.: Modeling human motion using binary latent variables. In: Advances in Neural Information Processing Systems (NIPS). pp. 1345–1352. Vancouver, Canada (December 2007)
55. Taylor, G.W., Sigal, L., Fleet, D.J., Hinton, G.E.: Dynamical binary latent variable models for 3D human pose tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 631–638. San Francisco, CA, USA (June 2010)
56. Thrun, S.: Is learning the n-th thing any easier than learning the first? In: Advances in Neural Information Processing Systems (NIPS). pp. 640–646. Denver, CO, USA (December 1996)
57. Thrun, S.: Lifelong learning algorithms. In: Learning to learn, pp. 181–209. Springer US, Boston, MA, USA (January 1998)
58. Thrun, S., Pratt, L.: Learning to learn: Introduction and overview. In: Learning To Learn, pp. 3–17. Springer US, Boston, MA, USA (January 1998)
59. Urtasun, R., Fleet, D.J., Geiger, A., Popović, J., Darrell, T.J., Lawrence, N.D.: Topologically-constrained latent variable models. In: International Conference on Machine Learning (ICML). pp. 1080–1087. Helsinki, Finland (July 2008)
60. Vernon, D., Von Hofsten, C., Fadiga, L.: A roadmap for cognitive development in humanoid robots, vol. 11. Springer Science & Business Media (2011)
61. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems (NIPS). pp. 3630–3638. Barcelona, Spain (December 2016)
62. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **30**(2), 283–298 (February 2008)
63. Wang, Y.X., Hebert, M.: Model recommendation: Generating object detectors from few samples. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1619 – 1628. Boston, MA, USA (June 2015)
64. Wang, Y.X., Hebert, M.: Learning by transferring from unsupervised universal sources. In: AAAI Conference on Artificial Intelligence (AAAI). pp. 2187–2193. Phoenix, AZ, USA (February 2016)
65. Wang, Y.X., Hebert, M.: Learning from small sample sets by combining unsupervised meta-training with CNNs. In: Advances in Neural Information Processing Systems (NIPS). pp. 244–252. Barcelona, Spain (December 2016)
66. Wang, Y.X., Hebert, M.: Learning to learn: Model regression networks for easy small sample learning. In: European Conference on Computer Vision (ECCV). pp. 616–634. Amsterdam, The Netherlands (October 2016)
67. Wang, Y.X., Girshick, R., Hebert, M., Hariharan, B.: Low-shot learning from imaginary data. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7278–7286. Salt Lake City, UT, USA (June 2018)

68. Wang, Y.X., Ramanan, D., Hebert, M.: Growing a brain: Fine-tuning by increasing model capacity. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3029–3038. Honolulu, HI, USA (2017)
69. Wang, Y.X., Ramanan, D., Hebert, M.: Learning to model the tail. In: Advances in Neural Information Processing Systems (NIPS). pp. 7029–7039. Long Beach, CA, USA (December 2017)
70. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems (NIPS). pp. 3320–3328. Montréal, Canada (December 2014)
71. Zhu, Y., Wang, Z., Merel, J., Rusu, A., Erez, T., Cabi, S., Tunyasuvunakool, S., Kramár, J., Hadsell, R., de Freitas, N., Heess, N.: Reinforcement and imitation learning for diverse visuomotor skills. arXiv preprint arXiv:1802.09564 (February 2018)