

Knowledge as Priors: Cross-Modal Knowledge Generalization for Datasets without Superior Knowledge

Long Zhao¹ Xi Peng² Yuxiao Chen¹ Mubbashir Kapadia¹ Dimitris N. Metaxas¹

¹Rutgers University ²University of Delaware

{lz311, yc984, mk1353, dnm}@cs.rutgers.edu, xipeng@udel.edu

Abstract

Cross-modal knowledge distillation deals with transferring knowledge from a model trained with superior modalities (Teacher) to another model trained with weak modalities (Student). Existing approaches require paired training examples exist in both modalities. However, accessing the data from superior modalities may not always be feasible. For example, in the case of 3D hand pose estimation, depth maps, point clouds, or stereo images usually capture better hand structures than RGB images, but most of them are expensive to be collected. In this paper, we propose a novel scheme to train the Student in a Target dataset where the Teacher is unavailable. Our key idea is to generalize the distilled cross-modal knowledge learned from a Source dataset, which contains paired examples from both modalities, to the Target dataset by modeling knowledge as priors on parameters of the Student. We name our method “Cross-Modal Knowledge Generalization” and demonstrate that our scheme results in competitive performance for 3D hand pose estimation on standard benchmark datasets.

1. Introduction

Leveraging multi-modal knowledge to boost the performance of classic computer vision problems, such as classification [28, 35, 50], object detection [14, 39, 51] and gesture recognition [1, 7, 40, 44, 54, 59, 60], has emerged as a promising research field in recent years. Current paradigms for transferring knowledge across modalities involve aligning feature representations from multiple modalities of data during training, and then improving the performance of a unimodal system during testing with the aligned feature representations. Several different schemes for learning these feature representations have been proposed over the years [1, 40, 49, 50], and all of these rely on the availability of paired training samples from different modalities.

Recently, Gupta *et al.* [14] have introduced *Cross-Modal Knowledge Distillation (CMKD)* which is a generic yet effi-

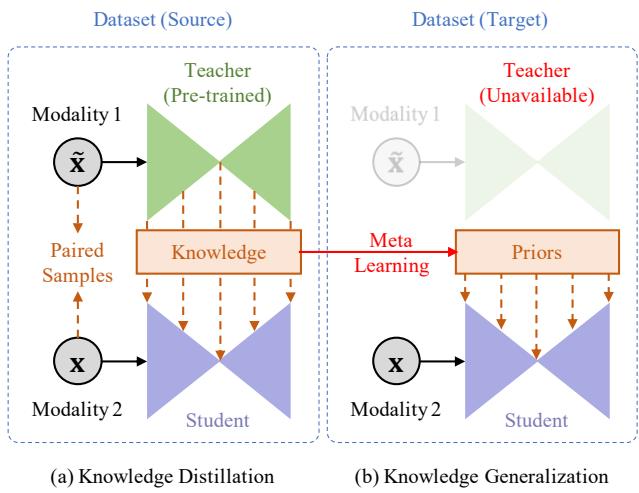


Figure 1. Cross-modal knowledge generalization. (a) Existing approaches distill cross-modal knowledge from the teacher to student in a source dataset. (b) We propose knowledge generalization which transfers learned knowledge in the source to a target dataset where the superior knowledge, *i.e.*, the teacher, is unavailable.

cient scheme among these. They transfer knowledge across different modalities by a *Teacher-Student* scheme [16, 41, 55]. Generally, teacher networks deliver excellent performance since they are trained on modalities with superior knowledge. However, data of these modalities may be limited or expensive to be collected. On the other hand, a student network is trained using a weak modality and thereby often results in lower performance. The goal of knowledge distillation is to transfer superior knowledge from teachers to the student by aligning their intermediate feature representations. For simplicity, in this paper, we consider a form of cross-modal knowledge distillation problems in datasets where only two modalities, *i.e.*, one teacher and one student, are involved as shown in Fig. 1 (a).

The question we ask in this work is, *what is the analogue of this paradigm for datasets which do not have modalities with superior knowledge?* As a motivating exam-

ple, consider the case of 3D hand pose estimation. There are a number of “superior” modalities beyond RGB images which capture more accurate 3D hand structures, *e.g.*, depth maps [32, 45, 61], point clouds [13, 23], or stereo images [56]. These data together with their paired RGB images can be collected by corresponding devices or synthesized using pre-defined hand shape models [12, 37]. However, most of real-world datasets still come with only a single weak modality, *i.e.*, RGB images, which raises the question: *is it possible for neural networks to transfer learned cross-modal knowledge to those target datasets where superior modalities are absent?*

We answer this question in this paper and propose a technique to transfer learned cross-modal knowledge from a source dataset, where both modalities are available, to the target dataset, where only one weak modality exists. Our technique uses “paired” data from the two modalities in the source dataset to distill cross-modal knowledge, and leverages meta-learning to generalize the knowledge to the target dataset by treating it as priors on the parameters of the student network. We call our scheme *Cross-Modal Knowledge Generalization (CMKG)*, which is illustrated in Fig. 1 (b). We further evaluate the performance of the proposed scheme in 3D hand pose estimation. We show that our generalized knowledge serves as a good regularizer to help the network learn better representations for 3D hands, and improves final results in the target dataset as well.

Our work makes the following contributions. First, unlike existing methods that distill knowledge across modalities in the same dataset, we introduce a novel method for Cross-Modal Knowledge Generalization, which generalizes the learned knowledge in the source to a target dataset where the superior modality is unavailable. Second, we introduce a novel meta-learning approach to transfer knowledge across datasets. Specifically, in Sect. 3, a simple yet powerful method is presented to distill cross-modal knowledge in the source dataset. The learned knowledge in the source dataset is then regarded as priors on network parameters during the training procedure in the target dataset. Sect. 4 describes the meta-learning algorithm for learning these priors. Third, we comprehensively evaluate our scheme in 3D hand pose estimation and demonstrate its comparable performance to the state-of-the-art methods in Sect. 5. Note that our scheme can be easily generalized to different tasks, and we leave this for future work.

2. Related Work

Knowledge Distillation. The concept of knowledge distillation was first shown by Hinton *et al.* [16]. Subsequent research [2, 6, 36] enhanced distillation by matching intermediate representations in the networks along with outputs using different approaches. Zagoruyko and Komodakis [55] proposed to align attentional activation maps

between networks. Srinivas and Fleuret [41] improved it by applying Jacobian matching to networks. Recently, cross-modal knowledge distillation [14, 50, 54] extended knowledge distillation by applying it to transferring knowledge across different modalities. Our approach generalizes cross-modal knowledge distillation to target datasets where superior modalities are missing.

Meta-Learning. Meta-learning is also known as “learning to learn”, which intends to learn how learning can be performed in a more efficient manner. Previous approaches studied this problem from a probabilistic modeling perspective [8, 21] or in metric spaces [25, 29, 38]. Recent remarkable advances in gradient-based optimization approaches have rekindled the interest in meta-learning. Among these, Model-Agnostic Meta-Learning (MAML) [9] is proposed to solve few-shot learning. Li *et al.* [22] extended MAML for domain generalization. Balaji *et al.* [3] introduced a meta-regularization function to train networks which can be easily generalized to different domains. Our meta-learning algorithm follows the spirit of these gradient-based methods but aims to learn cross-modal knowledge as priors.

3D Hand Pose Estimation. Estimating 3D hand poses from depth maps has made great progress in the past few years [10, 11, 26, 32, 43]. On the other hand, 3D hand pose estimation from RGB images is significantly more challenging. Zimmermann and Brox [61] first proposed a deep network to learn a network-implicit 3D articulation prior together with 2D key points for predicting 3D hand poses. Other studies [40, 52, 53] learned latent representations with a variational auto-encoder for inference of 3D hand poses. Note that some recent methods [4, 12, 31, 58] focused on recovering the full shapes of 3D hands other than locations of key hand joints, which have a different research target compared with our work.

Yuan *et al.* [54] is the most related work in spirit to ours. Like our work, they employed cross-modal knowledge distillation to improve the performance of RGB-based 3D hand pose estimation. Our method differs significantly in that in addition to knowledge distillation, we aim to address a more challenging problem of transferring cross-modal knowledge to target datasets where depth maps are unavailable.

3. Cross-Modal Knowledge Distillation

We assume that the input data is available in two modalities \mathbf{x}_i and $\tilde{\mathbf{x}}_i$, where $\tilde{\mathbf{x}}_i$ owns superior knowledge than \mathbf{x}_i . For each modality, one network is primarily trained with the data from its own modality. To be specific, we train a *teacher* network g using $\tilde{\mathbf{x}}_i$ and a *student* network f using \mathbf{x}_i . Given the ground truth \mathbf{y}_i , the teacher network parameterized by ψ minimizes the following ℓ^2 regression loss:

$$\mathcal{L}_{\text{REG}}(\tilde{\mathbf{x}}_i, \mathbf{y}_i; \psi) = \|g(\tilde{\mathbf{x}}_i; \psi) - \mathbf{y}_i\|^2. \quad (1)$$

During the training of the student network, the goal of

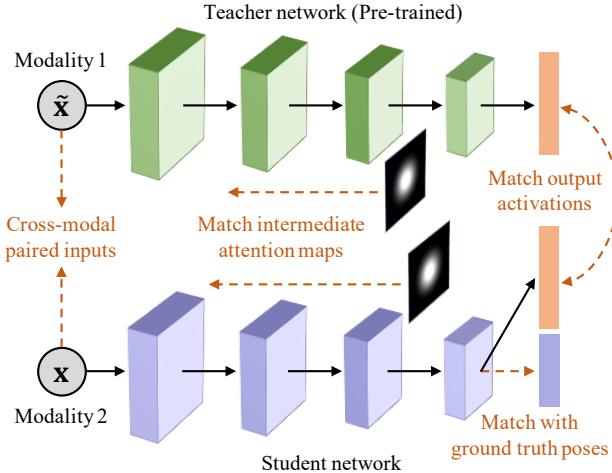


Figure 2. Illustration of our proposed approach for cross-modal knowledge distillation. For the student network, we match its outputs with the ground truth poses (\mathcal{L}_{REG}). Given cross-modal paired inputs, we match the final activations of a pre-trained teacher network (\mathcal{L}_{ACT}). We also match aggregated activations or ‘‘attention’’ maps between networks, similar to the work of [55] (\mathcal{L}_{ATT}). The distillation loss ($\mathcal{L}_{\text{DIST}}$) is a combination of the last two.

cross-modal knowledge distillation is to improve the learning process by transferring the knowledge from the teacher to student. The transferred knowledge can be viewed as an extra supervision in addition to the ground truth. To this end, the knowledge of networks is shared by aligning the semantics of the deep representations, *i.e.*, activation maps of intermediate layers, between the teacher and student.

Let $Q_j \in \mathbb{R}^{C \times H \times W}$ denote the activation map of the j -th layer in the network, which consists of C feature channels with spatial dimensions $H \times W$. We feed \mathbf{x}_i to the student network f and its paired $\tilde{\mathbf{x}}_i$ to the pre-trained teacher network g . Their last activation maps Q_l are aligned by:

$$\mathcal{L}_{\text{ACT}}(\mathbf{x}_i, \tilde{\mathbf{x}}_i; \boldsymbol{\theta}) = \|Q_l(\mathbf{x}_i; f) - Q_l(\tilde{\mathbf{x}}_i; g)\|^2, \quad (2)$$

where $\boldsymbol{\theta}$ are the parameters of the student network. Furthermore, we also match the attention maps [55] of the intermediate layers between the teacher and student. Specifically, let $A_j \in \mathbb{R}^{H \times W}$ be the channel-wise attention map of Q_j calculated by $A_j = \sum_{i=1}^C \|Q_j^{(i)}\|^2$, where $Q_j^{(i)}$ represents the i -th channel of Q_j . Then A_j is ℓ^2 -normalized using $\bar{A}_j = \frac{A_j}{\|A_j\|}$, and we define the attention loss as:

$$\mathcal{L}_{\text{ATT}}(\mathbf{x}_i, \tilde{\mathbf{x}}_i; \boldsymbol{\theta}) = \sum_{i \in \mathcal{I}} \|\bar{A}_i(\mathbf{x}_i; f) - \bar{A}_i(\tilde{\mathbf{x}}_i; g)\|^2, \quad (3)$$

where \mathcal{I} denote the indices of all teacher-student activation layer pairs for which we want to transfer attention maps. Our full knowledge distillation loss can be written as:

$$\mathcal{L}_{\text{DIST}}(\mathbf{x}_i, \tilde{\mathbf{x}}_i; \boldsymbol{\theta}) = \mathcal{L}_{\text{ACT}} + \lambda \cdot \mathcal{L}_{\text{ATT}}, \quad (4)$$

where λ is a hyper-parameter which is set to 1.0×10^3 empirically in the rest of the paper. The final student network is trained with the regression loss \mathcal{L}_{REG} in Eq. (1) together with the distillation loss $\mathcal{L}_{\text{DIST}}$ in Eq. (4). The whole pipeline of our approach is summarized in Fig. 2.

4. Cross-Modal Knowledge Generalization

Consider two datasets: $\mathcal{D}_S = \{\mathbf{x}_i^S, \tilde{\mathbf{x}}_i^S, \mathbf{y}_i^S\}_i$ is a source dataset while $\mathcal{D}_T = \{\mathbf{x}_i^T, \mathbf{y}_i^T\}_i$ denotes a target dataset. Cross-modal knowledge can be efficiently distilled in the source dataset by neural networks as shown in Sect. 3, since training pairs $(\mathbf{x}_i^S, \tilde{\mathbf{x}}_i^S)$ are available in \mathcal{D}_S . However, due to the absence of $\mathcal{K}_T = \{\tilde{\mathbf{x}}_i^T\}_i$, direct knowledge distillation is impossible in the target dataset \mathcal{D}_T .

In this paper, we address a novel and challenging task of *Cross-Modal Knowledge Generalization*. Specifically, we aim to learn the network parameters $\boldsymbol{\theta}_{\text{DIST}}$ which contain superior knowledge \mathcal{K}_T in the target dataset \mathcal{D}_T . As mentioned above, the main challenge is that \mathcal{K}_T is unavailable in \mathcal{D}_T . Our key idea is to generalize the learned knowledge from \mathcal{D}_S to \mathcal{D}_T . This is achieved by interpreting knowledge as priors on the network parameters, which can be learned in \mathcal{D}_S with meta-learning. In the following sections, we first derive our formulation from a probabilistic view. Then we present the meta-learning algorithm for knowledge generalization and theoretically show its connection to the expectation maximization (EM) algorithm.

4.1. Knowledge as Priors

From a Bayesian perspective, a neural network can be viewed as a probabilistic model $P(\mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\theta})$: given an input \mathbf{x}_i , the network assigns a probability to each possible $\mathbf{y}_i \in \mathcal{Y}$ with the parameters $\boldsymbol{\theta}$. Here, we consider a regression problem where $P(\mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\theta})$ is a Gaussian distribution which corresponds to a mean squared loss, and \mathbf{x}_i is mapped onto the parameters of a distribution on \mathcal{Y} using network layers parameterized by $\boldsymbol{\theta}$. Given a dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_i$, $\boldsymbol{\theta}$ can be learned by maximum likelihood estimation (MLE):

$$\begin{aligned} \boldsymbol{\theta}_{\text{MLE}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log P(\mathcal{D} | \boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_i \log P(\mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\theta}). \end{aligned} \quad (5)$$

We assume that $\log P(\mathcal{D} | \boldsymbol{\theta})$ is differentiable w.r.t. $\boldsymbol{\theta}$, and then Eq. (5) is typically solved by gradient descent.

The objective of cross-modal knowledge generalization is to find the parameters $\boldsymbol{\theta}_{\text{DIST}}$ by using the training examples in \mathcal{D}_T with intractable knowledge \mathcal{K}_T . This leads to maximize the posterior density of the parameters $\boldsymbol{\theta}$ directly depends on \mathcal{D}_T and implicitly depends on \mathcal{K}_T . In order to explicitly capture this dependence, we introduce a latent

variable ϕ summarizing the knowledge carried by \mathcal{K}_T :

$$\begin{aligned} P(\boldsymbol{\theta}|\mathcal{D}_T, \mathcal{K}_T) &= \int P(\boldsymbol{\theta}, \phi|\mathcal{D}_T, \mathcal{K}_T)d\phi \\ &= \int P(\boldsymbol{\theta}|\mathcal{D}_T, \mathcal{K}_T, \phi)P(\phi|\mathcal{D}_T, \mathcal{K}_T)d\phi \\ &= \int P(\boldsymbol{\theta}|\mathcal{D}_T, \phi)P(\phi|\mathcal{D}_T, \mathcal{K}_T)d\phi. \end{aligned} \quad (6)$$

Note that the last equation is the result of assuming that \mathcal{K}_T and $\boldsymbol{\theta}$ are conditionally independent given the latent variable ϕ . Since both \mathcal{K}_T and integrating Eq. (6) over ϕ are intractable, we make an approximation that uses a *point estimation* ϕ_{META} . This point estimation is obtained via the meta-learning approach described in Sect. 4.2, hence avoiding the need to perform integration over ϕ or interact \mathcal{K}_T . Consequently, maximizing the logarithm of the posterior density of Eq. (6) can be written as:

$$\begin{aligned} \boldsymbol{\theta}_{\text{DIST}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log P(\boldsymbol{\theta}|\mathcal{D}_T, \mathcal{K}_T) \\ &\approx \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log P(\boldsymbol{\theta}|\mathcal{D}_T, \phi_{\text{META}}) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \underbrace{\log P(\mathcal{D}_T|\boldsymbol{\theta})}_{\text{Likelihood}} + \underbrace{\log P(\boldsymbol{\theta}|\phi_{\text{META}})}_{\text{Prior (Knowledge)}}, \end{aligned} \quad (7)$$

where the last equality results from a direct application of Bayes rule. So, finding the parameters $\boldsymbol{\theta}_{\text{DIST}}$ involves a two step training procedure: (1) optimizing the prior term which obtains the point estimation ϕ_{META} using meta-learning and (2) optimizing the likelihood term which maximizes Eq. (7) using the learned parameters ϕ_{META} .

In a Bayesian setting, priors on the parameters can be interpreted as regularization. Thus the prior term in Eq. (7) is implemented as a regularizer during network training. Several other regularization schemes have been proposed in the literature such as weight decay [20], dropout [42, 48] and batch normalization [17]. While they aim to reduce error on examples drawn from the test distribution, the objective of our work is to learn a regularizer that captures cross-modal knowledge learned from the source dataset.

4.2. Learning Priors with Meta-Learning

As mentioned above, we model the prior term as a regularizer $\mathcal{R}(\boldsymbol{\theta}; \phi)$. Given the input $\boldsymbol{\theta}$, \mathcal{R} is implemented with a neural network parameterized by ϕ .

As described in Sect. 3, cross-modal knowledge distillation leads to optimize the following objective:

$$\mathcal{G}(\mathbf{x}_i, \tilde{\mathbf{x}}_i, \mathbf{y}_i; \boldsymbol{\theta}) = \mathcal{L}_{\text{REG}}(\mathbf{x}_i, \mathbf{y}_i; \boldsymbol{\theta}) + \mathcal{L}_{\text{DIST}}(\mathbf{x}_i, \tilde{\mathbf{x}}_i; \boldsymbol{\theta}), \quad (8)$$

where \mathcal{L}_{REG} is the regression loss minimizing the mean squared errors of the prediction and ground truth, and the distillation loss $\mathcal{L}_{\text{DIST}}$ distills knowledge from the teacher

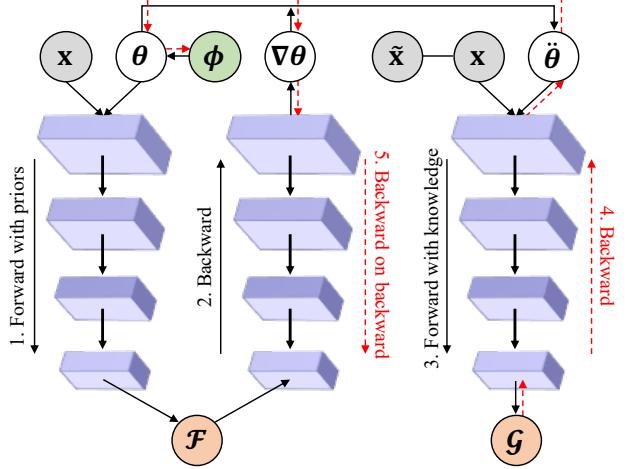


Figure 3. Computational graph of our meta-training algorithm (as shown in Algorithm 1) in a deep neural network, which can be efficiently implemented using the second order derivative.

to student. Using the regularizer \mathcal{R} , we introduce a regularized regression loss which is defined as:

$$\mathcal{F}(\mathbf{x}_i, \mathbf{y}_i; \boldsymbol{\theta}, \phi) = \mathcal{L}_{\text{REG}}(\mathbf{x}_i, \mathbf{y}_i; \boldsymbol{\theta}) + \mathcal{R}(\boldsymbol{\theta}; \phi). \quad (9)$$

During the training procedure on the source dataset, we aim to learn the regularizer \mathcal{R} in Eq. (9) which mimics the behavior of $\mathcal{L}_{\text{DIST}}$ in Eq. (8), so that \mathcal{F} can be applied to the target dataset where the superior knowledge is missing. We now describe the procedure for learning \mathcal{R} .

When training the student network on the source dataset, at iteration k , we begin by sampling a mini-batch from the dataset. Using this batch, l steps of gradient descent are first performed with the regularized regression loss \mathcal{F} . Let $\ddot{\boldsymbol{\theta}}_k$ denote the network parameters after these l steps. Then the full loss \mathcal{G} on the same batch computed using $\ddot{\boldsymbol{\theta}}_k$ is minimized w.r.t. the regularizer parameters ϕ . The regularizer \mathcal{R} is finally updated with the gradients which unroll through the l gradient steps. This ensures that \mathcal{G} can be approximated by \mathcal{F} using \mathcal{R} . After finishing the training, since the same regularizer is trained on every pair of $(\mathbf{x}_i, \tilde{\mathbf{x}}_i)$, the resulting \mathcal{R} captures the notion of cross-modal knowledge contained in the source dataset. Please refer to Fig. 3 for an illustration of the meta-training step. The entire algorithm is given in Algorithm 1. Note that l is set to 1 empirically in this paper, as we observe that a 1-step update is sufficient to achieve good performance.

Once the regularizer is learned, its parameters ϕ_{META} are frozen and the final student network initialized from scratch is trained on the target dataset using the regularized loss function \mathcal{F} . This meta-testing procedure generalizes the learned knowledge to the target dataset with \mathcal{R} parameterized by ϕ_{META} as summarized in Algorithm 2.

Our meta-learning approach is general and can be implemented by any type of regularizer. In this paper, we use

Algorithm 1 Meta-training for learning priors.

Input: Batch size N , # of iterations K , learning rate α .
Input: # of inner iterations l , meta learning rate β .

- 1: Initialize θ_0, ϕ_0
- 2: **for** $k = 0$ to $K - 1$ **do**
- 3: Sample N examples $\{(\mathbf{x}_n^S, \tilde{\mathbf{x}}_n^S, \mathbf{y}_n^S) \sim \mathcal{D}_S\}_{n=1}^N$
- 4: $\ddot{\theta}_0 \leftarrow \theta_k$
- 5: **for** $i = 0$ to $l - 1$ **do**
- 6: $\ddot{\theta}_{i+1} \leftarrow \ddot{\theta}_i - \alpha \nabla_{\ddot{\theta}_i} \mathcal{F}(\mathbf{x}_n^S, \mathbf{y}_n^S; \ddot{\theta}_i, \phi_k)$ ▷ E-step
- 7: **end for**
- 8: $\ddot{\theta}_k \leftarrow \ddot{\theta}_l$
- 9: $\phi_{k+1} \leftarrow \phi_k - \beta \nabla_{\phi_k} \mathcal{G}(\mathbf{x}_n^S, \tilde{\mathbf{x}}_n^S, \mathbf{y}_n^S; \ddot{\theta}_k)$ ▷ M-step
- 10: $\theta_{k+1} \leftarrow \theta_k - \alpha \nabla_{\theta_k} \mathcal{G}(\mathbf{x}_n^S, \tilde{\mathbf{x}}_n^S, \mathbf{y}_n^S; \theta_k)$
- 11: **end for**
- 12: $\phi_{\text{META}} \leftarrow \phi_K$

weighted ℓ^2 loss as our regularization function:

$$\mathcal{R}(\theta; \phi) = \sum_i \phi_i \|\theta_i\|^2, \quad (10)$$

where ϕ_i and θ_i are the i -th weight and parameter of the network. The use of weighted ℓ^2 loss can be interpreted as a learnable weight decay mechanism: weights θ_i for which ϕ_i is large will be decayed to zero and those for which ϕ_i is small will be boosted. By using our meta-learning approach, we select a set of weights that carry cross-modal knowledge across every pair of inputs $(\mathbf{x}_i, \tilde{\mathbf{x}}_i)$.

4.3. Theoretical Understanding

This section gives a theoretical understanding of Algorithm 1 in Sect. 4.2. We draw its connection to the expectation maximization (EM) algorithm and thus its convergence is theoretically guaranteed. To achieve this, we first derive the lower bound of the target objective and then show how it is solved by our meta-learning algorithm using EM.

In a Bayesian framework, given the evidence \mathcal{D}_S , learning the parameters ϕ of priors leads to maximize the likelihood $P(\mathcal{D}_S|\phi)$. Proposition 1 indicates its lower bound.

Proposition 1. Let q be any posterior distribution function over the latent variables θ given the evidence \mathcal{D}_S . Then, the marginal log-likelihood can be lower bounded:

$$\log P(\mathcal{D}_S|\phi) = \log \int P(\mathcal{D}_S, \theta|\phi) d\theta \geq \mathcal{E}(q, \phi), \quad (11)$$

where \mathcal{E} is the evidence lower-bound (ELBO) defined as:

$$\mathcal{E} \triangleq \mathbb{E}_q[\log P(\mathcal{D}_S|\theta)] - \text{KL}[q(\theta|\mathcal{D}_S)\|P(\theta|\phi)]. \quad (12)$$

Note that $\text{KL}[\cdot\|\cdot]$ in Eq. (12) represents the KL divergence between two distributions q and P . The proof to this proposition can be found in our supplementary material. According to Proposition 1, the following proposition shows that Algorithm 1 is an instance of EM maximizing \mathcal{E} .

Algorithm 2 Meta-testing for knowledge generalization.

Input: Batch size N , # of iterations K , learning rate α .
Input: Learned parameters ϕ_{META} from Algorithm 1.

- 1: Initialize θ_0
- 2: **for** $k = 0$ to $K - 1$ **do**
- 3: Sample N examples $\{(\mathbf{x}_n^T, \mathbf{y}_n^T) \sim \mathcal{D}_T\}_{n=1}^N$
- 4: $\theta_{k+1} \leftarrow \theta_k - \alpha \nabla_{\theta_k} \mathcal{F}(\mathbf{x}_n^T, \mathbf{y}_n^T; \theta_k, \phi_{\text{META}})$
- 5: **end for**
- 6: $\theta_{\text{DIST}} \leftarrow \theta_K$

Proposition 2. The parameters ϕ can be estimated by maximizing the evidence lower-bound of $\log P(\mathcal{D}_S|\phi)$ via expectation maximization (EM) as shown in Algorithm 1.

Proof. The EM algorithm can be viewed as two alternating maximization steps: E-step and M-step. In the k -th E-step, for fixed ϕ , the objective \mathcal{E} is bounded above by the first term in Eq. (12), and achieves that bound when the KL divergence term is zero. This is achieved if and only if q is equal to P . Therefore, the E-step sets q to P and estimates the posterior probability:

$$\ddot{\theta}_k = \underset{\theta_k}{\operatorname{argmax}} q_k = \underset{\theta_k}{\operatorname{argmax}} P(\theta_k|\phi_k). \quad (13)$$

And, after an E-step, the objective \mathcal{E} equals the likelihood term. In the k -th M-step, we fix $\ddot{\theta}$ and solve:

$$\phi_{k+1} = \underset{\phi_k}{\operatorname{argmax}} \mathcal{E}(q_k, \phi_k). \quad (14)$$

Both E-step and M-step are solved by gradient descent as commented in Algorithm 1. We have thus shown that Algorithm 1 is an instance of EM. \square

5. Experiments

The proposed approach is evaluated in 3D hand pose estimation. We aim to answer the following three questions: (1) Can our Cross-Modal Knowledge Distillation (CMKD) distill accurate cross-modal knowledge from the source dataset? (Sect. 5.3) (2) Does the proposed Cross-Modal Knowledge Generalization (CMKG) successfully transfer learned knowledge to the target dataset? (Sect. 5.4) (3) And what factors influence the effect of our CMKG? (Sect. 5.5)

5.1. Implementation Details

For simplicity, we use the same architecture for teacher and student networks. We choose ResNet [15] as the backbone, and adjust the final fully connected layer to output a vector representing the 3D positions of 21 hand joints. All corresponding depth maps of RGB images in the dataset are employed as the modality containing superior knowledge.

Data Augmentation. Recent methods [4, 12, 47] show that learning from synthetic data improves the performance

of 3D pose estimation, as it offers more effective hand variations than traditional data augmentation technologies, *e.g.*, random cropping and rotation. Hence, we create a synthetic dataset of paired hand images and depth maps with their 3D annotations using the MANO [37] hand model for synthetic data augmentation. Following the setting of [4], hand geometries are obtained by sampling pose and shape parameters from $[-2, 2]^{10}$ and $[-0.03, 0.03]^{10}$, respectively. Meanwhile, hand appearances are modeled by the original scans with 3D coordinates and RGB values from [37]. We create example hand appearances using these registered scan topologies. After rotations, translations and scalings are applied to hand models, the textured hands are finally rendered on background images which are randomly sampled and cropped from [18, 24]. In total, we synthesize 50,000 hand images with large variations for training.

Network Training. The input image is resized to 256×256 . For CMKD, all networks are trained using Adam [19] with mini-batches of size 32. The learning rate is set as 2.5×10^{-4} . The teacher is pre-trained for 200 epochs, while the student is trained with only the regression loss for 100 epochs and then fine-tuned with the full loss for another 100 epochs. For CMKG, the regularizer is optimized using Stochastic Gradient Descent (SGD) with the learning rate of 1.0×10^{-3} during the fine-tuning of the student network.

5.2. Datasets and Metrics

Our proposed approach is comprehensively evaluated on two publicly available datasets for 3D hand pose estimation: RHD [61] and STB [56] with the standard metrics.

Datasets. Rendered Hand Pose Dataset (RHD) [61] is a synthetic dataset built upon 20 different characters performing 39 actions. It provides 41,258 images for training and 2,728 images for evaluation with a resolution of 320×320 . All of them are fully annotated with a 21 joint skeleton hand model and additionally the depth map for each hand. This dataset is challenging due to the large variations in viewpoints and textures. We employ RHD for training and evaluating our knowledge distillation method.

Stereo Hand Pose Tracking Benchmark (STB) [56] is a real-world dataset which contains 18,000 stereo image pairs as well as the ground truth 3D positions of 21 hand joints from different scenarios. This benchmark has 12 different sequences and every sequence contains 1,500 stereo pairs. Following the evaluation protocol of [5, 12, 40, 61], we use the sequence of B1 for evaluation and the others for training. STB is utilized for evaluating the proposed cross-modal knowledge generation algorithm.

To make the joint definition consistent across different datasets, we reorganize the joints of each hand according to the layout of MANO [37]. Especially, we move the root joint location from palm center to wrist of each hand in STB. Following the same protocol used in [5, 12, 40, 61],

Settings	Backbone	EPE (RGB / Depth / KD)
\mathcal{L}_{ACT}	ResNet-18	24.68 / 13.60 / 23.41 _{↓1.27}
$\mathcal{L}_{ACT}, \mathcal{L}_{ATT}$	ResNet-18	24.68 / 13.60 / 22.19 _{↓2.49}
$\mathcal{L}_{ACT}, \mathcal{L}_{ATT}, \mathcal{A}$	ResNet-18	23.07 / 12.06 / 20.89 _{↓2.18}
$\mathcal{L}_{ACT}, \mathcal{L}_{ATT}, \mathcal{A}$	ResNet-50	20.74 / 10.78 / 18.06 _{↓2.68}

Table 1. Ablation study on the choices of loss terms used in Eq. (4), synthetic data augmentation denoted by \mathcal{A} , and network backbone for knowledge distillation. We also report the performance gain in EPE (mm) obtained by cross-modal knowledge distillation.

the absolute depth of root joint (wrist) and global hand scale, which is set as the bone length between MCP and PIP joints of the middle finger, are provided at test time.

Metrics. We evaluate the performance of 3D hand pose estimation with three common metrics in the literature: (1) EPE: the mean hand joint error which measures the average Euclidean distance in millimeters (mm) between the predicted 3D joints and the ground truth; (2) 3D PCK: the percentage of correct key points which are within the Euclidean distance of a certain threshold to its respective ground truth position; (3) AUC: the area under the curve on 3D PCK for different error thresholds.

5.3. Evaluation of Knowledge Distillation

To evaluate the performance of the proposed knowledge distillation approach for 3D hand pose estimation, we train three networks for each setting: a baseline network trained with RGB images (RGB), a teacher network trained with depth maps (Depth) and a student network trained using the knowledge distillation algorithm presented in Sect. 3 (KD). All the experiments are conducted on RHD dataset.

Ablation Study. We first evaluate the impacts of different losses used in knowledge distillation, data augmentation, and network architecture on the performance of 3D hand pose estimation. The results of EPE are presented in Table 1. We can see that the model trained with the full distillation loss (\mathcal{L}_{ACT} and \mathcal{L}_{ATT}) achieves higher performance improvement, from 1.27 (mm) to 2.49 (mm), which indicates that all the losses have contributions to distilling cross-modal knowledge from depth maps for 3D hand pose estimation. Moreover, synthetic data augmentation and employing deeper network during the training procedure can further boost the performance.

Comparison to State of the Art. We compare the 3D PCK curves with state-of-the-art methods [5, 40, 53, 54, 61] on RHD dataset in Fig. 4. We use ResNet-50 as the backbone. Note that some other works [4, 12, 58] aim to predict the 3D hand shape other than hand joint locations, which are with different research targets compared with ours. Therefore, they are not included here. In Fig. 4 (left), our method surpasses most existing methods except [5], which has a higher AUC of 0.015. However, it is not directly compa-

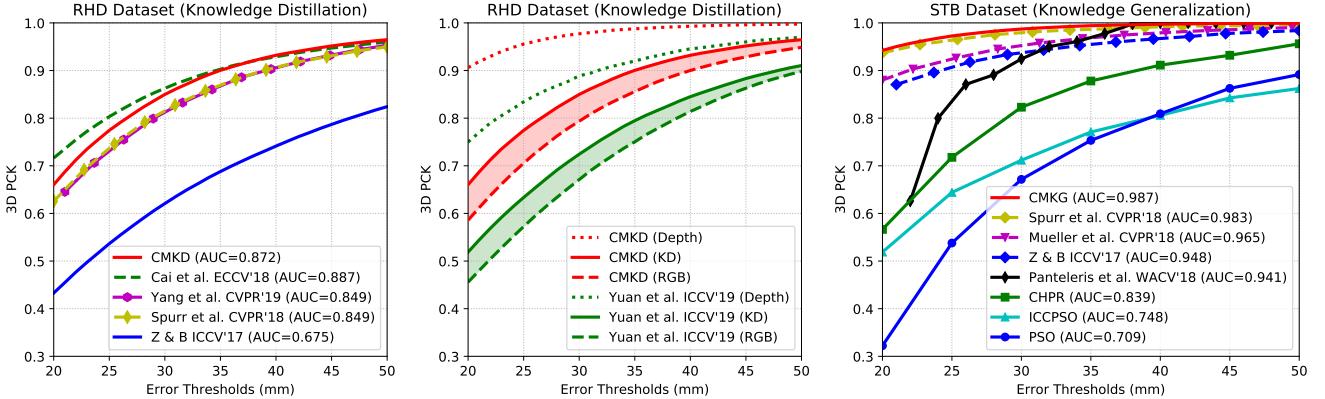


Figure 4. Comparisons with state of the art. Left: 3D PCK on RHD [61] of our knowledge distillation approach (CMKD). Our method has comparable performance to Cai *et al.* [5] which relies on additional 2D annotations for network training. Middle: Comparison with Yuan *et al.* [54] which also distills knowledge from depth. Our approach obtains a more significant improvement (red area, $\Delta_{\text{AUC}} = 0.045$) than [54] (green area, $\Delta_{\text{AUC}} = 0.041$). Right: Our knowledge generalization method (CMKG) obtains state-of-the-art results on STB [56].

table, as [5] incorporates 2D annotations as an additional supervision during network training.

In Fig. 4 (middle), we further compare our approach to Yuan *et al.* [54] which is the most related work also distilling cross-modal knowledge from depth maps for 3D hand pose estimation. We can find that our method substantially outperforms [54]. More importantly, the performance gain achieved by our approach ($\Delta_{\text{AUC}} = 0.045$) is larger than [54] ($\Delta_{\text{AUC}} = 0.041$), which shows that the proposed knowledge distillation algorithm is more efficient.

5.4. Evaluation of Knowledge Generalization

In order to evaluate the effectiveness of the proposed knowledge generalization algorithm, we transfer the learned cross-modal knowledge in RHD to STB and compare our approach to other regularization functions.

Effect of Regularizers. In this experiment, we study the effect of different regularizers including the proposed \mathcal{R} in Eq. (10) on the performance of network trained on STB. We compare our formulation with the default regularizers commonly used in the literature: $\sigma \sum_i \|\phi_i\|^p$, where $\|\cdot\|^p$ is the p -norm of the parameter and σ is a constant weight manually selected for each network. We experiment on the ℓ^1 and ℓ^2 regularizers (where p equals 1 or 2, respectively) and different choices of σ . We also implement a variant of the proposed \mathcal{R} which is ℓ^1 -regularized. The performance of these regularizers are reported in Table 2. We observe that our proposed regularizers outperform the default regularization functions by a large margin. Especially, our ℓ^2 -regularized \mathcal{R} achieves the best performance. These results demonstrate that \mathcal{R} carries effective knowledge learned from the source dataset which helps the training of the target network.

Visualization of Parameters. To give an intuitive understanding of how our regularizer \mathcal{R} affects the network learning, we plot the histograms of the parameters learned

Regularizer	EPE (mm)	AUC
None	15.67	0.915
$\ell^1, \sigma = 1.0 \times 10^{-4}$	11.41 _{4.26}	0.972 _{0.057}
$\ell^1, \sigma = 1.0 \times 10^{-6}$	11.82 _{3.85}	0.964 _{0.049}
$\ell^2, \sigma = 1.0 \times 10^{-3}$	12.28 _{3.39}	0.957 _{0.042}
$\ell^2, \sigma = 1.0 \times 10^{-5}$	12.02 _{3.65}	0.964 _{0.049}
\mathcal{R}, ℓ^1 -regularized	8.86 _{6.81}	0.985 _{0.070}
\mathcal{R}, ℓ^2 -regularized	8.18 _{7.49}	0.987 _{0.072}

Table 2. Effect of different classes of regularization functions on STB [56]. Note that σ denotes the constant weight manually chosen for the default ℓ^1 or ℓ^2 regularizer. We report EPE (mm) and AUC together with the performance gain for each method.

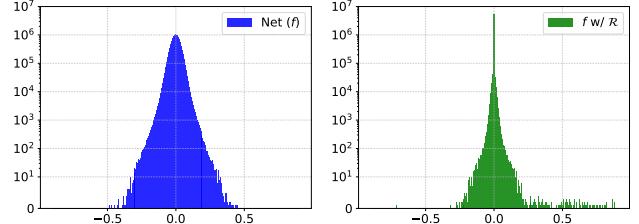


Figure 5. Histograms of the parameters learned by different regression networks on STB [56] dataset. Left: Histogram of the network f without any form of regularization. Right: Histogram of the network trained with the proposed regularizer \mathcal{R} in Eq. (10).

by the network with and without the use of \mathcal{R} in Fig. 5. We can make the following observations. First, for the network trained with regularization, there is a sharper peak at zero. This is due to the positive ϕ_i in Eq. (10) which decays the corresponding θ_i to zero. Second, on the other hand, the parameters of the network with regularization have wider spread, since they are boosted by the negative ϕ_i .

Comparison to State of the Art. We further compare

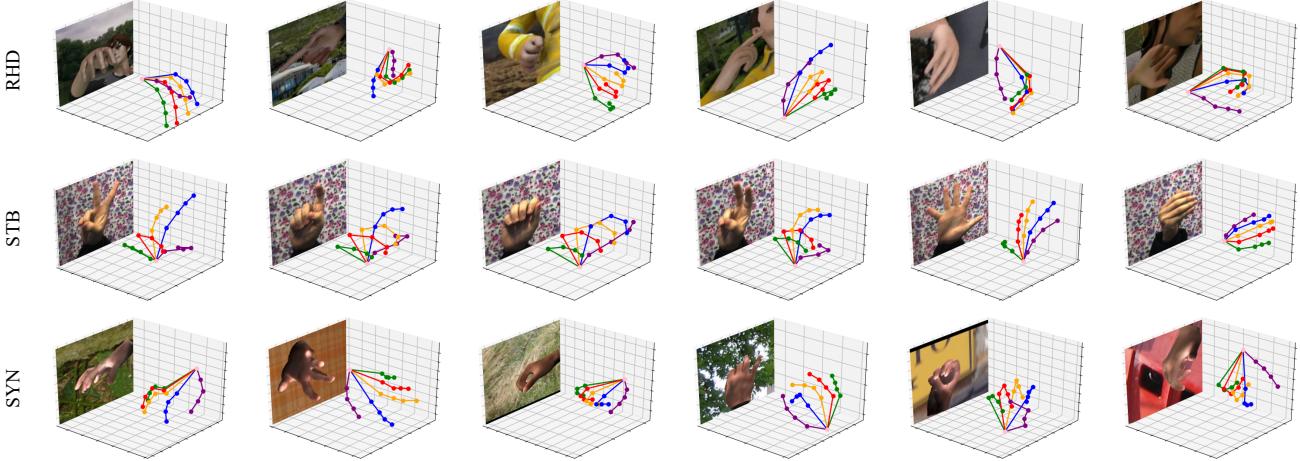


Figure 6. Visual results of our approach on RHD [61] (top) and STB [56] (middle). To demonstrate the generalizability of the proposed method, we also show the results after applying the network trained on STB to the synthetic dataset (bottom). Best viewed in color.

the proposed CMKG to other approaches [27, 31, 40, 61] on STB in Fig. 4 (right). We can see that our regularized network matches the state-of-the-art performance without using complex network architecture, loss functions or additional constraints like previous methods. Our visual results are shown in Fig. 6. As seen, our method is able to accurately predict 3D hand poses across different datasets and generalize the learned knowledge to some novel cases.

5.5. Discussion

One potential concern about the learned knowledge (regularizer) from the source dataset is how it performs when applied to different target datasets. First of all, we point out that it is impossible to learn a domain-independent regularizer from a single source which performs consistently well on all other datasets, since their data usually follow different statistics. Here, we hypothesize that the effect of the learned regularizer depends on two factors: (1) the domain shift between the source and target dataset, and (2) the effect of regularization on the target dataset.

The first factor is straightforward as large domain shifts always lead to difficulties in network generalization. This is a well-defined problem in transfer learning which is tackled by domain adaptation [30, 33, 34]. To illustrate the second factor, we conduct an additional experiment which applies the same regularizers in Sect. 5.4 to a number of different target datasets. Due to the space limitation, we ask the readers to refer to the supplementary material for detailed setups of this experiment. Looking at Table 3, we see a strong correlation between the default and the proposed regularizer: if there is a large increase obtained by the default regularizer, \mathcal{R} can boost the performance even further; otherwise, our improvement is limited. This is intuitive since our formulation is consistent with the default regularization technique.

Target Dataset	w/o \mathcal{R}	Default ℓ^2	\mathcal{R} in Eq. (10)
FreiHAND [62], \mathcal{G}	12.37	12.28 \downarrow 0.09	12.27 \downarrow 0.10
FreiHAND [62], \mathcal{H}	14.49	14.02 \downarrow 0.47	13.82 \downarrow 0.67
FreiHAND [62], \mathcal{S}	15.80	14.92 \downarrow 0.88	14.26 \downarrow 1.54
FreiHAND [62], \mathcal{A}	16.18	15.16 \downarrow 1.02	14.18 \downarrow 2.00
STB [56]	15.67	12.02 \downarrow 3.65	8.18 \downarrow 7.49

Table 3. Effect of regularizers on different target datasets. We report EPE (mm) and the performance gain for each setting. \mathcal{G} , \mathcal{H} , \mathcal{S} and \mathcal{A} are four different domains contained in FreiHAND [62].

Our findings suggest multiple directions of future work. For one, the proposed scheme currently has access to only one single source dataset; we believe that learning from multiple sources will result in better generalizability of the model. On the other hand, we treat target priors as a regularization term in this work, which is perhaps the simplest formulation. We believe that a further exploration on choices of this term will result in improved performance.

6. Conclusion

We introduce an end-to-end scheme for Cross-Modal Knowledge Generalization to transfer cross-modal knowledge between source and target datasets where superior modalities are missing. The core idea is to interpret knowledge as priors on the parameters of the student network which can be efficiently learned by meta-learning. Our method is comprehensively evaluated in 3D hand pose estimation. We show that our scheme can efficiently generalize cross-modal knowledge to the target dataset and significantly boost the network to match the state-of-the-art performance. We believe our work provides new insights in conventional cross-modal knowledge distillation tasks, and serves as a strong baseline in this novel research direction.

References

- [1] Mahdi Abavisani, Hamid Reza Vaezi Joze, and Vishal M. Patel. Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In *CVPR*, pages 1165–1174, 2019.
- [2] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NeurIPS*, pages 2654–2662, 2014.
- [3] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. MetaReg: Towards domain generalization using meta-regularization. In *NeurIPS*, pages 998–1008, 2018.
- [4] Adnane Boukhayma, Rodrigo de Bem, and Philip H.S. Torr. 3D hand shape and pose from images in the wild. In *CVPR*, pages 10843–10852, 2019.
- [5] Yujun Cai, Liuhan Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3D hand pose estimation from monocular RGB images. In *ECCV*, pages 666–682, 2018.
- [6] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *NeurIPS*, pages 742–751, 2017.
- [7] Yuxiao Chen, Long Zhao, Xi Peng, Jianbo Yuan, and Dimitris N. Metaxas. Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. In *BMVC*, 2019.
- [8] Li Fei-Fei, Rob Fergus, and Pietro Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, pages 1134–1141, 2003.
- [9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017.
- [10] Liuhan Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3D hand pose estimation in single depth images: from single-view CNN to multi-view CNNs. In *CVPR*, pages 3593–3601, 2016.
- [11] Liuhan Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 3D convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *CVPR*, pages 1991–2000, 2017.
- [12] Liuhan Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3D hand shape and pose estimation from a single RGB image. In *CVPR*, pages 10833–10842, 2019.
- [13] Liuhan Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3D hand pose estimation. In *ECCV*, pages 475–491, 2018.
- [14] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *CVPR*, pages 2827–2836, 2016.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Workshops*, 2014.
- [17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.
- [18] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, pages 304–317, 2008.
- [19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.
- [20] Anders Krogh and John A. Hertz. A simple weight decay can improve generalization. In *NeurIPS*, pages 950–957, 1992.
- [21] Neil D. Lawrence and John C. Platt. Learning to learn with the informative vector machine. In *ICML*, 2004.
- [22] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018.
- [23] Shile Li and Dongheui Lee. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In *CVPR*, pages 11927–11936, 2019.
- [24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014.
- [25] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018.
- [26] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In *CVPR*, pages 5079–5088, 2018.
- [27] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. GANerated hands for real-time 3D hand tracking from monocular RGB. In *CVPR*, pages 49–59, 2018.
- [28] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *ICML*, pages 689–696, 2011.
- [29] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, pages 721–731, 2018.
- [30] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.
- [31] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. Using a single RGB frame for real time 3D hand pose estimation in the wild. In *WACV*, pages 436–445, 2018.
- [32] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *CVPR*, pages 1106–1113, 2014.
- [33] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *CVPR*, 2020.
- [34] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.
- [35] Dhanesh Ramachandram and Graham W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017.
- [36] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *ICLR*, 2015.

- [37] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):245, 2017.
- [38] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, pages 4077–4087, 2017.
- [39] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3D object detection in RGB-D images. In *CVPR*, pages 808–816, 2016.
- [40] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, pages 89–98, 2018.
- [41] Suraj Srinivas and François Fleuret. Knowledge transfer with jacobian matching. In *ICML*, 2018.
- [42] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [43] Xiao Sun, Yichen Wei, Shuang Liang, Xiaou Tang, and Jian Sun. Cascaded hand pose regression. In *CVPR*, pages 824–832, 2015.
- [44] Yu Tian, Xi Peng, Long Zhao, Shaoting Zhang, and Dimitris N. Metaxas. CR-GAN: learning complete representations for multi-view generation. In *IJCAI*, pages 387–403, 2018.
- [45] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33(5):169, 2014.
- [46] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *CVPR*, pages 3789–3797, 2017.
- [47] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, pages 109–117, 2017.
- [48] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using drop-connect. In *ICML*, pages 1058–1066, 2013.
- [49] Chaoyang Wang, Chen Kong, and Simon Lucey. Distill knowledge from NRSfM for weakly supervised 3D pose learning. In *ICCV*, pages 743–752, 2019.
- [50] Lichen Wang, Jiaxiang Wu, Shao-Lun Huang, Lizhong Zheng, Xiangxiang Xu, Lin Zhang, and Junzhou Huang. An efficient approach to informative feature extraction from multimodal data. In *AAAI*, pages 5281–5288, 2019.
- [51] Dan Xu, Wanli Ouyang, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning cross-modal deep representations for robust pedestrian detection. In *CVPR*, pages 5363–5371, 2017.
- [52] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. Aligning latent spaces for 3D hand pose estimation. In *ICCV*, pages 2335–2343, 2019.
- [53] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *CVPR*, pages 9877–9886, 2019.
- [54] Shanxin Yuan, Bjorn Stenger, and Tae-Kyun Kim. RGB-based 3D hand pose estimation via privileged learning with depth images. In *ICCV Workshops*, 2019.
- [55] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.
- [56] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. A hand pose tracking benchmark from stereo matching. In *ICIP*, pages 982–986, 2017.
- [57] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S. Lin, Tianhe Yu, and Alexei A. Efros. Real-time user-guided image colorization with learned deep priors. In *SIGGRAPH*, 2017.
- [58] Xiong Zhang, Qiang Li, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular RGB image. In *ICCV*, pages 2354–2364, 2019.
- [59] Long Zhao, Xi Peng, Yu Tian, Mubbasis Kapadia, and Dimitris Metaxas. Learning to forecast and refine residual motion for image-to-video generation. In *ECCV*, pages 387–403, 2018.
- [60] Long Zhao, Xi Peng, Yu Tian, Mubbasis Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3D human pose regression. In *CVPR*, pages 3425–3435, 2019.
- [61] Christian Zimmermann and Thomas Brox. Learning to estimate 3D hand pose from single RGB images. In *ICCV*, pages 4903–4911, 2017.
- [62] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russel, Max Argus, and Thomas Brox. FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images. In *ICCV*, pages 813–822, 2019.

Appendix A. Supplementary Material

A.1. Proof of Proposition 1

Proposition 3 (Proposition 1 restated). *Let q be any posterior distribution function over the latent variables θ given the evidence \mathcal{D}_S . Then, the marginal log-likelihood can be lower bounded:*

$$\log P(\mathcal{D}_S|\phi) = \log \int P(\mathcal{D}_S, \theta|\phi) d\theta \geq \mathcal{E}(q, \phi),$$

where \mathcal{E} is the evidence lower-bound (ELBO) defined as:

$$\mathcal{E}(q, \phi) \triangleq \mathbb{E}_q[\log P(\mathcal{D}_S|\theta)] - \text{KL}[q(\theta|\mathcal{D}_S)\|P(\theta|\phi)].$$

Proof. The proposed meta-training as described in Algorithm 1 of the main paper makes a posterior inference based on the graphical model in Fig. 7. Given the evidence \mathcal{D}_S , learning the parameters ϕ leads to maximize the likelihood $P(\mathcal{D}_S|\phi)$:

$$\begin{aligned} \log P(\mathcal{D}_S|\phi) &= \log \int P(\mathcal{D}_S, \theta|\phi) d\theta \\ &= \log \int P(\mathcal{D}_S|\theta, \phi) P(\theta|\phi) d\theta \\ &= \log \int P(\mathcal{D}_S|\theta) P(\theta|\phi) d\theta \\ &= \log \int q(\theta|\mathcal{D}_S) \frac{P(\mathcal{D}_S|\theta) P(\theta|\phi)}{q(\theta|\mathcal{D}_S)} d\theta. \end{aligned}$$

By Jensen's inequality, we have:

$$\begin{aligned} \log P(\mathcal{D}_S|\phi) &= \log \int q(\theta|\mathcal{D}_S) \frac{P(\mathcal{D}_S|\theta) P(\theta|\phi)}{q(\theta|\mathcal{D}_S)} d\theta \\ &\geq \int q(\theta|\mathcal{D}_S) \log \frac{P(\mathcal{D}_S|\theta) P(\theta|\phi)}{q(\theta|\mathcal{D}_S)} d\theta \\ &\triangleq \mathcal{E}(q, \phi), \end{aligned}$$

where $\mathcal{E}(q, \phi)$ is the evidence lower-bound (ELBO) of the likelihood $\log P(\mathcal{D}_S|\phi)$. Then, we further have:

$$\begin{aligned} \mathcal{E}(q, \phi) &= \int q(\theta|\mathcal{D}_S) \log \frac{P(\mathcal{D}_S|\theta) P(\theta|\phi)}{q(\theta|\mathcal{D}_S)} d\theta \\ &= \int q(\theta|\mathcal{D}_S) \log P(\mathcal{D}_S|\theta) d\theta \\ &\quad + \int q(\theta|\mathcal{D}_S) \log \frac{P(\theta|\phi)}{q(\theta|\mathcal{D}_S)} d\theta \\ &= \int q(\theta|\mathcal{D}_S) \log P(\mathcal{D}_S|\theta) d\theta \\ &\quad - \int q(\theta|\mathcal{D}_S) \log \frac{q(\theta|\mathcal{D}_S)}{P(\theta|\phi)} d\theta \\ &= \mathbb{E}_{\theta \sim q(\theta|\mathcal{D}_S)} [\log P(\mathcal{D}_S|\theta)] \\ &\quad - \text{KL}[q(\theta|\mathcal{D}_S)\|P(\theta|\phi)]. \end{aligned}$$

We have thus proven Proposition 3. \square

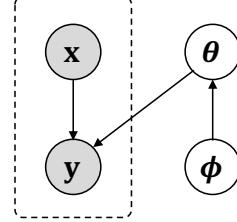


Figure 7. Graphical models for meta-training algorithm.

A.2. Efficient Implementation of Algorithm 1

In order to implement Algorithm 1 of the main paper, we need to compute the second order derivative of the network parameters when a set of ϕ are updated by gradient descent. This is computational expensive especially when the scale of the backbone network becomes very large. In this section, we provide an efficient implementation of Algorithm 1 when the derivative w.r.t. ϕ of the regularizer \mathcal{R} can be calculated directly.

As described in the main paper, we implement \mathcal{R} by a weighted ℓ^2 regularizer in this work. Therefore, the regularized objective function of Eq. (9) in the main paper can be rewritten by:

$$\mathcal{F}(\mathbf{x}_i, \mathbf{y}_i; \theta, \phi) = \mathcal{L}_{\text{REG}}(\mathbf{x}_i, \mathbf{y}_i; \theta) + \sum_i \phi_i \|\theta_i\|^2, \quad (15)$$

where ϕ_i is the i -th weight of the regularizer and θ_i is the i -th parameter of the student network. Then, the k -th gradient descent step of the network parameter θ_i^k is:

$$\begin{aligned} \theta_i^{k+1} &= \theta_i^k - \alpha \frac{\partial \mathcal{F}}{\partial \theta_i^k} = \theta_i^k - \alpha \frac{\partial (\mathcal{L}_{\text{REG}} + \sum_i \phi_i \|\theta_i^k\|^2)}{\partial \theta_i^k} \\ &= \theta_i^k - \alpha \frac{\partial \mathcal{L}_{\text{REG}}}{\partial \theta_i} - 2\alpha \phi_i \theta_i^k \\ &= \theta_i^k (1 - 2\alpha \phi_i) - \alpha \frac{\partial \mathcal{L}_{\text{REG}}}{\partial \theta_i^k}, \end{aligned} \quad (16)$$

where α is the learning rate of θ_i . We can see that Eq. (16) converts our regularizer formulation into the weight decay mechanism, where $2\phi_i$ turns into the decay rate. Since the second term of Eq. (16) is independent with ϕ_i , we only need to compute the first order derivative when updating ϕ_i of the regularizer \mathcal{R} . The modified meta-training approach is illustrated in Algorithm 3.

A.3. Experimental Setup on FreiHAND

FreiHAND [62] is a 3D hand pose dataset which records different hand actions performed by 32 people. For each hand image, MANO-based 3D hand pose annotations are provided. It currently contains 32,560 unique training samples and 3960 unique samples for evaluation. The training

Algorithm 3 Efficient implementation of meta-training.

Input: Batch size N , # of iterations K , learning rate α .

Input: # of inner iterations l , meta learning rate β .

```
Initialize  $\theta_0, \phi_0$ 
for  $k = 0$  to  $K - 1$  do
    Sample  $N$  examples  $\{(\mathbf{x}_n^S, \tilde{\mathbf{x}}_n^S, \mathbf{y}_n^S) \sim \mathcal{D}_S\}_{n=1}^N$ 
     $\ddot{\theta}_0 \leftarrow \theta_k$ 
    for  $i = 0$  to  $l - 1$  do
         $\ddot{\theta}_{i+1} \leftarrow \ddot{\theta}_i(1 - 2\alpha\phi_k) - \alpha\nabla_{\ddot{\theta}_i}\mathcal{L}_{\text{REG}}(\mathbf{x}_n^S, \mathbf{y}_n^S; \ddot{\theta}_i)$ 
    end for
     $\ddot{\theta}_k \leftarrow \ddot{\theta}_l$ 
     $\phi_{k+1} \leftarrow \phi_k - \beta\nabla_{\phi_k}\mathcal{G}(\mathbf{x}_n^S, \tilde{\mathbf{x}}_n^S, \mathbf{y}_n^S; \ddot{\theta}_k)$ 
     $\theta_{k+1} \leftarrow \theta_k - \alpha\nabla_{\theta_k}\mathcal{G}(\mathbf{x}_n^S, \tilde{\mathbf{x}}_n^S, \mathbf{y}_n^S; \theta_k)$ 
end for
 $\phi_{\text{META}} \leftarrow \phi_K$ 
```

samples are recorded with a green screen background allowing for background removal. In addition, it applies three different post processing strategies to training samples for data augmentation. However, these post processing strategies are not applied to evaluation samples.

In Sect. 5.5 of the main paper, we conduct the experiment to evaluate the performance of the learned regularizer when it is applied to different target datasets (domains). In this experiment, we treat the original images collected with the green screen background (\mathcal{G}) in FreiHAND, together with their post-processed results using three different strategies: harmonization [46] (\mathcal{H}), colorization auto [57] (\mathcal{A}), colorization sample [57] (\mathcal{S}), as three different domains contained by FreiHAND. However, since the domains of \mathcal{H} , \mathcal{A} and \mathcal{S} are not provided for the original evaluation samples, we create new training and evaluation splits from the original training data of FreiHAND. Therefore, for each domain, the first 30,000 training samples are used for network training while the rest 2,560 samples are leveraged for evaluation. We use the same setting as described in Sect. 5.1 of the main paper to train the network in this dataset.

A.4. Additional Visual Results

In Figs. 8 to 10, we show additional visual results predicted by our method on RHD [61], STB [56] and the synthetic dataset. We can see that our method is able to accurately estimate 3D hand poses across different datasets.

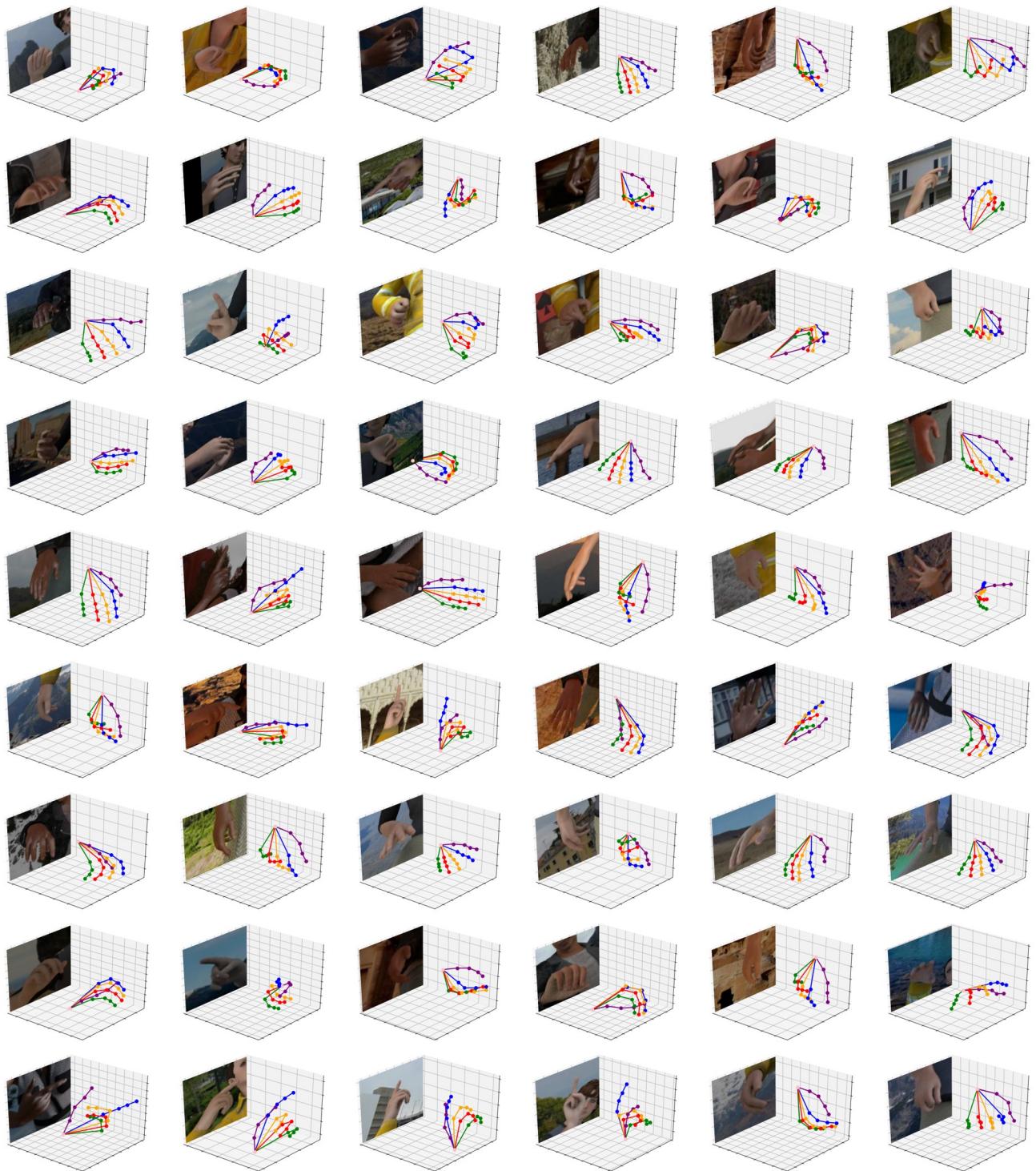


Figure 8. Additional visual results of our approach on RHD [61] dataset.

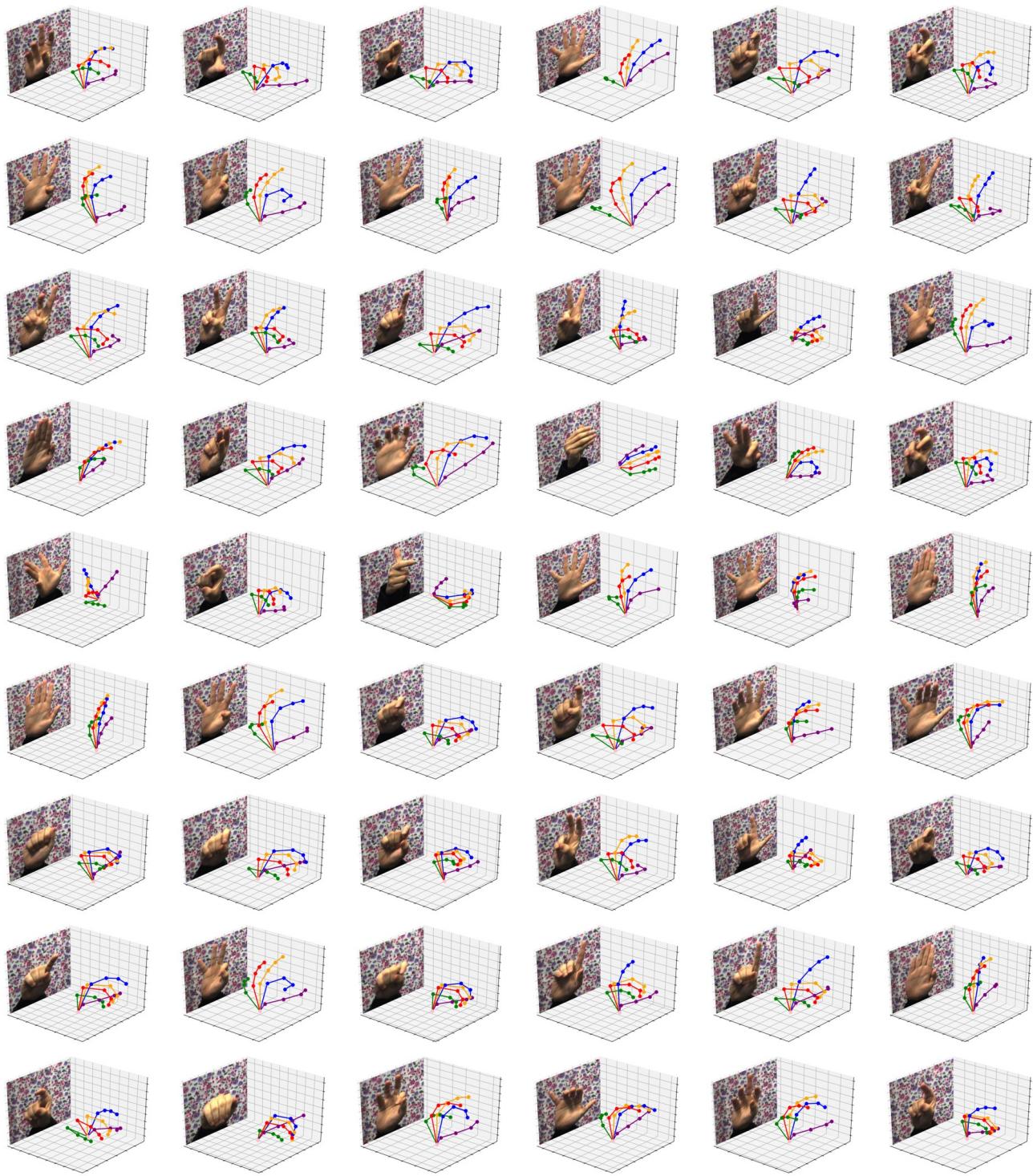


Figure 9. Additional visual results of our approach on STB [56] dataset.

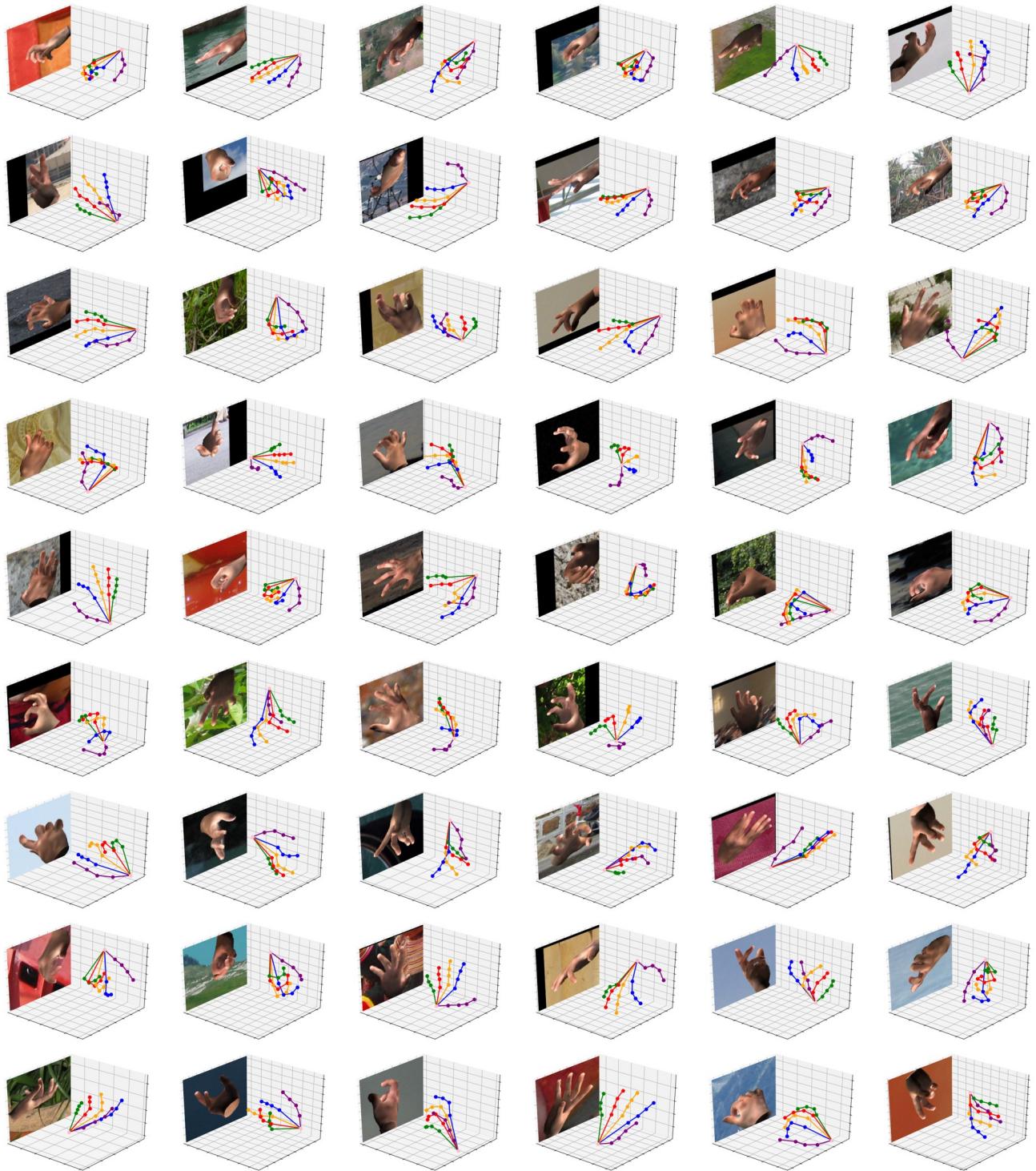


Figure 10. Additional visual results of our approach on synthetic dataset.