Few-shot Learning
(time series, skeletal sequences, images and keypoints)

Piotr Koniusz

Data61/CSIRO
Australian National University

August 17, 2023

Contents

- Uncertainty-DTW for Time Series and Sequences (oral paper, ECCV 2022)
- Temporal-Viewpoint Transportation Plan for Skeletal Few-shot Action Recognition (best student paper award, ACCV 2022)
- Few-shot Keypoint Detection with Uncertainty Learning for Unseen Species (CVPR, 2022)
- Transductive Few-shot Learning with Prototype-based Label Propagation by Iterative Graph Refinement (CVPR, 2023)

Works done with my fantastic PhD students:
Lei Wang, Hao Zhu, Changsheng Lu.

- Looking for strong PhD candidates.
  Talk to me if interested.

Overview

We are interested in few-shot learning (FSL) on 3D skeleton sequences of articulated body joints:

- Many robust pose estimation methods produced many 3D skeleton datasets.
- But labeling is expensive and fast adaptation to new classes underexplored.
- FSL must deal with temporal, viewpoint and geometric distortions.

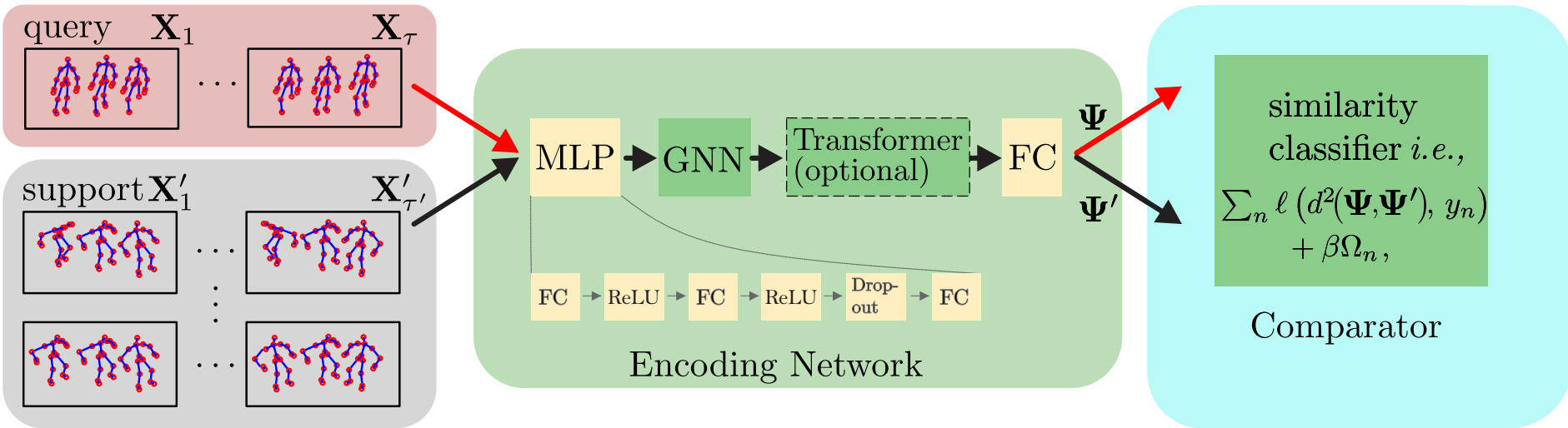


Figure 1: Example few-shot action recognition pipeline.

- We train the **Encoding Network**. The **comparator** captures the similarity between query-support pairs.
- The loss $\ell(\cdot) \to 0$ if query-support pair has the same class labels. For pairs with non-matching labels, $\ell(\cdot) \to \xi$ (FSL learns what is similar/dissimilar).
- Testing: given a set of support sequences+labels, $\ell(\cdot)$ decides which one matches the query.

Overview (cont.)

Formally, our **similarity learning** minimizes the empirical loss $\ell(\cdot)$ and some regularization term $\Omega(\cdot,\cdot)$ expressing our belief about the model:

$$\sum_n \ell(d^2(\mathbf{\Psi}_n, \mathbf{\Psi}'_n), y_n) + \beta\Omega_n(\mathbf{\Psi}_n, \mathbf{\Psi}'_n).$$

Query: $\mathbf{\Psi} \equiv [\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_\tau] \in \mathbb{R}^{d\times\tau}$ with τ temporal frames (or blocks).
Support: $\mathbf{\Psi}' \equiv [\boldsymbol{\psi}'_1, ..., \boldsymbol{\psi}'_{\tau'}] \in \mathbb{R}^{d\times\tau'}$ with τ' temporal frames (or blocks).

However, distance $d(\cdot,\cdot)$ is suboptimal for matching temporal sequences:

- **Temporal location** and **speed** of actions vary.
- Different **viewpoints** or **geometric** distortions..
- High intra-class variance: no two sequences are identical.
- Same/different actors never perform the same action exactly the same way.
- So-called (Soft-)Dynamic Time Warping (DTW) overcomes the temporal localization and speed issues[1]. We build on it.

[1] Cuturi, M., & Blondel, M. (2017, July). **Soft-DTW: a differentiable loss function for time-series**. In *International conference on machine learning* (pp. 894-903). PMLR.

Related work

Comparison of the Euclidean distance *vs.* (Soft-)Dynamic Time Warping (DTW):

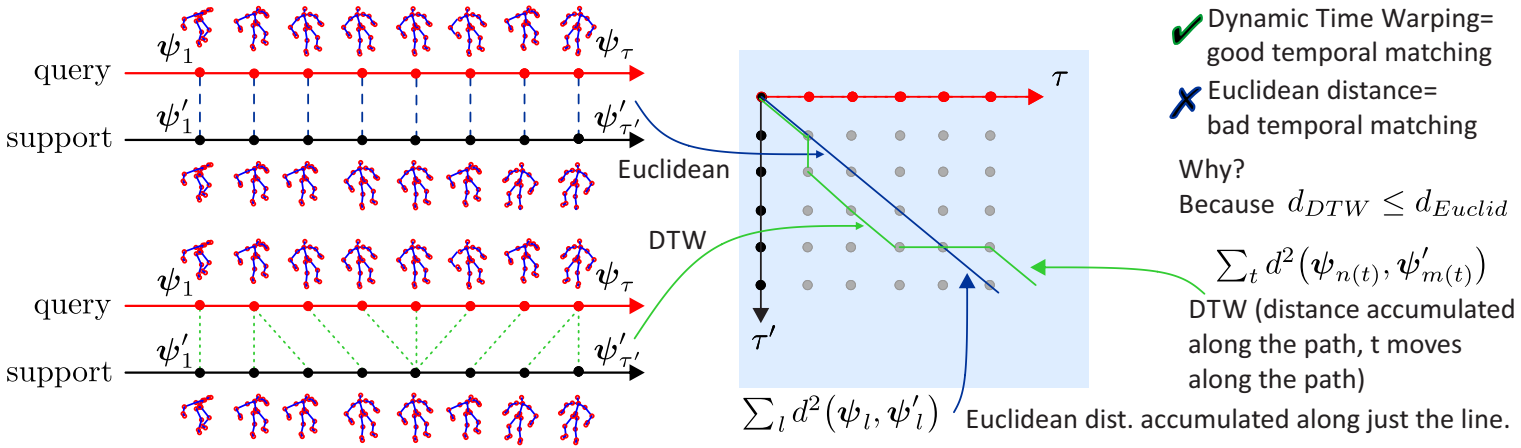


Figure 2: Euclidean dist. (top) *vs.* DTW (bottom). Corresponding matching paths (right).

- The Euclidean distance naively compares features of corresponding frames of two sequences $\mathbf{\Psi}$ and $\mathbf{\Psi}'$.
- DTW (bottom) matches human poses better by taking into account temporal location and speed variations. **DTW transportation plan: step $\downarrow$, $\searrow$, $\rightarrow$.**
- DTW performs 'better' matching (see the green matching path on the right) by factoring out temporal variations. The dark blue path is suboptimal.

Motivation (uncertainty-DTW)

However, sequences Ψ and Ψ' suffer from **the observation noise**.
Compare uncertainty-DTW *vs.* soft-DTW under the noise (indicated in gray):

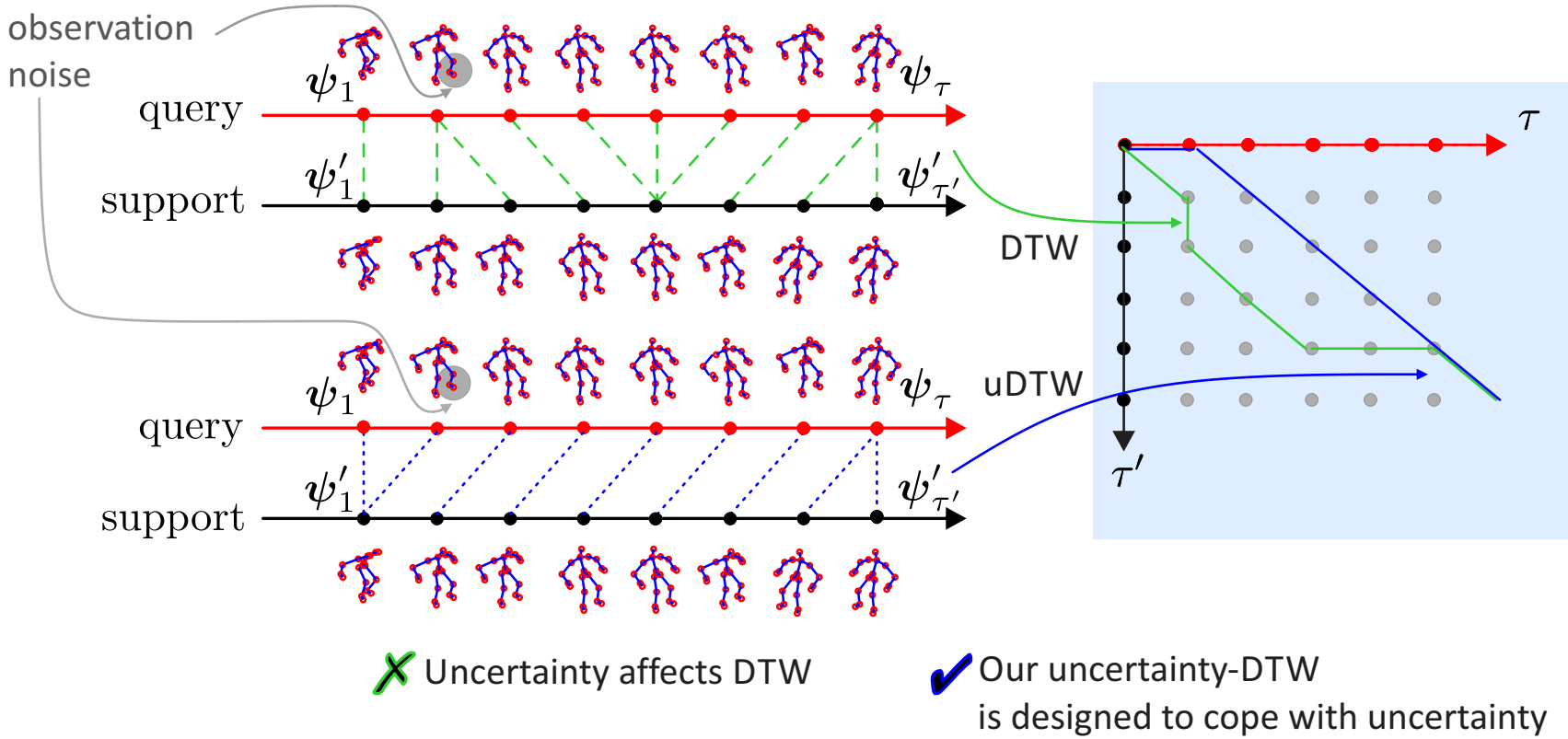


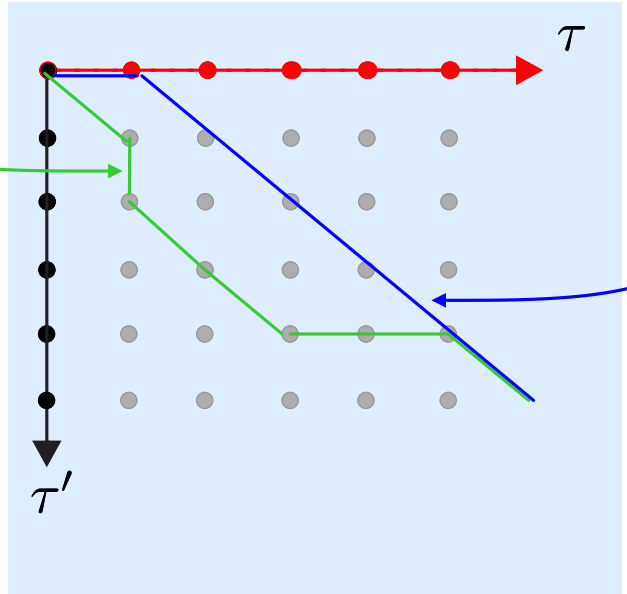
Figure 3: Soft-DTW. (top) *vs.* uncertainty-DTW (bottom).

- Blue path (right) takes uncertainty into account; green path does not.
- Thus, the blue path provides more robust distance for similarity learning.

Approach

$\sum_t d^2\left(\boldsymbol{\psi}_{n(t)}, \boldsymbol{\psi}'_{m(t)}\right)$

DTW
(distance accumulated along the path)



$\sum_t \frac{1}{2\sigma^2_{n(t),m(t)}} d^2\left(\boldsymbol{\psi}_{n(t)}, \boldsymbol{\psi}'_{m(t)}\right)$

$\Omega = \sum_t \log \sigma_{n(t),m(t)}$

uDTW
(uncertainty-weighted distance accumulated along the path)

$\Omega =$ uncertainty penalty
(regularization accumulated along the path)

Figure 4: Soft-DTW *vs.* uncertainty-DTW.

- Uncertainty-DTW models the uncertainty for each frame (or temporal block).
- Each path is a solution to the Maximum Likelihood Estimation (each node is Gaussian with variance): $\prod_t \mathcal{N}(\boldsymbol{\psi}_{n(t)}; \boldsymbol{\psi}'_{m(t)}, \sigma^2_{n(t)m(t)})$
- MLE 'explains' the distances on the path by the modelled distribution.
- Log-likelihood results in d_{uDTW} (see derivations in the paper).
- Additionally, Ω is penalty for selecting (trivially) large uncertainty.

Derivation of uDTW

We proceed by modeling an arbitrary path $\mathbf{\Pi}_i$ from the transportation plan of $\mathcal{A}_{\tau,\tau'}$ as the following Maximum Likelihood Estimation (MLE) problem:

$$\underset{\{\sigma_{mn}\}_{(m,n)\in\mathbf{\Pi}_i}}{\arg\max} \prod_{(m,n)\in\mathbf{\Pi}_i} p(\|\boldsymbol{\psi}_m - \boldsymbol{\psi}'_n\|, \sigma^2_{mn}), \tag{1}$$

where p may be some arbitrary distribution, σ are distribution parameters, $\|\cdot\|$ is an arbitrary norm. For the Normal distribution $\mathcal{N}$, we have:

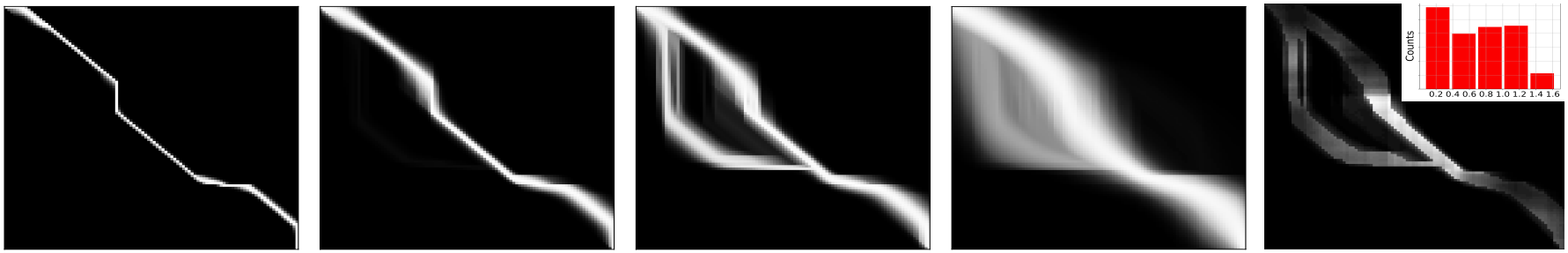
$$\underset{\{\sigma_{mn}\}_{(m,n)\in\mathbf{\Pi}_i}}{\arg\max} \prod_{(m,n)\in\mathbf{\Pi}_i} \mathcal{N}(\boldsymbol{\psi}_m; \boldsymbol{\psi}'_n, \sigma^2_{mn}) \tag{2}$$

$$= \underset{\{\sigma_{mn}\}_{(m,n)\in\mathbf{\Pi}_i}}{\arg\min} \sum_{(m,n)\in\mathbf{\Pi}_i} d' \log(\sigma) + \frac{\|\boldsymbol{\psi}_m - \boldsymbol{\psi}'_n\|_2^2}{\sigma^2_{mn}}, \tag{3}$$

where d' is the length of feature vectors $\boldsymbol{\psi}$.

Approach

Our **uncertainty-DTW** can capture 'alternative' paths:



(a) sDTW$_{\gamma=0.01}$ (b) sDTW$_{\gamma=0.1}$ (c) uDTW$_{\gamma=0.01}$ (d) uDTW$_{\gamma=0.1}$ (e) uncertainty

Figure 5: With higher γ controlling softness, in (b) & (d) more paths become 'active'. In (c) & (d), uDTW has two possible routes due to uncertainty modeling.

- Soft-DTW (plots (a) & (b)) produces single paths ('fuzziness is due to soft-maximum operator selecting the best path).
- Uncertainty-DTW (plots (c) & (d)) produces alternative paths merging where the uncertainty $\sigma_{n,m}$ (plot (e)) is large.
- $\sigma_{n,m}$ is obtained from a small MLP called SigmaNet (we have observed it is better to optimize over SigmaNet parameters than directly over $\sigma_{n,m}$.

Pipeline: Supervised Few-shot Action Recognition

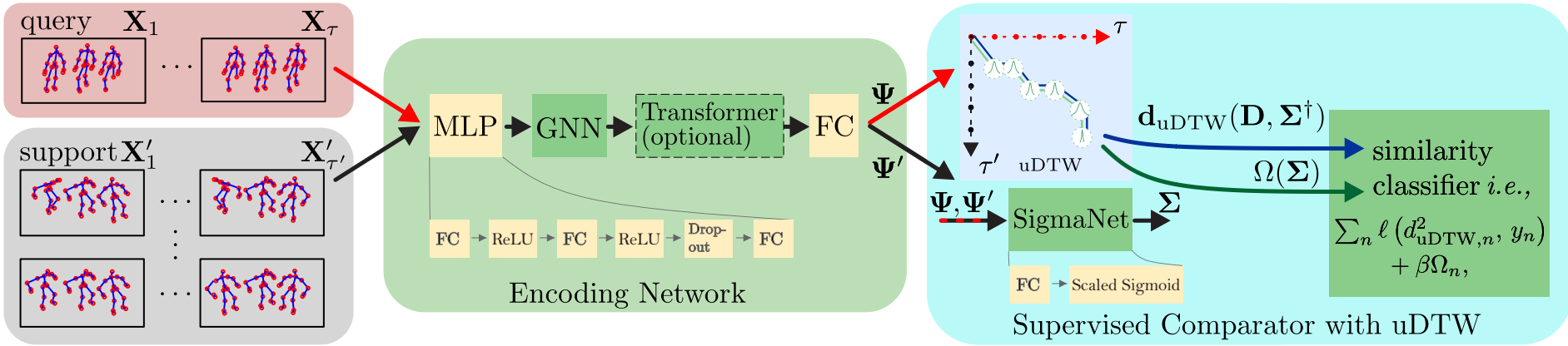


Figure 6: Supervised few-shot action recognition with the uncertainty-DTW (uDTW).

Our model contains:

- Encoding Network (backbone); each sequence is split into temporal blocks.
- Comparator has access to each temporal block features $\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_\tau$ and $\boldsymbol{\psi}'_1, ..., \boldsymbol{\psi}'_{\tau'}$ of query-support pairs.
- SigmaNet produces the uncertainty variable $\mathbf{\Sigma}$
- The objective function is a trade-off between the empirical loss $\ell(\cdot)$ with uncertainty-DTW and the uncertainty penalty (regularization) $\Omega(\cdot)$.

Pipeline: Unsupervised Few-shot Action Recognition

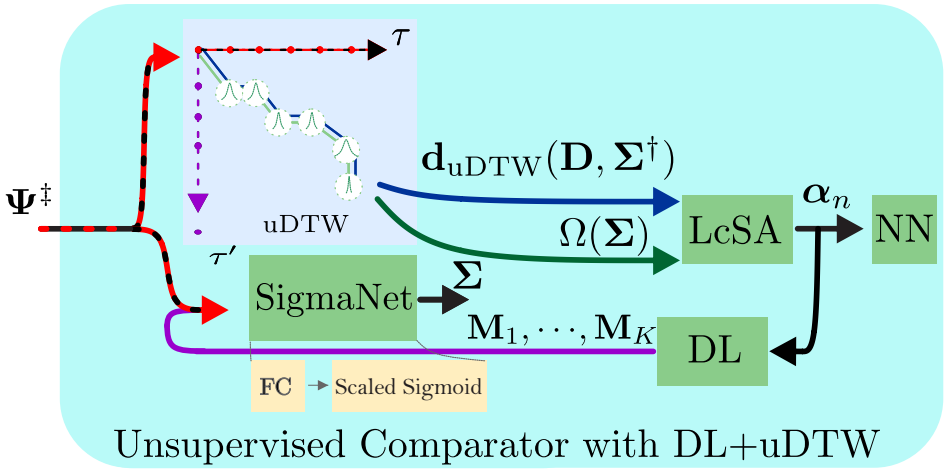


Figure 7: Unsupervised few-shot action recognition with the uncertainty-DTW (uDTW).

- We train Encoding Network (backbone) but in an unsupervised manner.
- Comparator learns a dictionary (DL) which contains 'abstract' dictionary sequences (clusters).
- LcSA is an encoder of sequences into the dictionary space.
- Interaction between LcSA encoder and dictionary can be thought as soft clustering that uses the uncertainty-DTW distance.
- At the test time, the nearest neighbor on encoded sequences is used to match support sequence (known labels) with the query (unknown label).

Pipeline: Forecasting the Evolution of Time Series

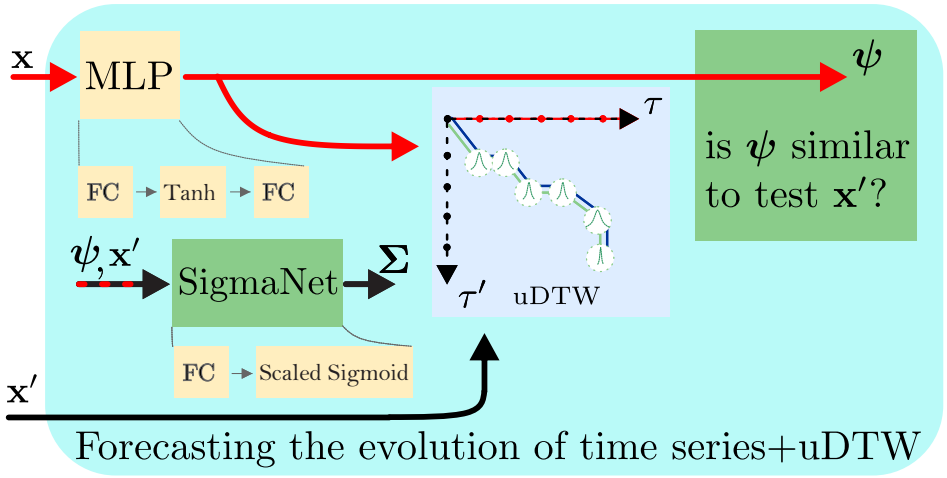
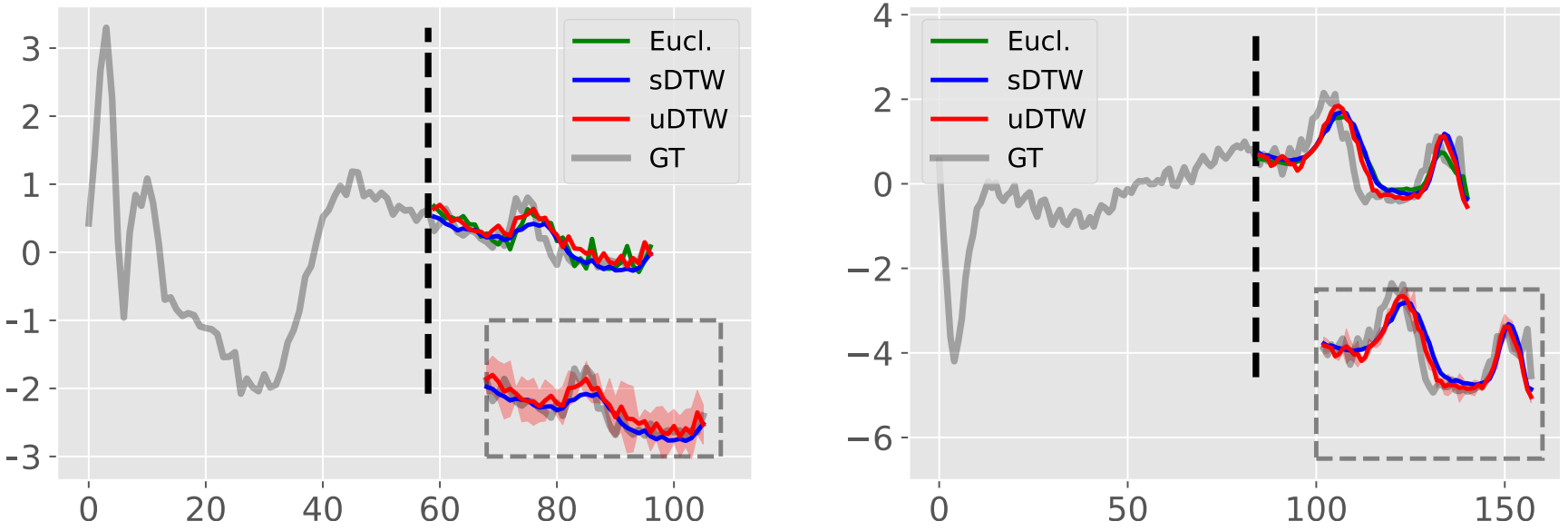


Figure 8: Predicting Evolution of Time Series.

- Variable $\mathbf{x}$ is the first half of time series, and $\mathbf{x}'$ is the second half of time series.
- MLP learns to predict $\mathbf{x}'$ with MLP+uncertainty-DTW from $\mathbf{x}$.

Results: Forecasting the Evolution of Time Series

- Given the first part of a time series, we
  - train 3 multi-layer perception (MLP) to predict the remaining part
  - use the Euclidean, sDTW or uDTW distance per MLP



(a) ECG200 (b) ECG5000

Figure 9: We use ECG200 and ECG5000 in UCR archive, and display the prediction obtained for the given test sample and the ground truth (GT). Oftentimes, we observe that uDTW helps predict the sudden changes well.

Results: Few-shot Action Recognition

For more details, results and discussions, please refer to our paper.

Table 1: Evaluations on NTU-60.

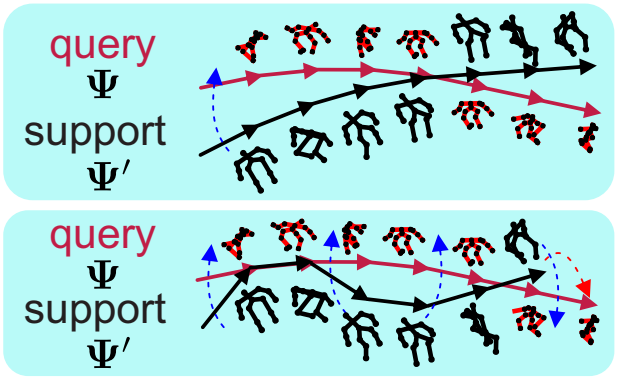
#classes	10	20	30	40	50
			Supervised		
MatchNets	46.1	48.6	53.3	56.3	58.8
ProtoNet	47.2	51.1	54.3	58.9	63.0
TAP	54.2	57.3	61.7	64.7	68.3
Euclidean	38.5	42.2	45.1	48.3	50.9
sDTW	53.7	56.2	60.0	63.9	67.8
sDTW div.	54.0	57.3	62.1	65.7	69.0
uDTW	56.9	61.2	64.8	68.3	72.4
			Unsupervised		
Euclidean	20.9	23.7	26.3	30.0	33.1
sDTW	35.6	45.2	53.3	56.7	61.7
sDTW div.	36.0	46.1	54.0	57.2	62.0
uDTW	37.0	48.3	55.3	58.0	63.3

Table 2: Evaluations on NTU-120.

#classes	20	40	60	80	100
			Supervised		
MatchNets	20.5	23.4	25.1	28.7	30.0
ProtoNet	21.7	24.0	25.9	29.2	32.1
TAP	31.2	37.7	40.9	44.5	47.3
Euclidean	18.7	21.3	24.9	27.5	30.0
sDTW	30.3	37.2	39.7	44.0	46.8
sDTW div.	30.8	38.1	40.0	44.7	47.3
uDTW	32.2	39.0	41.2	45.3	49.0
			Unsupervised		
Euclidean	13.5	16.3	20.0	24.9	26.2
sDTW	20.1	25.3	32.0	36.9	40.9
sDTW div.	20.8	26.0	33.2	37.5	42.3
uDTW	22.7	28.3	35.9	39.4	44.0

sDTW div.: Blondel *et al.*, **Differentiable divergences between time series**. *AISTATS 2021*.
TAP: Bing Su & Ji-Rong Wen, **Temporal Alignment Prediction for Supervised Representation Learning and Few-Shot Sequence Classification**, *ICLR 2022*.

Motivation (JEANIE)



Matching query-support features under varying viewpoints of 3D poses:

- (*top*) rotate a support trajectory onto a query trajectory (naive).
- (*bottom*) advanced viewpoint alignment strategy is needed: locally follow complicated non-linear paths but **assume viewpoints change smoothly in time**, *e.g.*, no large abrupt changes along the path.

To **learn similarity/dissimilarity** between pairs of query-support sequences:

- find a smooth joint viewpoint-temporal alignment.
- minimize/maximize d_{JEANIE} for same/different support-query labels.

A viewpoint invariant distance can be defined as:

$$d_{\text{inv}}(\mathbf{\Psi}, \mathbf{\Psi}') = \inf_{\gamma,\gamma' \in T} d\big(\gamma(\mathbf{\Psi}), \gamma'(\mathbf{\Psi}')\big), \tag{4}$$

- T is a set of transformations required to achieve a viewpoint invariance.
- T may include 3D rotations to rotate one trajectory onto the other (or each 3D pose onto the corresponding 3D pose).
- Such global viewpoint alignment of two sequences or local alignment of 3D poses are **suboptimal**. T may realise better transformation strategies...

Thus, we propose a FSAR approach that learns on skeleton-based 3D body joints by **Joint tEmporal and cAmera viewpoiNt alIgnmEnt (JEANIE)**.

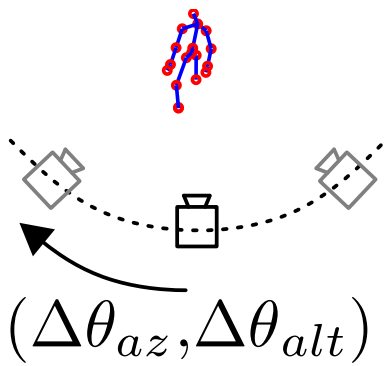
JEANIE

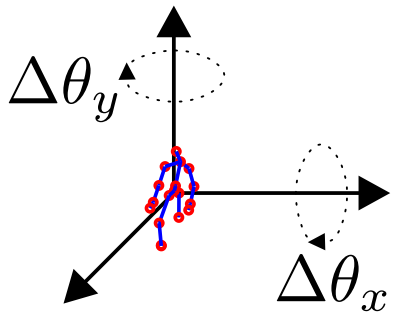
Sequences that are being matched might have been captured under different camera viewpoints or subjects might have followed different trajectories.
Thus, to model 3D pose variations, we:

- exploit the **projective camera geometry**.
- propose **the smooth path** in DTW should **simultaneously perform temporal & viewpoint alignment**

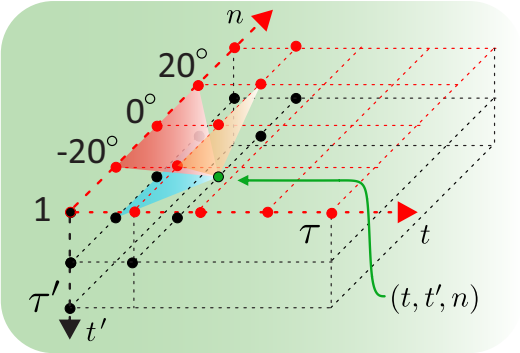
JEANIE has the transportation plan $\mathcal{A}'$where apart of steps $\downarrow$, $\searrow$, $\rightarrow$ for temporal axes (indicated as τ and τ'), JEANIE can also take **additional steps on the viewpoint axis, e.g., step inward, inward-down, etc.**.
Thus, apart from temporal block counts τ (query) & τ' (support), for query sequences we simulate $K=2\eta_{az}+1$, $K'=2\eta_{alt}+1$ camera viewpoints (or Euler angles). We have:

- possible η_{az} left and η_{az} right steps from the **initial camera azimuth**,
- and η_{alt} up and η_{alt} down steps from the **initial camera altitude**.





JEANIE (cont.)



JEANIE is given as:

$$d_{\text{JEANIE}}(\boldsymbol{\Psi}, \boldsymbol{\Psi}') = \underset{\mathbf{A} \in \mathcal{A}'}{\text{SoftMin}_\gamma} \left\langle \mathbf{A}, \boldsymbol{\mathcal{D}}(\boldsymbol{\Psi}, \boldsymbol{\Psi}') \right\rangle, \tag{5}$$

where $\boldsymbol{\mathcal{D}} \in \mathbb{R}_+^{K \times K' \times \tau \times \tau'} \equiv \left[d_{\text{base}}(\boldsymbol{\psi}_{m,k,k'}, \boldsymbol{\psi}'_n) \right]_{\substack{(m,n) \in \mathcal{I}_\tau \times \mathcal{I}_{\tau'} \\ (k,k') \in \mathcal{I}_K \times \mathcal{I}_{K'}}}$.

Algorithm 1 Joint tEmporal and cAmera viewpoiNt alIgnmEnt (JEANIE).

Input (forward pass): $\boldsymbol{\Psi}, \boldsymbol{\Psi}'$, $\gamma > 0$, $d_{\text{base}}(\cdot,\cdot)$, ι-max shift.

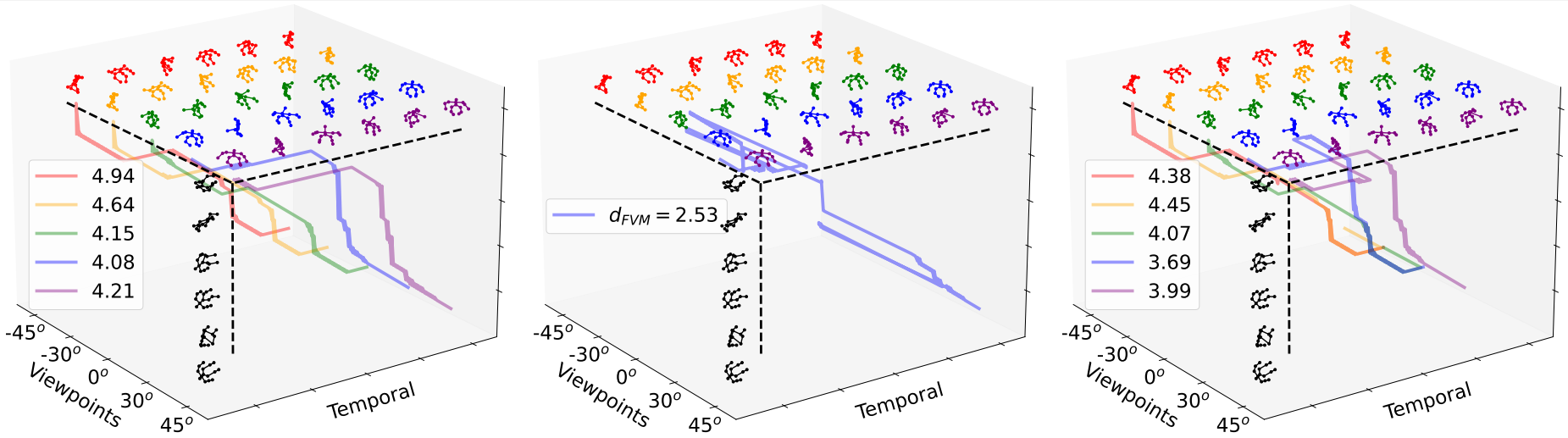
1: $r_{:,:,:} = \infty$, $r_{n,1,1} = d_{\text{base}}(\boldsymbol{\psi}_{n,1}, \boldsymbol{\psi}'_1)$, $\forall n \in \{-\eta, ..., \eta\}$
2: $\Pi \equiv \{-\iota, ..., 0, ..., \iota\} \times \{(0,1), (1,0), (1,1)\}$
3: **for** $t \in \mathcal{I}_\tau$:
4: **for** $t' \in \mathcal{I}_{\tau'}$:
5: **if** $t \neq 1$ or $t' \neq 1$:
6: **for** $n \in \{-\eta, ..., \eta\}$:
7: $r_{n,t,t'} = d_{\text{base}}(\boldsymbol{\psi}_{n,t}, \boldsymbol{\psi}'_{t'}) + \text{SoftMin}_\gamma \left([r_{n-i,t-j,t'-k}]_{(i,j,k) \in \Pi} \right)$

Output: $\text{SoftMin}_\gamma \left([r_{n,\tau,\tau'}]_{n \in \{-\eta,...,\eta\}} \right)$

- Suppose we tackle the camera viewpoint alignment in just camera azimuth.
- The maximum viewpoint change from block to block is ι-max shift (smoothness).

- We initialize all possible origins of shifts in accumulator $r_{n,1,1}$.
- A phase related to soft-DTW (temporal-viewpoint alignment) takes place.
- We choose the path with the smallest distance (of matched features) over all possible viewpoint ends by selecting a soft-minimum over $[r_{n,\tau,\tau'}]_{n \in \{-\eta,...,\eta\}}$.

View-wise Soft-DTW *vs.* FVM *vs.* JEANIE



(a) soft-DTW (view-wise) (b) FVM (c) JEANIE(1-max shift)

Figure 10: The support & query sequence are shown in green & black respectively.

- soft-DTW finds each individual alignment **per viewpoint fixed** throughout alignment: $d_{\text{shortest}} = 4.08$. Too pessimistic!
- FVM is a **greedy matching algorithm** which leads to unrealistic zigzag path: $d_{\text{FVM}} = 2.53$. Overoptimistic!
- JEANIE (1-max shift) is able to find **smooth joint viewpoint-temporal alignment** between support and query sequences: $d_{\text{JEANIE}} = 3.69$.

Free Viewpoint Matching (FVM) **seeks the best local viewpoint alignment** **for every step of DTW**, thus resulting in a **non-smooth path along viewpoint axis** in contrast to JEANIE.

Pipeline: further details

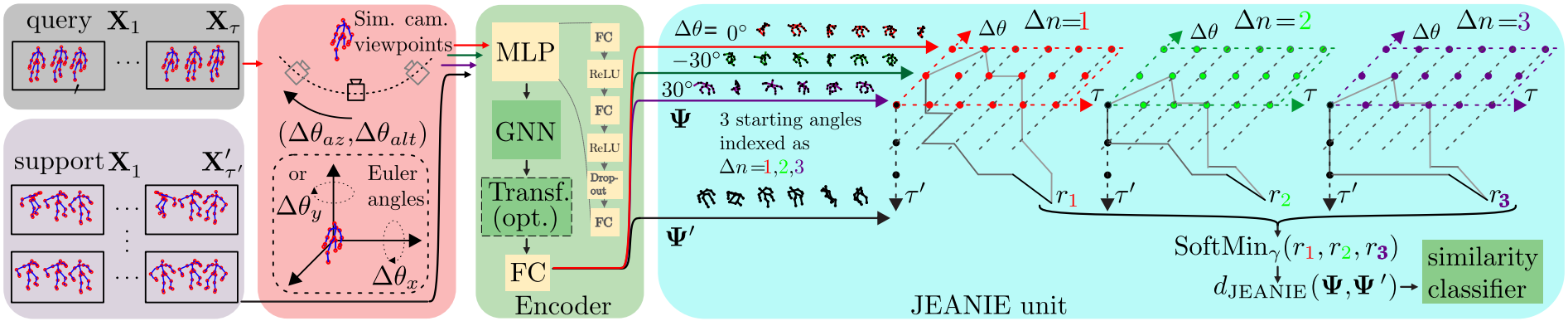
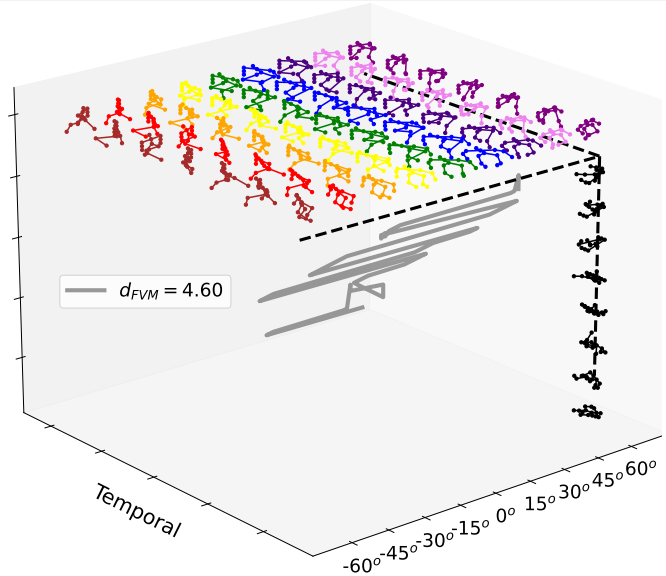


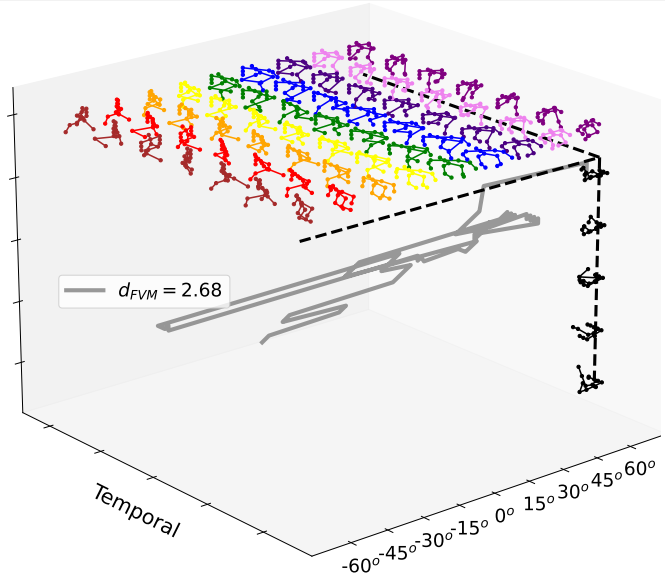
Figure 11: Our 3D skeleton-based FSAR with JEANIE.

- Generate multiple rotations by $(\Delta\theta_x, \Delta\theta_y)$ of each query by
  - **Euler angles** (baseline approach) or
  - **simulated camera views** (gray cameras) by camera shifts $(\Delta\theta_{az}, \Delta\theta_{alt})$.
- Temporal-viewpoint alignment takes place in 4D space (we show a 3D case).
- **Temporally-wise**, JEANIE starts from the same $t=(1,1)$ & finishes at $t=(\tau,\tau')$.
- **Viewpoint-wise**, JEANIE starts from **every possible camera shift** & finishes at one of possible camera shifts.
- At each step, the step may be no larger than $(\pm\Delta\theta_{az}, \pm\Delta\theta_{alt})$ to prevent erroneous alignments.

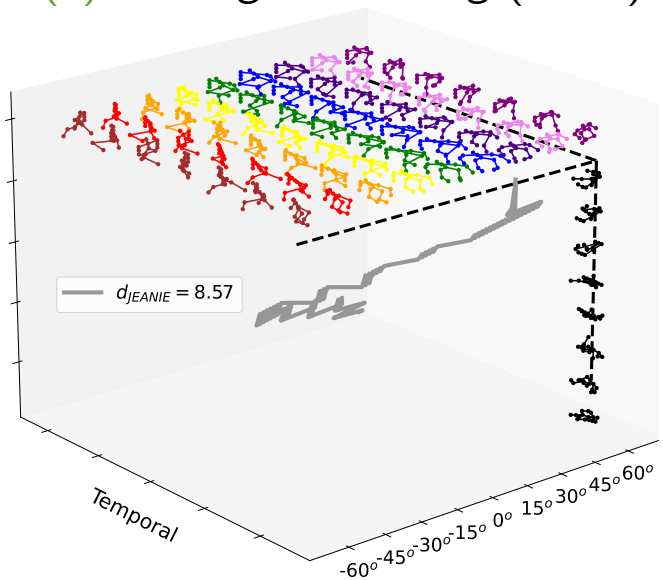
Results & Discussions



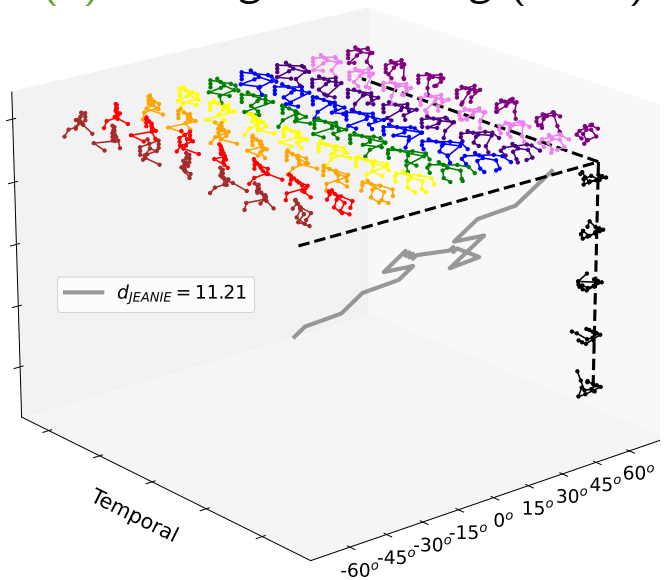
(a) *walking vs. walking* (**FVM**)



(b) *walking vs. running* (**FVM**)



(c) *walking vs. walking* (**JEANIE**)



(d) *walking vs. running* (**JEANIE**)

Results & Discussions (cont.)

Table 3: Results on NTU-120 (multiview classification).

Training view / Testing view	bott. cent.	bott. top	bott.& cent. top	left cent.	left right	left & cent. right
100/same 100 (baseline)	74.2	73.8	75.0	58.3	57.2	68.9
100/same 100 (*FVM*)	79.9	78.2	80.0	65.9	63.9	75.0
100/same 100 (*JEANIE*)	**81.5**	**79.2**	**83.9**	**67.7**	**66.9**	**79.2**
100/novel 20 (baseline)	58.2	58.2	61.3	51.3	47.2	53.7
100/novel 20 (*FVM*)	66.0	65.3	68.2	58.8	53.9	60.1
100/novel 20 (*JEANIE*)	**67.8**	**65.8**	**70.8**	**59.5**	**55.0**	**62.7**

Table 4: Experiments on 2D and 3D Kinetics-skeleton.

	S^2GC (no soft-DTW)	soft-DTW	*FVM*	JEANIE	JEANIE +Transf.
2D skel.	32.8	34.7	-	-	-
3D skel.	35.9	39.6	44.1	**50.3**	**52.5**

Discussion.

- Few-shot multi-view classification.
  - Adding more camera viewpoints helps.
  - Even with (*novel 20*) (not used in training), we still achieve 62.7% & 70.8%.
- JEANIE on the Kinetics-skeleton dataset.
  - We use Euler angles.
  - 3D outperforms 2D by 3–4%.
  - With Transformer, JEANIE further boosts results by 2%.

Few-shot Keypoint Detection with Uncertainty Learning for Unseen Species

Motivation

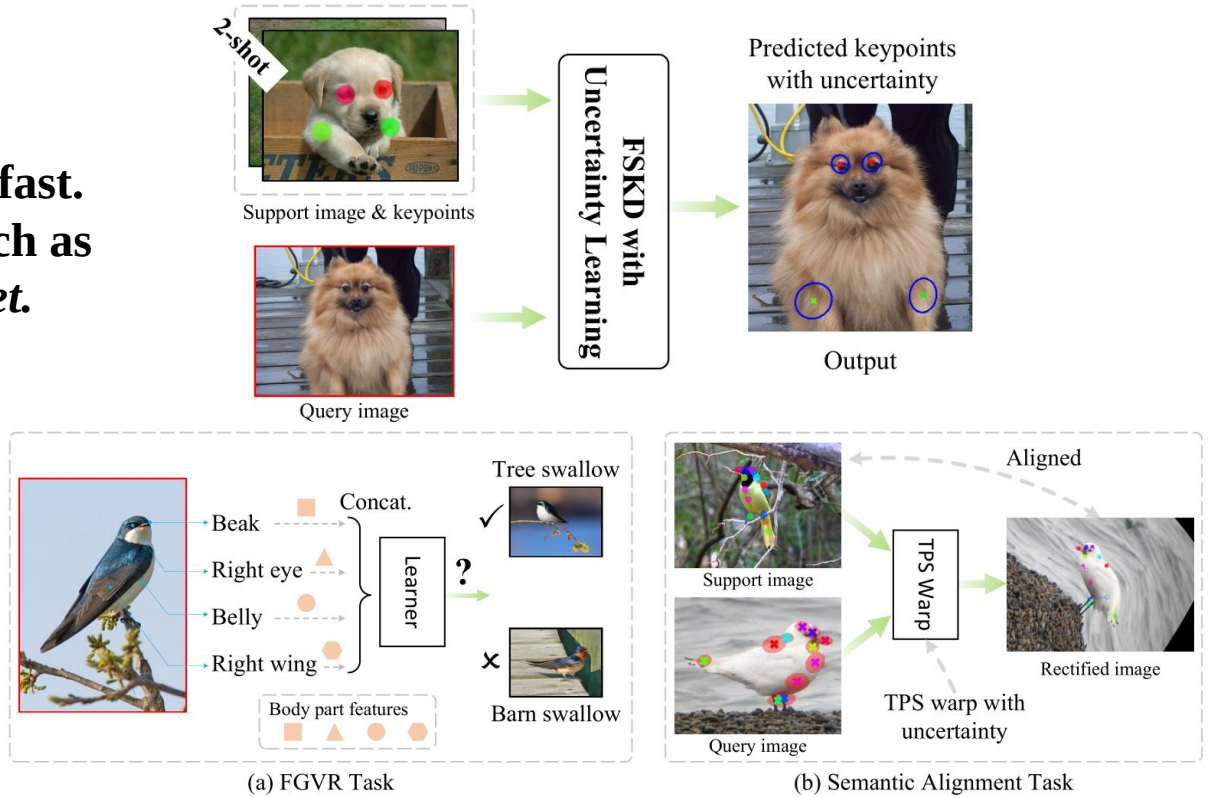
- **Humans learn to recognize/generalize keypoints fast.**
- **We are inspired by *few-shot learning methods* such as *RelationNet, ProtoNet, Matching Net., Siamese Net.***

Applications

- Fine-Grained Visual Recognition (FGVR).
- Semantic Alignment (SA).
- Semi-automatic labelling.
- Animal behavior analysis.

Challenges

- Generalizing based on a few of samples is hard.
- Large amounts of interfering noise and similar local patterns affect keypoint detection.
- Inherent keypoint uncertainty, existed in GTs/Predictions.



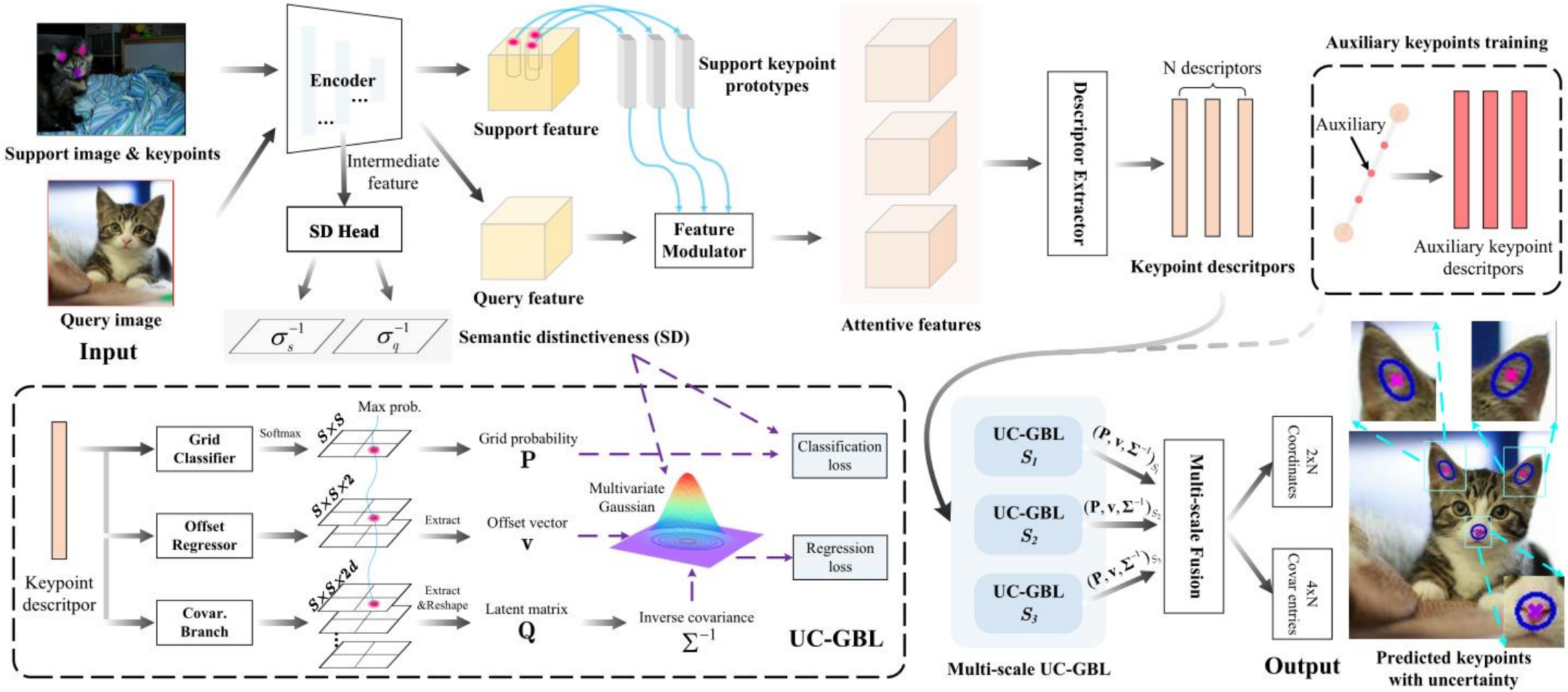
Contributions

- A flexible few-shot keypoint detector (FSKD) is proposed.
- We model both localization and semantic uncertainty within our localization networks (UC-GBL, muti-scale UC-GBL).
- We employ low-quality auxiliary keypoints during learning.
- Our FSKD model can successfully detect novel keypoints, and be applied to *FGVR* and *SA*.

Proposed Approach

Problem Definition & Pipeline

Given *N support keypoints* and *K support images*, a problem of detecting the corresponding keypoints in a query image is dubbed as *N-way K-shot* detection problem.



FSKD Experiments

Base and novel keypoint splits

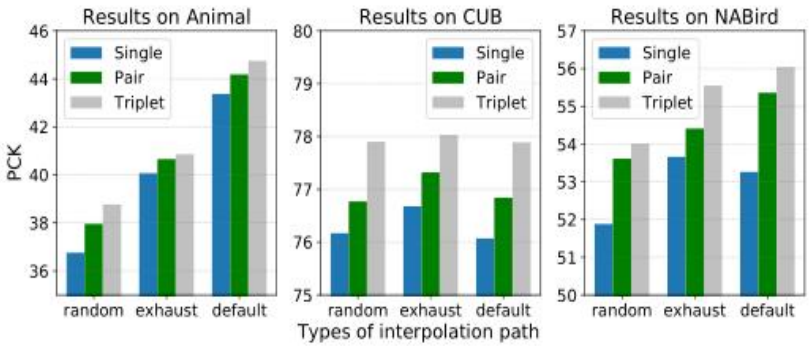
Dataset	Base Keypoint Set	Novel Keypoint Set
Animal	*two ears, nose, four legs, four paws*	*two eyes, four knees*
CUB	*beak, belly, back, breast, crown, two legs, nape, throat, tail*	*forehead, two eyes, two wings*
NABird	*beak, belly, back, breast, crown, nape, tail*	*two eyes, two wings*

One-shot novel keypoint detection

Method	Animal Pose Dataset						CUB	NABird
	Cat	Dog	Cow	Horse	Sheep	Avg		
Baseline	27.30	24.40	19.40	18.25	21.22	22.11	66.12	39.14
ProbIntr	28.54	23.20	19.55	17.94	17.03	21.25	68.07	48.70
TFA	19.40	20.00	20.85	17.99	19.54	19.56	50.12	30.16
ProtoNet	19.68	16.18	14.39	12.05	15.06	15.47	51.32	36.65
RelationNet	22.15	17.19	15.47	13.58	16.55	16.99	56.59	34.02
WG (w/o Att.)	21.86	17.11	16.19	16.34	16.13	17.53	52.66	33.31
WG	22.47	19.39	16.82	16.40	16.94	18.40	54.75	34.19
FSKD (rand) (Ours)	46.05	40.66	37.55	38.09	31.50	38.77	77.90	54.01
FSKD (default) (Ours)	52.36	47.94	44.07	42.77	36.60	44.75	77.89	56.04

PCK scores (the higher, the better)

FSKD with	Animal	CUB	NABird	DeepF.2	AwA
ResNet50	44.75	77.89	56.04	33.04	**64.76** (*27.75)
HRNet-W32	47.61	78.24	56.89	33.67	**70.99**
HRNet-W48	**48.81**	**79.45**	**57.11**	**34.29**	**72.20**



Downstream Tasks

Few-shot Fine Grained Recognition

- Pose normalization, i.e., using the concatenation of body part features for FGVR.
- We build UKPs (Universal Keypoint Prototypes) during training, thus in testing no longer needs support keypoints as reference.

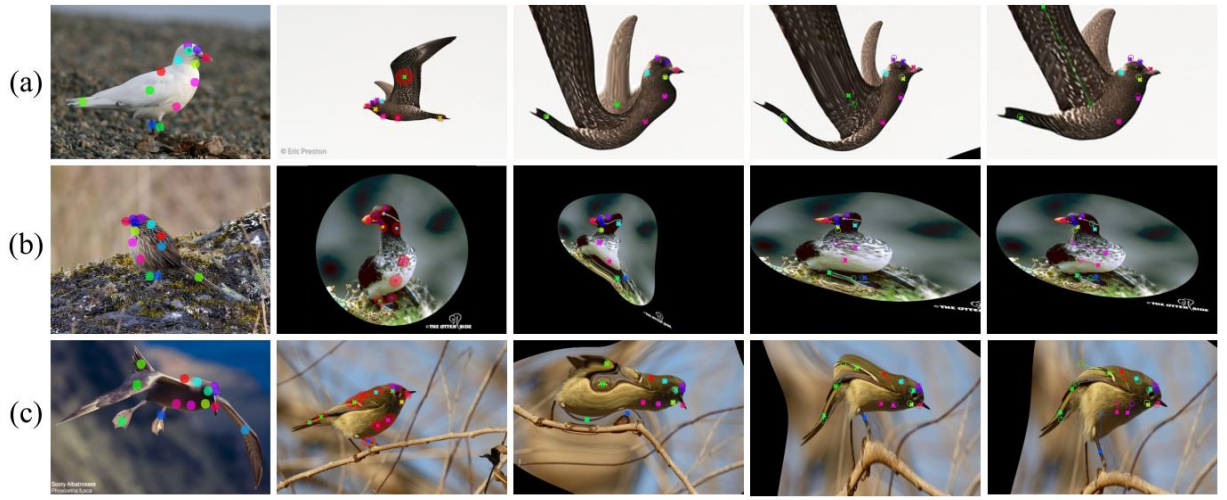
Datasets	Models		1-shot	5-shot	all-shot
CUB	Proto	[42]	23.03	38.05	41.79
	Proto+BP	[31]	18.43	33.63	38.34
	Proto+bbN	[46]	23.97	40.22	44.61
	Proto+PN	[46]	35.92	58.66	63.51
	Ours		**37.45**	**61.22**	**66.25**
	Ours+*AuxKps*		**38.04**	**61.74**	**66.37**
NABird	Proto+PN	[46]	26.17	50.55	60.03
	Ours		**27.68**	**51.81**	**61.56**

AuxKps means adding auxiliary keypoints to form augmented prototypes for classification.

Semantic Alignment

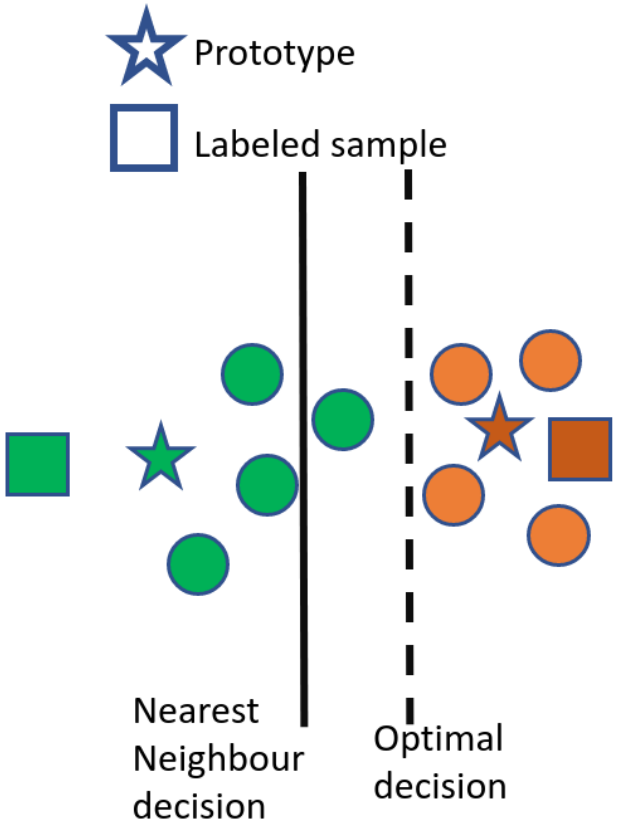
$$\mathbf{T} = \left(\begin{bmatrix} \mathbf{R} + \lambda \mathbf{D}^2 & \hat{\mathbf{P}}^\intercal \\ \hat{\mathbf{P}} & \mathbf{0}^{3\times 3} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{P}'^\intercal \\ \mathbf{0}^{3\times 2} \end{bmatrix} \right)^\intercal$$

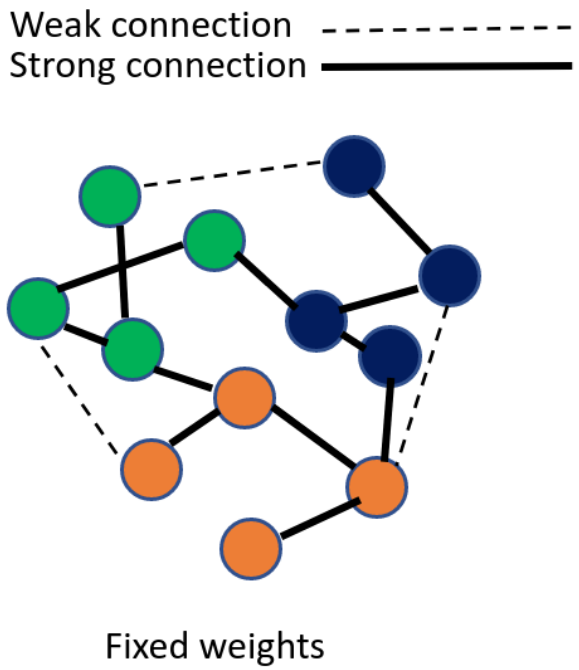
- *P is support keypoints, P' is predicted query keypoints, D is diagonal matrix of 'uncertainty strength'.*



Code: https://github.com/AlanLuSun/Few-shot-keypoint-detection

Transductive Few-shot Learning with Prototype-based Label Propagation by Iterative Graph Refinement





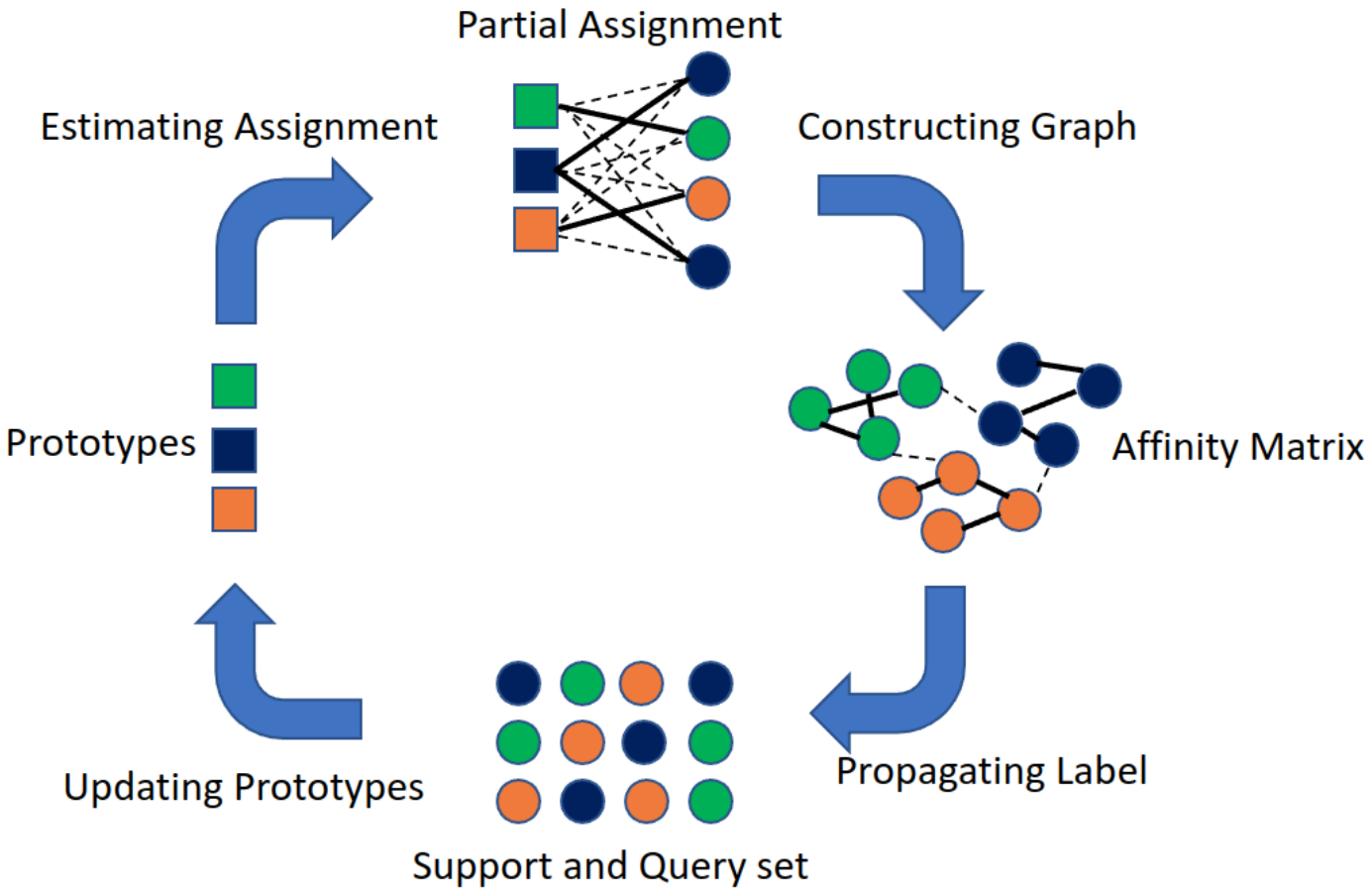
Prototype-based methods:

1. sensitive to the large within-class variance and low between-class variance
2. estimate prototypes inaccurately by the soft-label assignment alone.

Graph-based methods:

1. determined graph with noisy links
2. propagating labels based on the graph

Method: Prototypes-based Label Propagation



Algorithm 1: Prototype-based Label Propagation.

Input: $\boldsymbol{X}, \boldsymbol{Y}, \lambda, \alpha, n_{step}$

Init: $\tilde{\mathbf{c}}_k = \frac{1}{|S_k|} \sum_{(\boldsymbol{x}_i, y_i) \in S_k} \boldsymbol{x}_i, k = 0;$

while $k < n_{step}$ **do**

Estimating Assignment:

$$Z_{ij} = \frac{\exp(-\left\|\boldsymbol{x}_i - \tilde{\mathbf{c}}_j\right\|_2^2)}{\sum_{j'} exp(-\left\|\boldsymbol{x}_i - \tilde{\mathbf{c}}_{j'}\right\|_2^2)};$$

Constructing Graph:

$$\Lambda_{kk} = \sum_i Z_{ik} \text{ and } \boldsymbol{W} = \boldsymbol{Z}_t \boldsymbol{\Lambda}^{-1} \boldsymbol{Z}_t^\top;$$

Propagating Label:

$$\tilde{\mathbf{Y}} = \boldsymbol{Z}_t \left(\boldsymbol{Z}_L^\top \boldsymbol{Z}_L + \lambda \boldsymbol{Z}_t^\top (\boldsymbol{I} - \boldsymbol{W}) \boldsymbol{Z}_t \right)^{-1} \boldsymbol{Z}_t^\top \boldsymbol{Y};$$

Updating Prototypes:

$$\tilde{\mathbf{C}} \leftarrow (1 - \alpha) \tilde{\mathbf{C}} + \alpha \tilde{\boldsymbol{Y}} \boldsymbol{X};$$

$k \leftarrow k + 1$

end

return $y_i = \arg\max_j \tilde{Y}_{i,j}$

Results

Table 1. Comparison of test accuracy against state-of-the-art methods for 1-shot and 5-shot classification. (*: inference aug., §4.2.3)

			mini-ImageNet		tiered-ImageNet	
Methods	Setting	Network	1-shot	5-shot	1-shot	5-shot
MAML [8]	Inductive	ResNet-18	49.61 ± 0.92	65.72 ± 0.77	–	–
RelationNet [45]	Inductive	ResNet-18	52.48 ± 0.86	69.83 ± 0.68	–	–
MatchingNet [47]	Inductive	ResNet-18	52.91 ± 0.88	68.88 ± 0.69	–	–
ProtoNet [44]	Inductive	ResNet-18	54.16 ± 0.82	73.68 ± 0.65	–	–
TPN [29]	transductive	ResNet-12	59.46	75.64	–	–
TEAM [35]	transductive	ResNet-18	60.07	75.9	–	–
Transductive tuning [6]	Transductive	ResNet-12	62.35 ± 0.66	74.53 ± 0.54	–	–
MetaoptNet [24]	Transductive	ResNet-12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
CAN+T [11]	Transductive	ResNet-12	67.19 ± 0.55	80.64 ± 0.35	73.21 ± 0.58	84.93 ± 0.38
DSN-MR [43]	Transductive	ResNet-12	64.60 ± 0.72	79.51 ± 0.50	67.39 ± 0.82	82.85 ± 0.56
ODC* [34]	Transductive	ResNet-18	77.20 ± 0.36	87.11 ± 0.42	83.73 ± 0.36	90.46 ± 0.46
MCT* [21]	Transductive	ResNet-12	78.55 ± 0.86	86.03 ± 0.42	82.32 ± 0.81	87.36 ± 0.50
EASY* [1]	Transductive	ResNet-12	82.31 ± 0.24	88.57 ± 0.12	83.98 ± 0.24	89.26 ± 0.14
protoLP (ours)	Transductive	ResNet-12	70.77 ± 0.30	80.85 ± 0.16	84.69 ± 0.29	89.47 ± 0.15
protoLP* (ours)	Transductive	ResNet-12	84.35 ± 0.24	90.22 ± 0.11	86.27 ± 0.25	91.19 ± 0.14
protoLP (ours)	Transductive	ResNet-18	75.77 ± 0.29	84.00 ± 0.16	82.32 ± 0.27	88.09 ± 0.15
protoLP* (ours)	Transductive	ResNet-18	85.13 ± 0.24	90.45 ± 0.11	83.05 ± 0.25	88.62 ± 0.14
ProtoNet [44]	Inductive	WRN-28-10	62.60 ± 0.20	79.97 ± 0.14	–	–
MatchingNet [47]	Inductive	WRN-28-10	64.03 ± 0.20	76.32 ± 0.16	–	–
SimpleShot [50]	Inductive	WRN-28-10	65.87 ± 0.20	82.09 ± 0.14	70.90 ± 0.22	85.76 ± 0.15
S2M2-R [31]	Inductive	WRN-28-10	64.93 ± 0.18	83.18 ± 0.11	-	-
Transductive tuning [6]	Transductive	WRN-28-10	65.73 ± 0.68	78.40 ± 0.52	73.34 ± 0.71	85.50 ± 0.50
SIB [13]	Transductive	WRN-28-10	70.00 ± 0.60	79.20 ± 0.40	–	–
BD-CSPN [27]	Transductive	WRN-28-10	70.31 ± 0.93	81.89 ± 0.60	78.74 ± 0.95	86.92 ± 0.63
EPNet [38]	Transductive	WRN-28-10	70.74 ± 0.85	84.34 ± 0.53	78.50 ± 0.91	88.36 ± 0.57
LaplacianShot [61]	Transductive	WRN-28-10	74.86 ± 0.19	84.13 ± 0.14	80.18 ± 0.21	87.56 ± 0.15
ODC [34]	Transductive	WRN-28-10	80.22	88.22	84.70	91.20
iLPC [22]	Transductive	WRN-28-10	83.05 ± 0.79	88.82 ± 0.42	88.50 ± 0.75	92.46 ± 0.42
protoLP (ours)	Transductive	WRN-28-10	83.07 ± 0.25	89.04 ± 0.13	89.04 ± 0.23	92.80 ± 0.13
protoLP* (ours)	Transductive	WRN-28-10	84.32 ± 0.21	90.02 ± 0.12	89.65 ± 0.22	93.21 ± 0.13

Table 8. Test accuracy against the state of the art in the class-unbalanced setting (WRN-28-10, 1- and 5-shot protocols).

	mini-ImageNet		tiered-ImageNet	
Methods	1-shot	5-shot	1-shot	5-shot
Entropy-min	60.4	76.2	62.9	77.3
PT-MAP	60.6	66.8	65.1	71.0
LaplacianShot	68.1	83.2	73.5	86.8
TIM	69.8	81.6	75.8	85.4
BD-CSPN	70.4	82.3	75.4	85.9
α-TIM	69.8	84.8	76.0	87.8
protoLP (ours)	73.7	85.2	81.0	89.0

Table 9. Test accuracy against the state of the art in the class unbalanced setting (ResNet-12, 1-shot protocols, CUB).

CUB	unbalanced	balanced
Method	1-shot	1-shot
PT-MAP [14]	65.1	85.5
LaplacianShot [61]	73.7	78.9
BD-CSPN [27]	74.5	77.9
TIM [2]	74.8	80.3
α-TIM [46]	75.7	–
protoLP	82.22	90.13

Code: https://github.com/allenhaozhu/protoLP