Unsupervised Motion Representation Learning with Capsule Autoencoders

Ziwei Xu

ziwei-xu@comp.nus.edu.sg

Xudong Shen

xudong.shen@u.nus.edu

Yongkang Wong

yongkang.wong@nus.edu.sg

Mohan Kankanhalli

mohan@comp.nus.edu.sg

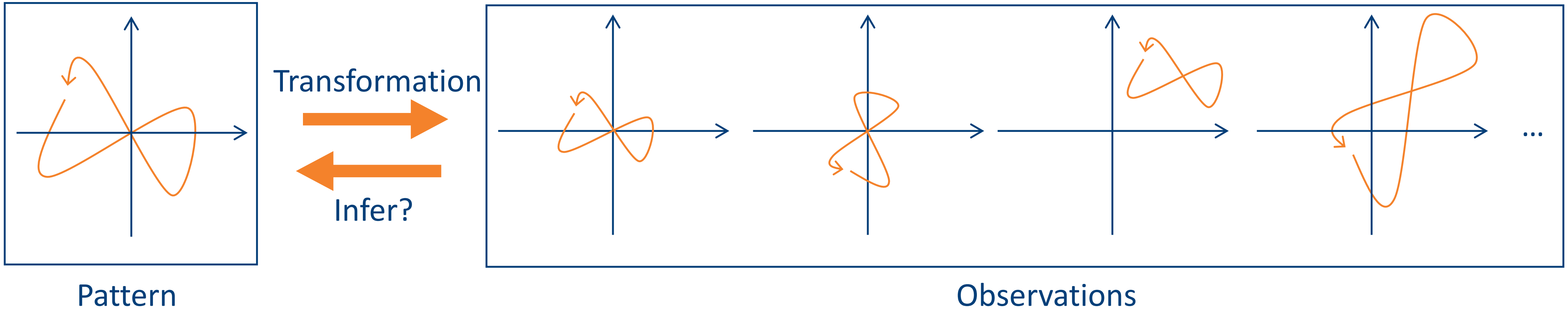
NEURAL INFORMATION PROCESSING SYSTEMS

-CRiPT

NUS National University of Singapore

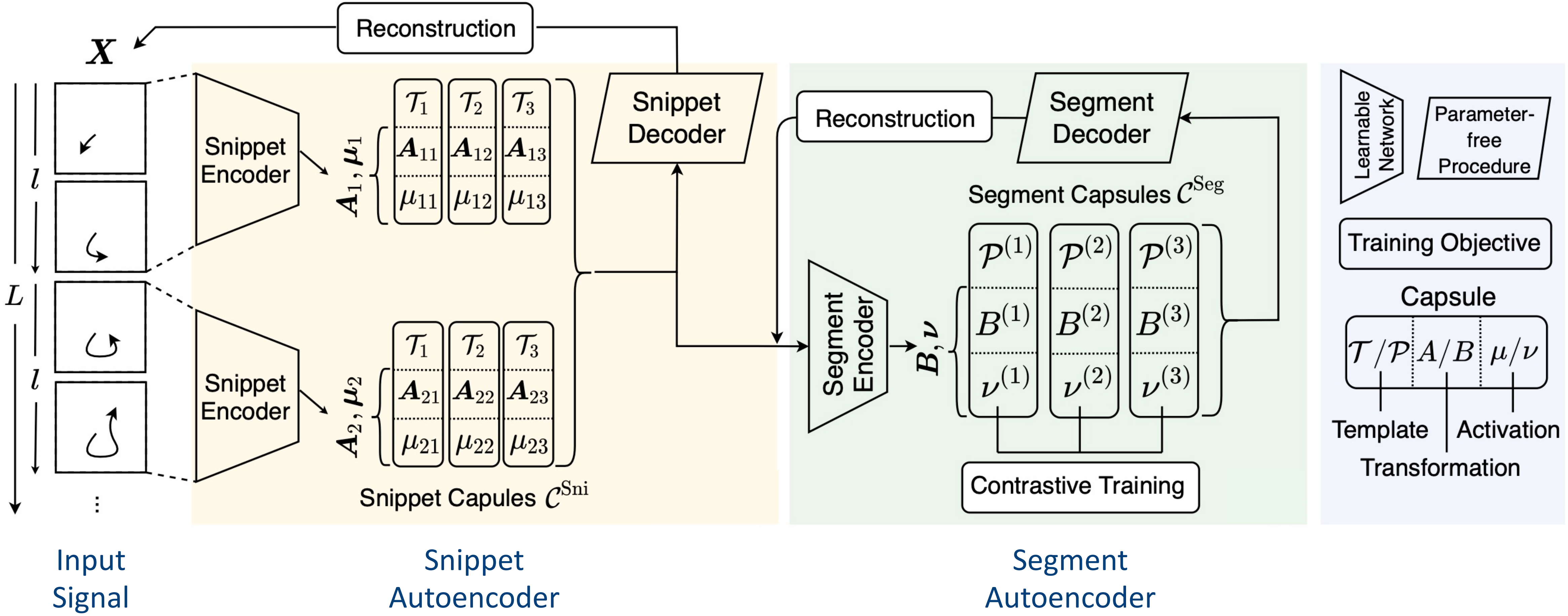
Problem Formulation

- A trajectory described as $X = \{x_i | i = 1, \ldots, L\}, x_i \in \mathbb{R}^d$
- The trajectory belongs to one motion pattern
  - Subject to an arbitrary and unknown transformation
- Given sufficient Xs, can we infer their patterns without supervision?
  - Key: separation of identity and transformation



Method – Framework

MCAE: Motion Capsule Autoencoder



Method – Snippet Autoencoder

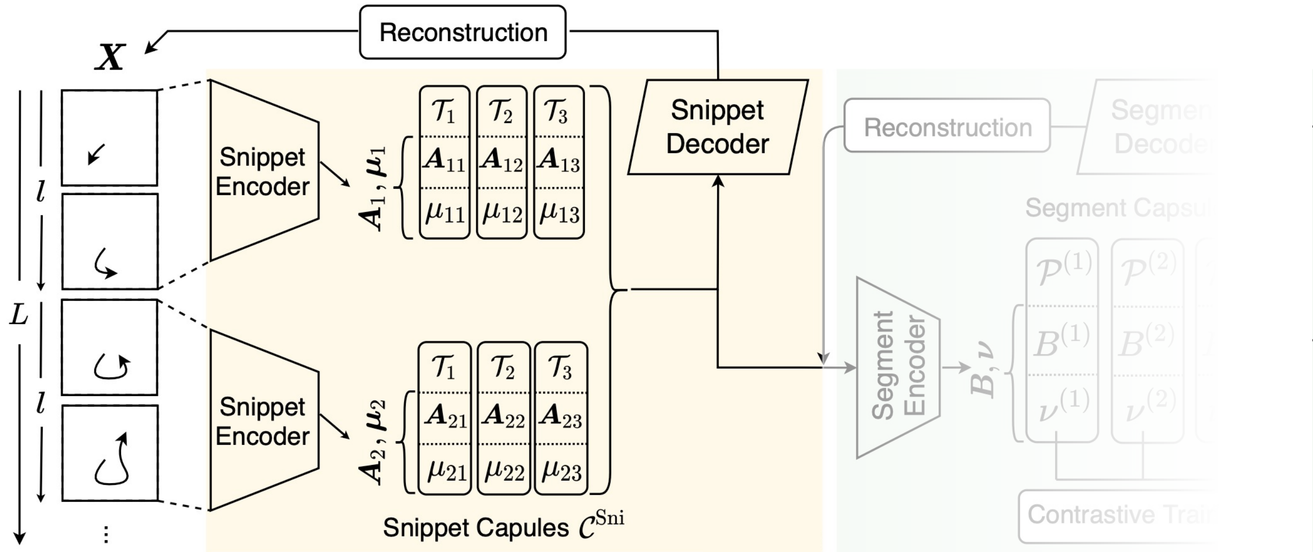
- SniCap = trainable template $\mathcal{T}$ + data-dependent parameters (A, μ)
- Divide input X into l-long snippets
- For each snippet, a snippet encoder predicts (A, μ)
- (A, μ) + template $\mathcal{T}$: reconstructs input snippets

Decoding:

$$\mathcal{T} = \{\boldsymbol{t}_i | \boldsymbol{t}_i \in \mathbb{R}^d, i = 1, ..., l\}$$

$$\begin{pmatrix} \hat{\boldsymbol{t}}_{ij} \\ 1 \end{pmatrix} = \boldsymbol{A}_i \begin{pmatrix} \boldsymbol{t}_j \\ 1 \end{pmatrix}, \quad i = 1, \dots, N, \quad j = 1, \dots, l$$

$$\hat{\boldsymbol{x}}_j = \sum_{i=1}^{N} \mu_i \hat{\boldsymbol{t}}_{ij}, \quad j = 1, \dots, l$$



Method – Segment Autoencoder

- SegCap = trainable template $\mathcal{P}$ + data-dependent parameters (B, ν)
- Template $\mathcal{P}$ defines the relation between a SegCap and $\mathcal{T}$
- For input snippet parameters, a segment encoder predicts (B, ν)
- (B, ν) + $\mathcal{P}$: reconstructs input snippet parameters

Encoding:

Flattened version of (A, μ)

$$\boldsymbol{h} = f_{\mathtt{LSTM}}\Big(\big[\boxed{\mathcal{C}_1^{\mathtt{Sni}}}, \ldots, \mathcal{C}_S^{\mathtt{Sni}}\big]\Big),$$

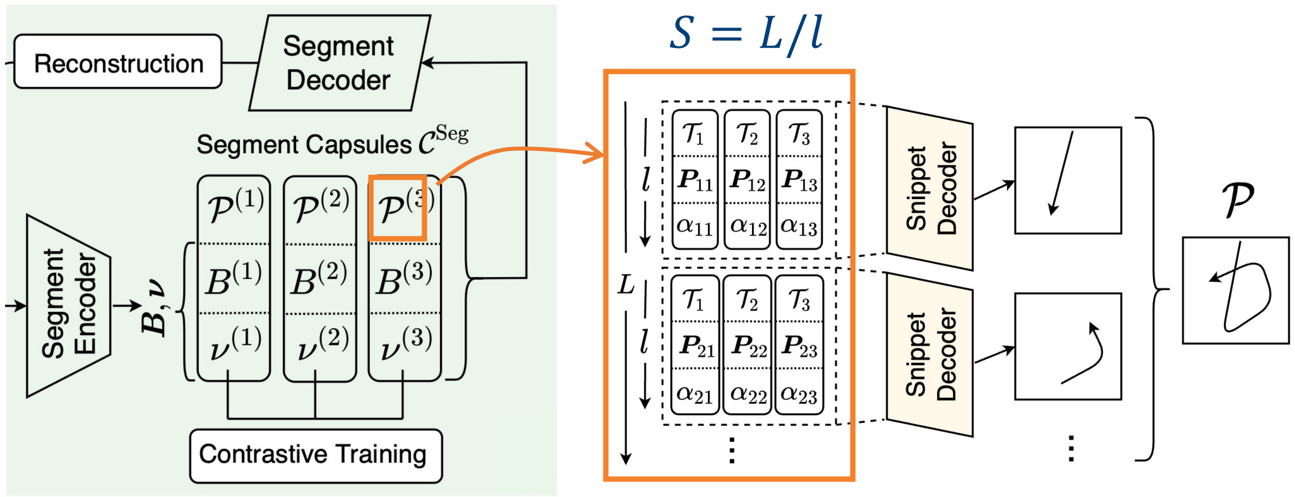
$$\{\boldsymbol{B}^{(k)}, \nu^{(k)}\} = f_{\mathtt{FC}}^{(k)}(\boxed{\boldsymbol{T}}, \boldsymbol{h}), \quad k = 1, \ldots, M,$$

Concatenated snippet templates

Decoding:

$$\hat{\boldsymbol{P}}_{ij}^{(k)} = \boldsymbol{B}^{(k)} \times \boldsymbol{P}_{ij}^{(k)}, \quad i = 1, \ldots, S, \quad j = 1, \ldots, N, \quad k = 1, \ldots, M,$$

$$\hat{\mathcal{C}}_i^{\mathtt{Sni}} = (\hat{\boldsymbol{A}}_i, \hat{\boldsymbol{\mu}}_i) = \Big(\sum_{k=1}^{M} \nu^{(k)} \hat{\boldsymbol{P}}_i^{(k)}, \sum_{k=1}^{M} \nu^{(k)} \boldsymbol{\alpha}_i^{(k)}\Big), \quad i = 1, \ldots, S,$$



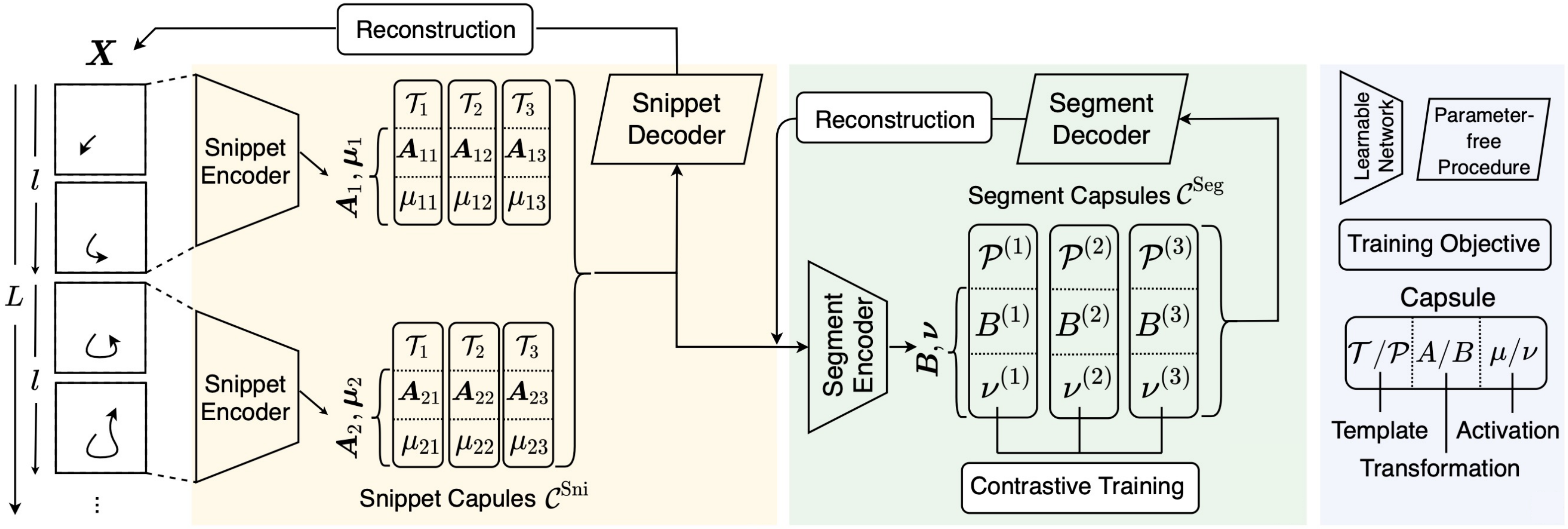
Method – Training

$$\mathcal{L} = \underbrace{\lambda^{\mathtt{Sni}}\mathcal{L}^{\mathtt{Sni}}_{\mathtt{Rec}} + \lambda^{\mathtt{Seg}}\mathcal{L}^{\mathtt{Seg}}_{\mathtt{Rec}}} + \underbrace{\mathcal{L}^{\mathtt{Seg}}_{\mathtt{Con}}} + \underbrace{0.5\mathcal{L}^{\mathtt{Reg}}_{\mathtt{Smt}} + 0.05\mathcal{L}^{\mathtt{Reg}}_{\mathtt{Sps}}}$$

$$\mathcal{L}^{\mathtt{Reg}}_{\mathtt{Smt}} = \sum_{i=2}^{L} ||\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_{i-1}||_2^2$$

$$\mathcal{L}^{\mathtt{Sni}}_{\mathtt{Rec}} = \sum_{i=1}^{L} ||(\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i)||_2^2$$

$$\mathcal{L}^{\mathtt{Seg}}_{\mathtt{Rec}} = \sum_{i=1}^{S} ||(\hat{\boldsymbol{A}}_i - \boldsymbol{A}_i)||_2^2 + ||(\hat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)||_2^2$$



$$\mathcal{L}^{\mathtt{Seg}}_{\mathtt{Con}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp\left(\mathtt{cossim}(\boldsymbol{\nu}'_i, \boldsymbol{\nu}''_i)/\tau\right)}{\sum_{j=1}^{B} \exp\left(\mathtt{cossim}(\boldsymbol{\nu}'_i, \boldsymbol{\nu}''_j)/\tau\right)}$$

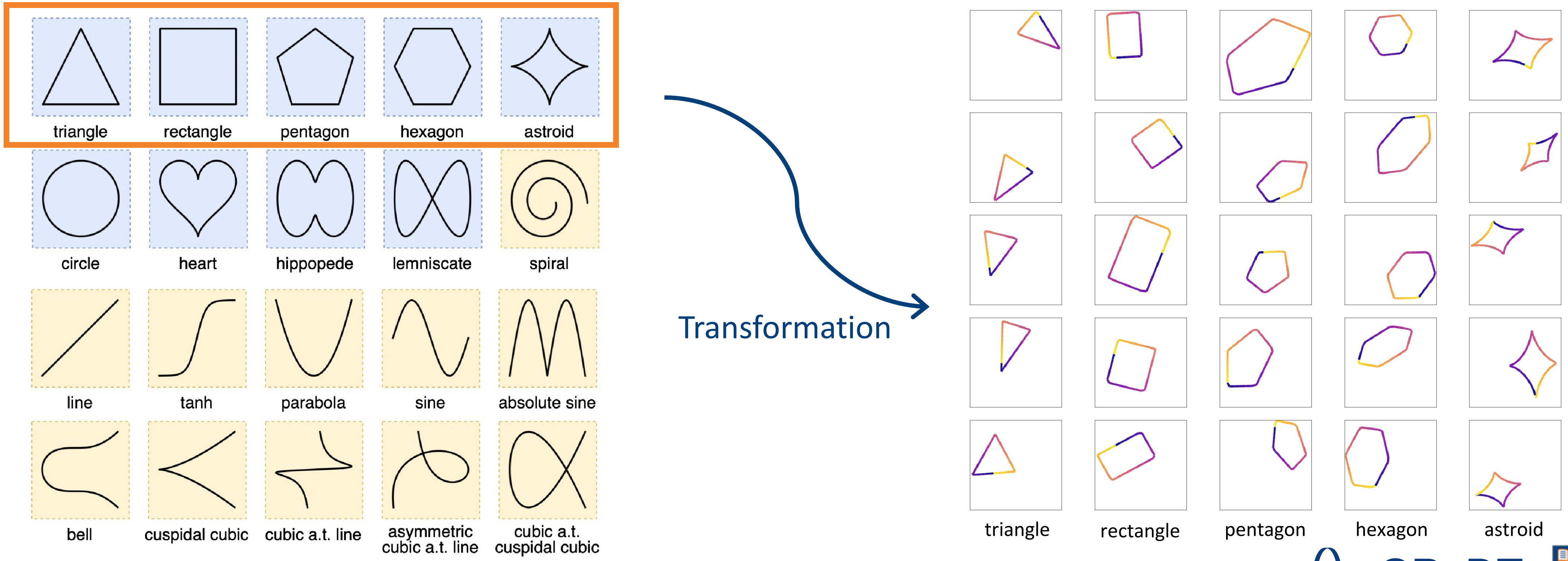
$$\mathcal{L}^{\mathtt{Reg}}_{\mathtt{Sps}} = ||\boldsymbol{\nu}||_2^2$$

Experiment

- **Questions to answer:**
  - What are the effects of hyperparameters?
  - How effective and efficient is MCAE?
  - What does MCAE learn?
  - Does MCAE work well for real-world systems?
- **Datasets:**
  - Trajectory20
  - NWUCLA, NTU-RGBD 60/120

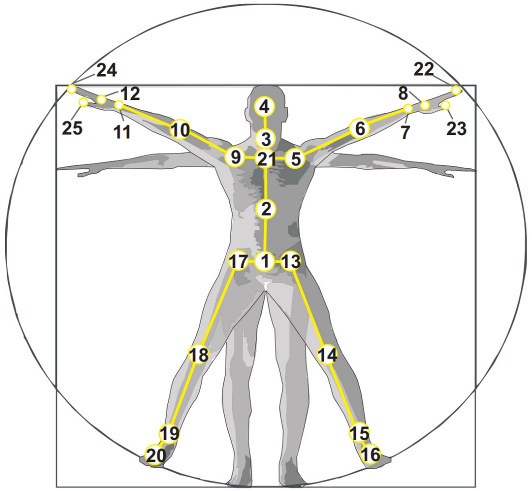
Experiment – Datasets (1)

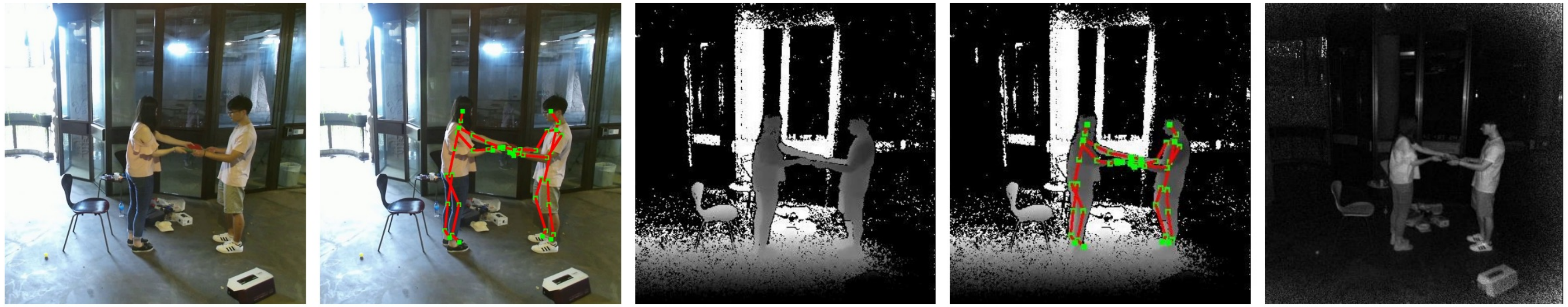
- **Trajectory20 (T20)**
  - Twenty motion patterns in 2D space spanning 32 timesteps
  - Transformation: rotation, scaling, translation, initial and end points



Experiment – Datasets (2)

- **Skeleton-based action recognition datasets:**
  - NW-UCLA, NTU-RGBD 60/120
  - Multi-point motion system in 3D space





NEURAL INFORMATION PROCESSING SYSTEMS

Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, Alex C. Kot, "NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding", TPAMI, 2019.

NUS National University of Singapore

Experiment – Learning Motion on T20

Table 1: Ablation study on T20.

Reg.	l	#Sni	#Seg	Acc. (%)
Full	8	8	80	69.30 ± 0.76
	4	8	80	41.01 ± 8.81
	16	8	80	45.83 ± 8.36
	8	2	80	64.02 ± 2.10
	8	4	80	68.17 ± 0.36
	8	16	80	48.11 ± 1.60
	8	8	32	42.36 ± 3.15
	8	8	64	63.94 ± 1.41
	8	8	128	69.44 ± 1.69
w/o $\mathcal{L}^{\mathtt{Reg}}_{\mathtt{Smt}}$	8	8	80	67.60 ± 1.69
w/o $\mathcal{L}^{\mathtt{Reg}}_{\mathtt{Sps}}$	8	8	80	65.92 ± 1.63

l: length of snippets

#Sni: number of snippet capsules

#Seg: number of segment capsules

Table 2: Unsupervised learning performance of MCAE and baselines on T20.

	Hidden Param.	#Param.	Acc. (%)
KMeans	–	–	8.57 ± 0.04
DTW-KMeans	–	–	9.12 ± 0.20
k-Shape [31]	–	–	12.94 ± 0.34
LSTM	128	600k	29.17 ± 2.45
	256	669k	40.03 ± 0.57
	512	805k	45.59 ± 1.37
	1,024	1,078k	53.47 ± 1.52
	2,048	1,625k	54.32 ± 0.55
1D-Conv	128	588k	44.78 ± 0.57
	256	787k	53.69 ± 0.53
	512	1,185k	57.57 ± 0.56
	1,024	1,982k	57.58 ± 0.08
	(#Sni, #Seg)	#Param.	Acc. (%)
MCAE	(8, 80)	**277k**	**69.30** ± **0.76**

Experiment – Look into Capsules (1)

- Visualization of snippet/segment templates
  - Snippet templates: lines and "hooks"
  - Segment templates: higher resemblance with patterns in T20

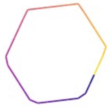
(a) Snippet templates $\mathcal{T}$. (b) Samples of segment templates $\mathcal{P}$.

Figure 4: Templates learned from Trajectory20 dataset. Color indicates time.

Experiment – Look into Capsules (2)

- Changes of segment parameters in response to input perturbation
  - Translate input by $(\Delta x, \Delta y)$
  - Check the translation component of segment parameters B.

Table 5: Top-5 segment templates (sorted by segment activation ν then segment ID for better visualization), and the translation (x, y) calculated from their parameters $\boldsymbol{B}$.

	$(\Delta x, \Delta y)$ = (-0.2, 0)			$(\Delta x, \Delta y)$ = (-0.1, 0)			$(\Delta x, \Delta y)$ = (0, 0)			$(\Delta x, \Delta y)$ = (0, 0.1)			$(\Delta x, \Delta y)$ = (0, 0.2)		
Input	ID	x	y	ID	x	y	ID	x	y	ID	x	y	ID	x	y
hexagon 	**2**	0.05	0.18	**2**	0.17	0.19	**2**	0.27	0.19	**2**	0.28	0.28	**2**	0.27	0.37
	8	0.01	-0.07	**8**	0.09	-0.06	**8**	0.18	-0.04	**8**	0.19	0.04	**8**	0.19	0.12
	12	-0.09	0.13	**12**	0.00	0.13	**12**	0.09	0.13	**12**	0.09	0.23	**12**	0.09	0.32
	37	0.10	-0.11	**37**	0.18	-0.11	**37**	0.27	-0.11	**37**	0.27	-0.03	**37**	0.27	0.05
	66	-0.12	0.16	**66**	-0.03	0.16	**66**	0.05	0.17	**66**	0.06	0.26	**66**	0.06	0.35
abs_sine 	**2**	0.04	0.2	**2**	0.14	0.19	**2**	0.24	0.19	**2**	0.24	0.28	**2**	0.23	0.38
	5	-0.01	0.30	**5**	0.07	0.29	**5**	0.16	0.29	**5**	0.16	0.38	**5**	0.15	0.46
	7	0.20	-0.16	**7**	0.28	-0.16	**7**	0.37	-0.15	**7**	0.36	-0.06	**7**	0.36	0.04
	37	0.04	-0.17	**37**	0.12	-0.16	**37**	0.21	-0.16	**37**	0.20	-0.07	**37**	0.20	0.01
	46	0.02	0.01	**46**	0.13	0.02	**46**	0.23	0.04	**46**	0.23	0.13	**46**	0.22	0.23

NEURAL INFORMATION PROCESSING SYSTEMS

NUS National University of Singapore

Experiment – Skeleton Actions

- MCAE -> MCAE-MP
- Multiple joints: encoded independently, use concatenated representation
- 3D trajectory: projected to three orthogonal 2D planes

Table 3: Performance (%) for skeleton-based action classification. Column "Mod." shows the data modality, where "S" indicates skeleton and "D" indicates depth map. Column "Cls." shows the auxiliary classifier used for supervised training. We also report supervised SOTAs for completeness.

				NTU60		NTU120		NW-UCLA
	Model	Mod.	Cls.	XSUB	XVIEW	XSUB	XSET	V1&V2 → V3
Unsupervised	Luo *et al.* [27]	S+D	SLP	61.4	53.2	–	–	50.7
	Li *et al.* [20]	S+D	SLP	68.1	63.9	–	–	62.5
	SeBiReNet [29]	S	LSTM	–	79.7	–	–	80.3
	LongT GAN [63]	S	SLP	39.1	48.1	–	–	74.3
	MS^{2}L [24]	S	SLP	52.6	–	–	–	76.8
	CAE+ [33]	S	SLP	58.5	64.8	48.6	49.2	–
	MCAE-MP (SLP)	S	SLP	**65.6**	74.7	**52.8**	**54.7**	83.6
	P&C [41]	S	1-NN	50.7	76.1	–	–	**84.9**
	MCAE-MP (1-NN)	S	1-NN	51.9	**82.4**	42.3	46.1	79.1
Supv.	DropGraph [2]	S	–	90.5	96.6	82.4	84.3	93.8
	JOLO-GCN [1]	S	–	93.8	98.1	87.6	89.7	–

Conclusion

- MCAE learns representation for motion that is
  - Discriminative: segment activation reveals semantic information
  - Efficient: requires significantly less parameters compared with baselines
  - Robust against transformation
- Works well on both synthetic and real-world scenarios
- MCAE can be helpful in other sequence analysis tasks
  - Joint modeling of visual appearance and motion in video
  - As mid-level feature in other models (e.g., GCN) for tasks beyond classification

Thank you!

- See you in the poster session…
- Source code and Trajectory20 👇

https://github.com/ZiweiXU/CapsuleMotion

NEURAL INFORMATION PROCESSING SYSTEMS

-CRiPT

NUS National University of Singapore