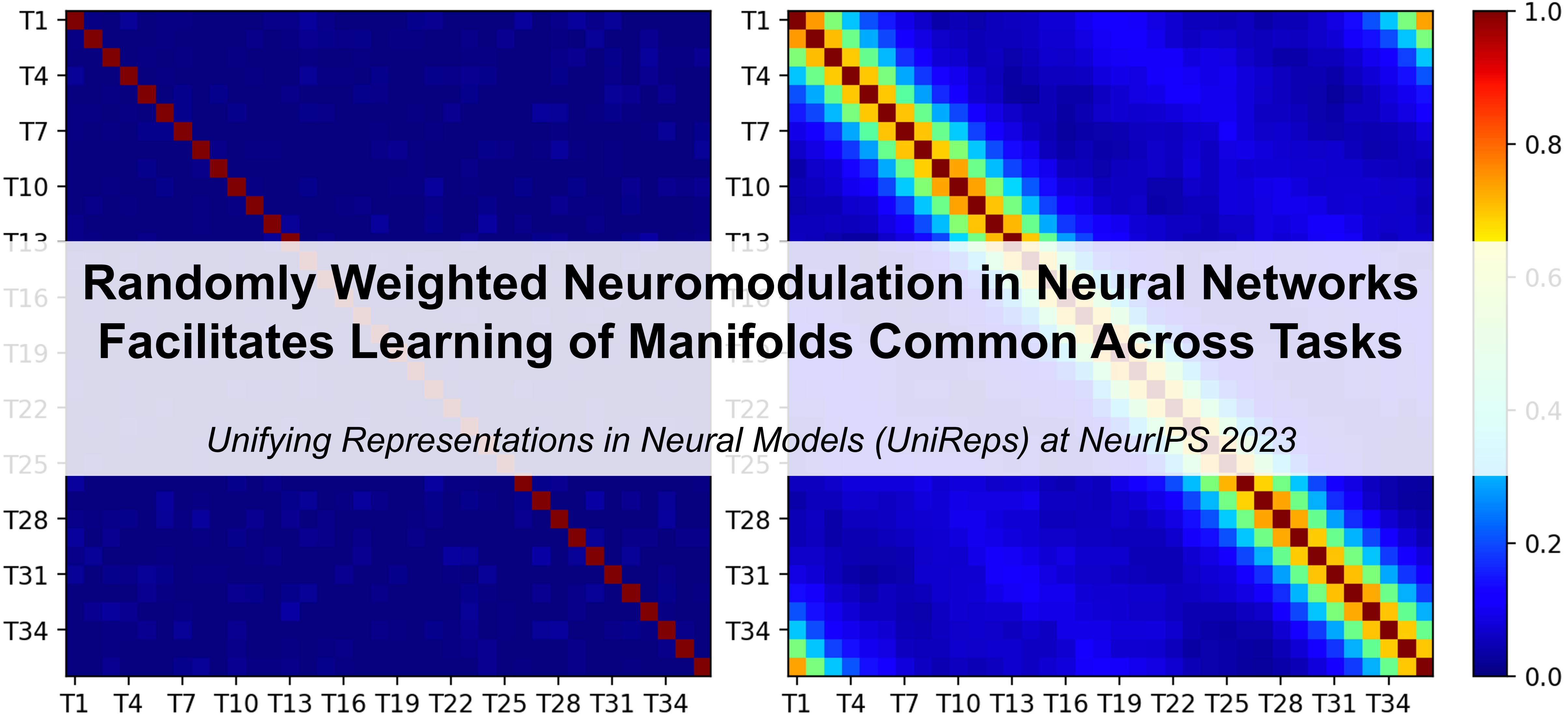
Randomly Weighted Neuromodulation in Neural Networks Facilitates Learning of Manifolds Common Across Tasks

Unifying Representations in Neural Models (UniReps) at NeurIPS 2023



Arizona State University

SEADS Lab

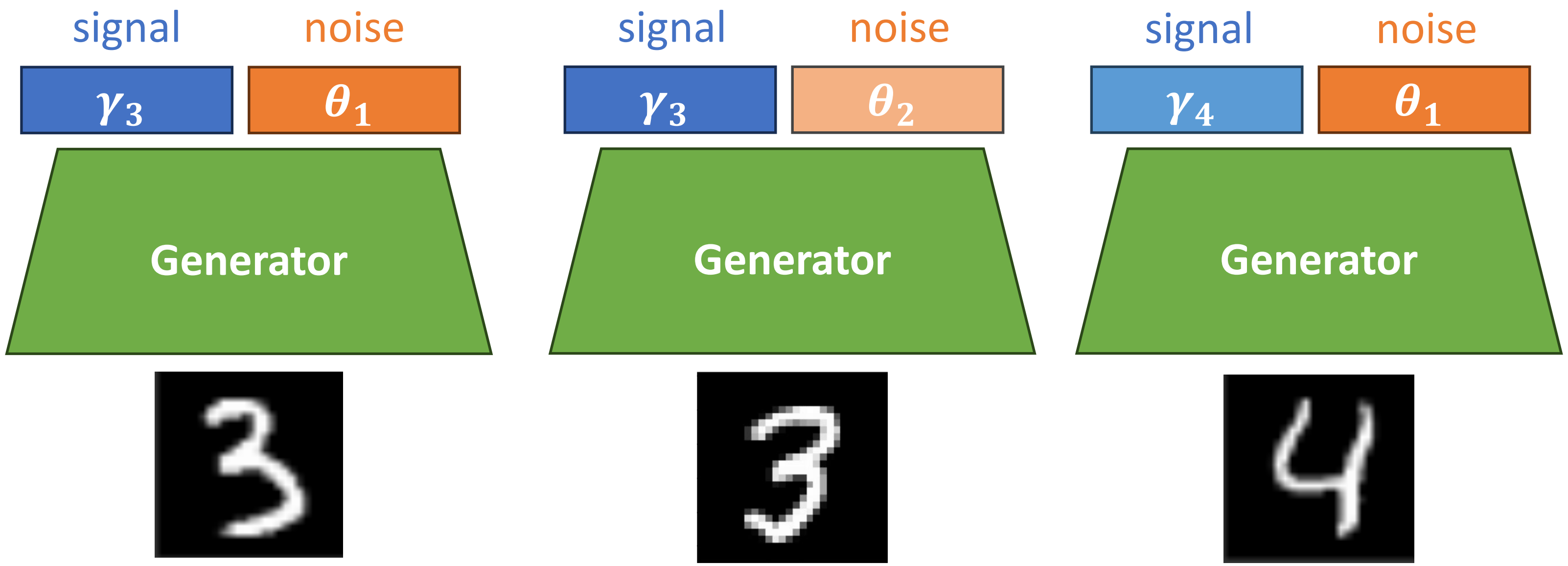
Science &
Engineering of
Autonomous
Decision-making
Systems

Jinyung Hong, Theodore P. Pavlic
School of Computing and Augmented Intelligence, Arizona State University

- **Representation Learning in Supervised Learning**
  - Learn **a representation of relevant features** *and* **parameters characterizing corruption from "noise"**
  - *Example*: Learn a function that maps every "3" to the same value but maps "4" to another value
  - *Do neural representations filter out noise and preserve uncorrupted signal at DNN representation layer?*
- **Why is this important?**
  - Transfer Learning with a few samples (representation can generalize across variations)
  - Improve robustness (separating representation and noise reduces sensitivity to noise)

$\boldsymbol{\gamma}$: relevant features or signal

$\boldsymbol{\theta}$: spurious features or noise

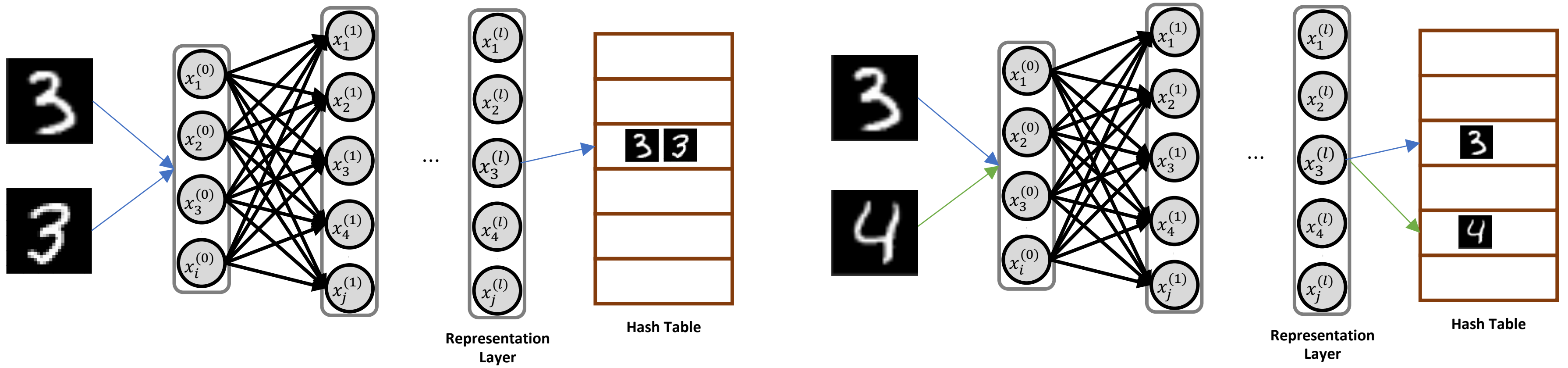


Dikkala, N., Kaplun, G. and Panigrahy, R., 2021. For manifold learning, deep neural networks can be locality sensitive hash functions. arXiv preprint arXiv:2103.06875.

INTRODUCTION: *MANIFOLD* LEARNING

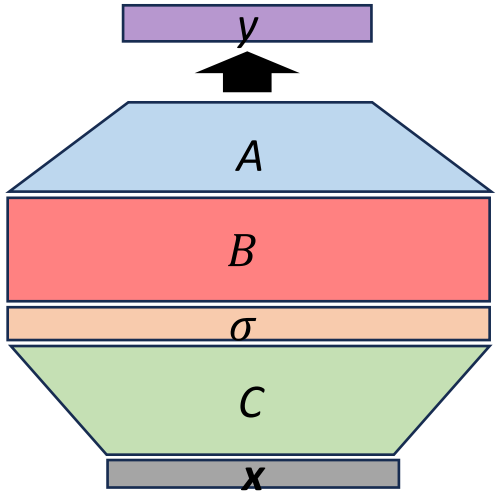
- **Each Label is a Manifold in High-dimensional Space**
  - Input $\boldsymbol{x} \in \mathbb{R}^d$ drawn from a set of manifolds with ***a shared geometry***
  - *Shared Geometry*
    - *f is an (unknown) vector-valued, bounded-norm, analytic function that maps latents $\boldsymbol{\gamma}$, $\boldsymbol{\theta}$* to input $\boldsymbol{x} = f(\boldsymbol{\gamma}, \boldsymbol{\theta})$
  - *Label* is a function g that maps $\boldsymbol{\gamma}$ (*only*) to manifold identity $y = g(\boldsymbol{\gamma})$
  - *Supervised Learning Task*
    - Given: m manifolds $(\gamma_1, \dots, \boldsymbol{\gamma}_m)$ and n samples from each (m is the number of classes)
    - Supervised learning of $g(\cdot)$: Learn to map an input $\boldsymbol{x}$ to the manifold $\boldsymbol{\gamma}$ it came from

- **Geometric Sensitive Hashing (GSH)**
  - Representations of same class cluster together ($\boldsymbol{\theta}$ independence)
  - Representations of different classes are well separated ($\boldsymbol{\gamma}$ sensitivity)



Dikkala, N., Kaplun, G. and Panigrahy, R., 2021. For manifold learning, deep neural networks can be locality sensitive hash functions. arXiv preprint arXiv:2103.06875.

- **Model Architecture for Geometric Sensitive Hashing (GSH)**
  - A single-hidden-layer architecture is sufficient for a GSH:
    - $y = A \cdot B \cdot \sigma(C\boldsymbol{x})$
      - $C \in \mathbb{R}^{D \times d}$ is **non-trainable, randomly weighted** matrix $(D \gg d)$
      - σ: ReLU activation function
      - $A \in \mathbb{R}^{m \times p}, B \in \mathbb{R}^{p \times D}$ are **trainable** matrices (m is the number of classes)
        - A, B are linear layers with ***no non-linearity*** between them
      - $y \in \mathbb{R}^{m}$ (one-hot encoding label)
  - *Loss function and regularization*: Square loss and L2 norm on A, B
    - $\mathcal{L}(A, B) = \mathbb{E}_n\left(\|A \cdot B \cdot \sigma(C\boldsymbol{x}) - y_{\text{true}}\|_{\text{F}}^2\right) + \lambda_1 \|A\|_{\text{F}}^2 + \lambda_2 \|B\|_{\text{F}}^2$
  - *Main results*: DNN can provably exhibit GSH on manifold data



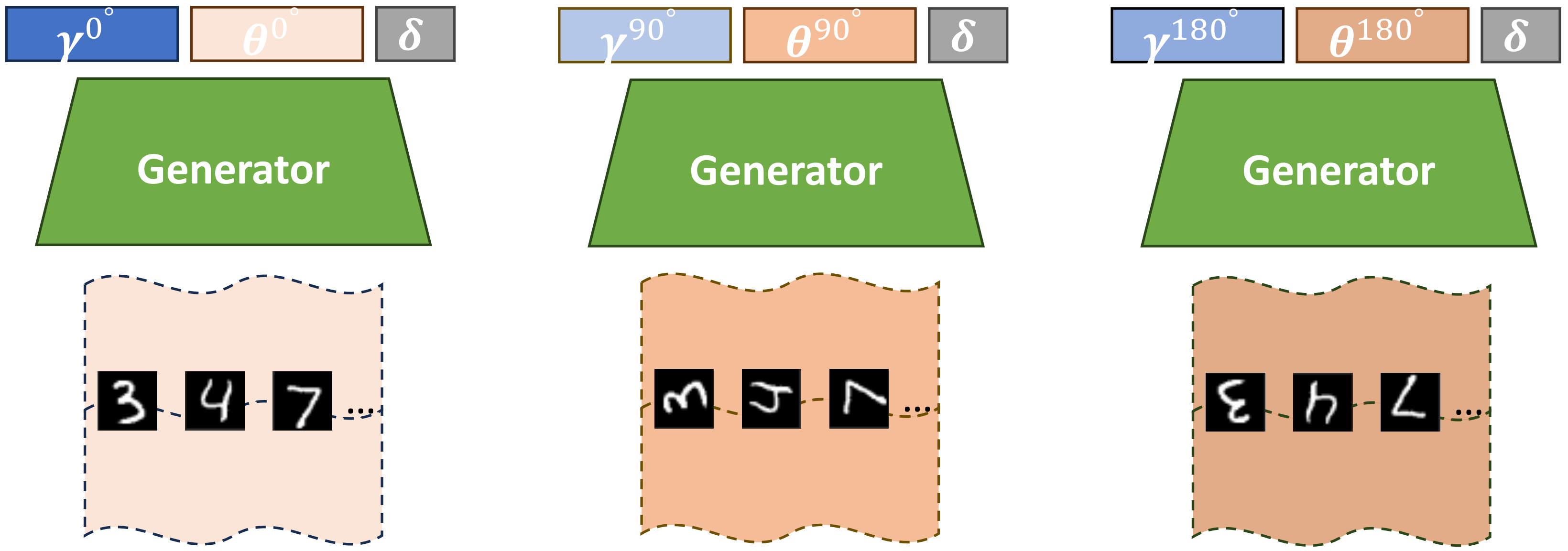
How that GSH can be extended to understand the manifold geometries in a series of supervised learning tasks?

Will manifold comparisons reflect task similarities?

Dikkala, N., Kaplun, G. and Panigrahy, R., 2021. For manifold learning, deep neural networks can be locality sensitive hash functions. arXiv preprint arXiv:2103.06875.

PROBLEM DEFINITION: REPRESENTATION LEARNING ACROSS MULTIPLE RELATED TASKS

- **Supervised Continual Learning**
  - $\mathcal{T}$ tasks arrive to a learner in sequential order
  - $\mathcal{D}_t = \{\boldsymbol{x}_{i,t}, y_{i,t}\}_t^{n_t}$ is the dataset of task t, composed of n_t pairs of input and labels
    - For simplicity, $\mathcal{C}$ is the number of classes for every task
- **Representation Learning in Supervised Continual Learning**
  - *Goal*: A function that is constant among digits with the same rotated angle but sensitive to the rotation angle of digits



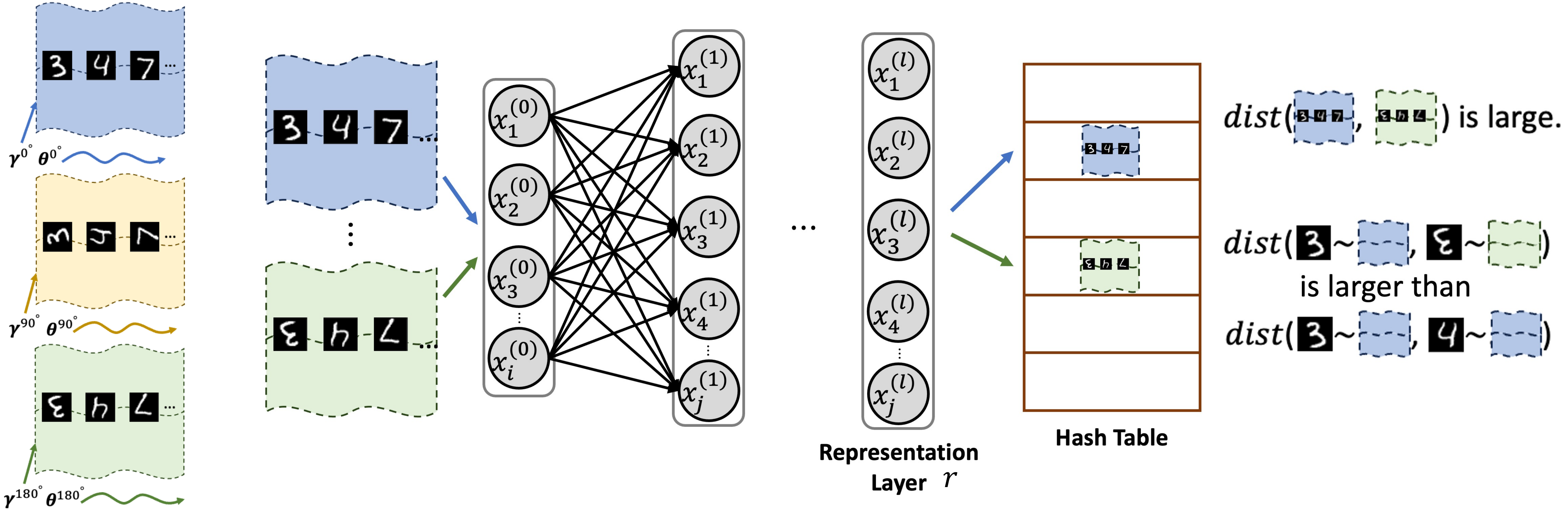
γ^t: fundamental features of the task t
θ^t: spurious features or noise of the task t
δ: shared features for classes

PROBLEM DEFINITION: GEOMETRIC SENSITIVE HASHING ACROSS MULTIPLE RELATED TASKS

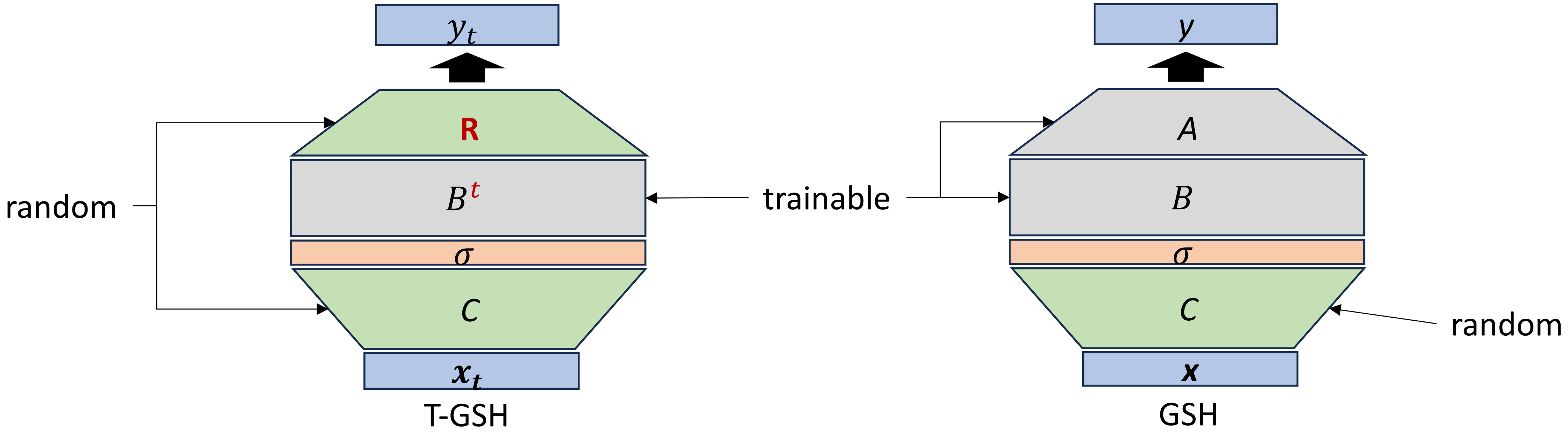
- **Each Task is a Manifold in High-dimensional Space**
  - $\mathcal{T}$ tasks arrive to a learner in sequential order
  - $\mathcal{D}_t = \{\boldsymbol{x}_{i,t}, y_{i,t}\}_t^{n_t}$ is the dataset of task t, composed of n_t pairs of input and labels
  - Each input $\boldsymbol{x}_{i,t} \in \mathbb{R}^d$ drawn from a set of manifolds with a ***task-specific*** shared geometry
  - ***Task-specific*** *Shared Geometry*
    - *f is an (unknown) vector-valued bounded norm analytic function that maps latents* $\boldsymbol{\gamma}^t, \boldsymbol{\delta}, \boldsymbol{\theta}^t$ to input $\boldsymbol{x}_t = \mathbf{f}(\boldsymbol{\gamma}^t, \boldsymbol{\delta}, \boldsymbol{\theta}^t)$
  - *Label* is a function of $\boldsymbol{\gamma}^t, \boldsymbol{\delta}$: $y_t = g(\boldsymbol{\gamma}^t, \boldsymbol{\delta})$
  - ***A Set of Supervised Learning Tasks***
    - Given $\mathcal{T}$ manifolds $(\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_{\mathcal{T}})$ and n_t samples from each task t, learn to map an input of the task t to the ***task-specific*** **manifold** it came from
    - Finding $g(\cdot)$ is **the training process in continual learning setup**

TASK-SPECIFIC GEOMETRIC SENSITIVE HASHING (T-GSH)

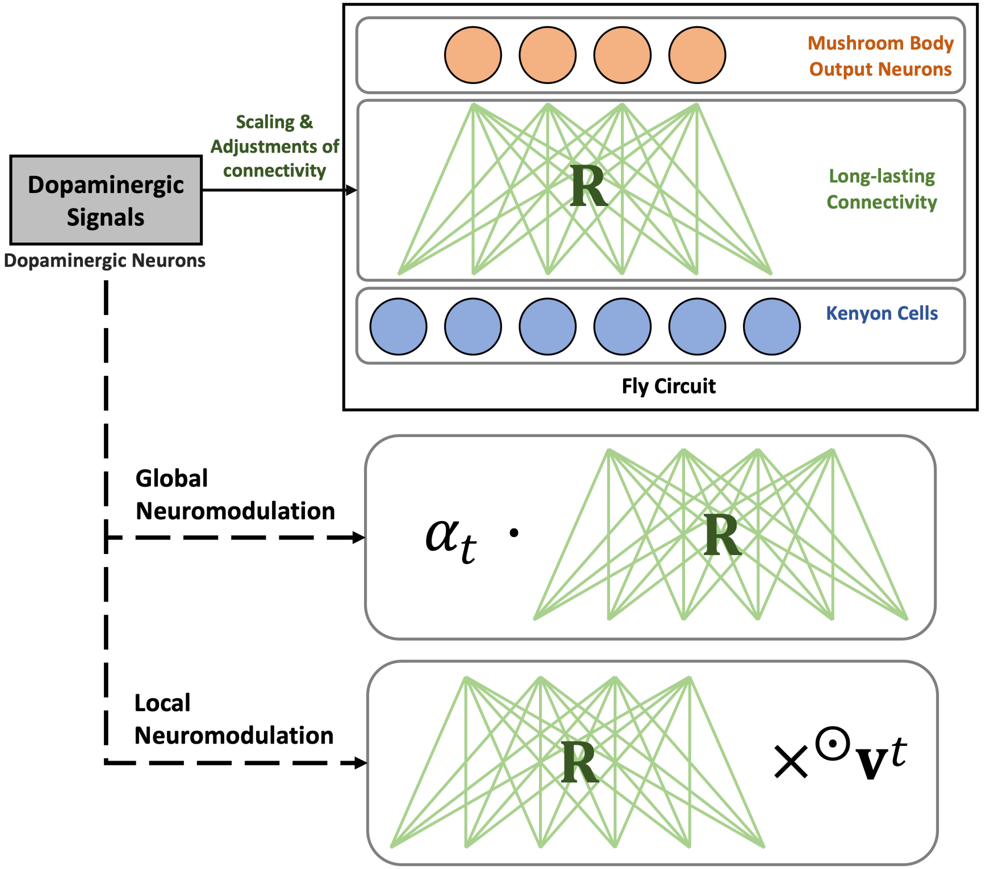
- Regardless of the associated labels:
  - Representations of any data points on the ***same*** task **cluster together**
  - Representations of any data points on the ***different* tasks** are **well separated**



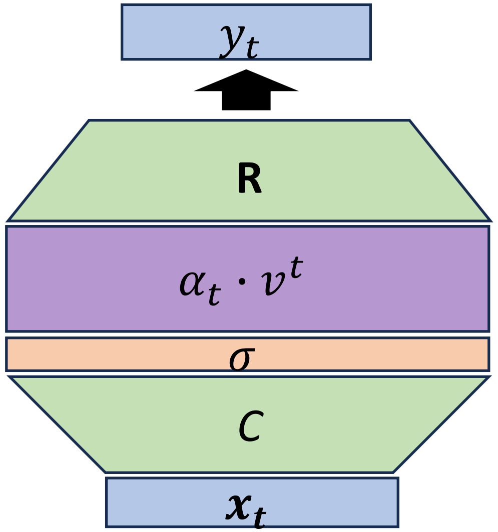
- **Model Architecture for Task-specific Geometric Sensitive Hashing (T-GSH)**
  - Model for conventional GSH: $y = A \cdot B \cdot \sigma(C\boldsymbol{x})$
  - Model for a **T-GSH** for task t: $y_t = \mathbf{R} \cdot B^t \cdot \sigma(C\boldsymbol{x}_t)$
    - σ: ReLU activation function
    - $C \in \mathbb{R}^{D\times d}$ is non-trainable, randomly weighted matrix $(D \gg d)$
    - $\mathbf{R} \in \mathbb{R}^{C\times p}$ is also non-trainable, randomly weighted matrix, representing $\boldsymbol{\delta}$
    - $B^t \in \mathbb{R}^{p\times D}$ is a trainable matrix for the task t
    - $\mathbf{R}, B$ are linear layers with ***no non-linearity*** between them
  - Can leverage the Loss function and regularization of the model for GSH



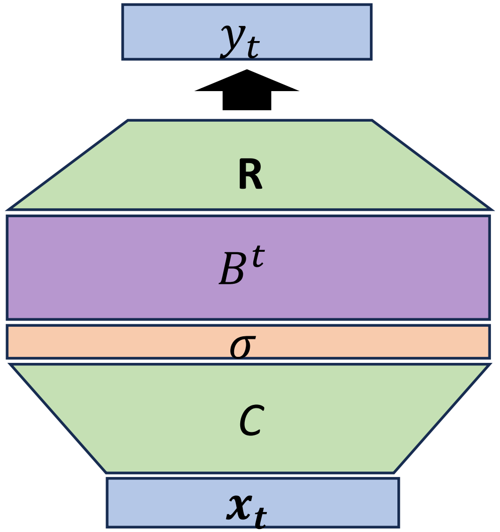
- **Configurable Random Weighted Networks (CRWNs)**
  - Simple yet efficient **neuromodulation-inspired** DNNs for continual learning
  - $y_t = \alpha_t \cdot \mathbf{R} \cdot (v^t \odot \sigma(C\boldsymbol{x}_t)) = \mathbf{R} \cdot ((\alpha_t \cdot v^t) \odot (\sigma(C\boldsymbol{x}_t)))$
    - $\alpha_t \in \mathbb{R}$ is a learnable constant acting as ***global* neuromodulation**
    - $v^t \in \mathbb{R}^D$ is a learnable vector mimicking ***local* neuromodulation**



$$y_t = \mathbf{R} \cdot \left(\underbrace{(\alpha_t \cdot v^t)}_{B^t} \odot \left(\sigma(C\boldsymbol{x}_t)\right) \right) \quad = \mathbf{R} \cdot B^t \cdot \sigma(C\boldsymbol{x}_t)$$



CRWNs

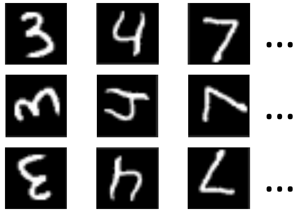


T-GSH configuration

Hong, J. and Pavlic, T.P., 2022. Learning to modulate random weights: neuromodulation-inspired neural networks for efficient continual learning. arXiv preprint arXiv:2204.04297.

EXPERIMENT 1-1: CRWN IS A GSH and T-GSH FUNCTION

- **RotationMNIST**
  - A total 36 tasks exist and each of which corresponds to images counterclockwise rotated by a multiple of 10 degrees



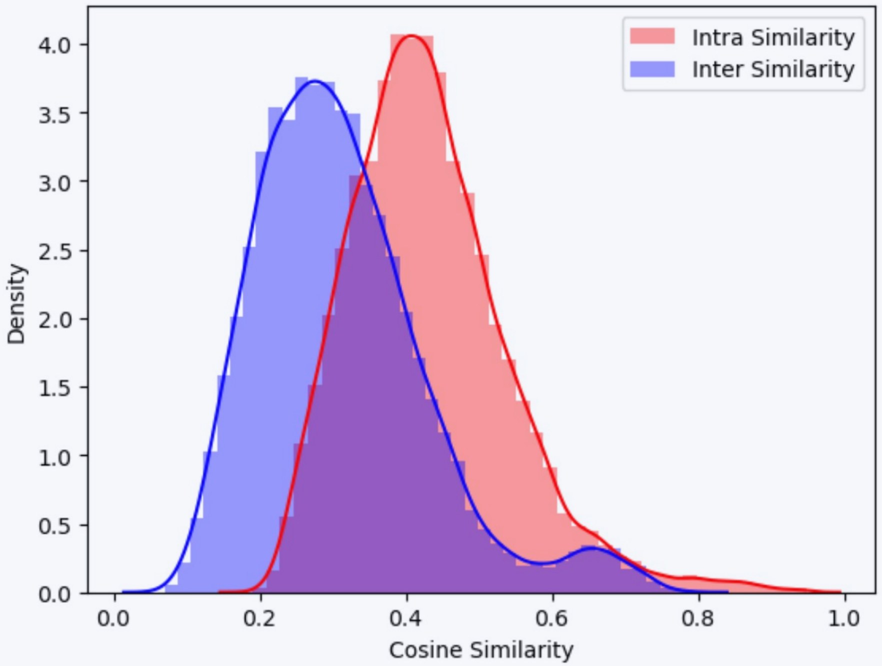
T1 (0°)
T10 (90°)
T19 (180°)

1) CRWN is a GSH Function

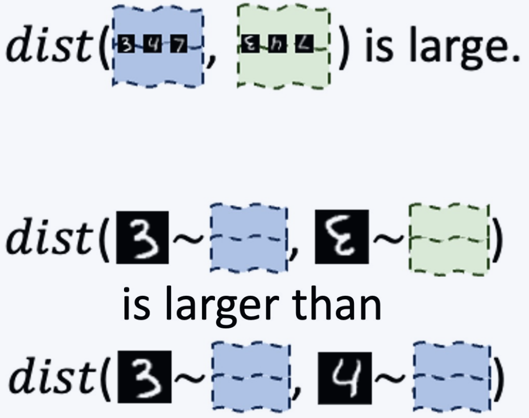
	Methods	R-MNIST Acc. (%)
	MTL, FC256	$97.18_{\pm 0.06}$
	PSP [10], FC256	$96.16_{\pm 0.06}$
	BatchE [11], FC256	$89.21_{\pm 0.17}$
	SupSup [12], FC256	$94.22_{\pm 0.03}$
	FlyNet, 10d	$94.18_{\pm 0.02}$
	FlyNet, 20d	$94.74_{\pm 0.07}$
	FlyNet, 30d	$94.90_{\pm 0.13}$
	FlyNet, 40d	$94.84_{\pm 0.10}$
CRWNs	NeuroModNet, FC256	$90.59_{\pm 0.06}$
	NeuroModNet, FC512	$92.14_{\pm 0.05}$
	NeuroModNet, FC1024	$93.51_{\pm 0.06}$
	NeuroModNet, FC2048	$94.62_{\pm 0.08}$
	NeuroModNet, FC4096	$95.49_{\pm 0.06}$

CRWNs achieved ~95% test accuracy average over all 36 tasks.
(FlyNet: 94.9% and NeuroModNet: 95.5%)

2) CRWN is a T-GSH Function



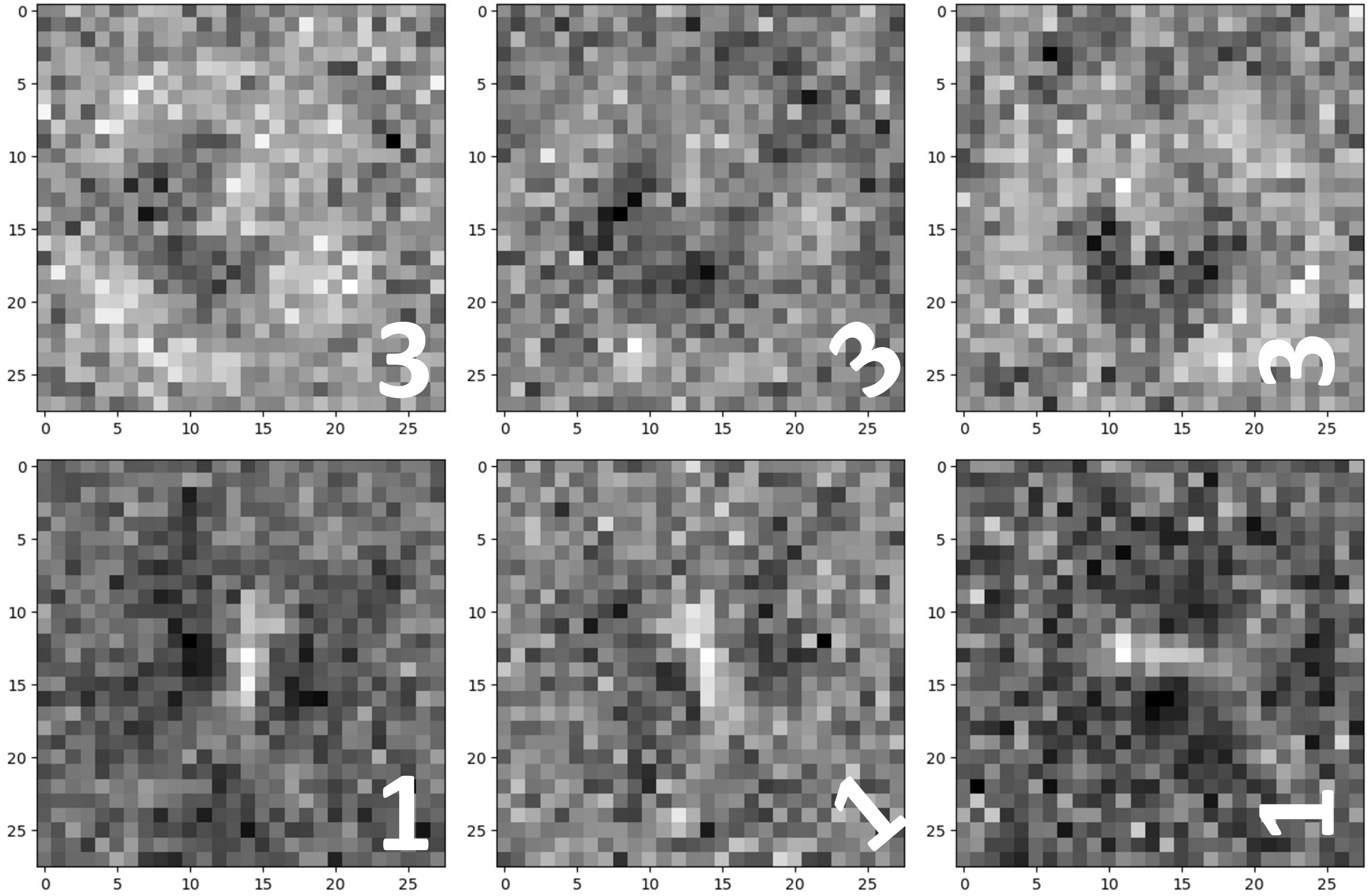
(a) A comparison of cosine similarities of the points. (**Intra similarity**): the cosine similarities of the points with the *different labels* on the *same task manifold*. (**Inter similarity**): the cosine similarities of the points with the *same label* on the *different task manifolds*.



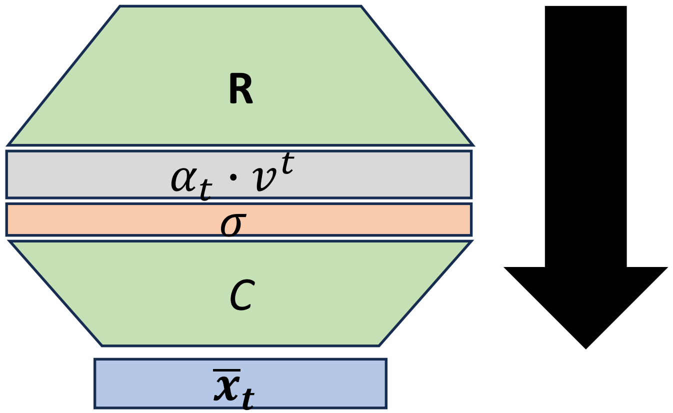
Hong, J. and Pavlic, T.P., 2022. Learning to modulate random weights: neuromodulation-inspired neural networks for efficient continual learning. arXiv preprint arXiv:2204.04297.

EXPERIMENT 1-2: APPROXIMATING/RECONSTRUCTING DATA FROM LATENT REPRESENTATION

- *Task-specific Shared Geometry*
  - *f is an (unknown) vector-valued bounded norm analytic function that maps latents $\boldsymbol{\gamma}^t, \boldsymbol{\delta}, \boldsymbol{\theta}^t$ to input* $\boldsymbol{x}_t = \mathbf{f}(\boldsymbol{\gamma}^t, \boldsymbol{\delta}, \boldsymbol{\theta}^t)$
- ***Can reconstruct data on the desired task manifold!***
  - Finding $y_t = g(\boldsymbol{\gamma}^t, \boldsymbol{\delta})$ is the training process in continual learning setup
  - Because of using ReLU, inverse of the trained CRWNs can reconstruct an approximation of data



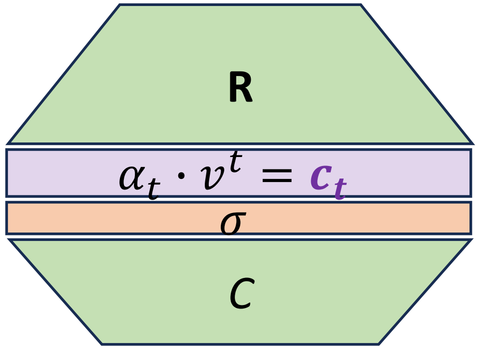
$$\bar{\mathbf{x}}_{i,t} = \mathbf{f}(\mathbf{s}) \text{ where } \mathbf{s} \sim \mathcal{N}(\mu_{i,t}, \sigma_{i,t}), \mu_{i,t} = R_i^\top \odot (\alpha_t \cdot v_t), \sigma_{i,t} = \mathbf{1}_D \cdot 1/D$$



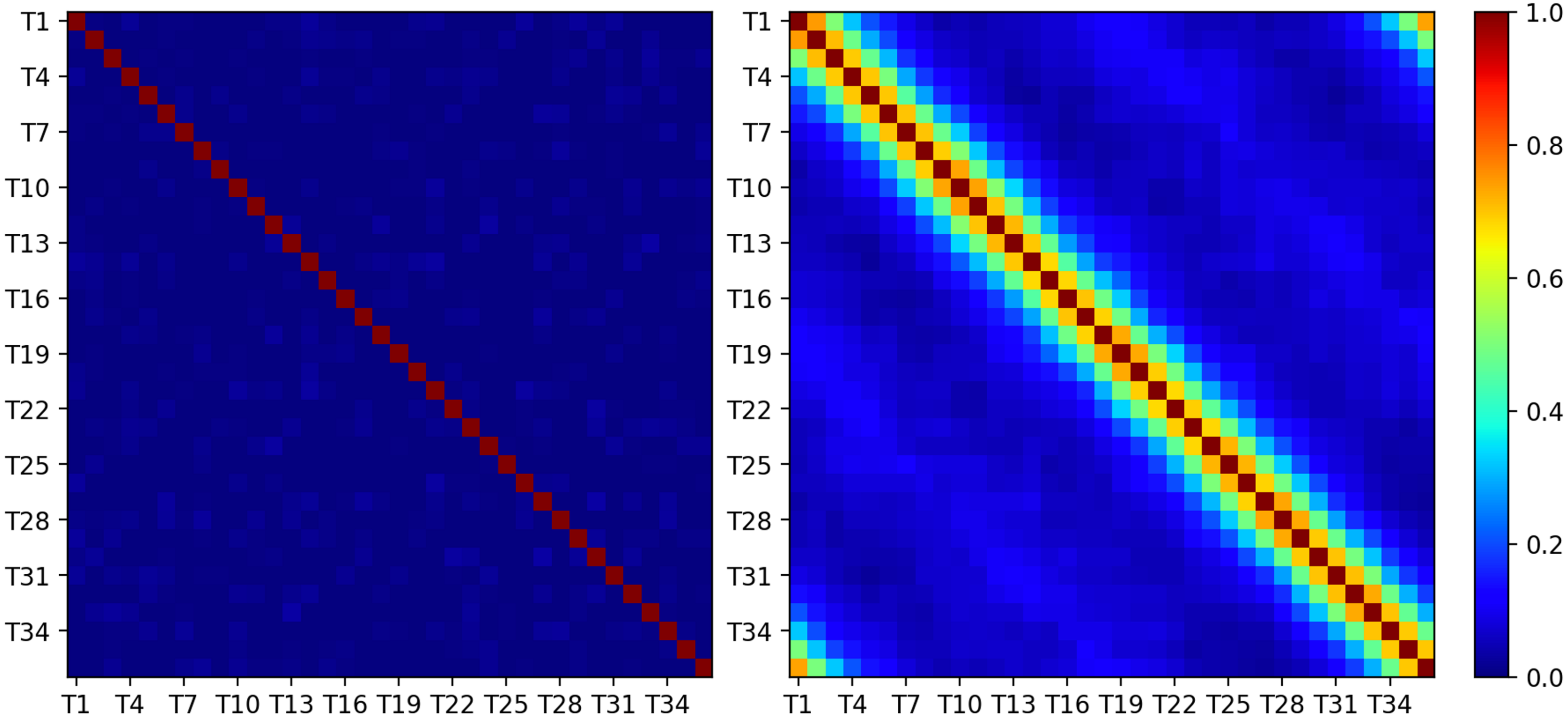
(Top Row) Reconstructed samples of digit “3”. **(Bottom Row)** Reconstructed samples of digit “1”. **(Left Column)** Reconstructed samples of digits from the task manifold T1, which is 0°rotation. **(Middle Column)** the samples of digits from the task manifold T5, which is 40 °counterclockwise rotation. **(Right Column)** the samples of digits from the task manifold T10, which is 90 °counterclockwise rotation.

EXPERIMENT 2-1: MEASURING REPRESENTATIONAL SIMILARITIES ON ROTATIONMNIST TASKS

- **Configurable Random Weighted Networks (CRWNs)**
  - $y_t = \mathbf{R} \cdot ((\alpha_t \cdot v^t) \odot \sigma(C\boldsymbol{x}_t))$
    - $\alpha_t \in \mathbb{R}$ is a learnable constant acting as *global* neuromodulation
    - $v^t \in \mathbb{R}^D$ is a learnable vector mimicking *local* neuromodulation
  - $y_t = \mathbf{R} \cdot (c_t \odot \sigma(C\boldsymbol{x}_t))$
    - $c_t \triangleq \alpha_t \cdot v^t$ is called **a context vector**



- **The learned task manifolds can represent the *relationships between the tasks*!**

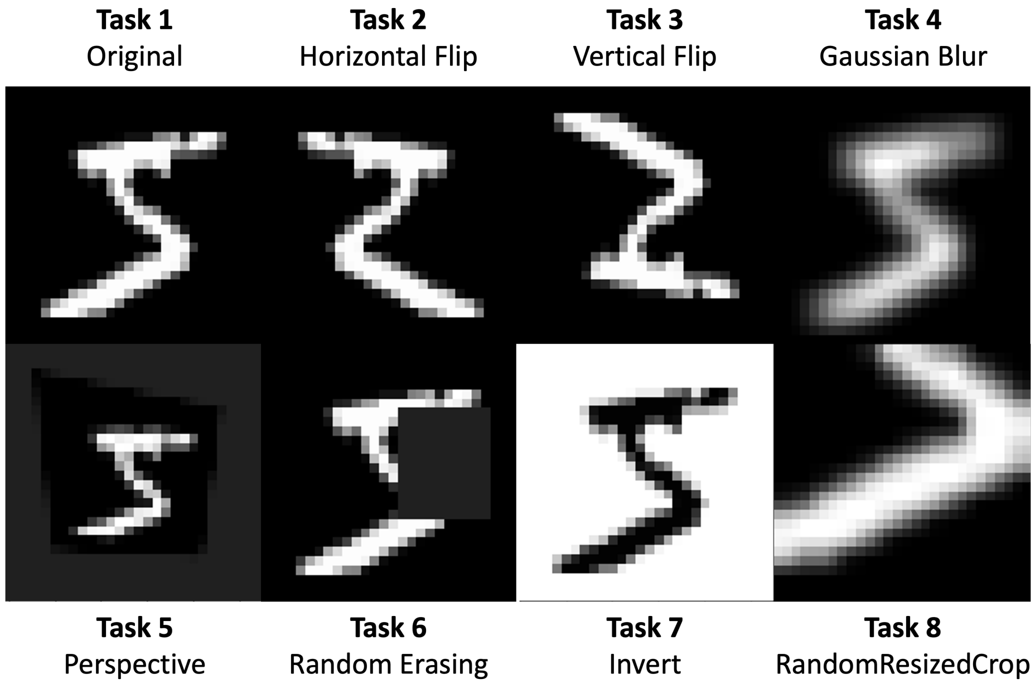


<u>Context-vector comparison across 36 tasks</u>

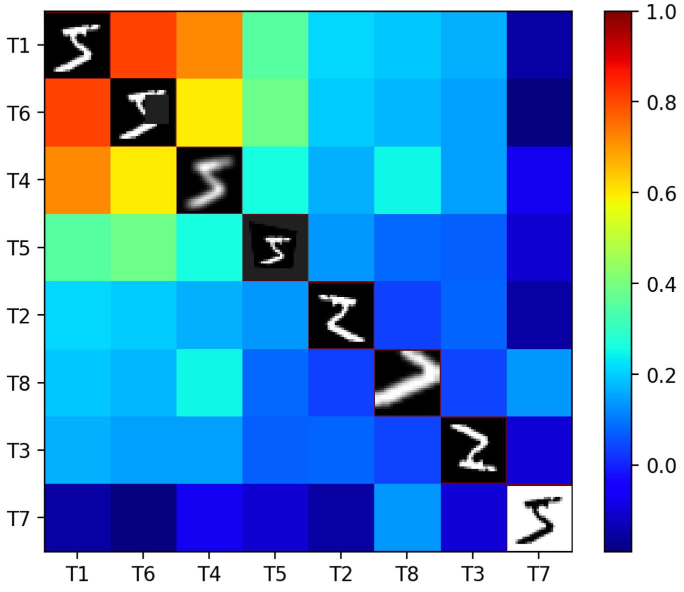
A confusion matrix of intra (same task manifold) VS inter (different task manifolds) cosine similarity of task representations trained on *RotationMNIST*. Cosine similarity between task context vectors before **(left)** and after training **(right)**.

EXPERIMENT 2-2: MEASURING REPRESENTATIONAL SIMILARITIES ON AUGMENTMNIST TASKS

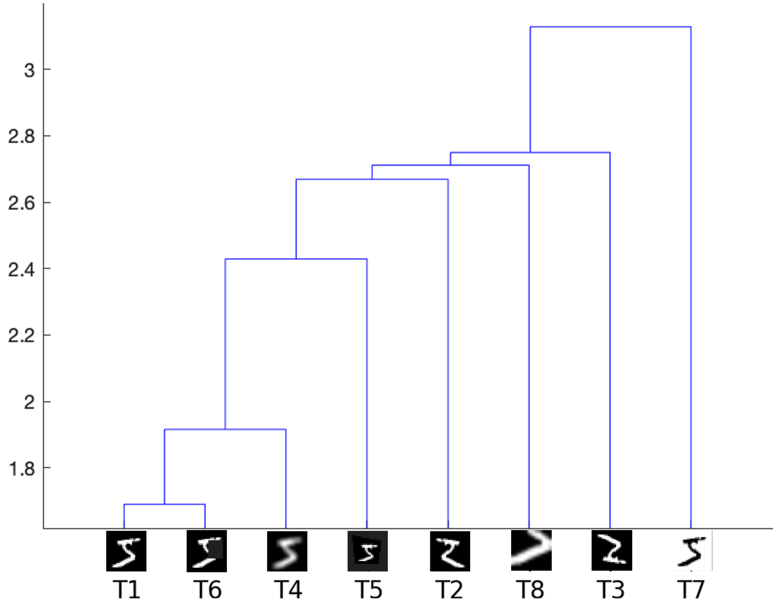
- **AugmentMNIST**
  - A sequence of 8 off-the-shelf, commonly used data-augmentation tasks
  - After training on each of the 8 tasks, use *hierarchical agglomerative clustering* to **sort the task context vectors** so that adjacent tasks tend to have highest similarity



Cosine similarities



Inferred task phylogeny



Representational differences reflect <u>fundamental relationships</u> between tasks

- **Key Results:**
  - Proposed T-GSH, an extension of GSH, to understand the manifold geometries in a ***series*** of supervised learning tasks
  - Used T-GSH to connect neuromodulation-inspired neural networks for continual learning and task-specific geometric manifold learning
    - Closing a gap between representational learning and neuroscience
  - Demonstrated that each of the learned task manifolds can represent (possibly unappreciated) relationships between the tasks based on them
- **Future Research Directions**
  - Enhance theoretical support for learning in various continual-learning setups, such as *class-incremental* and *domain-incremental* learning

Thank you so much ☺

UniReps Workshop

Unifying Representations in Neural Models

NSF

ERDC

ENGINEER RESEARCH & DEVELOPMENT CENTER

DISCOVER | DEVELOP | DELIVER

ASU

SEADS Lab

Science &
Engineering of
Autonomous
Decision-making
Systems

Randomly Weighted Neuromodulation in Neural Networks Facilitates Learning of Manifolds Common Across Tasks

Jinyung Hong[1] **Theodore P. Pavlic**[1,2]

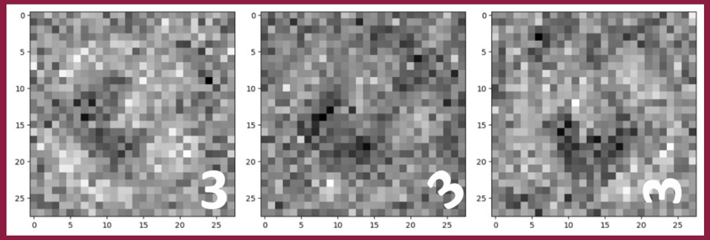
[1]School of Computing and Augmented Intelligence

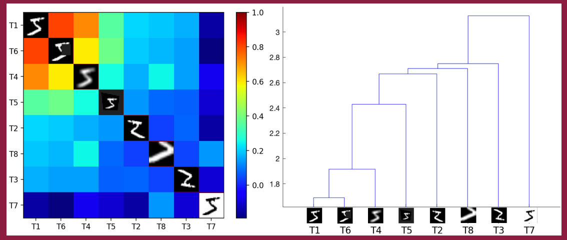
[2]School of Life Sciences

Arizona State University

Tempe, AZ 85281

{ jhong53, tpavlic }@asu.edu





jhong53@asu.edu

Google Scholar

My Webpage

…Questions?