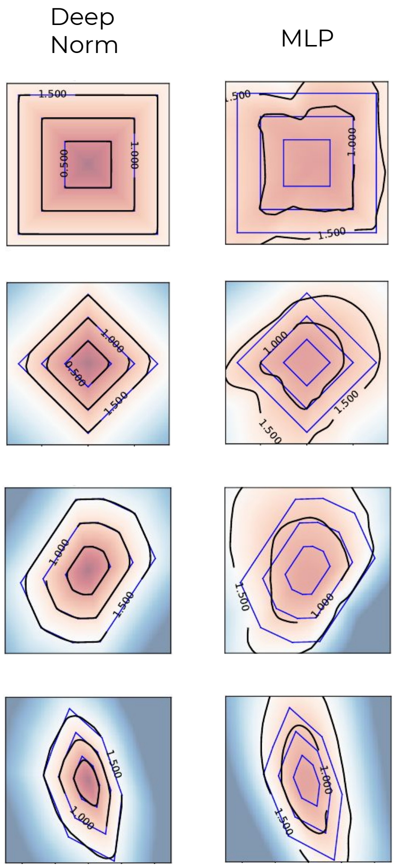


An Inductive Bias for Distances: Neural Nets that Respect the Triangle Inequality

Deep Norms, Wide Norms, and Neural Metrics

Silviu Pitis* **Harris Chan*** **Kiarash Jamali** **Jimmy Ba**

University of Toronto, Vector Institute

*Equal contribution

ICLR 2020, April 26-30

We Seek to Model Distances on Some Set

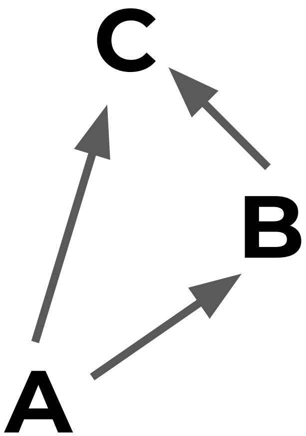
- Distance tells us how close or far apart things are

- Examples of things:
  - Nodes in a graph (shortest path length)
  - States in reinforcement learning (optimal value function)
  - Items in a recommender system
  - Images in computer vision

Hypothesis: Triangle Inequality is a Good Inductive Bias

The (shortest) distance from A to C is *no greater* than the distance from A to B plus the distance from B to C.

$$d(A, C) \leq d(A, B) + d(B, C)$$



Many have found this to be a useful inductive bias:

Reinforcement Learning

- Kaelbling's DG Learning (1993)
- Schaul et al.'s Universal Value Functions (2015)
- He et al.'s Optimality Tightening (2016)

Supervised Learning

- Hsieh et al.'s Collaborative Metric Learning (2017)
- Snell et al.'s Prototypical Networks (2017)

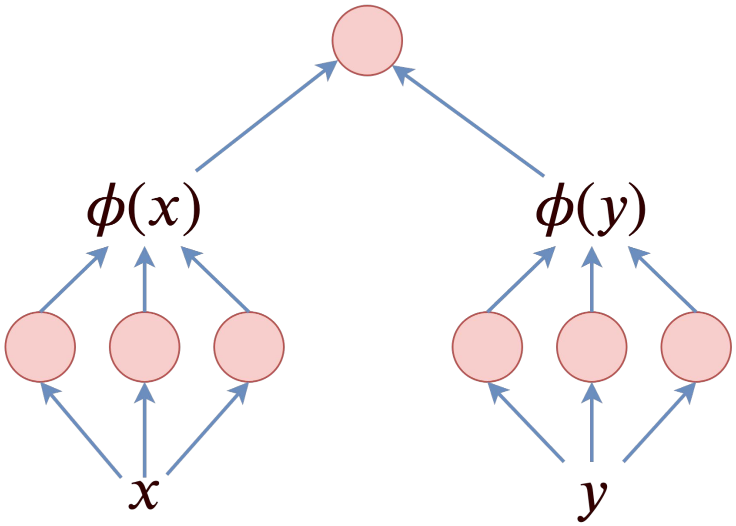
Deep Metric Learning

Map inputs to a latent metric space, e.g., R^n with Euclidean metric (right), and apply the metric in the latent space.

AKA **Siamese network** (Bromley 1994).

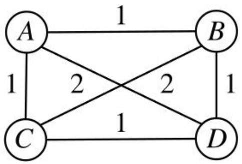
This architecture satisfies the triangle inequality.

$$d(x, y) = \|\phi(x) - \phi(y)\|$$

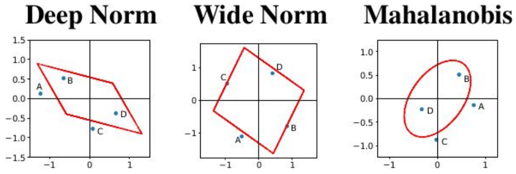


Where <u>Euclidean</u> Metric Learning Fails

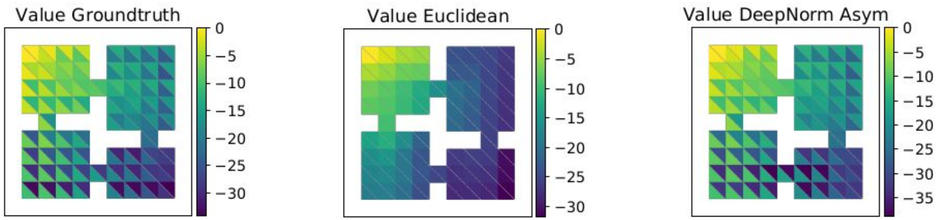
1. **Many simple metrics cannot be embedded into any Euclidean space.**



Norm	MSE
Euclidean, $\mathbb{R}^n$	0.057
Deep Norm, $\mathbb{R}^2$	0.000
Wide Norm, $\mathbb{R}^2$	0.000



2. **Many metrics we care about (e.g., almost all RL tasks) are *asymmetric!***



Comparison of Architectures

	Metric / Norm Properties				
	N1 / M1-2 Positive Definite	N2 Positive Homogenous	**N3 / M3 Triangle Inequality**	N4 / M4 Symmetric	**Universal Norm Approximator**
Euclidean	✔	✔	✔	✔	✘
Unconstrained NN	✘	✘	✘	✘	✔

Comparison of Architectures

	Metric / Norm Properties				
	N1 / M1-2 Positive Definite	N2 Positive Homogenous	**N3 / M3 Triangle Inequality**	N4 / M4 Symmetric	**Universal Norm Approximator**
Euclidean	✓	✓	✓	✓	✗
Unconstrained NN	✗	✗	✗	✗	✓
Deep Norm	✓/✗	✓	✓	✓/✗	✓
Wide Norm	✓/✗	✓	✓	✓/✗	✓
Neural Metric	✓/✗	✓/✗	✓	✓/✗	✓

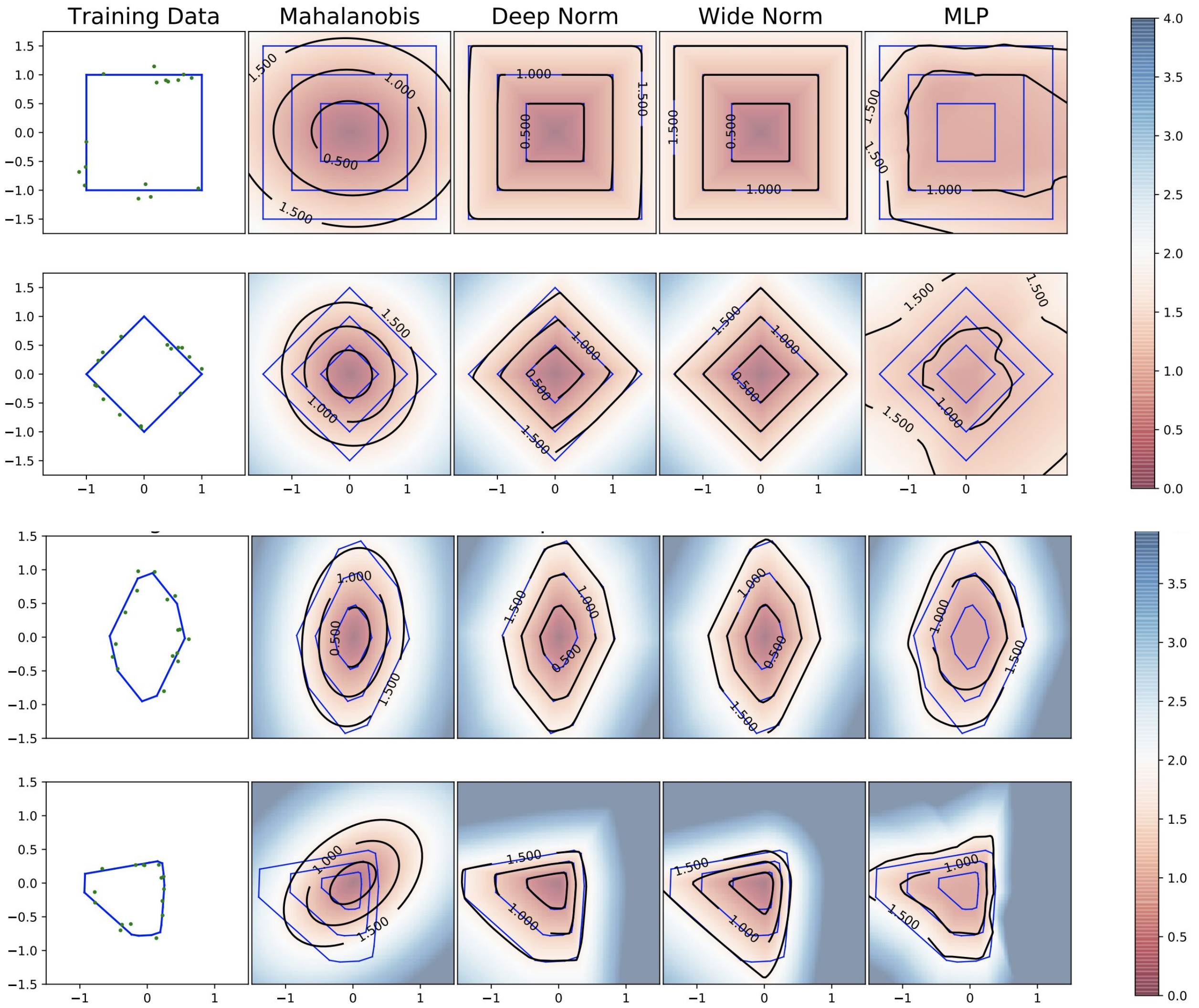
Deep → more expressive

Fast for pairwise distances in large mini-batches!

Can use either DN or WN as base architecture.

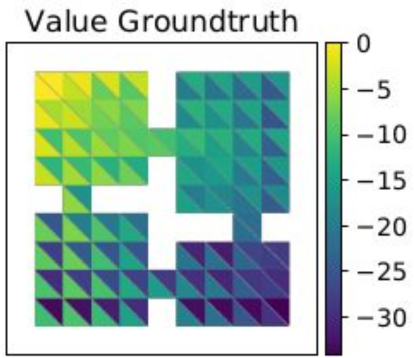
✓/✗: *optionally enforced via propositions (see paper)*

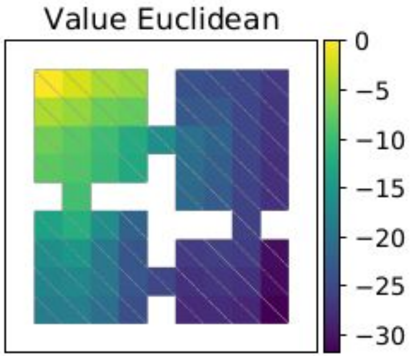
Modeling 2D Norms

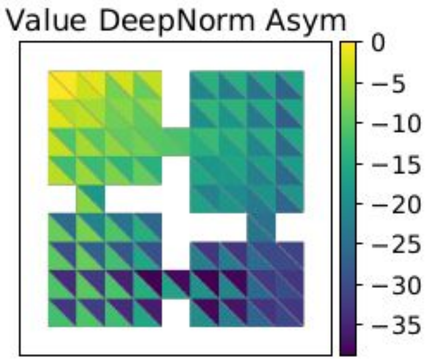


Learning General Value Functions

Problem: Learn a **Universal Value Function Approximator (UVFA)** $V_\theta(s, g)$ in goal-oriented RL environment: $R(s, s') = -1$ eps terminates @ **s = g**







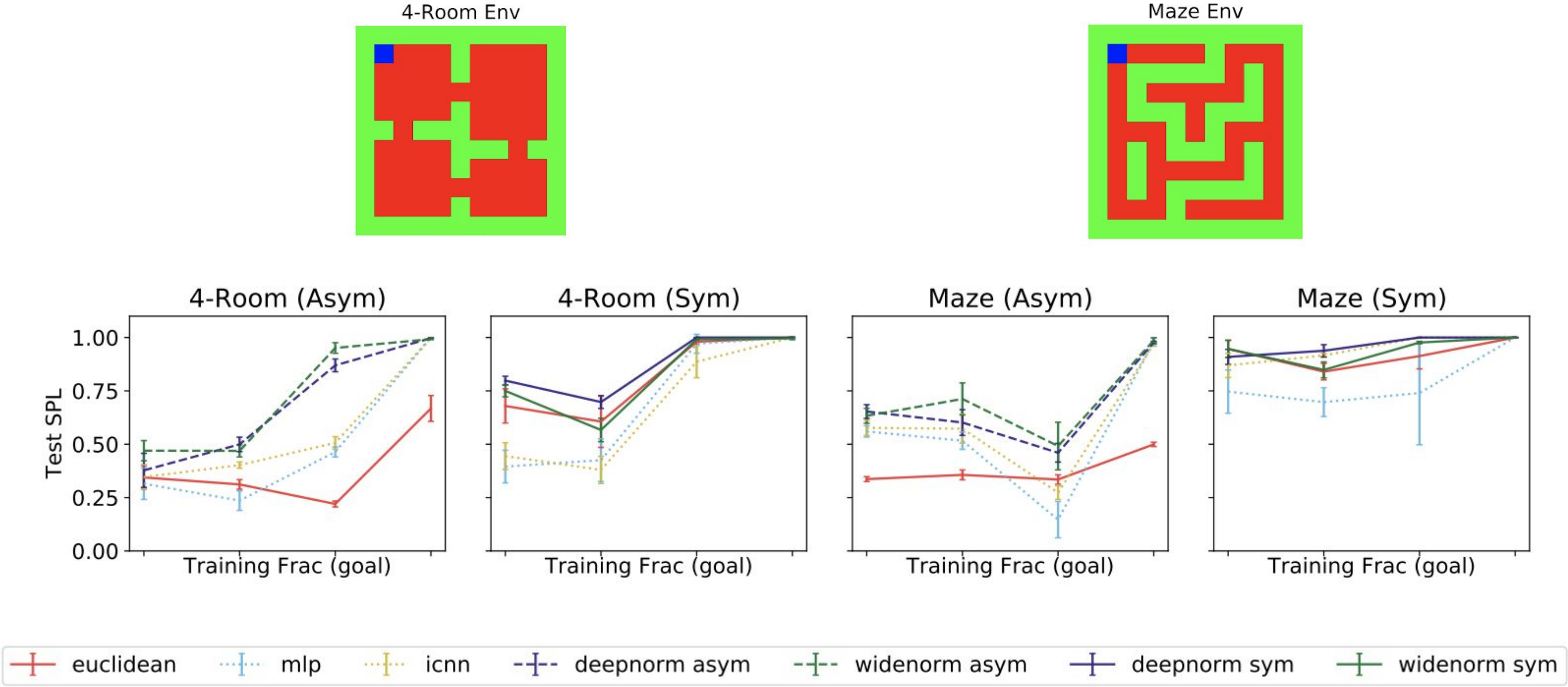


Fig. 5: GVF results. Generalization as measured by SPL metric (higher is better) on held out (s, g) pairs as function of fraction of goals seen during training. Results averaged over 3 seeds and error bar indicates standard deviation. For fraction = 1 we evaluate on entire data.

Application: Norm/Metric Substitution

Can be used in any architecture / algorithm that uses a Euclidean norm or metric

- Clustering & retrieval
- Collaborative metric learning
- Few-shot learning (e.g., prototypical networks)

e.g., asymmetric node2vec?

May be useful for ***asymmetric*** applications (e.g., DAGs, ordered embeddings, entailment)

Caveat: triangle inequality not always necessary / sensible

Thanks for watching!

- Check out our paper to learn more!

- Check our Github:

  https://github.com/spitis/deepnorms

  for Tensorflow (v1) and Pytorch implementations.

- Happy to answer any questions by email:

spitis@cs.toronto.edu hchan@cs.toronto.edu