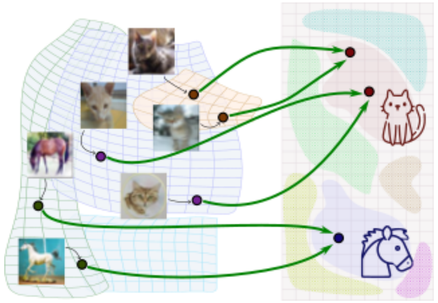


NEURAL INFORMATION PROCESSING SYSTEMS

Unveiling Transformer Perception by Exploring Input Manifolds

Alessandro Benfenati♤✶, Alfio Ferrara♧✶, Alessio Marta♡✶, Davide Riva♢✧, Elisabetta Rocchetti♧✶ ✉

Departments of Environmental Science and Policy♤, Computer Science♧, Mathematics♡, Control and Computer Engineering♢

Università degli Studi di Milano✶, Politecnico di Torino✧

{alessandro.benfenati, alfio.ferrara, alessio.marta, elisabetta.rocchetti}@unimi.it ; davide.riva@polito.it

UNIVERSITÀ DEGLI STUDI DI MILANO

LA STATALE

IS Lab information systems and knowledge management

NEURAL INFORMATION PROCESSING SYSTEMS

How to understand which inputs a Transformer treats as *equivalent*?

How to understand which inputs a Transformer treats as *equivalent*?

NEURAL INFORMATION PROCESSING SYSTEMS

📚 Existing methods rely on ad-hoc perturbations (heuristics, gradients) [1, 2, 3, 4]

VERIX: Towards Verified Explainability of Deep Neural Networks

Min Wu
Department of Computer Science
Stanford University
minwu@cs.stanford.edu

Haoze Wu
Department of Computer Science
Stanford University
haozewu@cs.stanford.edu

Clark Barrett
Department of Computer Science
Stanford University
barrett@cs.stanford.edu

The Limitations of Deep Learning in Adversarial Settings

Nicolas Papernot*, Patrick McDaniel*, Somesh Jha†, Matt Fredrikson‡, Z. Berkay Celik*, Ananthram Swami§
*Department of Computer Science and Engineering, Penn State University
†Computer Sciences Department, University of Wisconsin-Madison
‡School of Computer Science, Carnegie Mellon University
§United States Army Research Laboratory, Adelphi, Maryland
{ngp5056,mcdaniel}@cse.psu.edu, {jha,mfredrik}@cs.wisc.edu, zbc102@cse.psu.edu, ananthram.swami.civ@mail.mil

Intriguing Equivalence Structures of the Embedding Space of Vision Transformers

Shaeke Salman[1], Md Montasir Bin Shams[1], Xiuwen Liu[1]
[1]Department of Computer Science, Florida State University, FL 32306, USA
{salman, liux}@cs.fsu.edu, mshams@fsu.edu,

↳Bounded and optimised perturbations

↳Gradient-based exploration

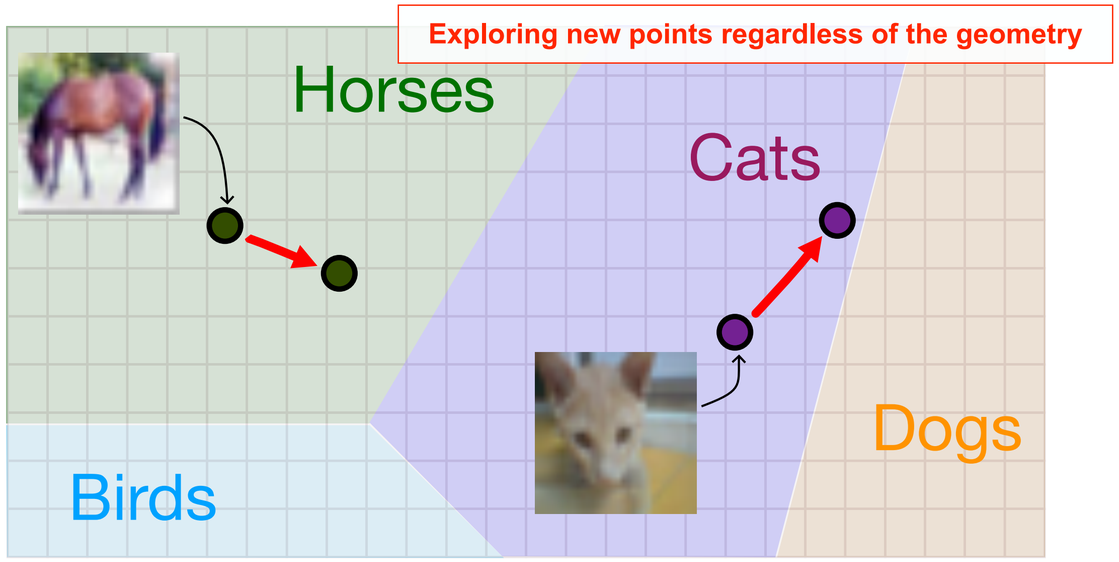
[1] X. Cai, J. Huang, Y. Bian, and K. Church (2020). Isotropy in the contextual embedding space: Clusters and manifolds. In International conference on learning representations.
[2] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The Limitations of Deep Learning in Adversarial Settings (2016). In IEEE European Symposium on Security and Privacy (EuroS&P), pages 372–387, Mar. 2016. doi: 10.1109/EuroSP.2016.36.
[3] S. Salman, M. M. B. Shams, and X. Liu (2024). Intriguing equivalence structures of the embedding space of vision transformers.
[4] M. Wu, H. Wu, and C. Barrett. Verix: Towards verified explainability of deep neural networks (2024). Advances in neural information processing systems, 36.

How to understand which inputs a Transformer treats as *equivalent*?

NEURAL INFORMATION PROCESSING SYSTEMS

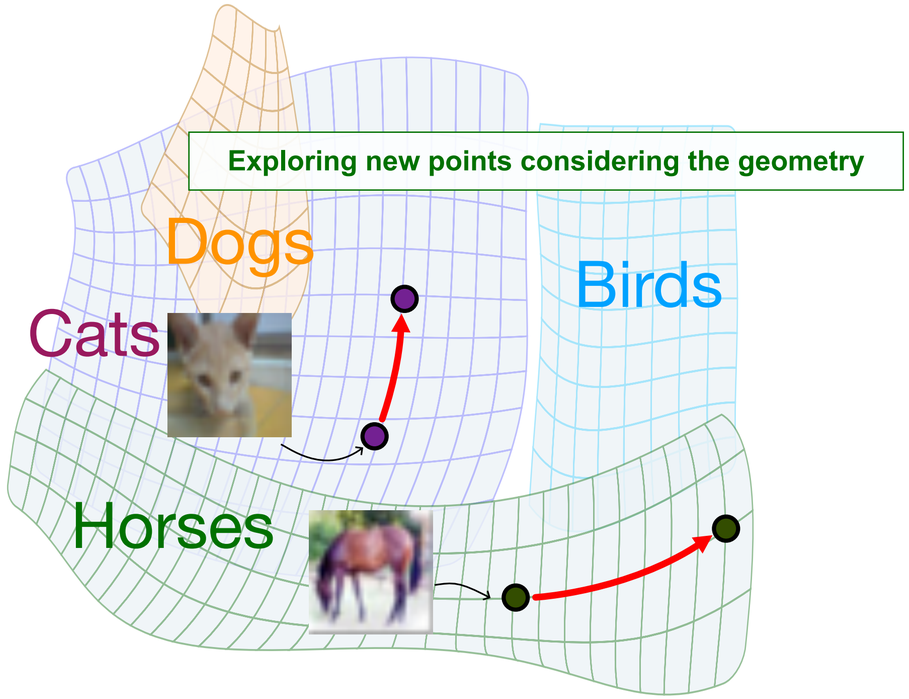
📚 Existing methods rely on ad-hoc perturbations (heuristics, gradients) [1, 2, 3, 4]

❗ Issue: the *real* geometric structure of Transformer representations is ignored



Embedding space assuming Euclidean structure

≠



Embedding space ***not*** assuming Euclidean structure

NEURAL INFORMATION PROCESSING SYSTEMS

How to understand which **inputs** a Transformer treats as ***equivalent***?

📚 Existing methods rely on ad-hoc perturbations (heuristics, gradients) [1, 2, 3, 4]

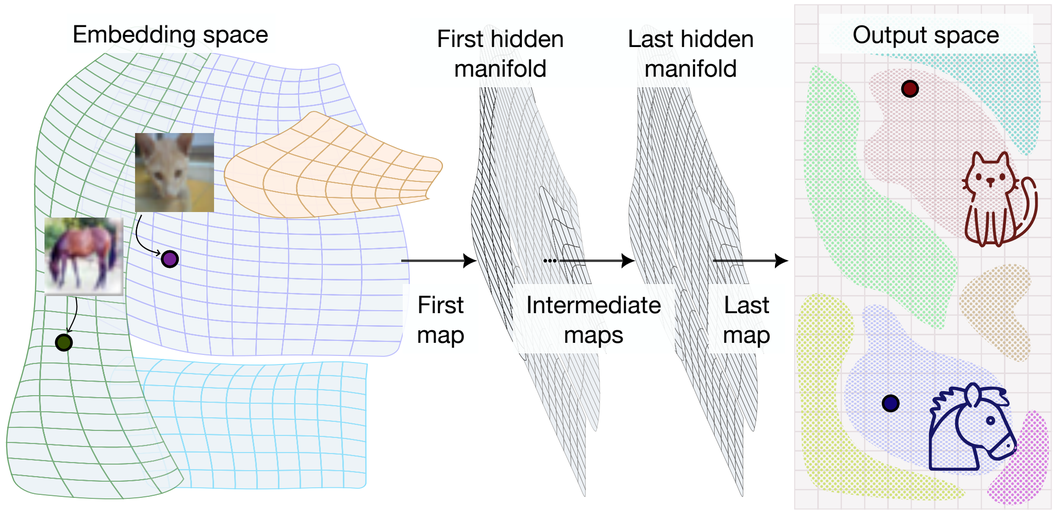
❗ Issue: the *real* geometric structure of Transformer representations is ignored

We use ***Riemannian geometry*** techniques to navigate the model's **equivalence classes**

Preliminaries

Neural networks as *sequences of smooth geometric maps* between manifolds.

$$M_0 \overset{\Lambda_1}{\to} M_1 \overset{\Lambda_2}{\to} \cdots M_{n-1} \overset{\Lambda_n}{\to} M_n$$



Preliminaries

Neural networks as sequences of smooth geometric maps between manifolds.

Modelling distances with ***Riemannian geometry***

Preliminaries

Neural networks as sequences of smooth geometric maps between manifolds.

Modelling distances with Riemannian geometry

- ***Singular Riemannian metric***

$$g : M \to Bil(\mathbb{R}^n \times \mathbb{R}^n), \qquad g_p : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$$

Preliminaries

Neural networks as sequences of smooth geometric maps between manifolds.

Modelling distances with Riemannian geometry

- Singular Riemannian metric
- ***Pseudodistance***: infimum of Pseudolength

$$\gamma : [a,b] \to \mathbb{R}^n, \quad s \in [0,1] \qquad Pl(\gamma) = \int_a^b \|\dot{\gamma}(s)\|_{\gamma(s)} ds = \int_a^b \sqrt{g_{\gamma(s)}(\dot{\gamma}(s), \dot{\gamma}(s))} ds$$

- Zero Pd points in the same equivalence class $M_i / \sim_i$.

Preliminaries

Neural networks as sequences of smooth geometric maps between manifolds.

Modelling distances with Riemannian geometry

- Singular Riemannian metric
- Pseudodistance
  - Zero Pd points in the same equivalence class $M_i/\sim_i$.
- ***Pullback function***

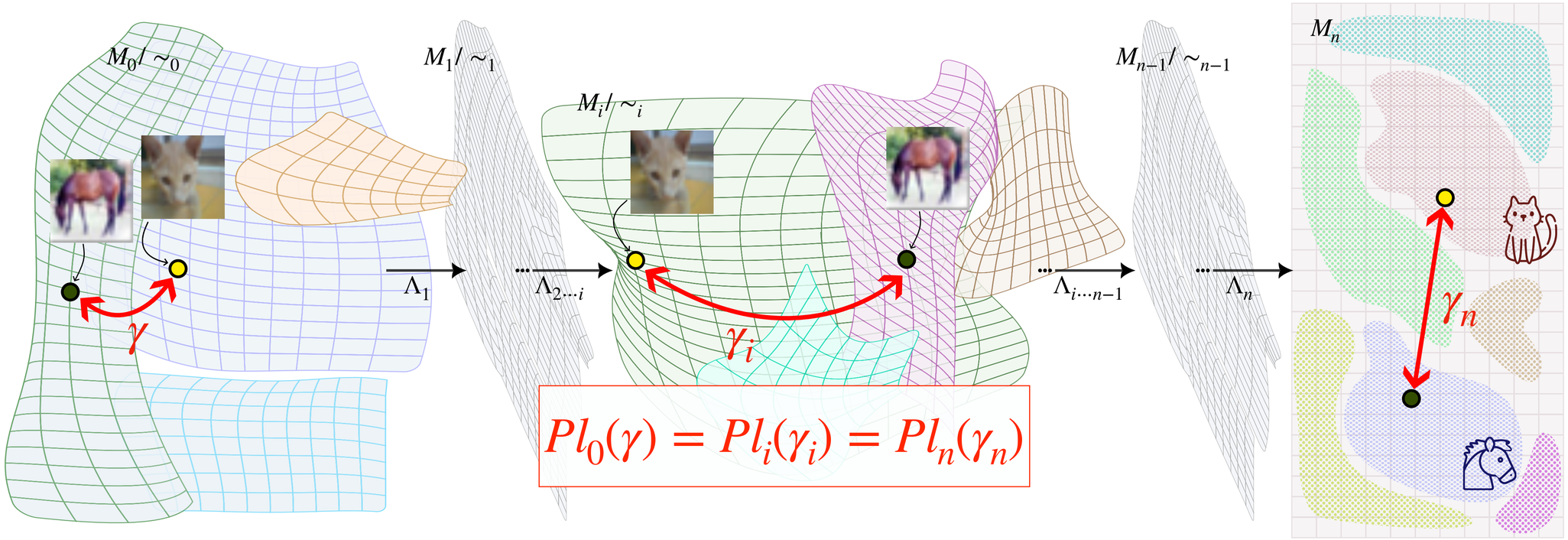
$$\left(f^{*}g\right)_{ij} = \sum_{h,k=1}^{q} \left(\frac{\partial f_h}{\partial x_i}\right) g_{hk} \left(\frac{\partial f_k}{\partial x_j}\right)$$

We take $f = \Lambda_n \circ \Lambda_{n-1} \circ \ldots \circ \Lambda_1$ and $M_0 = \mathbb{R}^p, M_n = \mathbb{R}^q$ to get the pullback for an neural network.

General Results

Proposition. **Distances** measured by singular metrics g_i **are preserved all the way to** M_n: this is because we have constructed $\left(f^*g\right)_{ij}$ to naturally allow this.

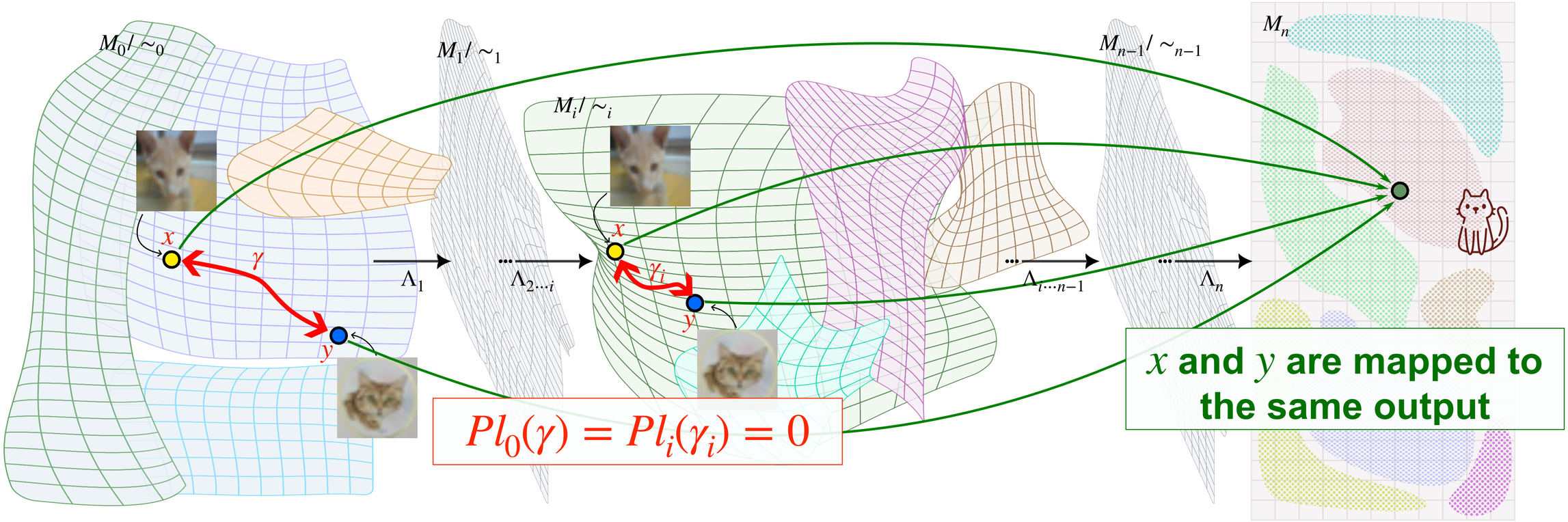
$$Pl_i(\gamma) = Pl_j(\gamma_j) \qquad \left(f^*g\right)_{ij} = \sum_{h,k=1}^{q} \left(\frac{\partial f_h}{\partial x_i}\right) g_{hk} \left(\frac{\partial f_k}{\partial x_j}\right) \qquad f = \Lambda_n \circ \Lambda_{n-1} \circ \ldots \circ \Lambda_1$$



General Results

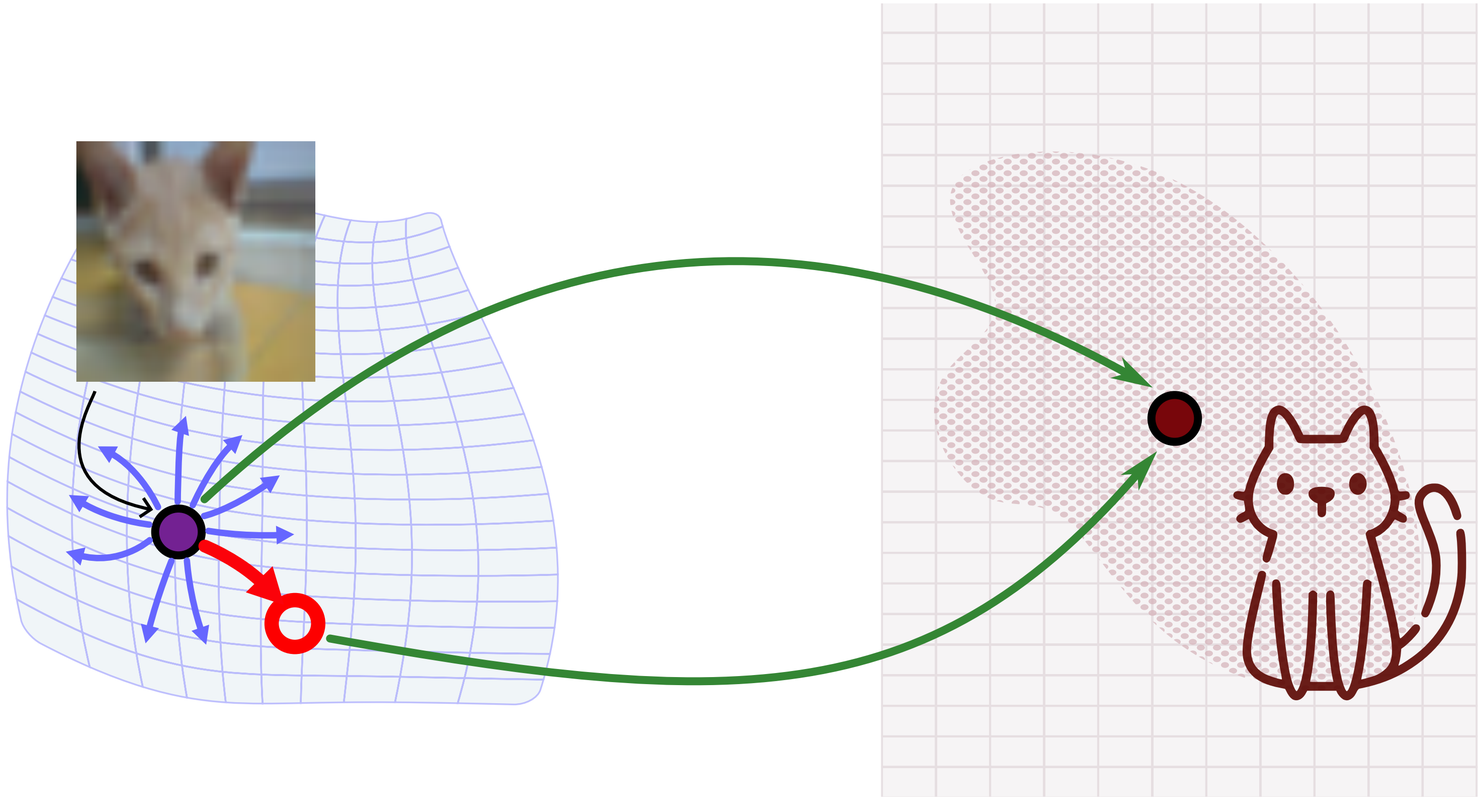
Proposition. **Two points have zero pseudodistance** $\iff$ any point in the path connecting them is **mapped into the same output.**

$$x \sim_i y \iff x \sim_{\mathscr{N}_i} y \qquad \mathscr{N}_{(i)} = \Lambda_n \circ \ldots \circ \Lambda_i : M_i \to M_n$$



General Results

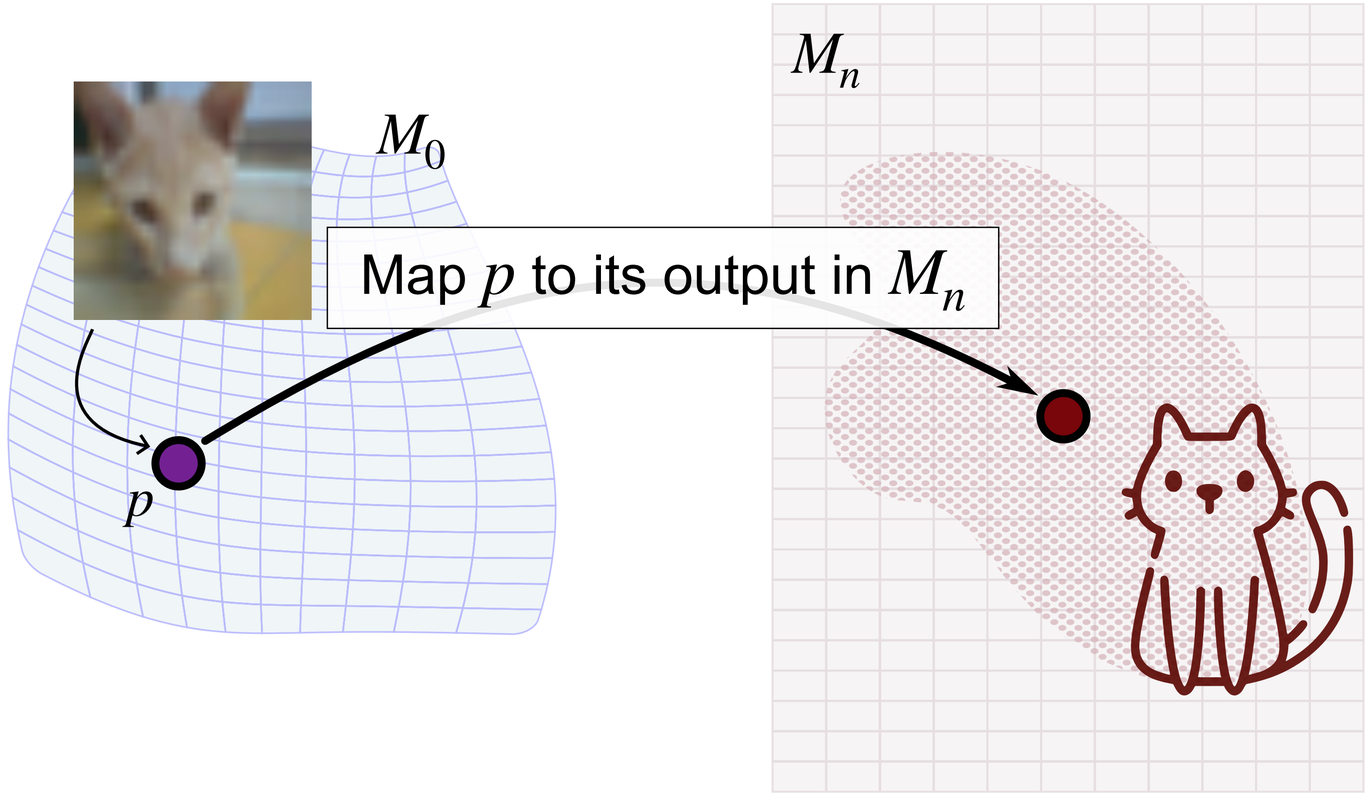
Proposition. **Each equivalence class is a smooth manifold.** The **directions tangent** to these manifolds are precisely the directions of **“zero change”.**



Movement along tangent directions is ignored by the network

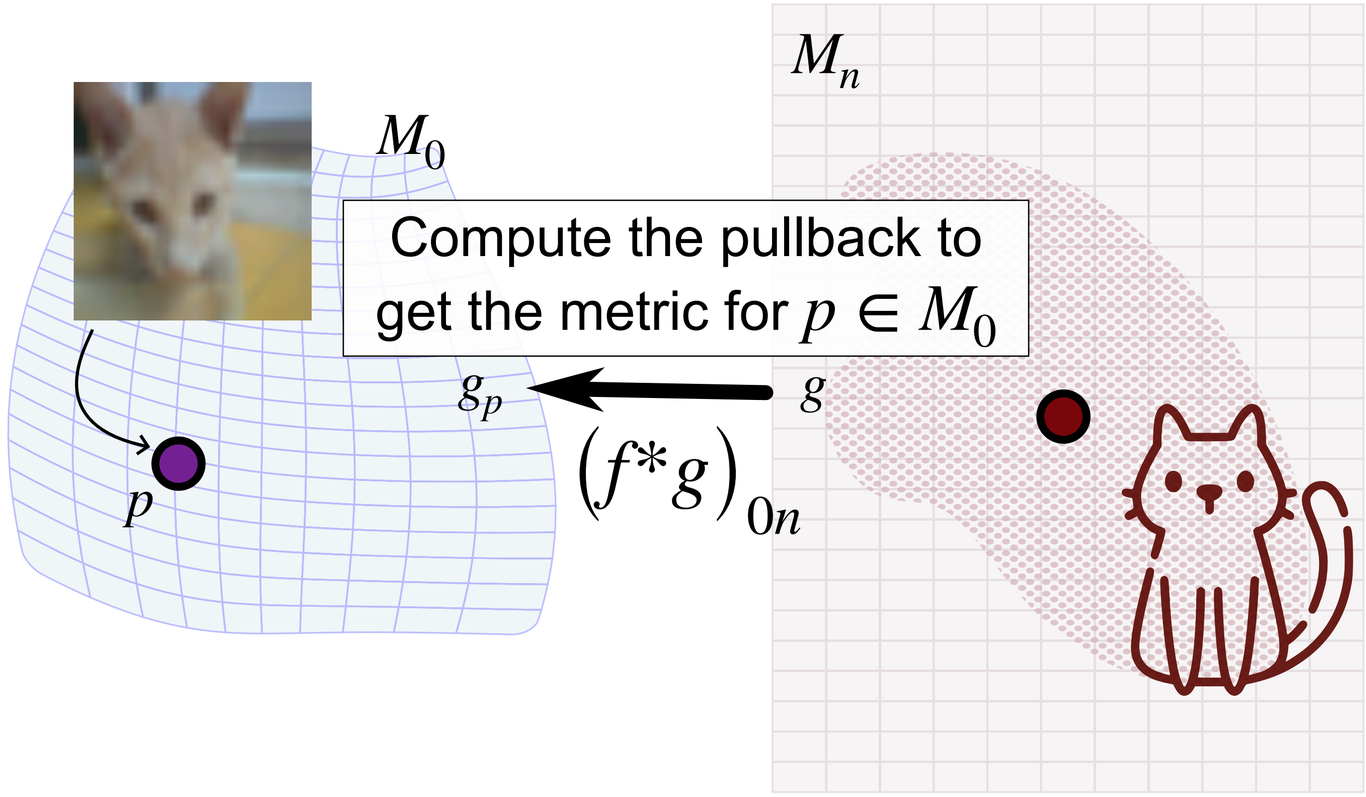
Methodology

Exploring an equivalence class: SiMEC



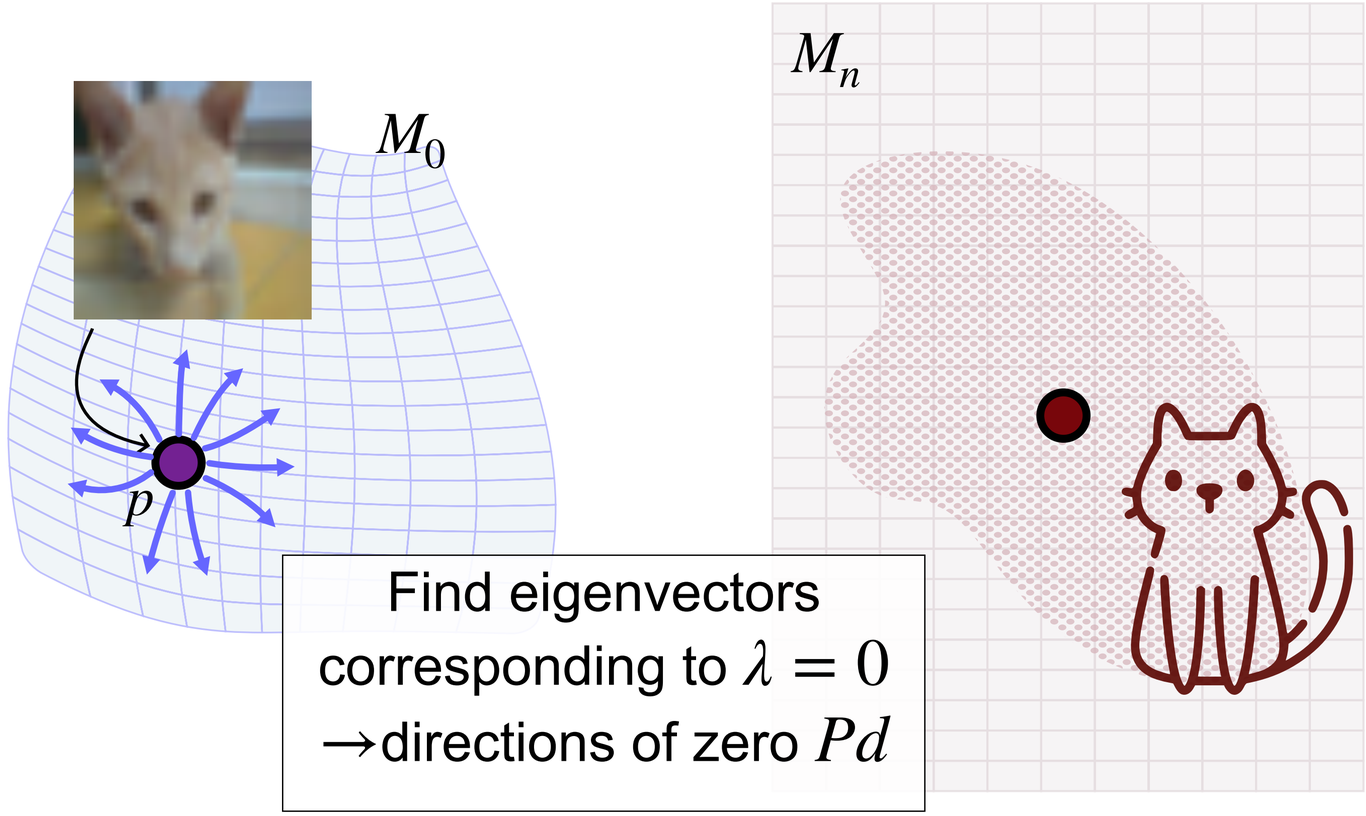
Methodology

Exploring an equivalence class: SiMEC



Methodology

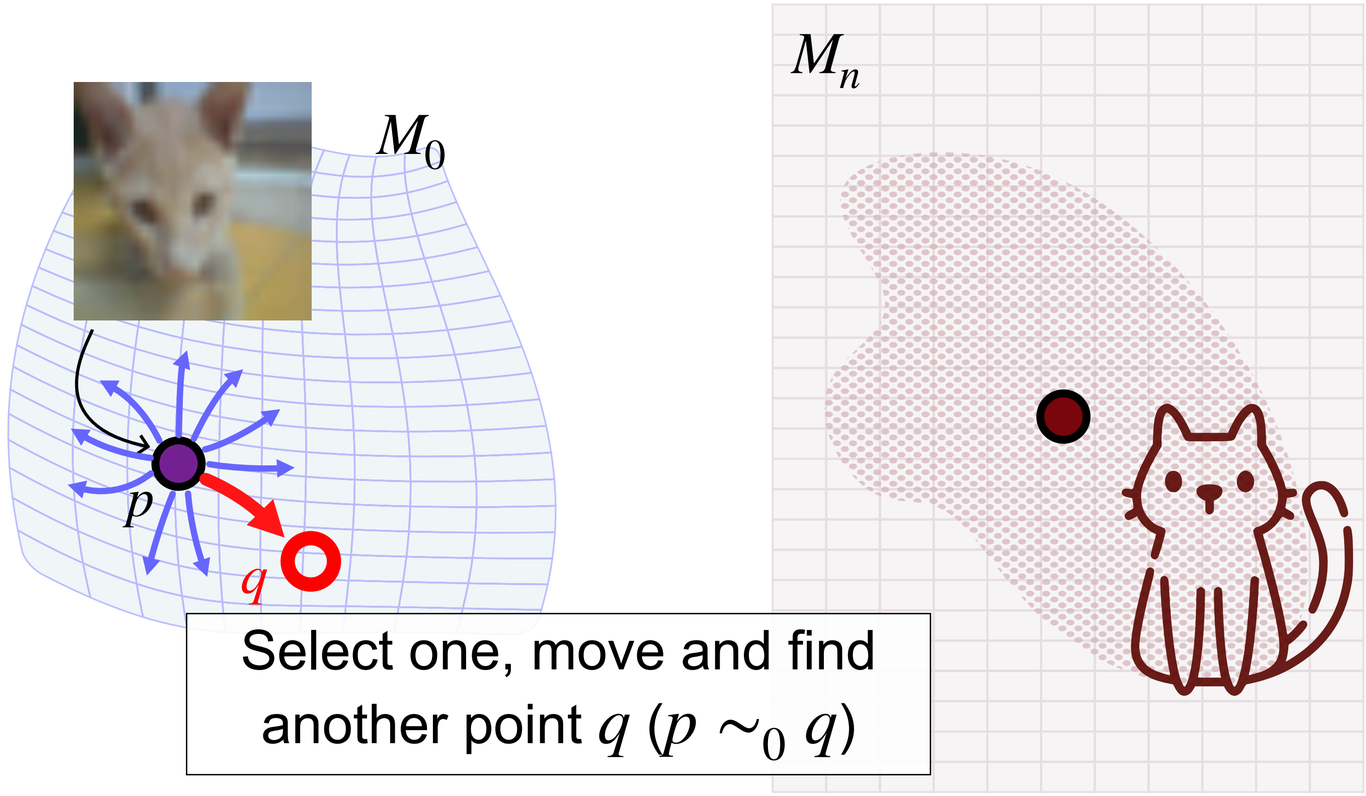
Exploring an equivalence class: SiMEC



Find eigenvectors corresponding to $\lambda = 0$
$\rightarrow$directions of zero Pd

Methodology

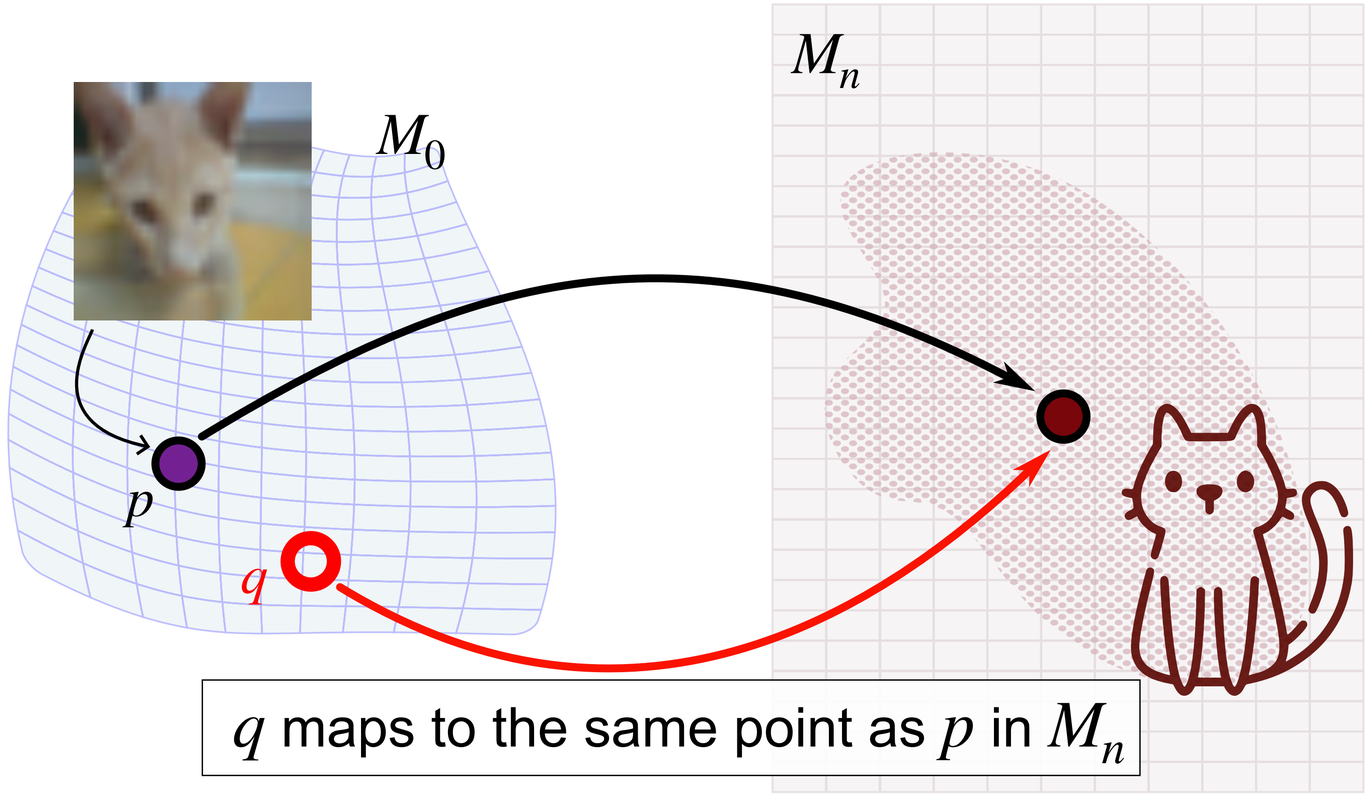
Exploring an equivalence class: SiMEC



Select one, move and find another point q ($p \sim_0 q$)

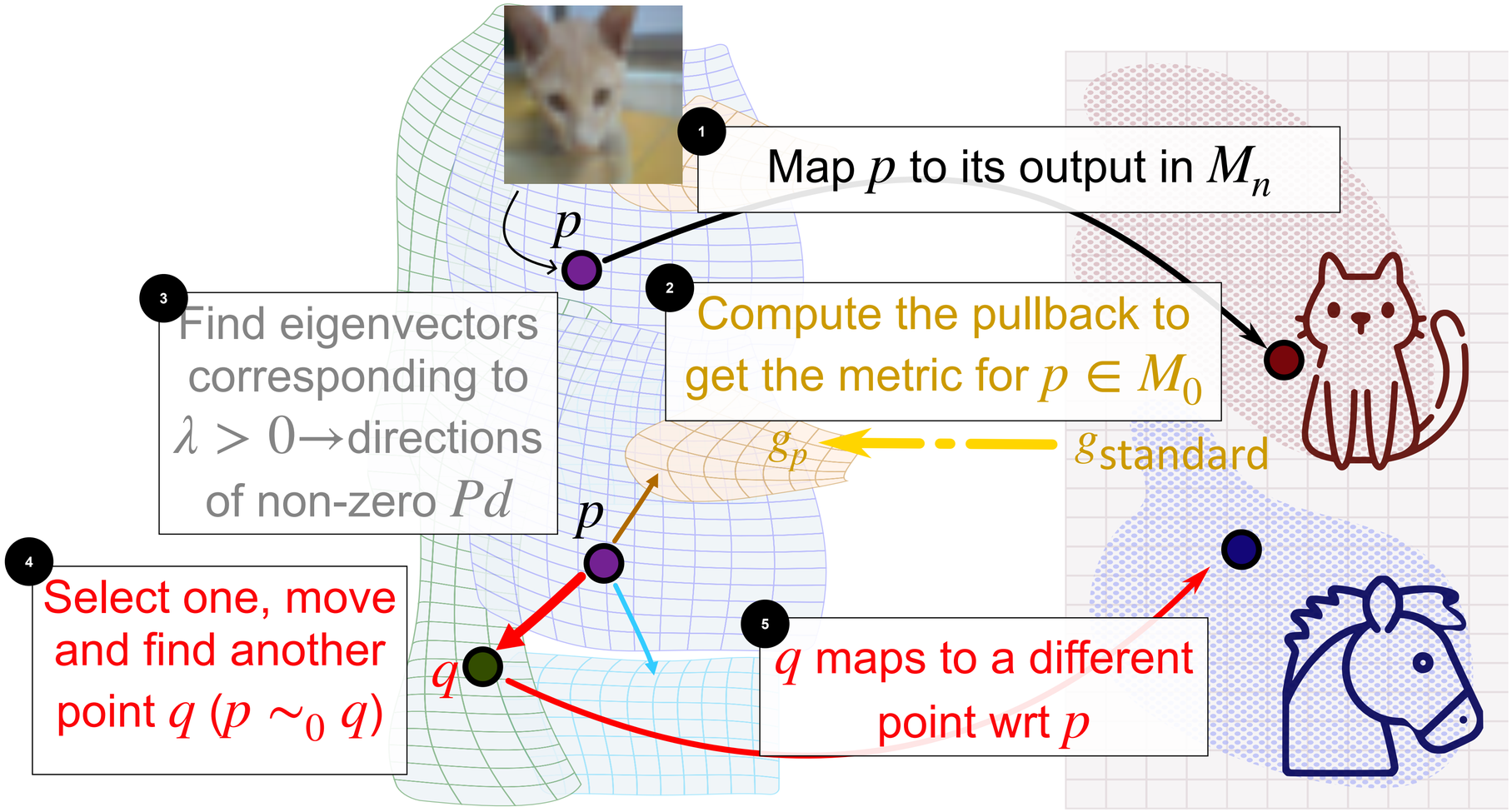
Methodology

Exploring an equivalence class: SiMEC



Methodology

Exploring toward a *different* equivalence class: SiMExp



Experiments

SiMEC and SiMExp on different models and data modalities:

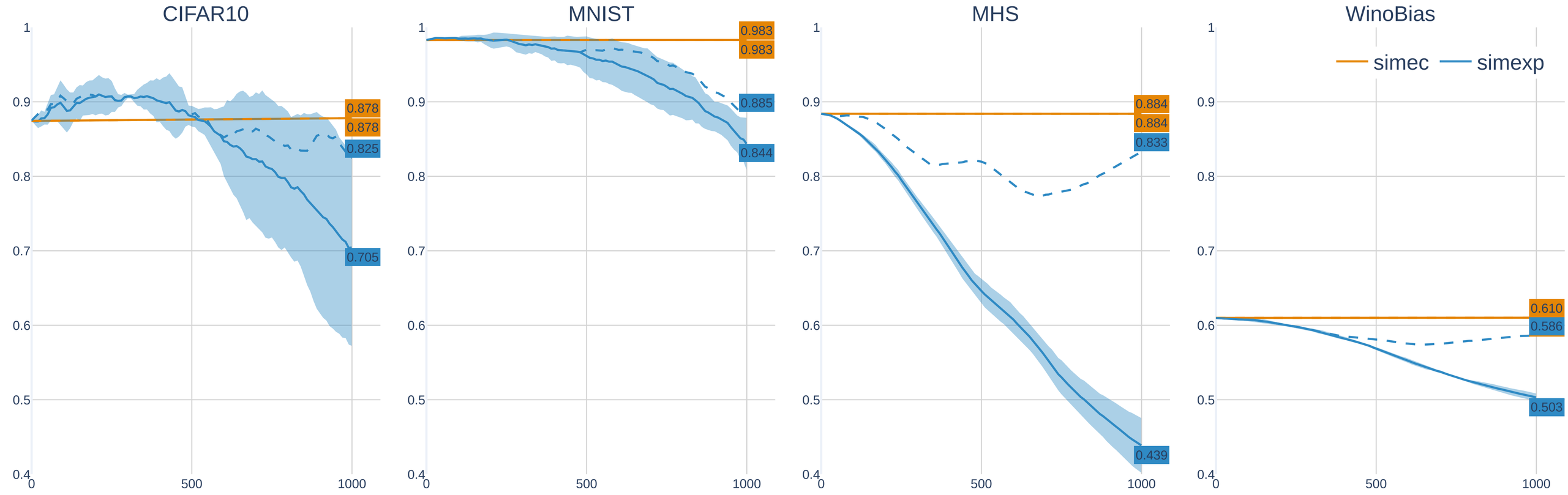
- ViT on CIFAR and MNIST image classification
- BERT on Measuring Hate Speech (classification) and WinoBias (masked prediction)

NEURAL INFORMATION PROCESSING SYSTEMS

Experiments

Empirical validation for SiMEC and SiMExp

Original and top classes' probabilities from embeddings, across datasets



Conclusions

- Introduced a Riemannian geometry-based framework for exploring Transformer input spaces
- Developed SiMEC and SiMExp for the exploration of equivalence classes
- Validated on different models and data modalities

Thank you for listening!