# Persistent Topological Features in Large Language Models

Matteo Biagetti

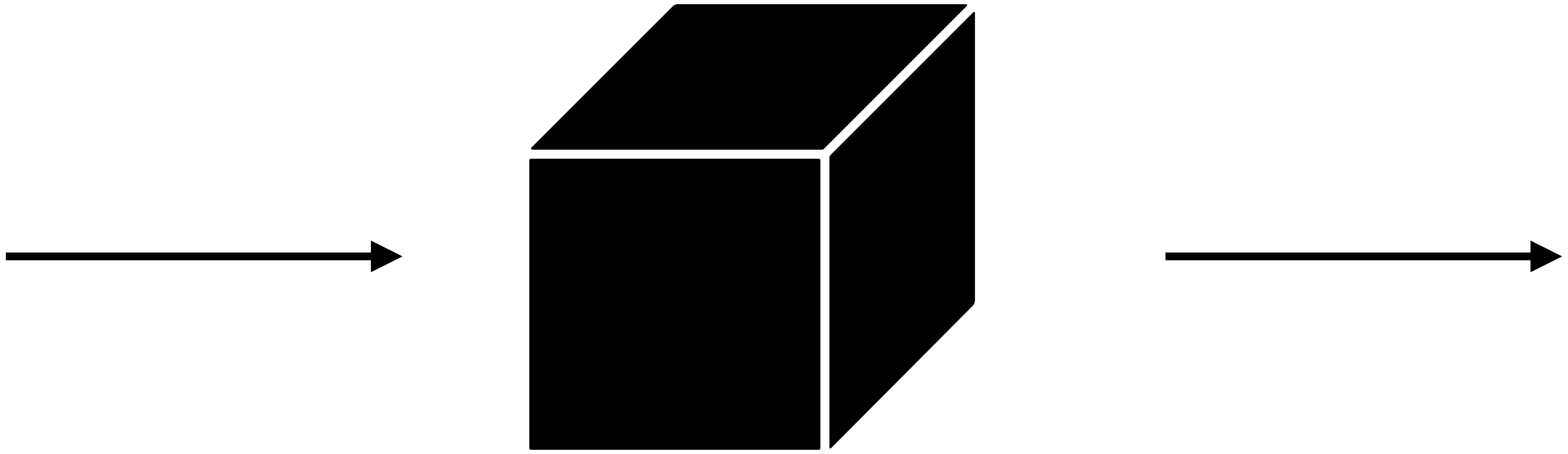
AREA SCIENCE PARK

1st SMASHING Workshop - 9th October 2024

PNR Programma Nazionale per la Ricerca
FAIR-by-design

With Yuri Gardinazzi, Giada Panerai, Karthik Viswanathan, Alberto Cazzaniga (arXiv:2410.XXXXX)

# Large Language Models are black boxes

Text  Output

***Problem***: black box system with $\mathcal{O}\left(10^9\right)$ tuned parameters. Not really possible to

1. Understand what goes on inside
2. Evaluate incorrect or unsafe behaviour
3. Optimize inefficiency in a systematic way

Given the widespread applications, we need to understand the **decision-making** process

Singh et al. 2024

# Internal Representations of LLMs

input

The quick brown fox jumps over the lazy dog

# Internal Representations of LLMs

input

The quick brown fox jumps over the lazy dog

map into $\mathbb{R}^d$

$\vec{x}_1$ $\vec{x}_2$ $\vec{x}_N$

Each input: $\vec{x}_i \in \mathbb{R}^d$ **token**

Sequence $\{\vec{x}_1, \ldots, \boxed{\vec{x}_N}\}$ **prompt**

# Internal Representations of LLMs

input

The quick brown fox jumps over the lazy dog

map into $\mathbb{R}^d$

$\vec{x}_1$ $\vec{x}_2$ $\vec{x}_N$

Each input: $\vec{x}_i \in \mathbb{R}^d$ **token**

Sequence $\{\vec{x}_1, \ldots, \boxed{\vec{x}_N}\}$ **prompt**

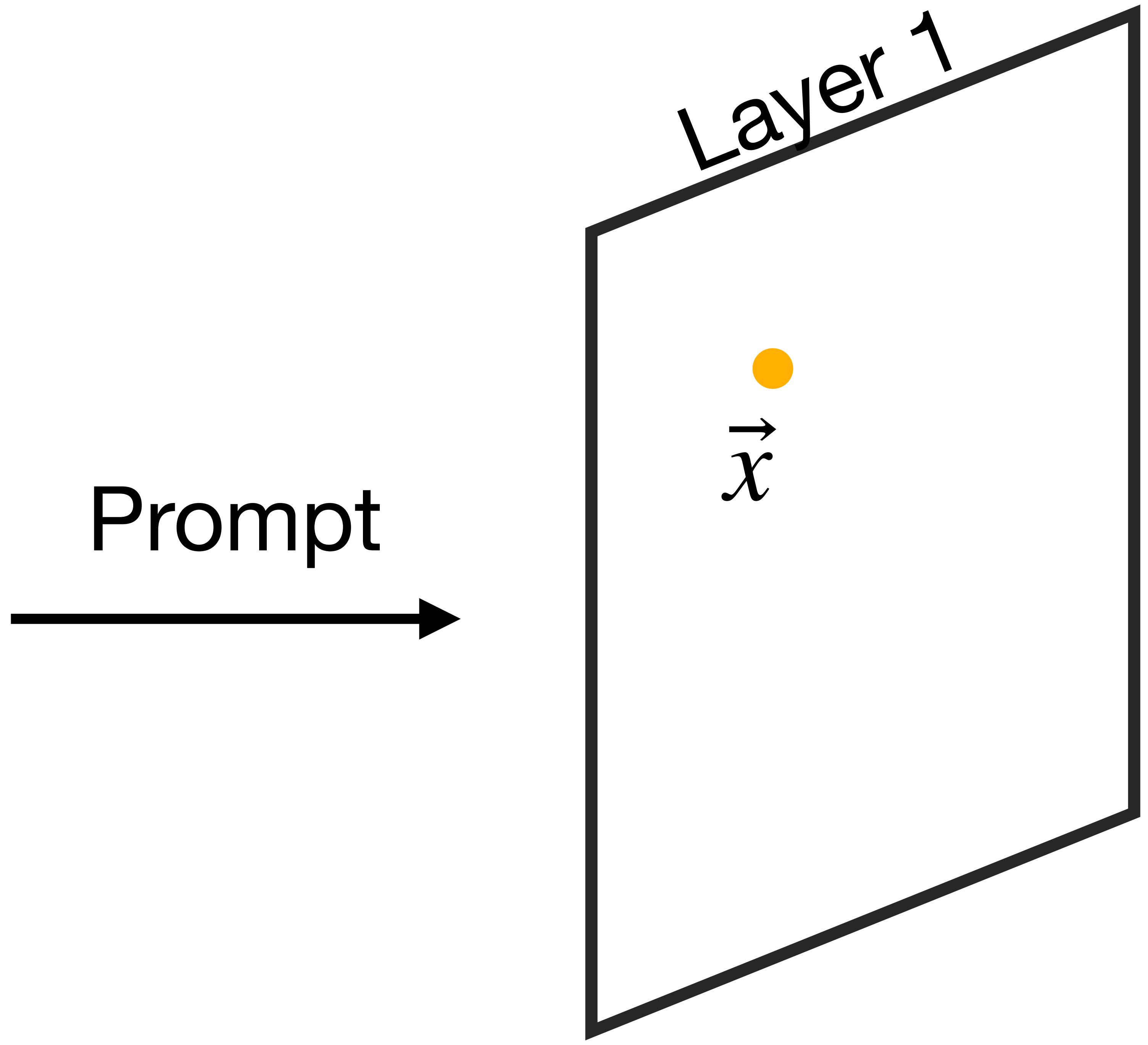
Task: predict next token

$d \approx \mathcal{O}(10^3)$

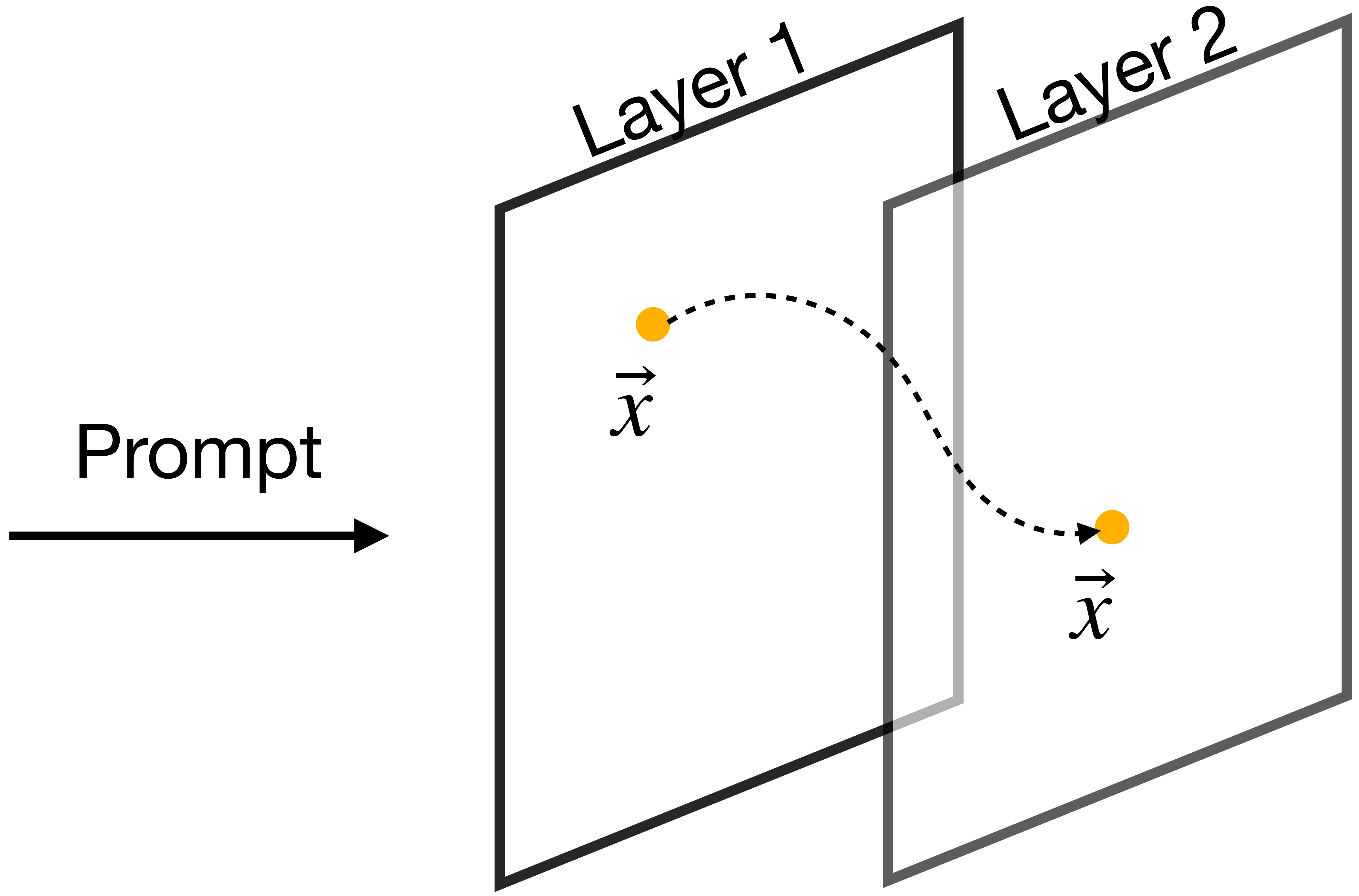
$\boxed{\vec{x}_N \equiv \vec{x} \quad \textbf{Last token}}$

It contains most information on whole sequence

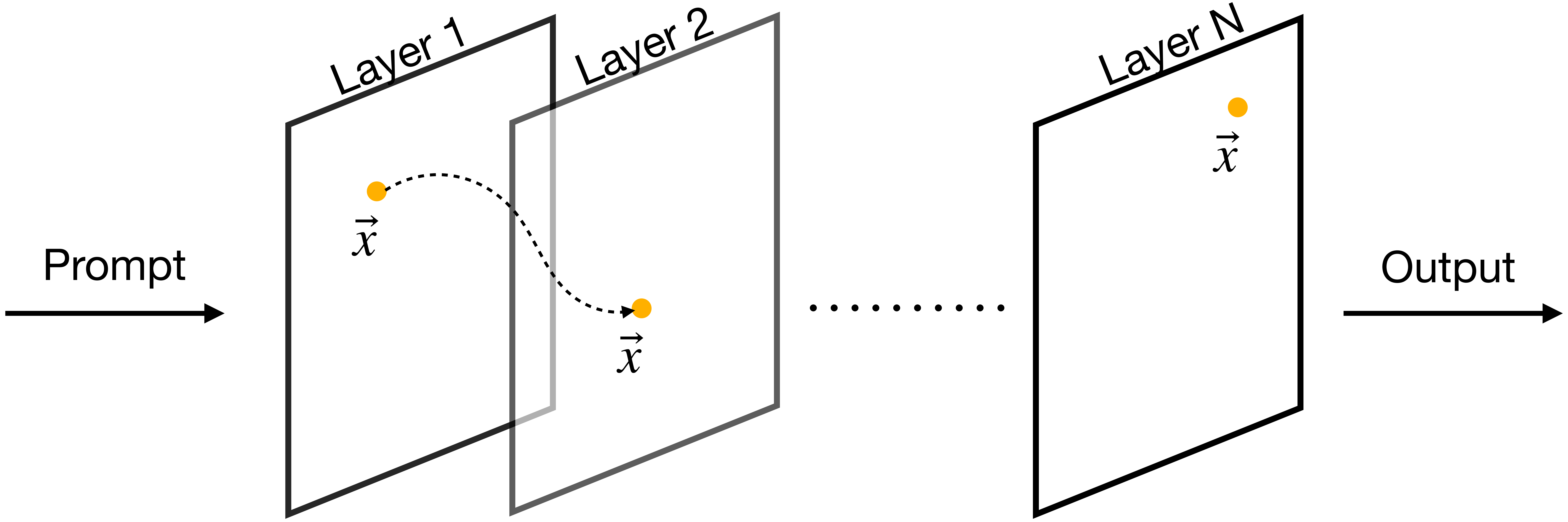
# Internal Representations of LLMs



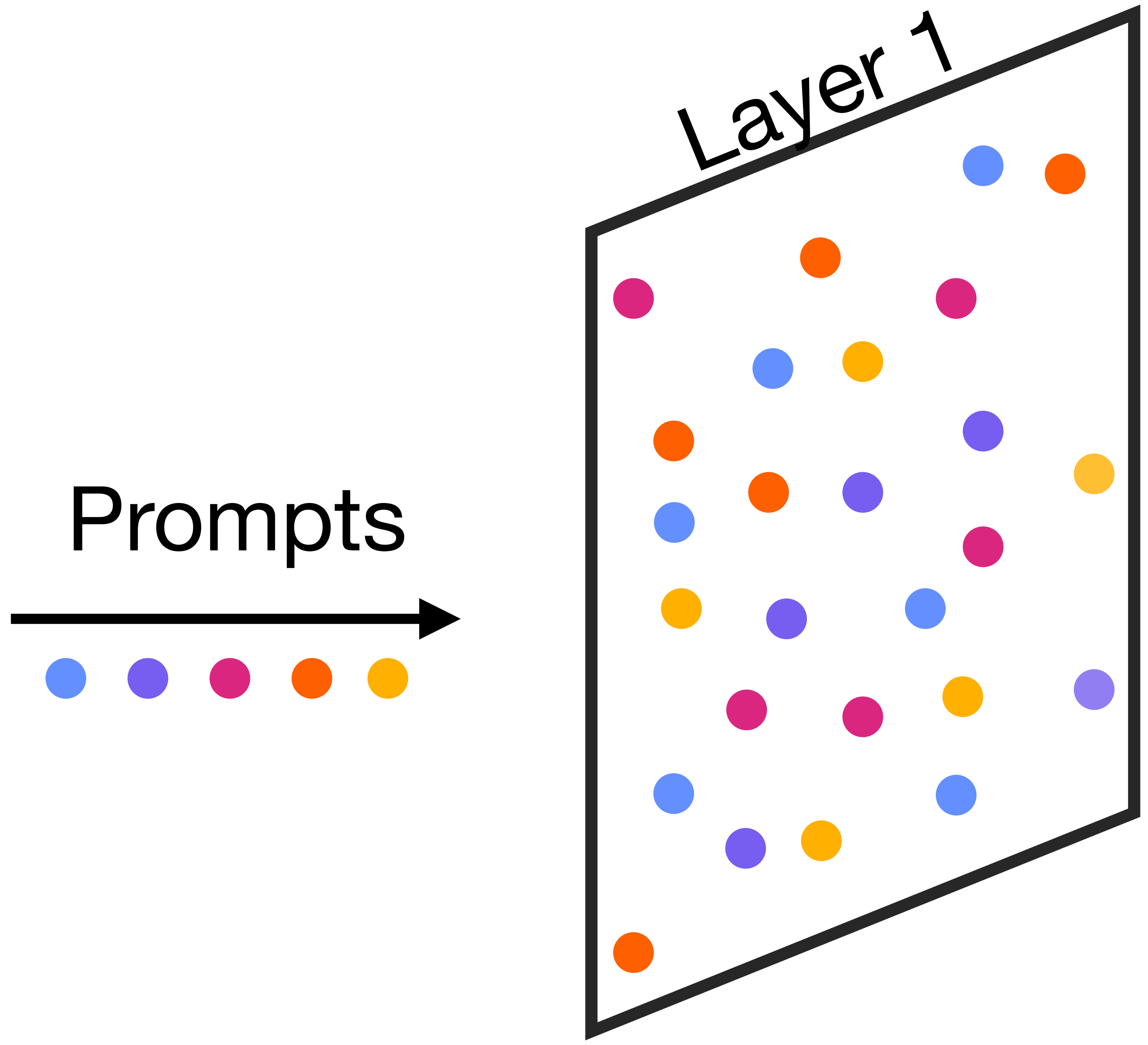
# Internal Representations of LLMs



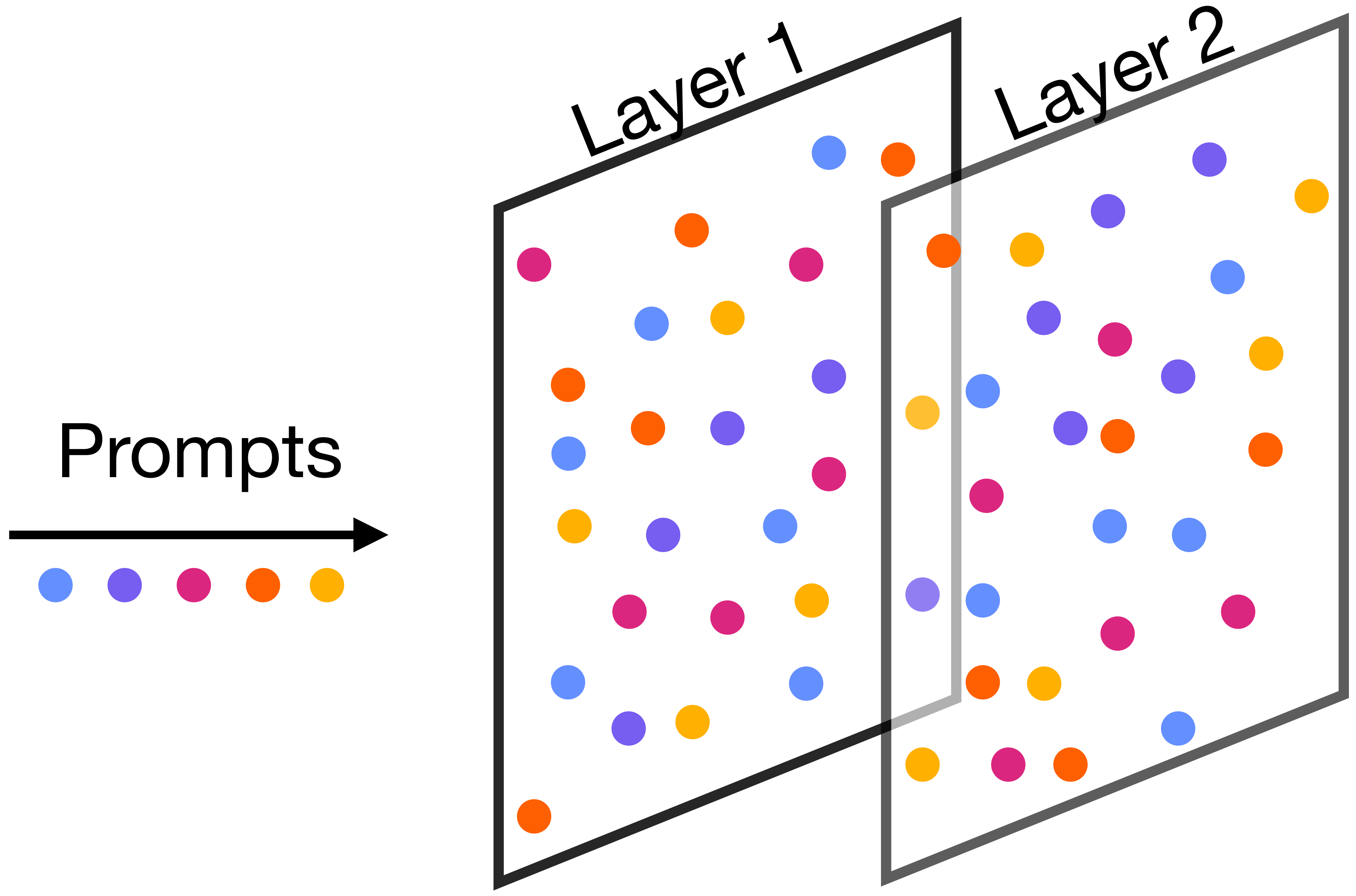
# Internal Representations of LLMs



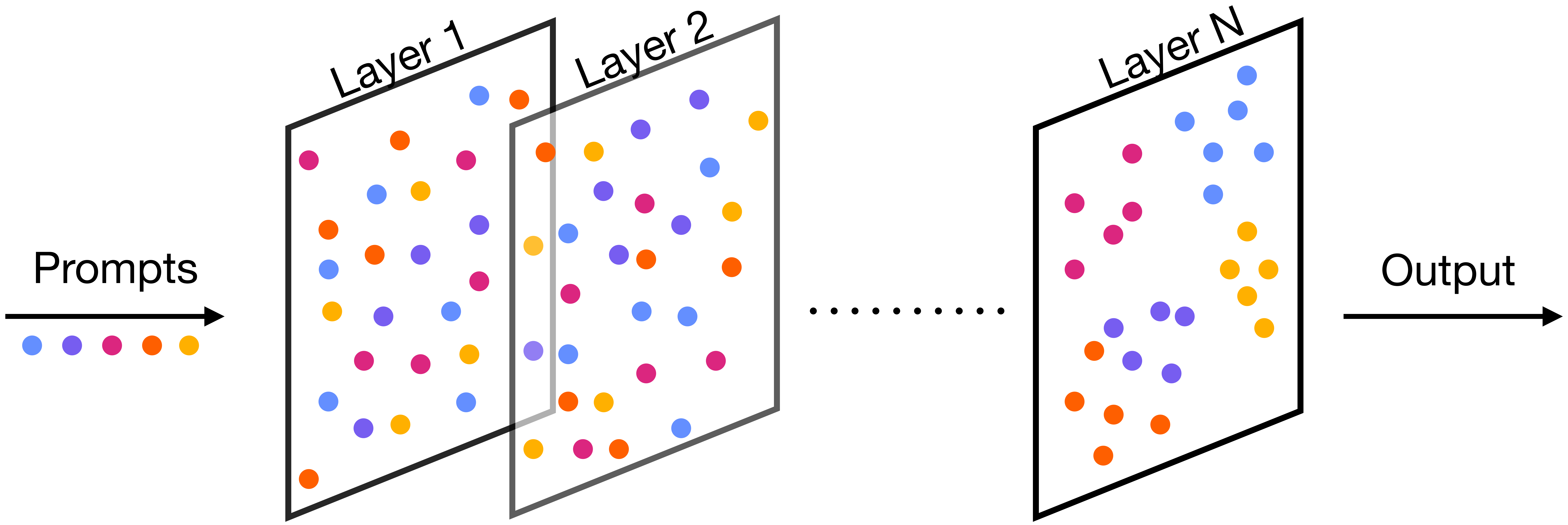
# Internal Representations of LLMs



# Internal Representations of LLMs



# Internal Representations of LLMs




***Hypothesis***: distribution of prompts in representation space related to model's inner workings

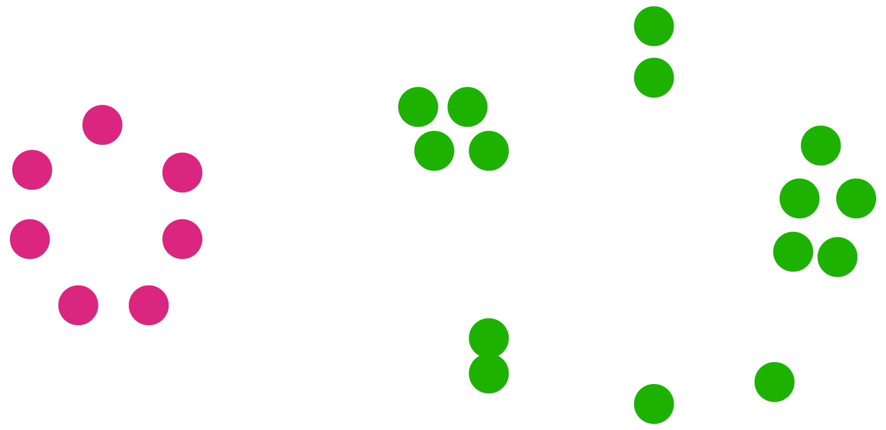
***Strategy***: Analyse representations using *topological data analysis*

***Goal***: describe global features of LLMs

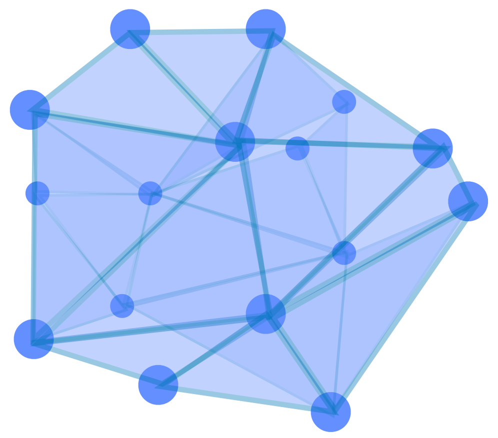
# Topological Data Analysis

Calculating the shape of data

Clustering of points



Loops or rings

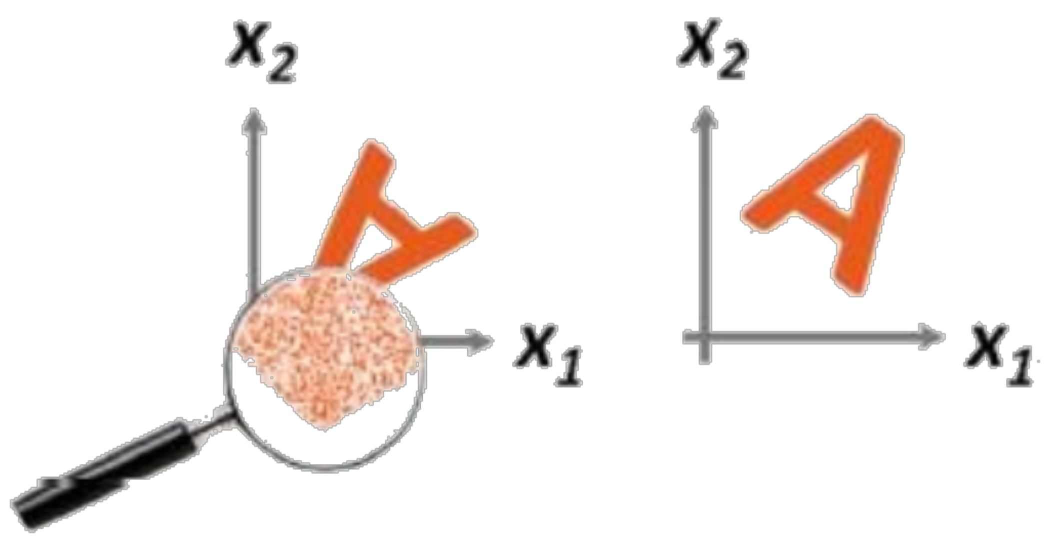


Voids in higher dimensions

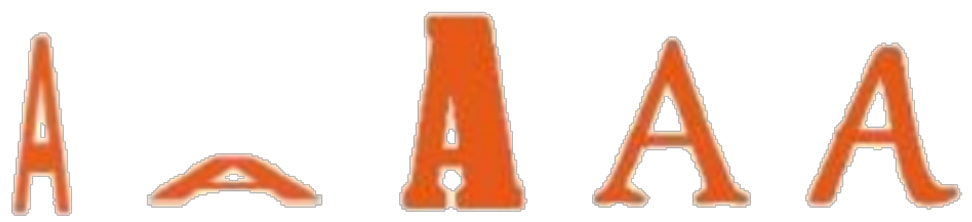
# Topological Data Analysis

Calculating the shape of data

- Coordinate Invariance




- Deformations Invariance



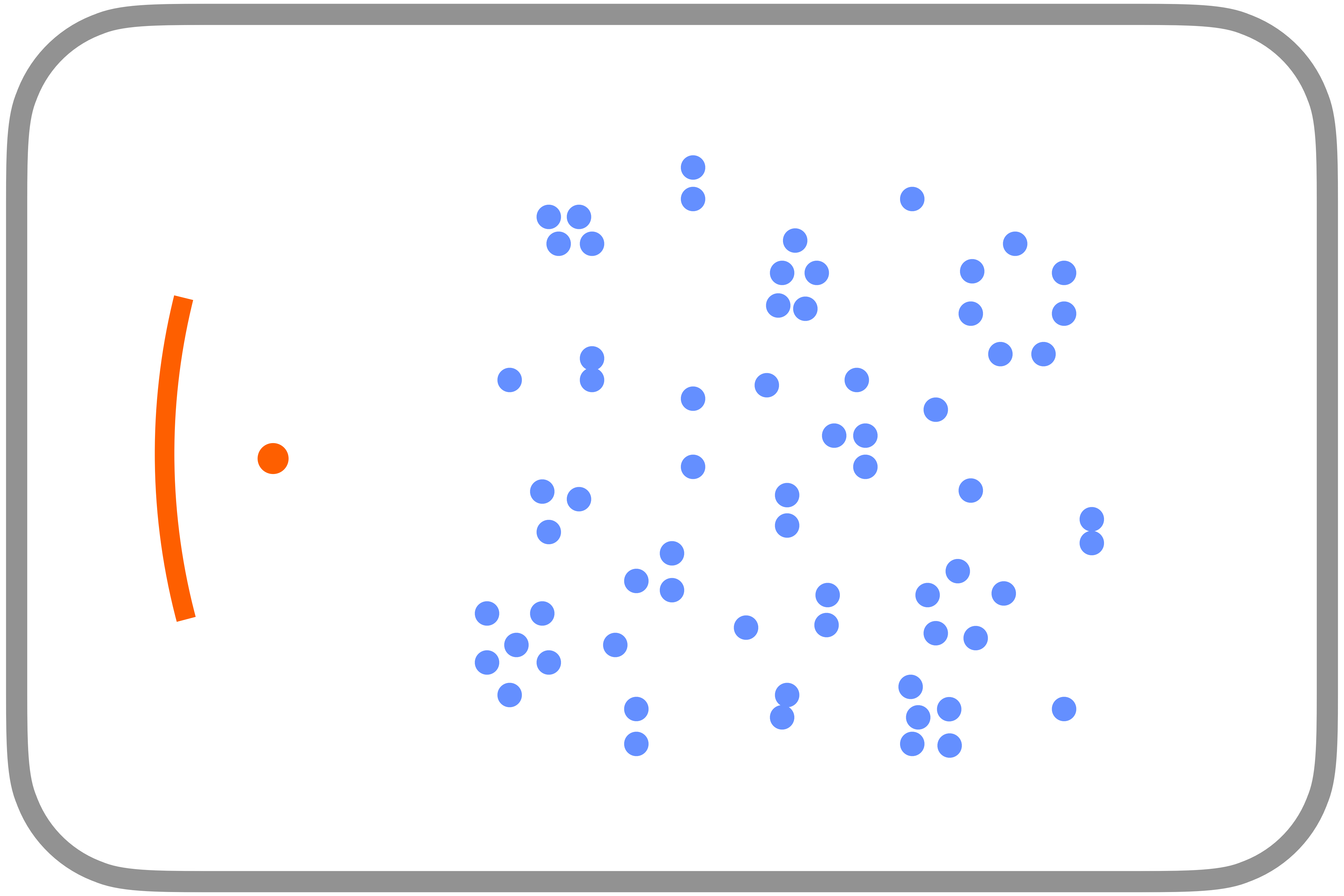

- Information compression


*Input: millions of data points with similarity relationships.*

(Ohanuba et al. 2021)

# Topological Data Analysis

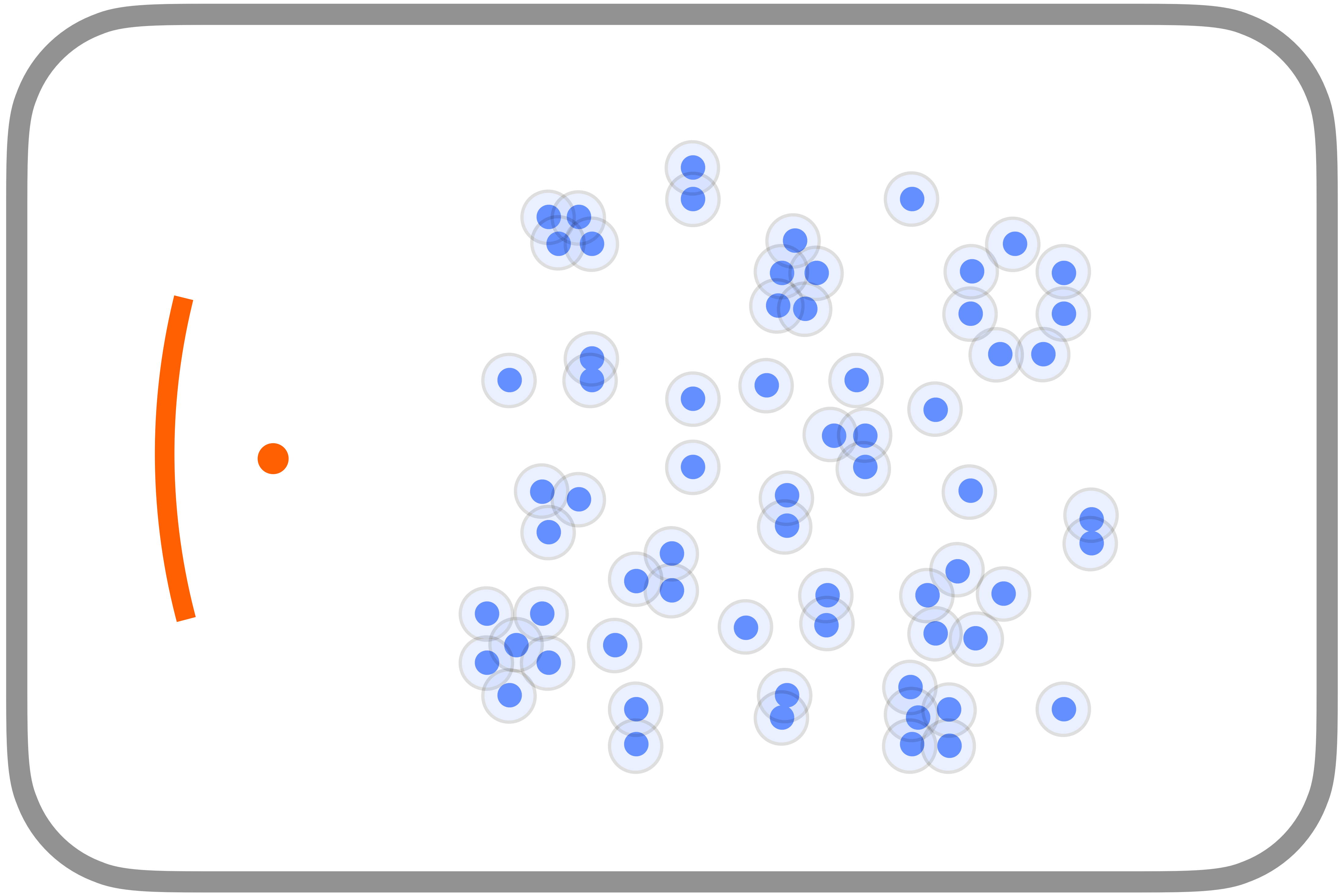
Calculating the shape of data



Example: audience distribution at the SMASHING workshop

# Topological Data Analysis

Calculating the shape of data

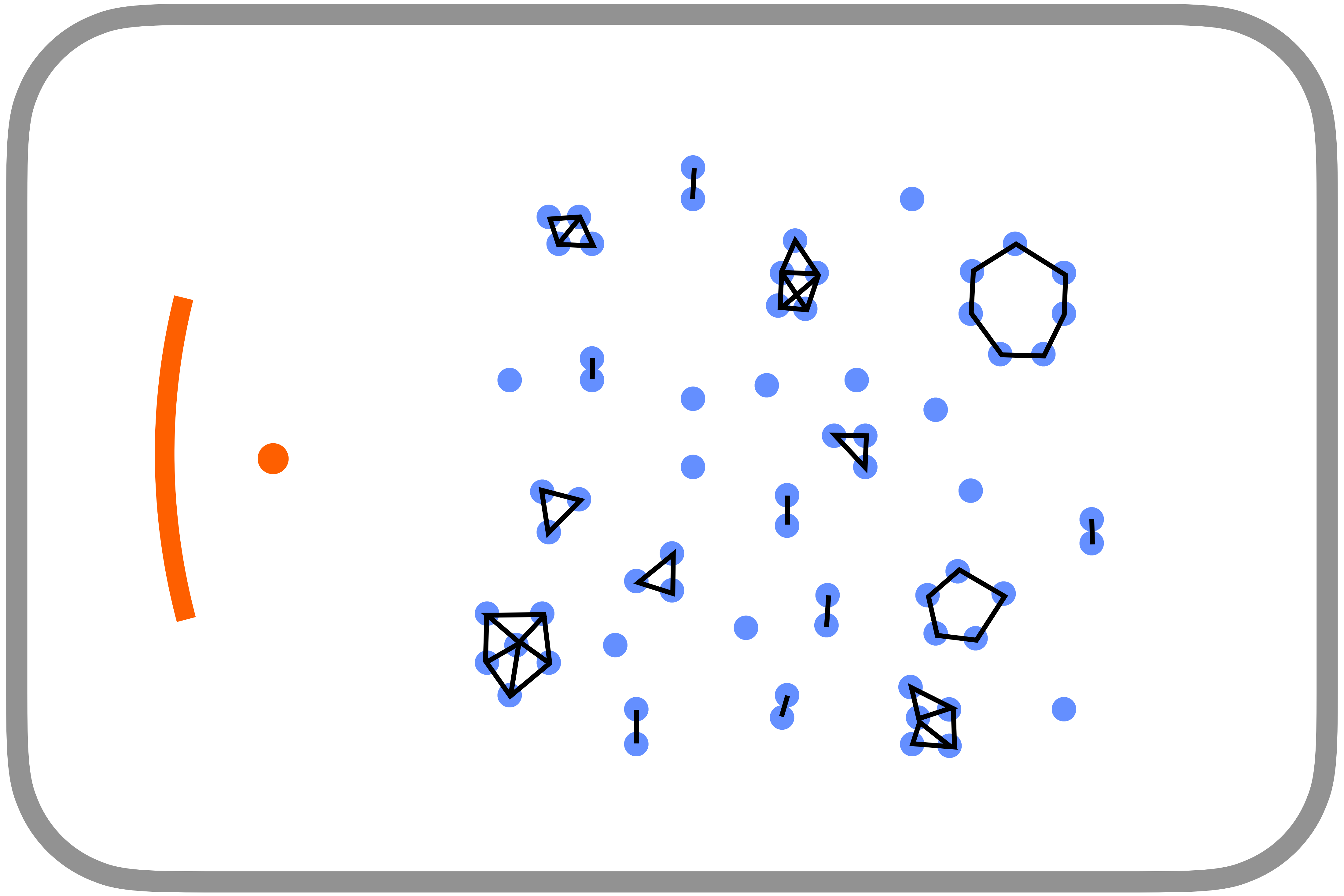


If we stick our elbows out, which neighbours do we touch?

# Topological Data Analysis

Calculating the shape of data

How many clusters?

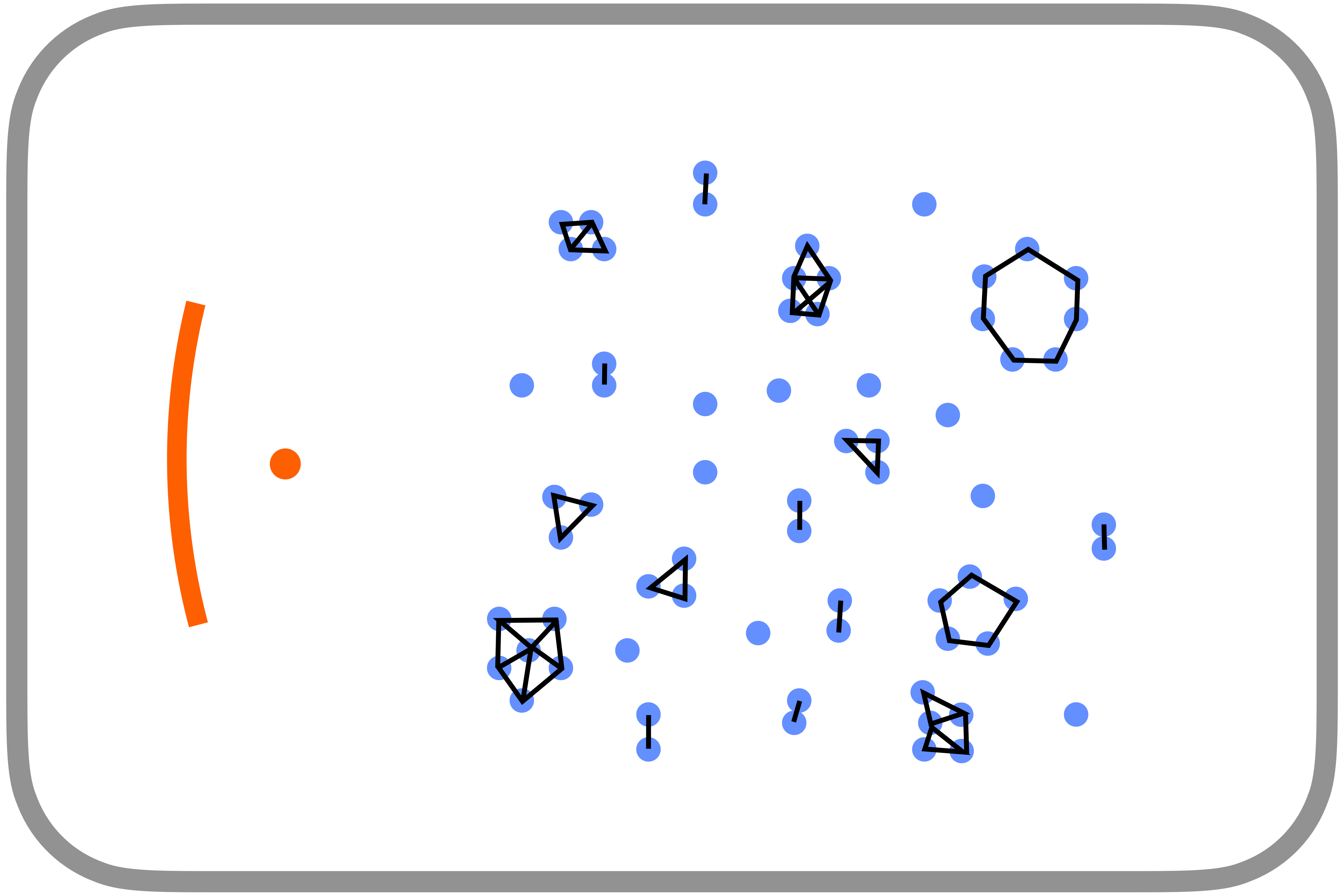


How many loops?

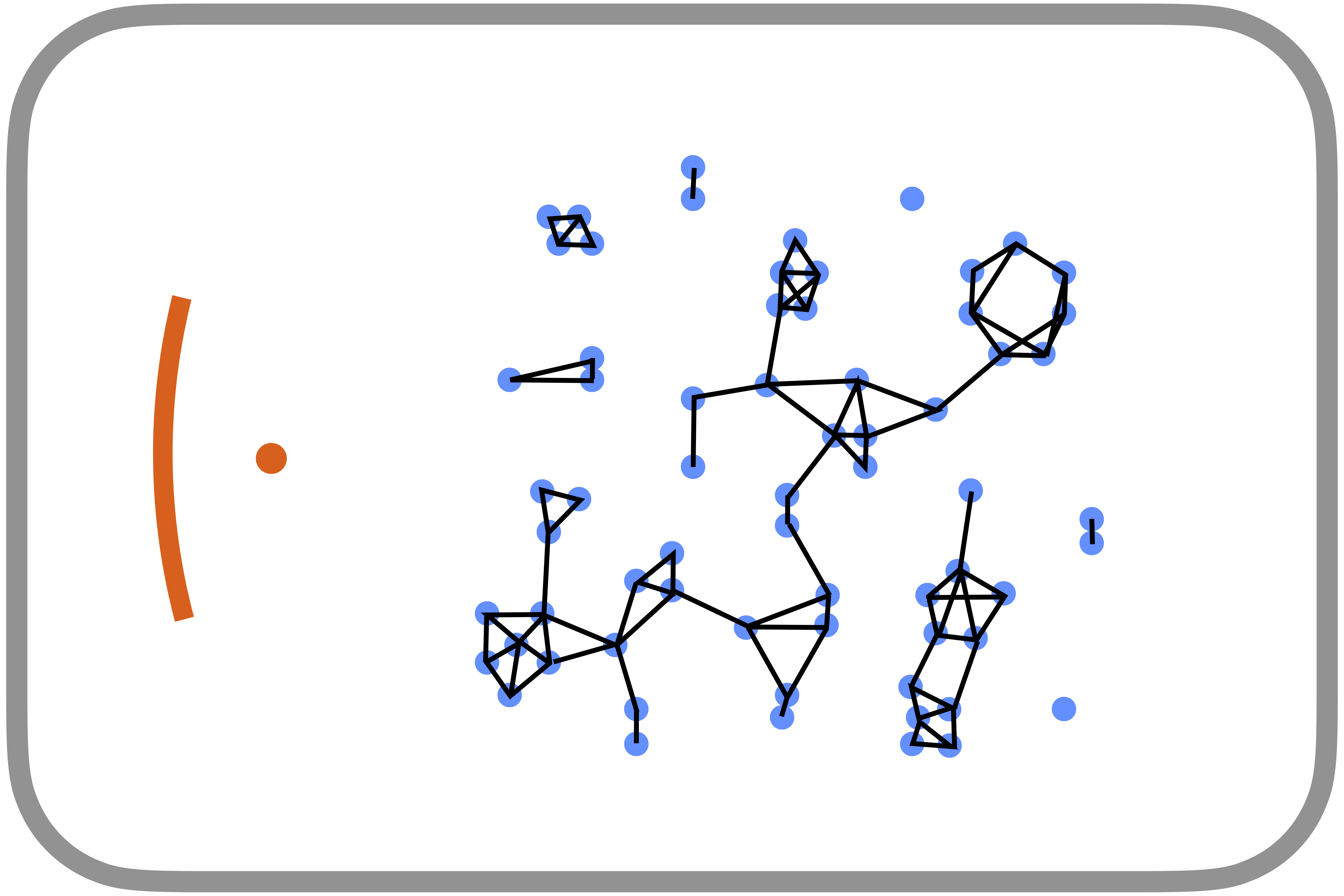
If we stick our elbows out, which neighbours do we touch?

# Topological Data Analysis

Elbows



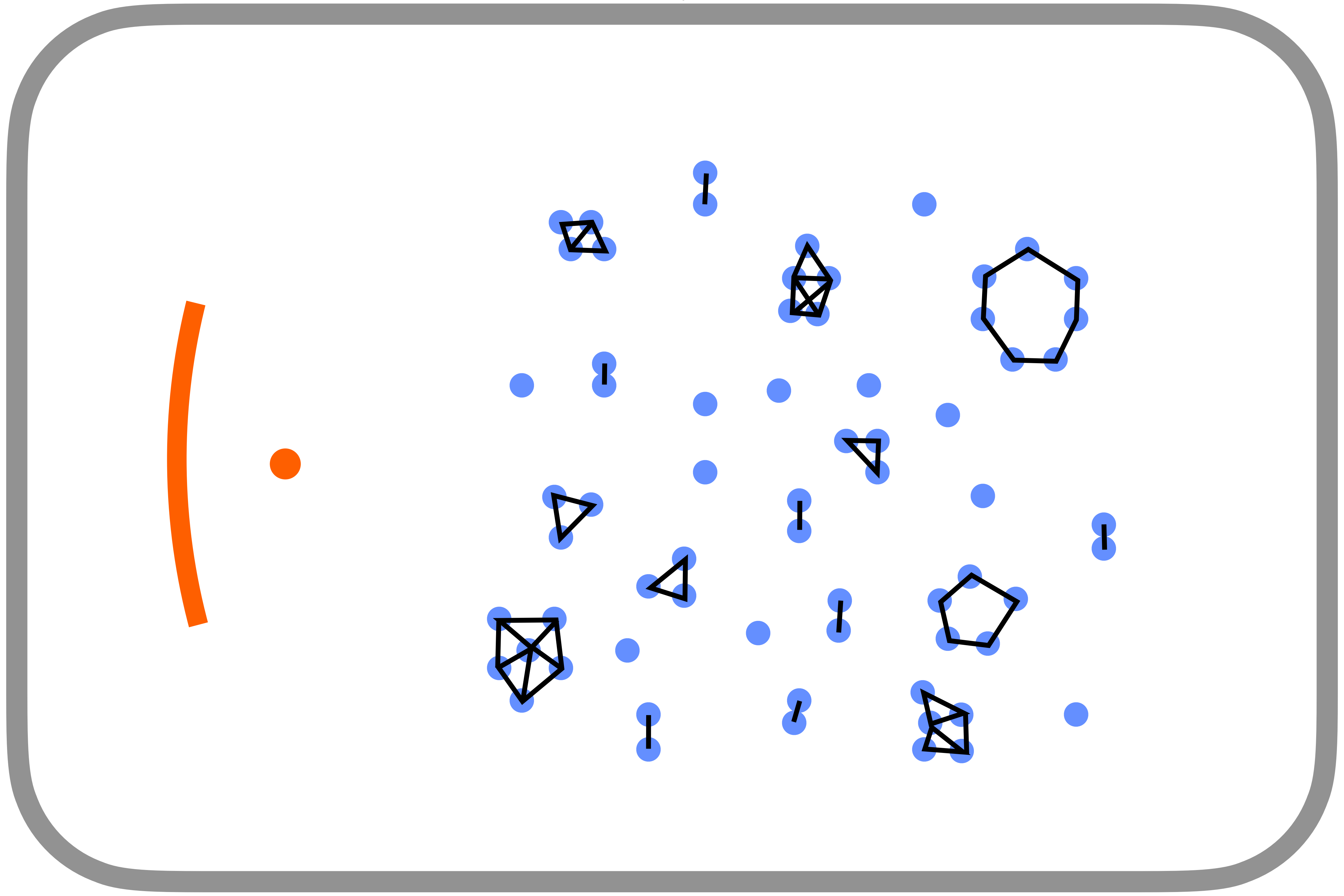
Arms



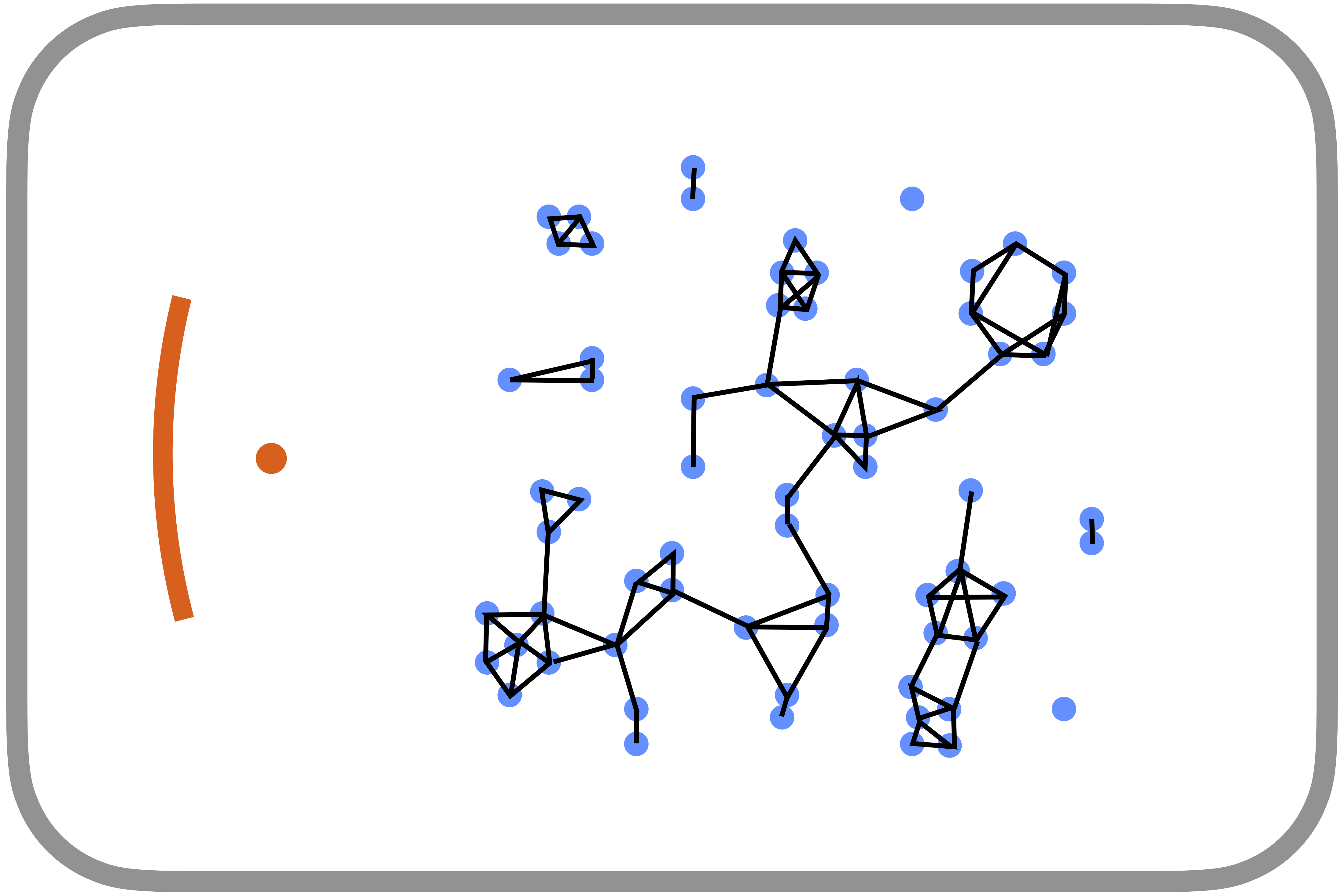
Clusters and loops that remain at varying radius are defined as persistent and considered relevant features of the dataset

# Topological Data Analysis

Layer 1

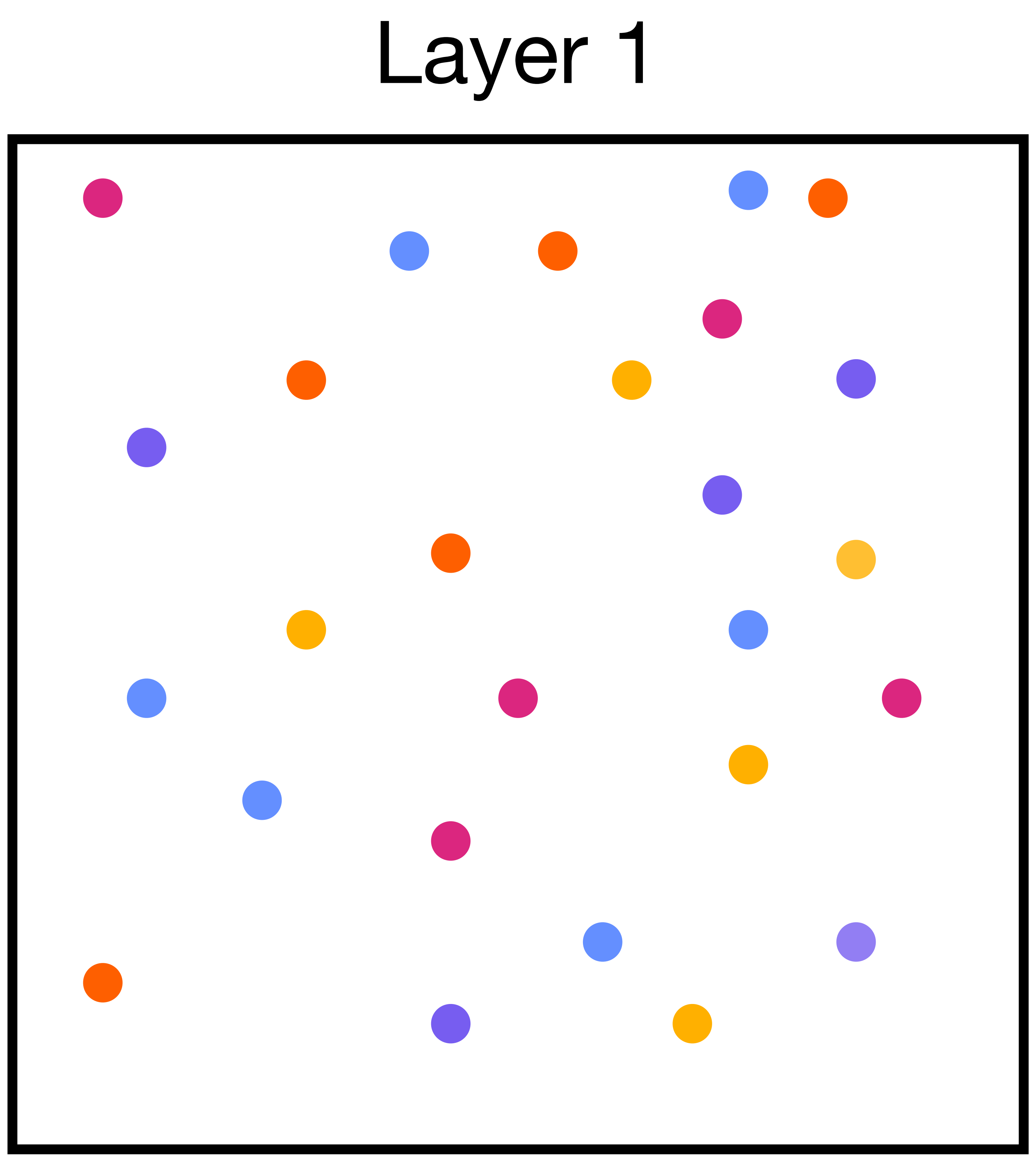


Layer 2



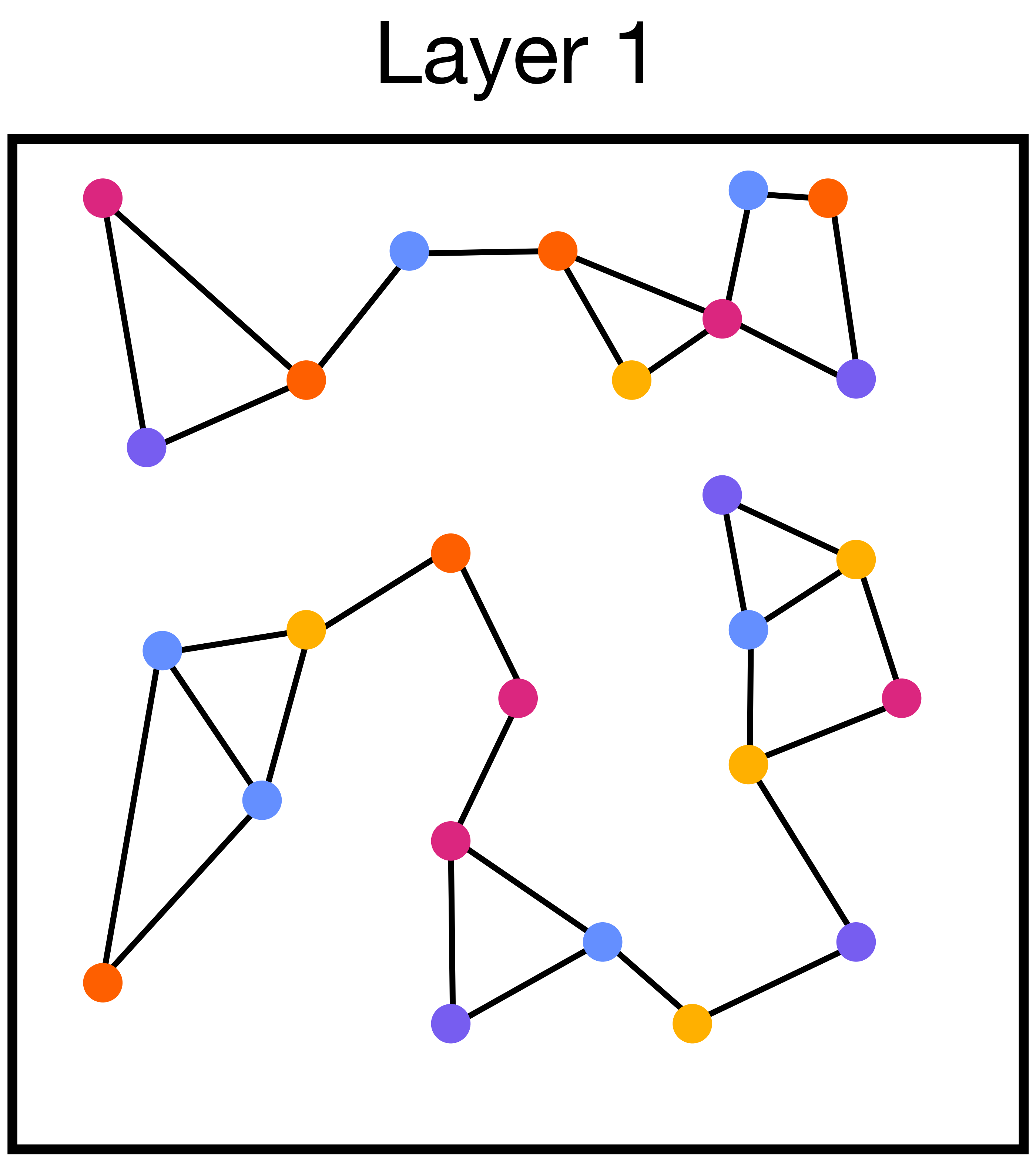
Clusters and loops that remain at varying time are defined as persistent and considered relevant features of the dataset

# The Zig Zag Algorithm




**First step**: connecting points

# The Zig Zag Algorithm

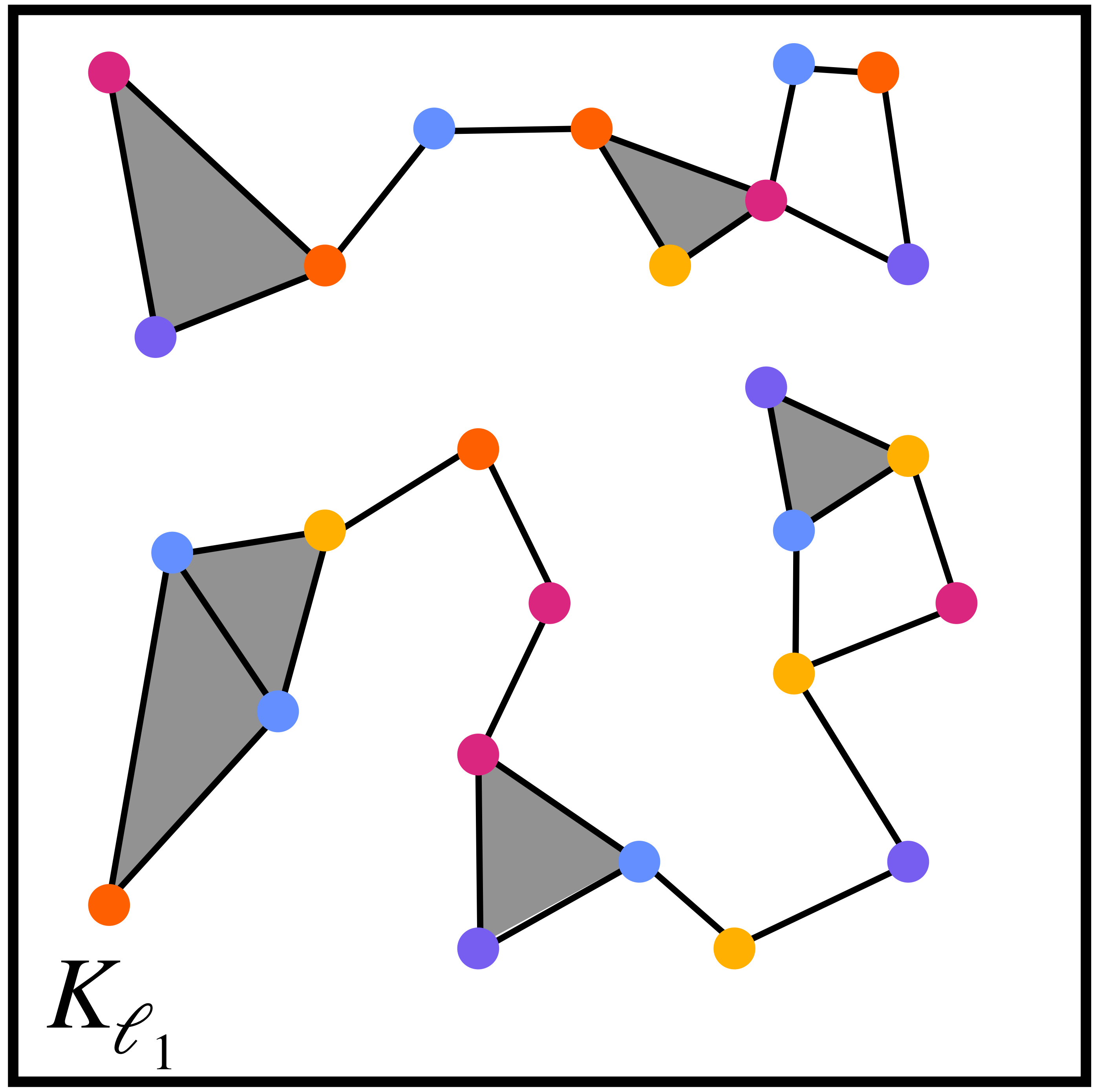



**First step**: connecting points

$k$-Nearest-Neighbours graph

# The Zig Zag Algorithm

Layer 1

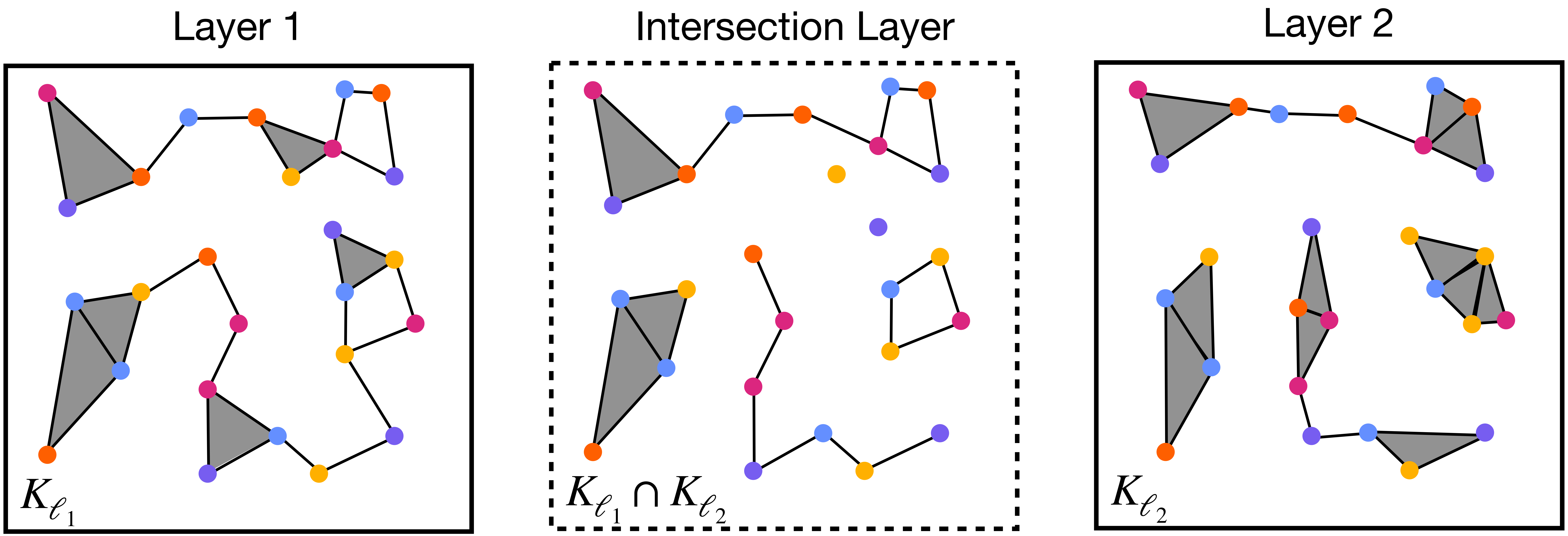



**Second step**: Simplicial Complex

Three adjacent edges are triangles
Six adjacent edges are tetrahedra
...

$$K_{\ell_i} = \bigcup_{S \subseteq V_{\ell_i}} \left\{ S \,\middle|\, \forall x_s, x_l \in S,\ (x_s, x_l) \in E_{\ell_i} \text{ and } |S| \leq m+1 \right\}$$

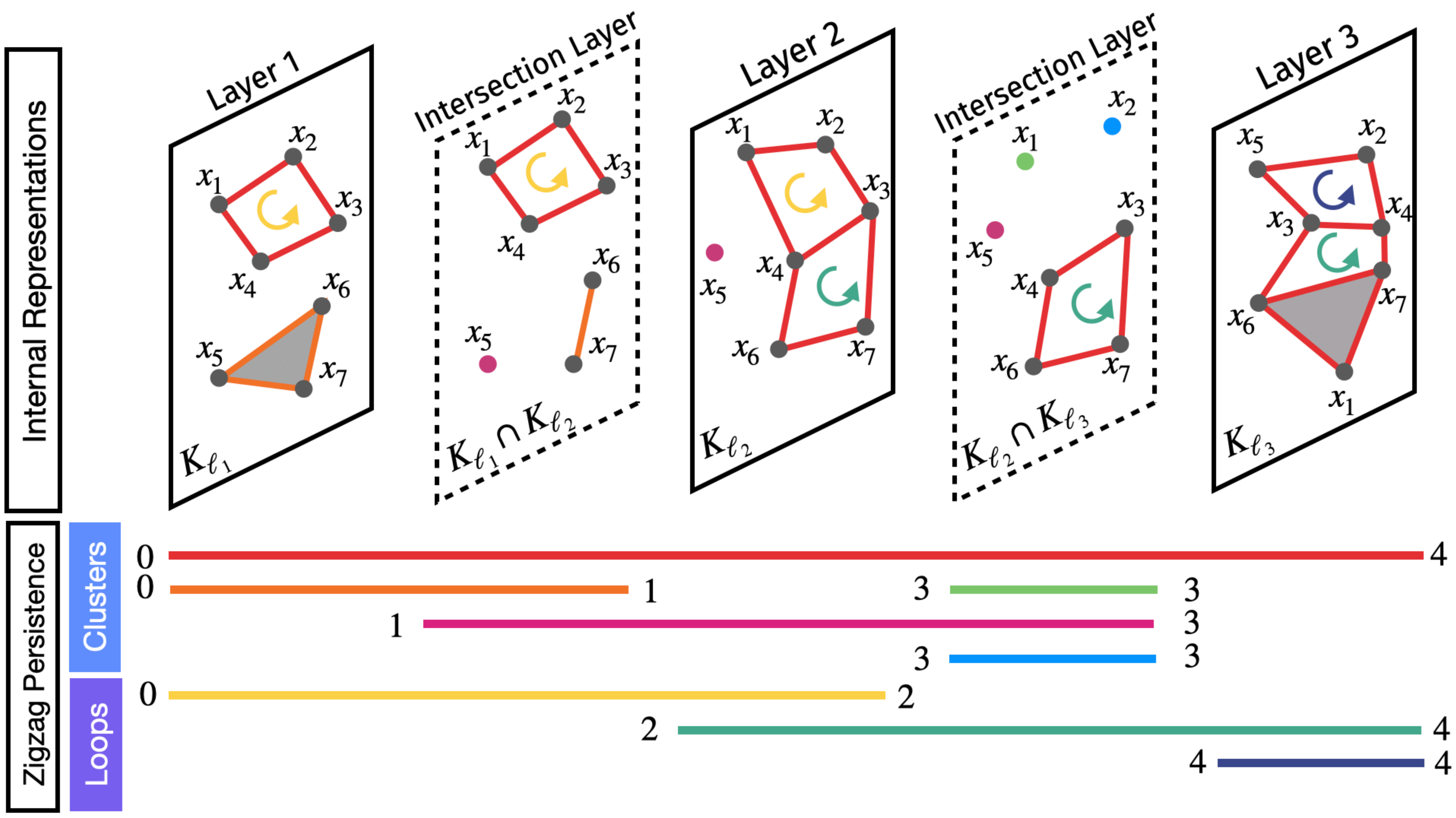
# The Zig Zag Algorithm




$$K_{\ell_1} \leftarrow K_{\ell_1} \cap K_{\ell_2} \rightarrow K_{\ell_2} \leftarrow \cdots \rightarrow K_{\ell_{L-1}} \leftarrow K_{\ell_{L-1}} \cap K_{\ell_L} \rightarrow K_{\ell_L}$$

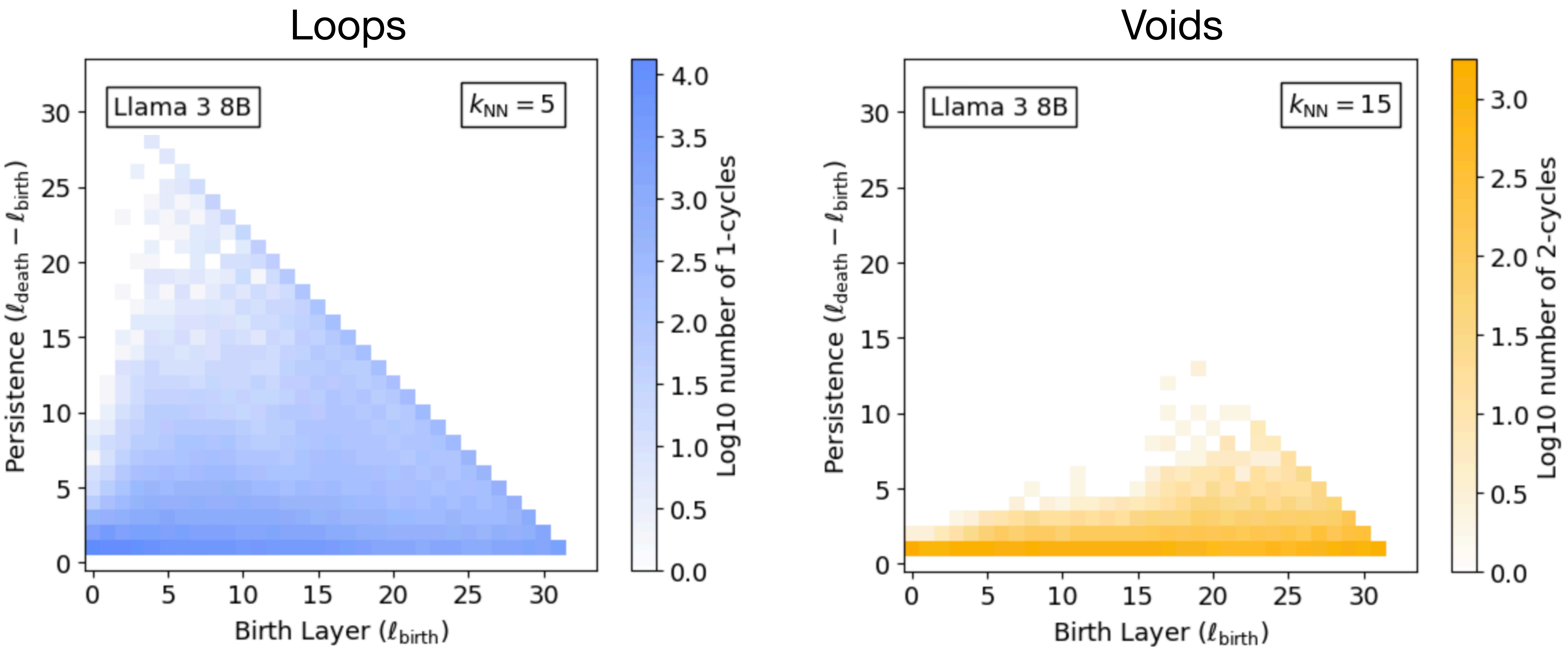
**Third step**: Intersection Layers

# The Zig Zag Algorithm



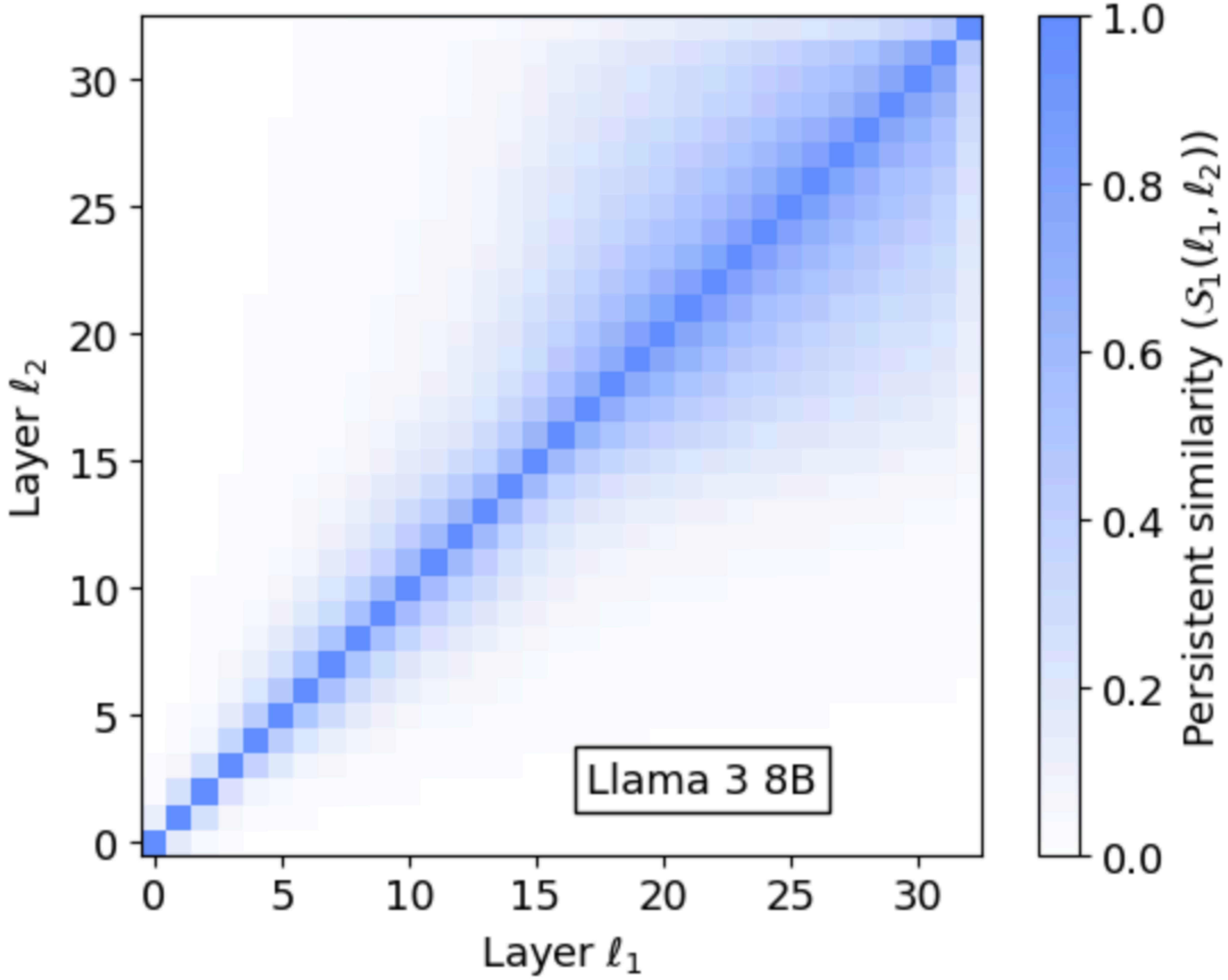

$$\mathrm{Pers}_p(\Phi) = \left\{ \left[\mathrm{birth}, \mathrm{death}\right] \mid \mathrm{birth}, \mathrm{death} \in \{0, \ldots, 2\mathrm{N}_{\mathrm{layers}} - 1\} \right\}$$

# Effective Persistence Images




Build density grid from persistence diagram

# Persistent Similarity




$$\mathcal{S}_p(\ell_1, \ell_2) = \frac{\sum_{\ell_1 \leq M_1, \ell_2 > M_2} \widehat{PI}_p(\ell_1, \ell_2)}{\beta_p(\ell_1)}$$

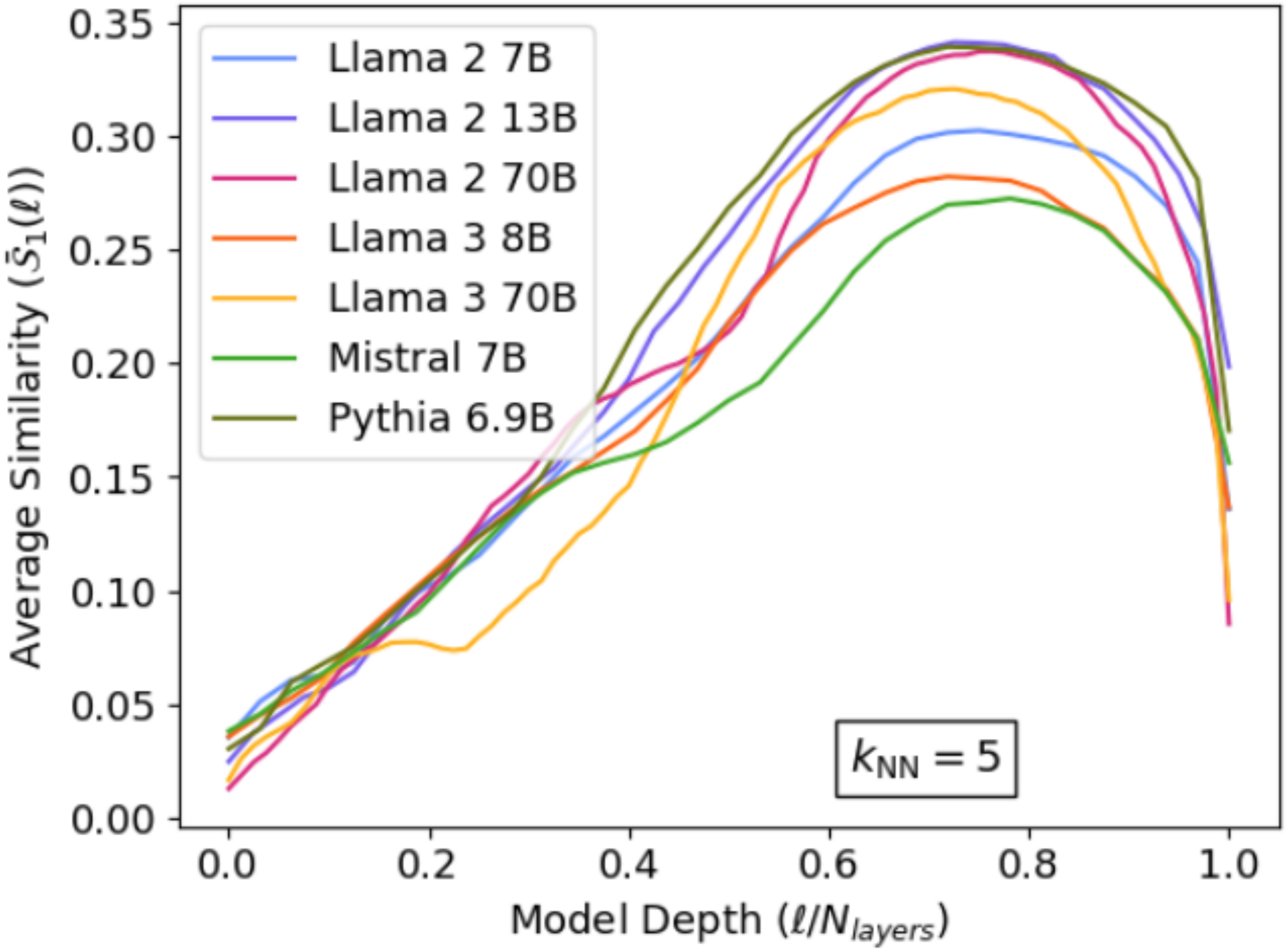
$M_1 = \min(\ell_1, \ell_2) \qquad M_2 = \max(\ell_1, \ell_2)$

Similarity measure sensitive to the features' trajectories

↓

Fraction of loops alive at layer $\ell_1$ that are still alive at layer $\ell_2$ (and were alive the whole path)

# Average Persistent Similarity




Average retention of features in each layer

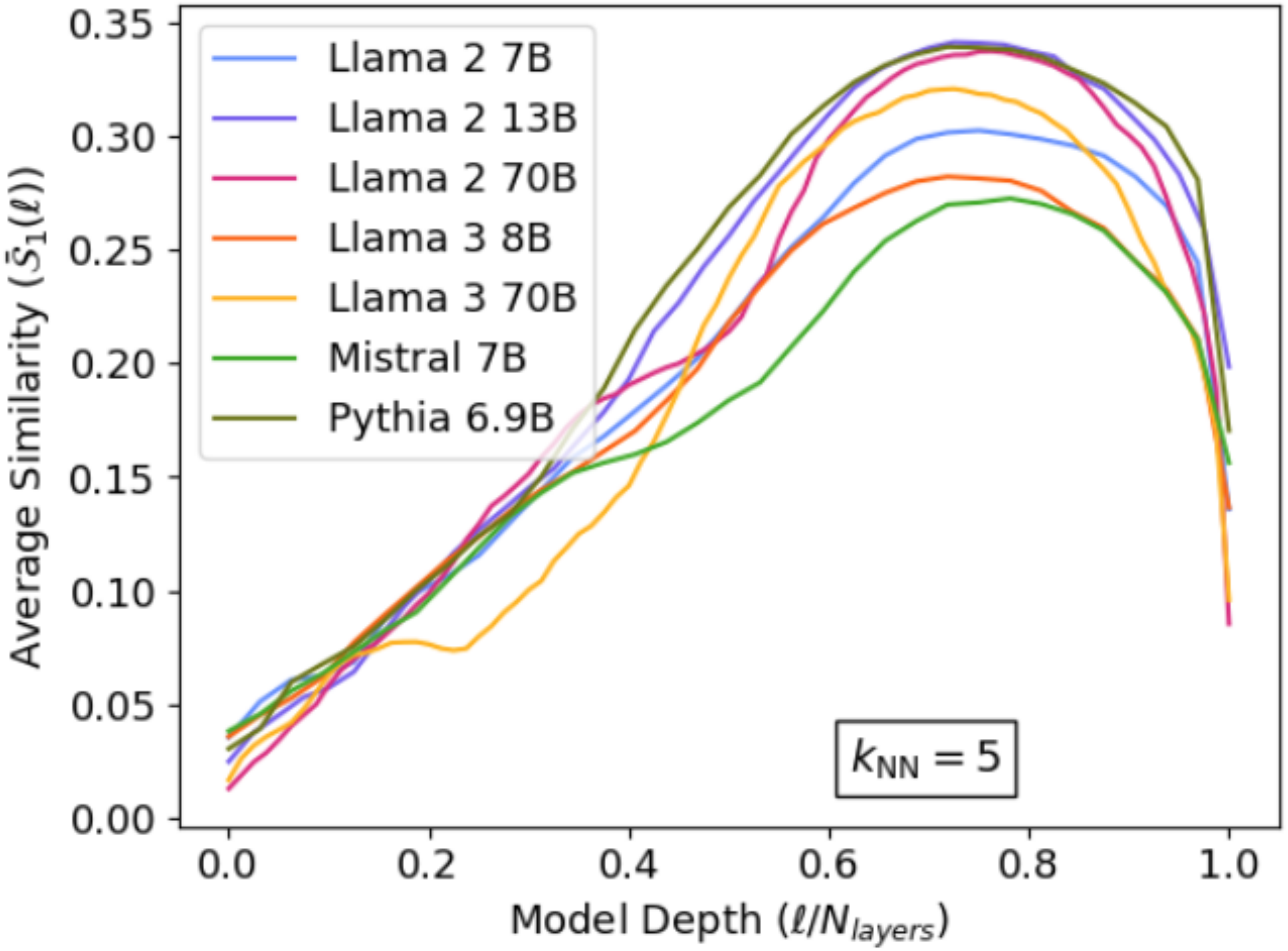
$$\bar{\mathcal{S}}_p(\ell) = \frac{1}{N_{\text{layers}}} \sum_{\ell_i=1}^{N_{\text{layers}}} \mathcal{S}_p(\ell, \ell_i),$$

***low value*** represents a phase of change of relative positions among points

***high value*** the relations among points are relatively stationary.

# Application: layer pruning

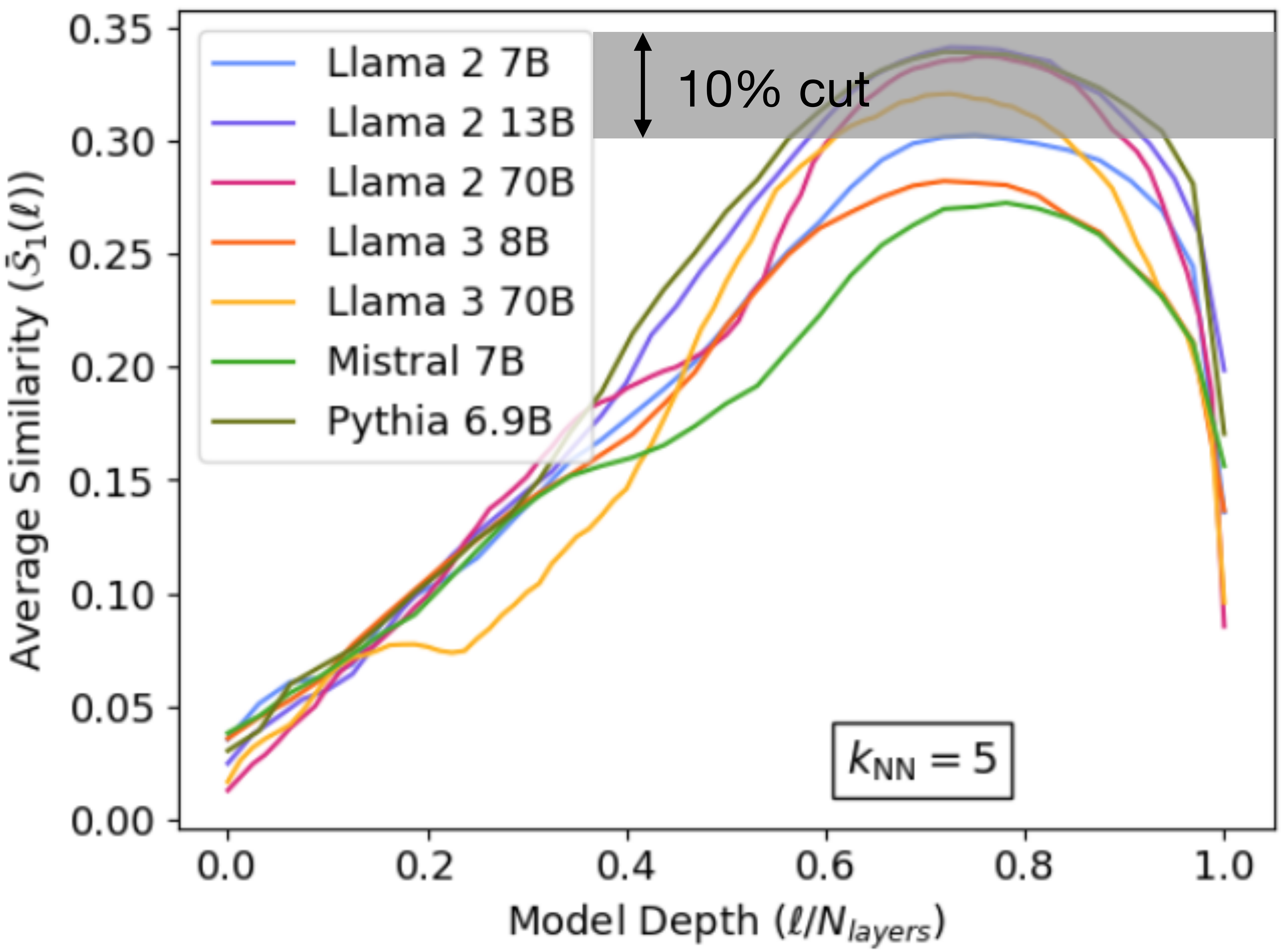
Do we need layers were similarity is high?




***Proposal***: Prune layers based on similarity

# Application: layer pruning

Do we need layers were similarity is high?




***Proposal***: Prune layers based on similarity

1. Decide a cut (e.g. 10% of max avg similarity)
2. Remove layers falling in that range
3. Rebuild model without those layers
4. Compute performance degradation

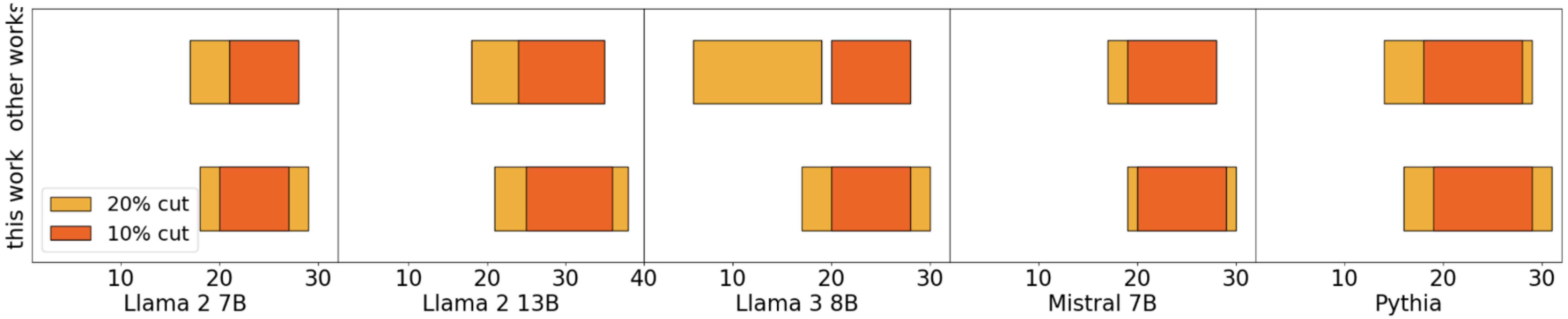
Compare to other methods of layer pruning by similarity

Gromov et al. 2024
Men at al. 2024

# Application: layer pruning

***Proposal***: Prune layers based on similarity

| Models | MMLU | | | HellaSwag | | | WinoGrande | | |
|---|---|---|---|---|---|---|---|---|---|
| | Full | This work | Other works | Full | This work | Other works | Full | This work | Other works |
| Llama 2 7B | 45.74 | 37.38 (39.32) | **43.95** (34.35) | 58.54 | **44.71** (32.10) | 42.78 (35.10) | 74.43 | **68.67** (59.67) | 67.72 (62.67) |
| Llama 2 13B | 54.60 | 50.16 (36.45) | **50.71** (37.91) | 61.43 | **48.60** (34.35) | 47.84 (34.52) | 76.72 | 71.67 (63.21) | **73.15** (61.47) |
| Llama 3 8B | 65.07 | **53.44** (23.16) | **53.44** (24.33) | 61.37 | **41.60** (29.69) | **41.60** (27.10) | 77.10 | **70.00** (59.75) | **70.00** (50.58) |
| Mistral 7B | 62.40 | **53.17** (24.26) | 38.20 (37.86) | 62.83 | **36.67** (26.26) | 34.45 (28.10) | 77.35 | **66.50** (57.76) | 63.76 (55.96) |
| Pythia | - | - | - | 49.70 | 31.43 (31.23) | **34.96** (26.84) | 63.30 | 55.71 (54.84) | **58.09** (51.07) |




Gromov et al. 2024
Men at al. 2024

# Conclusions

- ***Zig Zag Persistence:*** Novel framework based on TDA to analyse internal representations of LLMs
- ***Persistence Similarity***: new metric to measure changes in relative positions across the layers of an LLM. It tracks the entire trajectory of transformations between two layers.
- ***Model Pruning***: Prune layers with high persistence similarity without significantly degrading performance
- ***Consistency Across Models and Hyperparameters***: Persistent topological features and their similarities are consistent across differen models, layers, and choices of hyperparameters of the framework.