

# KHÓA LUẬN TỐT NGHIỆP

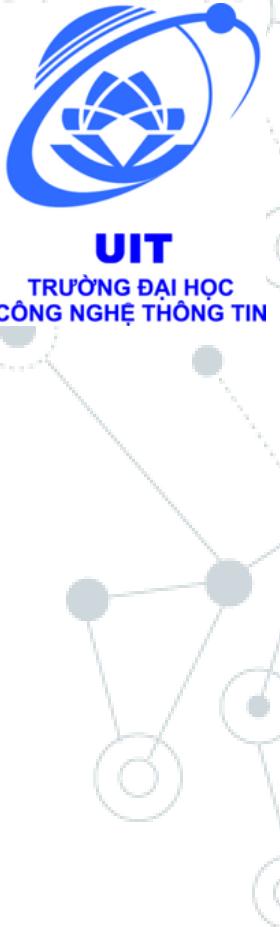
## CẢI TIẾN CÁC PHƯƠNG PHÁP •DATA VALUATION VÀ ỨNG DỤNG

NGUYỄN TRƯỜNG THỊNH - 21520110

GVHD: TS.VÕ NGUYỄN LÊ DUY



# Mục tiêu đề tài



- Tìm hiểu về **data valuation**.
- **Phân tích ưu nhược điểm** hai phương pháp **KNN-shapley** và **LAVA**. (Dựa trên backbone Data shapley và Optimal transport)
- **Đề xuất cải tiến** của hai phương pháp.
- **Thực nghiệm** trên nhiều lĩnh vực.



**Giới thiệu**

**01**

**Phương pháp**

**02**

**Đề xuất**

**03**

# **NỘI DUNG**

**04**

**Thực nghiệm**

**05**

**Thảo luận**

**06**

**Appendix**



# Giới thiệu về bài toán

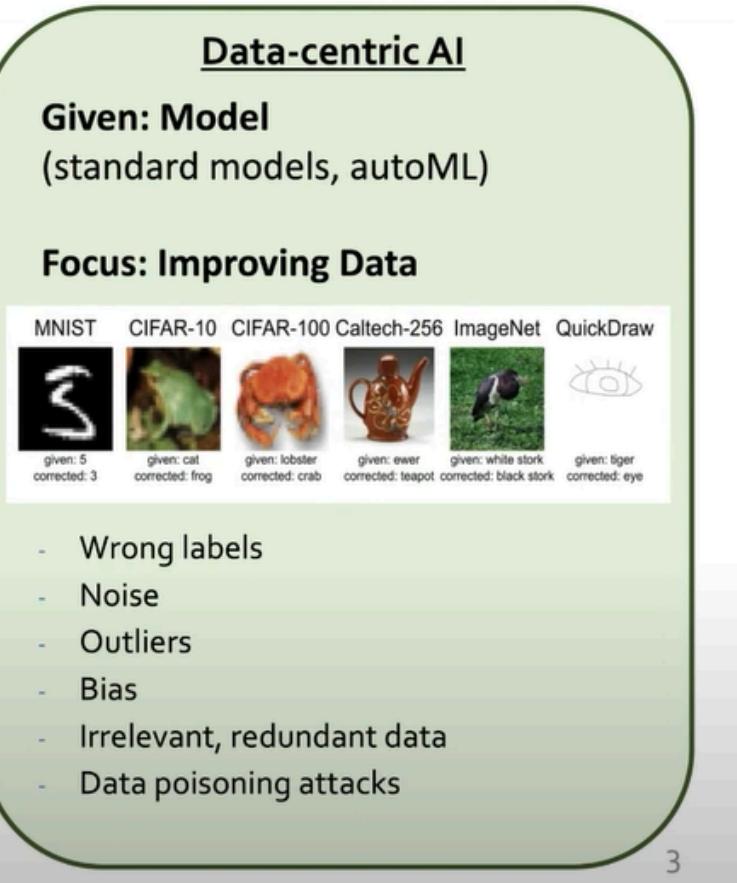
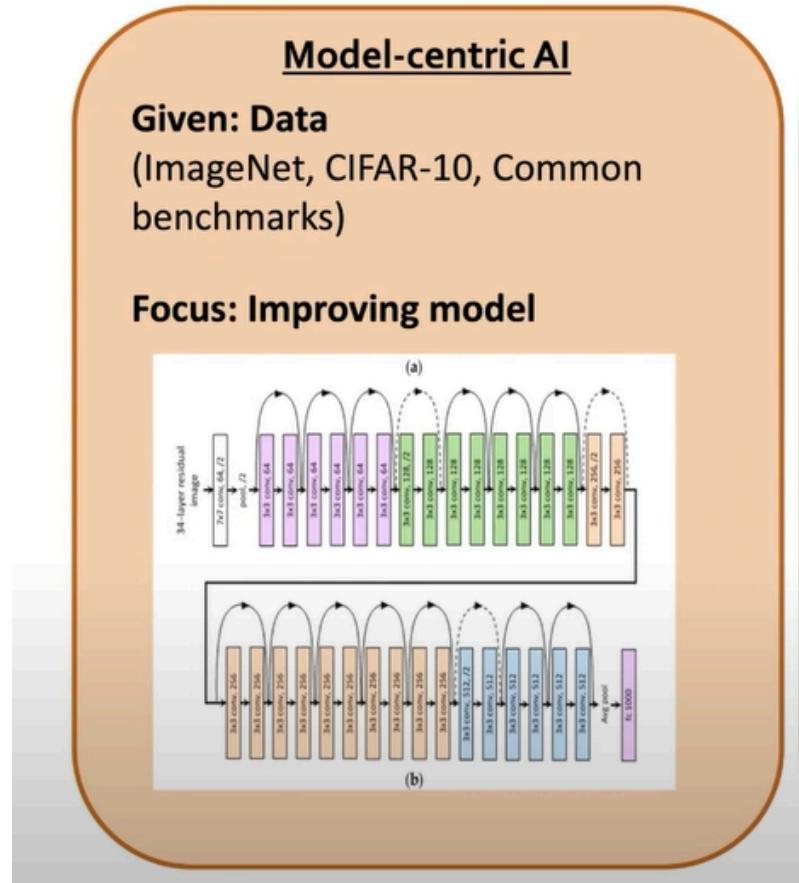


# Động lực

**Chất lượng dữ liệu:** đóng vai trò quan trọng trong việc **quyết định hiệu suất của mô hình.**

Tuy nhiên, không phải tất cả dữ liệu đều có giá trị như nhau.

**Data Valuation (Định giá dữ liệu)** ra đời nhằm **xác định đóng góp** thực sự của **mỗi điểm dữ liệu** vào quá trình huấn luyện và kết quả của mô hình học máy.

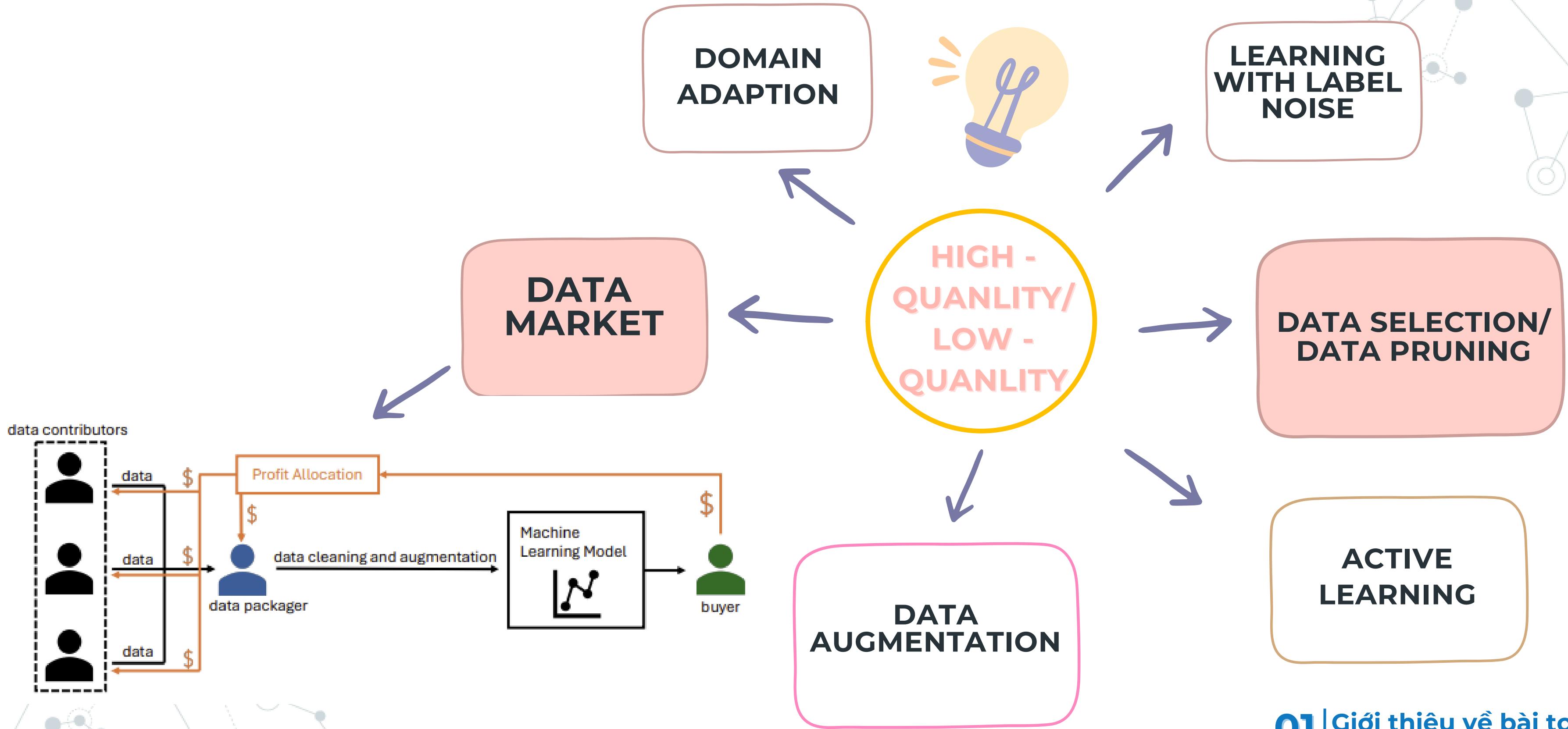


AI systems run both on code and data. "All that progress in algorithms means it's actually time to spend more time on the data," **Andrew Ng said**

5



# Ứng dụng



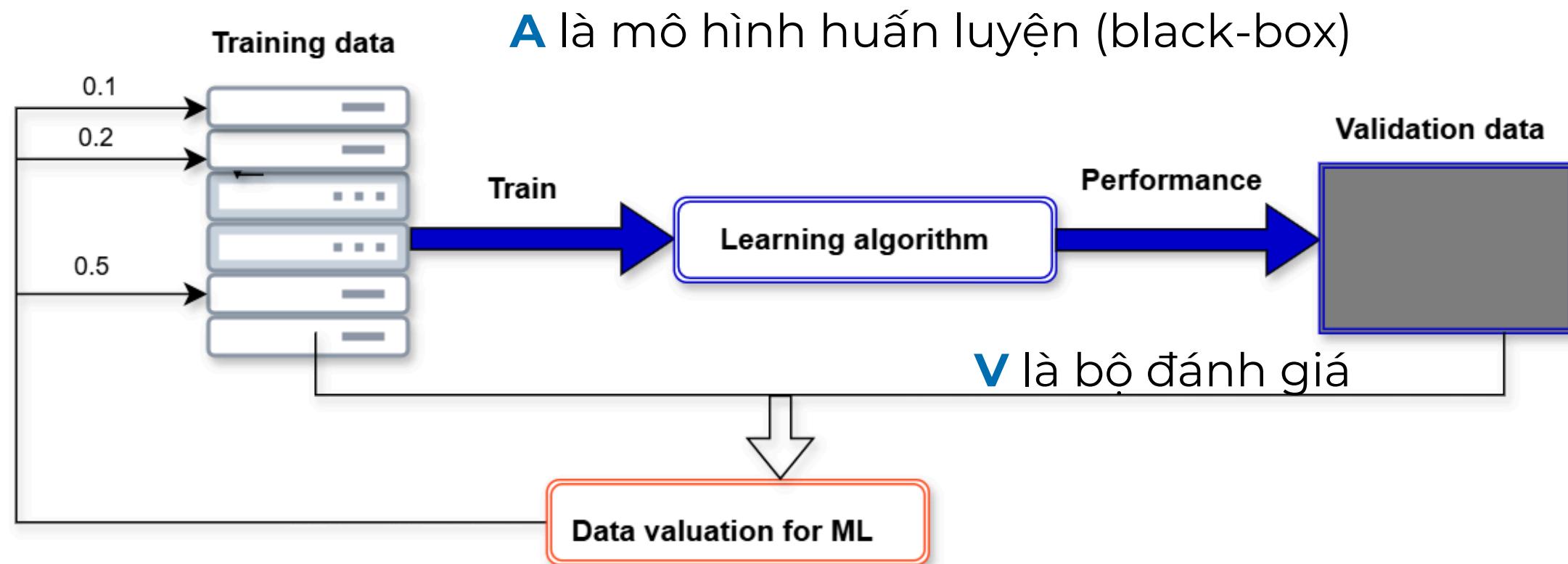


# Phát biểu bài toán



Bài toán bao gồm 3 phần chính:

$$D = \{(x_i, y_i)\}_i^n : \text{ Nguồn dữ liệu}$$



Với **mỗi điểm**  $(x_i, y_i)$  mục tiêu là mỗi điểm phải **mang một giá trị** (gọi là **data value**) thể hiện mức độ quan trọng **đóng góp** của nó **trong tập D**.

Kí hiệu:

$$\phi_i(D, A, V) \text{ viết tắt } \phi_i$$



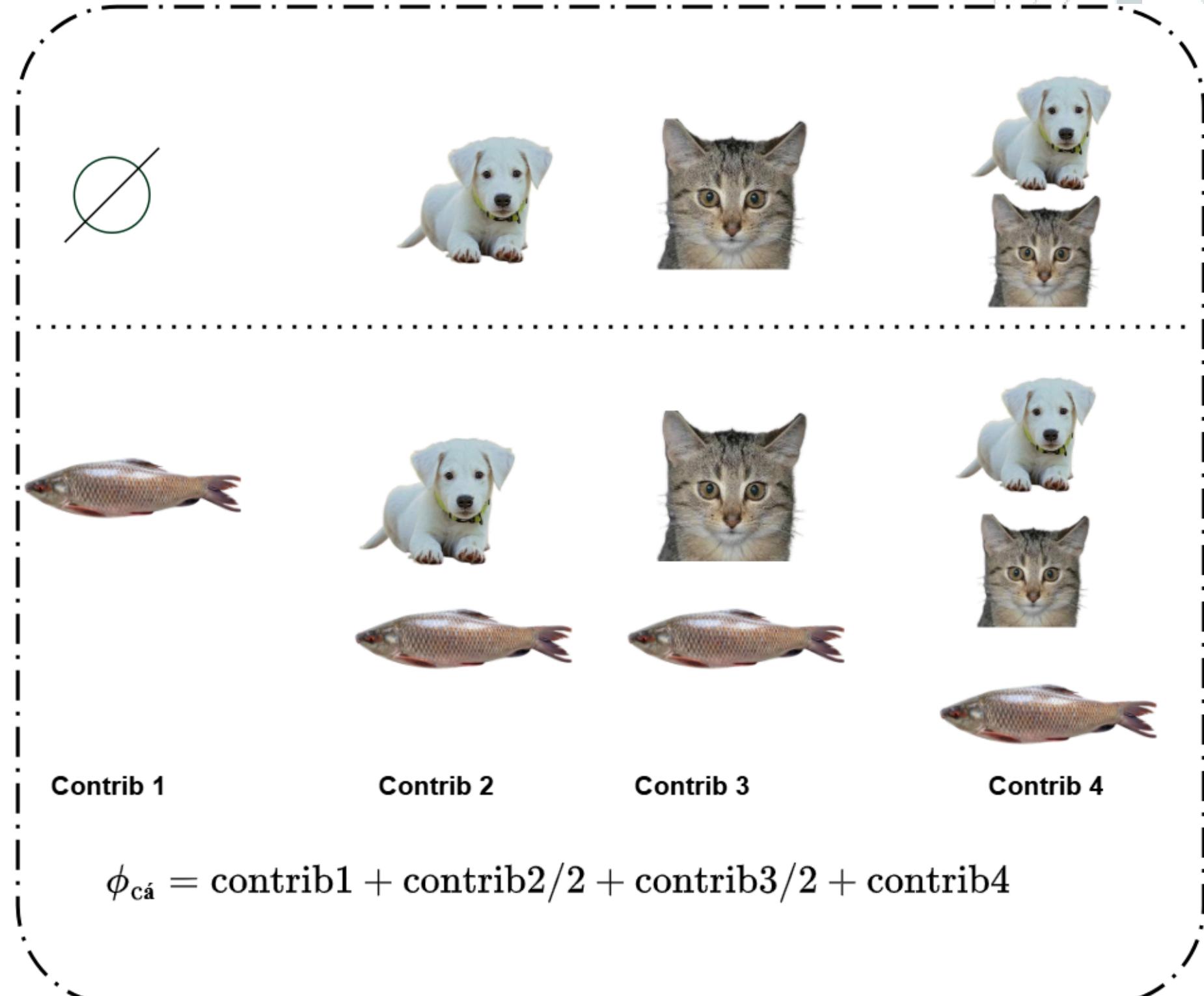
# Shapley values

Công thức:

$$\phi_i = C \sum_{S \subseteq D \setminus \{i\}} \frac{V(S \cup \{i\}) - V(S)}{\binom{n-1}{|S|}}$$

S là tập con của D

V(S) hàm đánh giá mức độ đóng góp của tập con S.





# Khó khăn

Gánh nặng tính toán

Phụ thuộc vào  
thuật toán học máy (A)

Nhu cầu đánh giá dữ liệu trước khi chọn  
thuật toán học máy

02

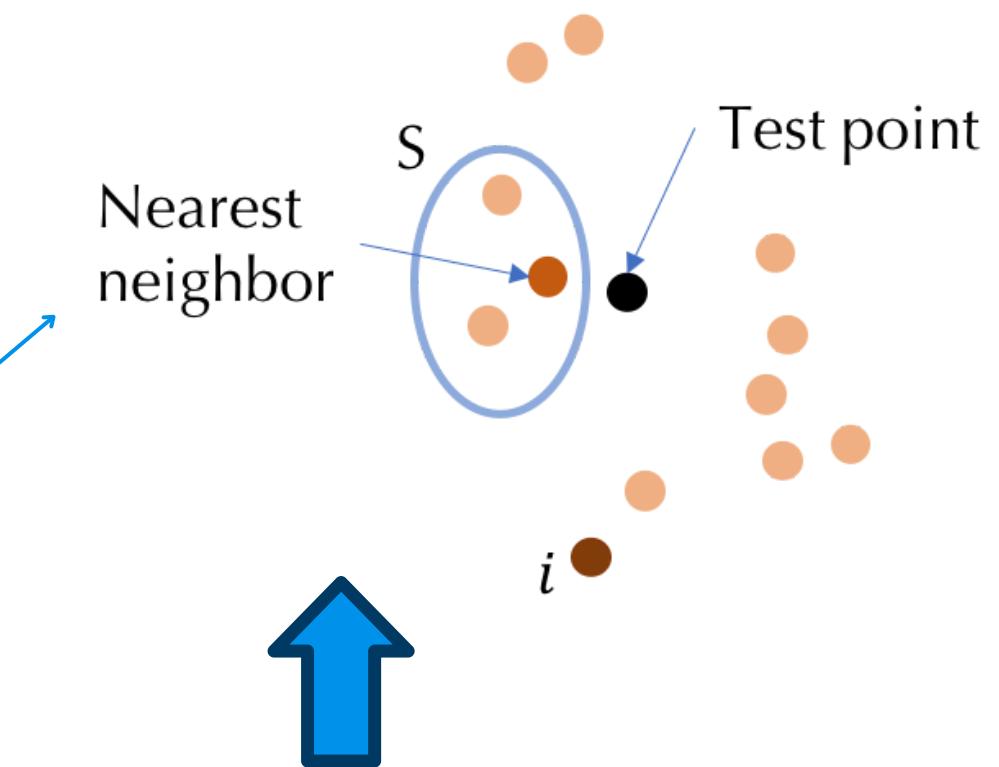
# Phương pháp



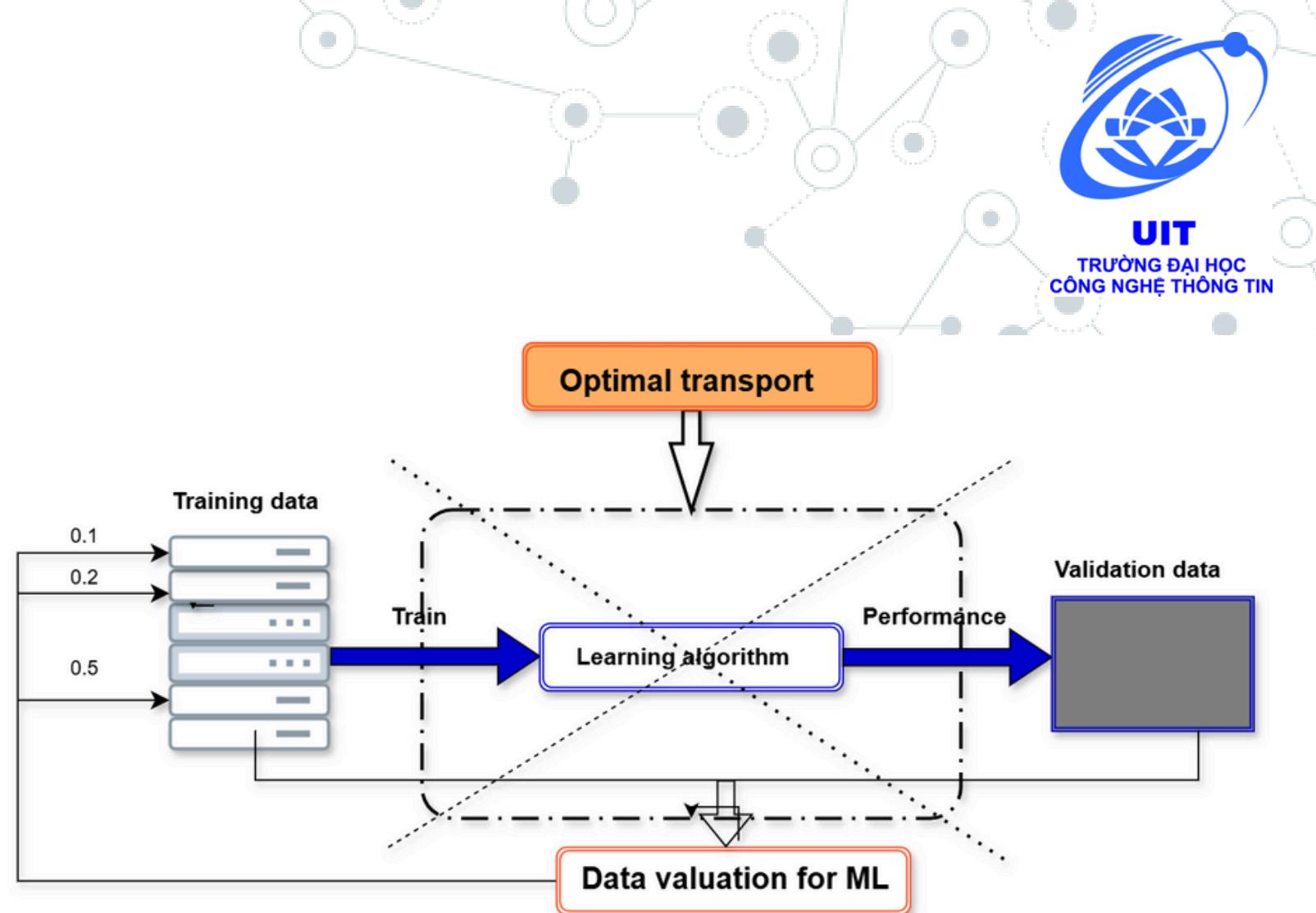
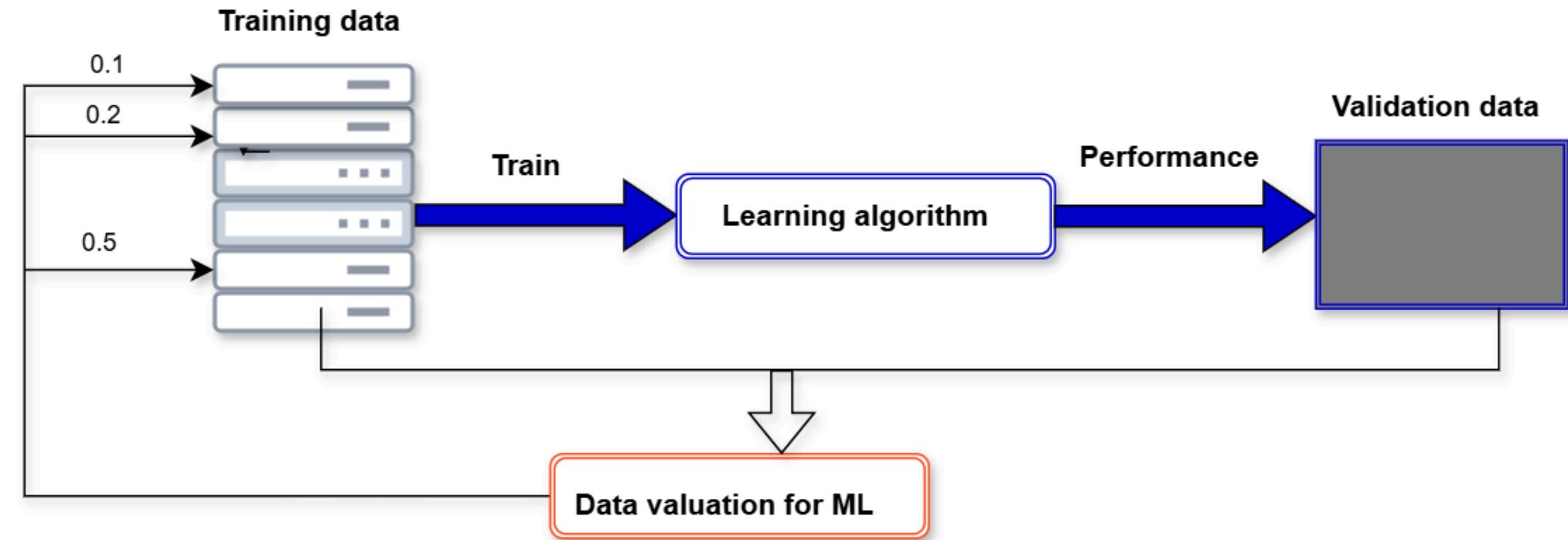
# Phương pháp

Tìm hiểu các phương pháp xử lí giá trị data value độc lập với thuật toán học máy

KNN Shapley  
(version 2020)  
[2]



LAVA(ICLR 2023) [3]





# KNN Shapley

**B1) Thiết kế hàm đánh giá (Utility)  
dựa vào bộ phân loại KNN:**

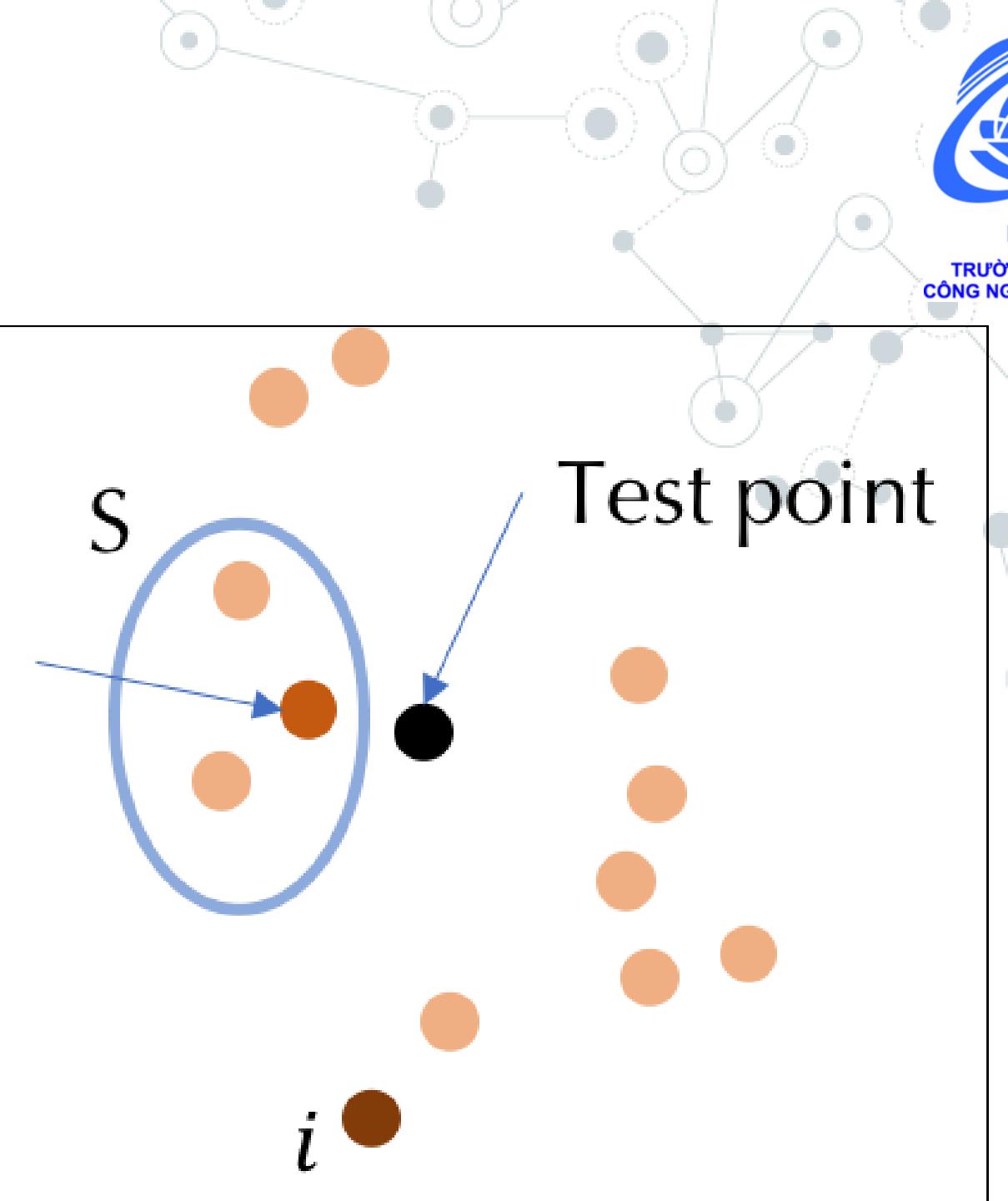
$$V(S) = \frac{1}{\min(K, |S|)} \sum_{k=1}^{\min(K, |S|)} 1_{y_{\alpha_k(S)} = y_{\text{test}}}$$

**B2) Áp dụng vào tính data shapley**

$$\phi_i = C \sum_{S \subseteq D \setminus \{i\}} \frac{V(S \cup \{i\}) - V(S)}{\binom{n-1}{|S|}}$$

**B3) Tạo công thức:**

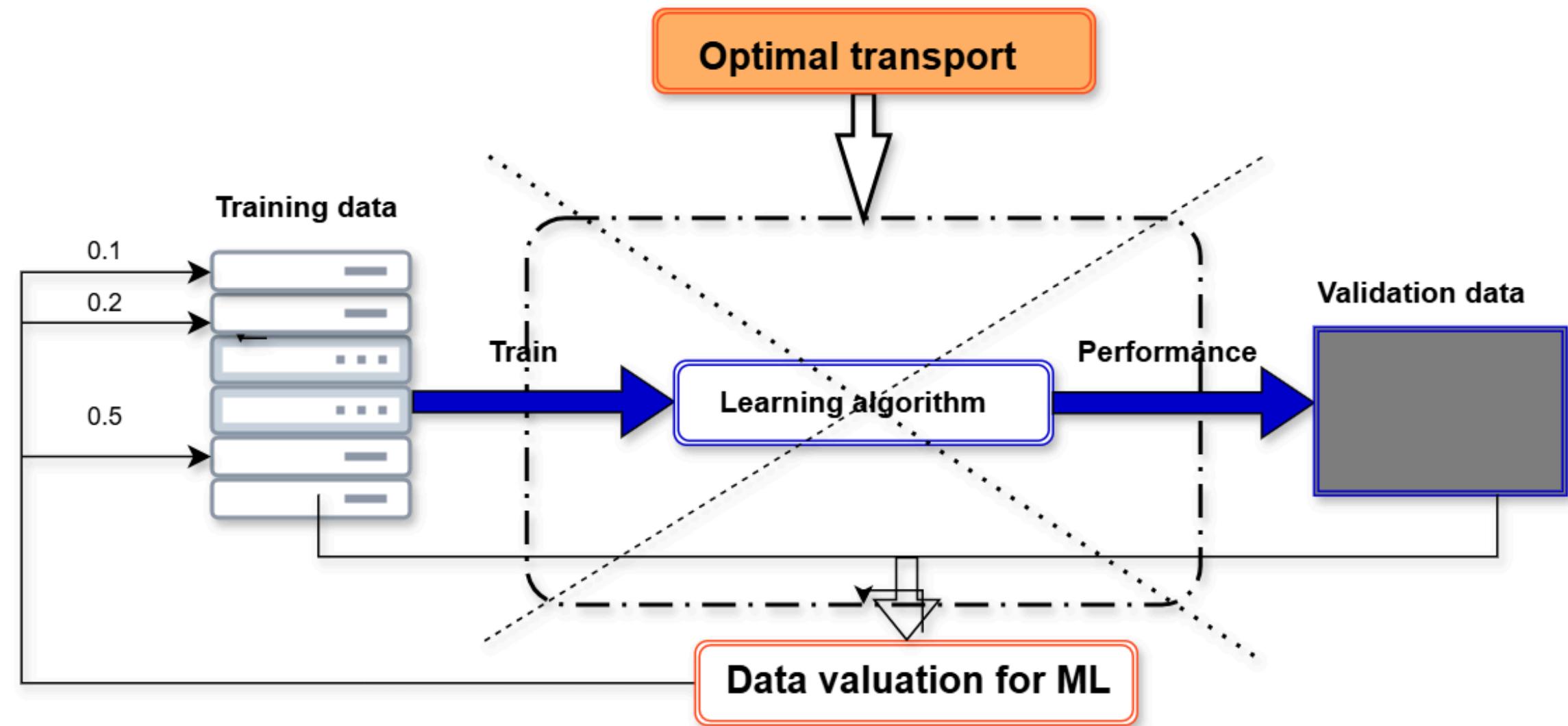
$$\phi_i - \phi_{i+1} = \frac{1[y_i = y_{\text{test}}] - 1[y_{i+1} = y_{\text{test}}]}{K} \frac{\min(K, i)}{i}.$$



$i$  và  $i + 1$  là hai điểm liền kề nhau khi sắp xếp theo khoảng cách đến test point



**Smaller dataset distance ~ High performance**  
**Bigger dataset distance ~ Lower performance**





# Calibrated Gradients



**Đạo hàm của khoảng cách OT trên phân phối xác suất của mỗi điểm dữ liệu:**

$$\frac{\partial \text{OT}(\mu_t, \mu_\nu)}{\partial \mu_t(z_i)} = f_i^* - \sum_{j \in \{1, \dots, N\} \setminus i} \frac{f_j^*}{N-1}, \quad \frac{\partial \text{OT}(\mu_t, \mu_\nu)}{\partial \mu_\nu(z'_j)} = g_j^* - \sum_{i \in \{1, \dots, M\} \setminus j} \frac{g_i^*}{M-1}.$$

**Đạo hàm dương lớn** khi **tăng khối lượng** ở điểm đó, OT tăng mạnh  
=> **nên giảm** để thu hẹp khoảng cách.

**Đạo hàm âm lớn** => khi **tăng khối lượng** ở điểm đó, OT giảm mạnh  
=> **nên bổ sung** để hai phân phối lại gần nhau hơn.

Như vậy giá trị **data value** tính được trong **lava ngược** với **data value** trong **KNN-shapley**



# Phân cấp trong Optimal Transport

**Ý tưởng:** Muốn tính “khoảng cách” giữa hai điểm  $(x_t, y_t)$  và  $(x_v, y_v)$  không chỉ xét tọa độ (phần x) mà còn xét nhãn (phần y):

$$C((x_t, y_t), (x_v, y_v)) := d(x_t, x_v) + cOT_d(\mu(\cdot | y_t), \mu(\cdot | y_v))$$

Trong đó:

$d(x_t, x_v)$  : khoảng cách Euclidian giữa hai vector đặc trưng.

$OT_d(\mu(\cdot | y_t), \mu(\cdot | y_v))$  : khoảng cách OT giữa hai phân bố theo nhãn  $y_t$  và  $y_v$ .



03

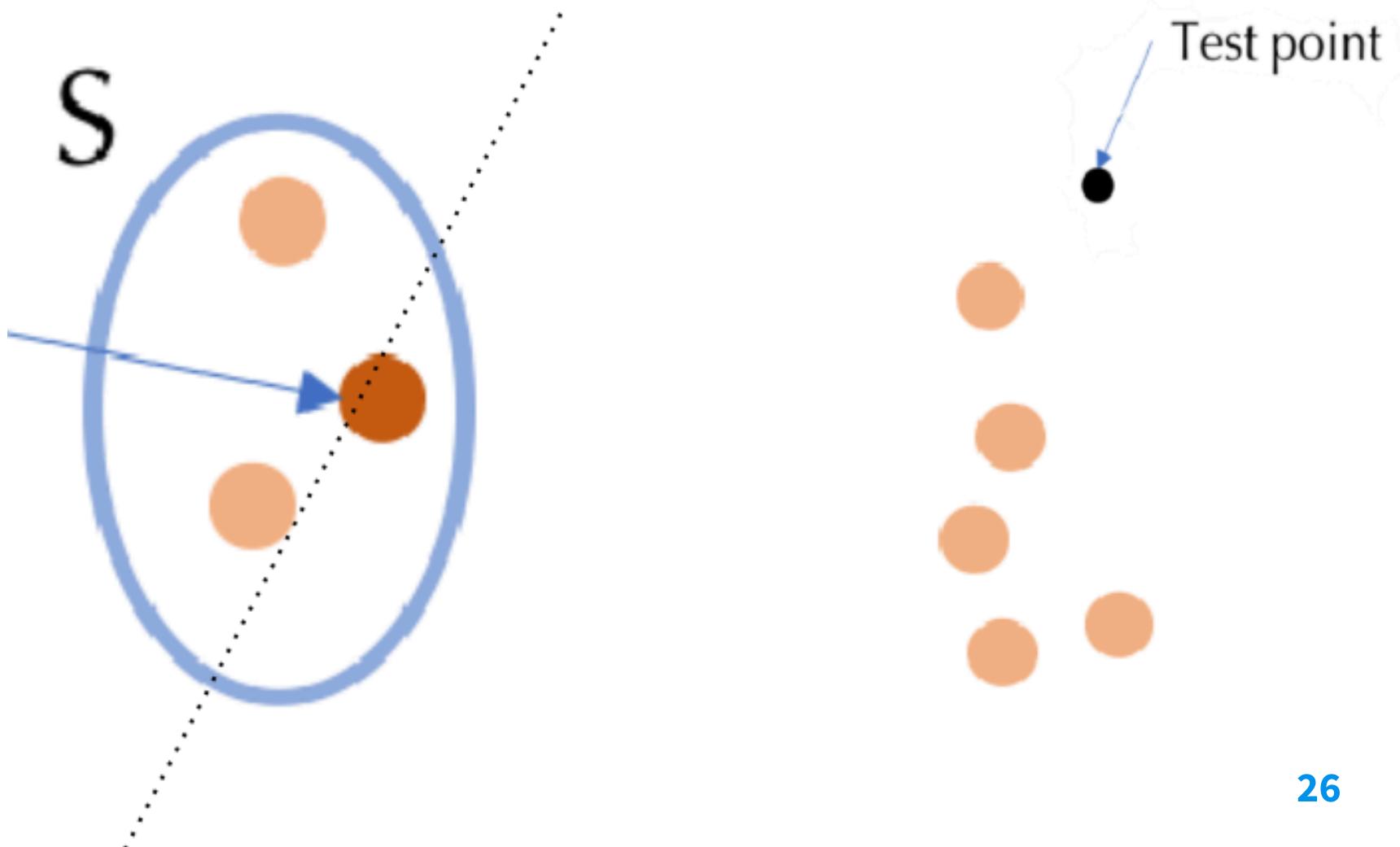
**Đề xuất**



## Đề xuất cho KNN - Shapley

**Vấn đề:** Một khi tập con  $S$  chứa các điểm xa *test point*, việc đưa các điểm xa này vào quá trình tính toán đóng góp (Shapley value) có thể làm cho việc đánh giá sai hoặc nhiễu.

Vì vậy, **cắt giảm hoặc vô hiệu hóa** những điểm nằm quá xa bằng một **ngưỡng** (*threshold*) nhằm tránh “thổi phồng” đóng góp của các điểm ngoại biên.





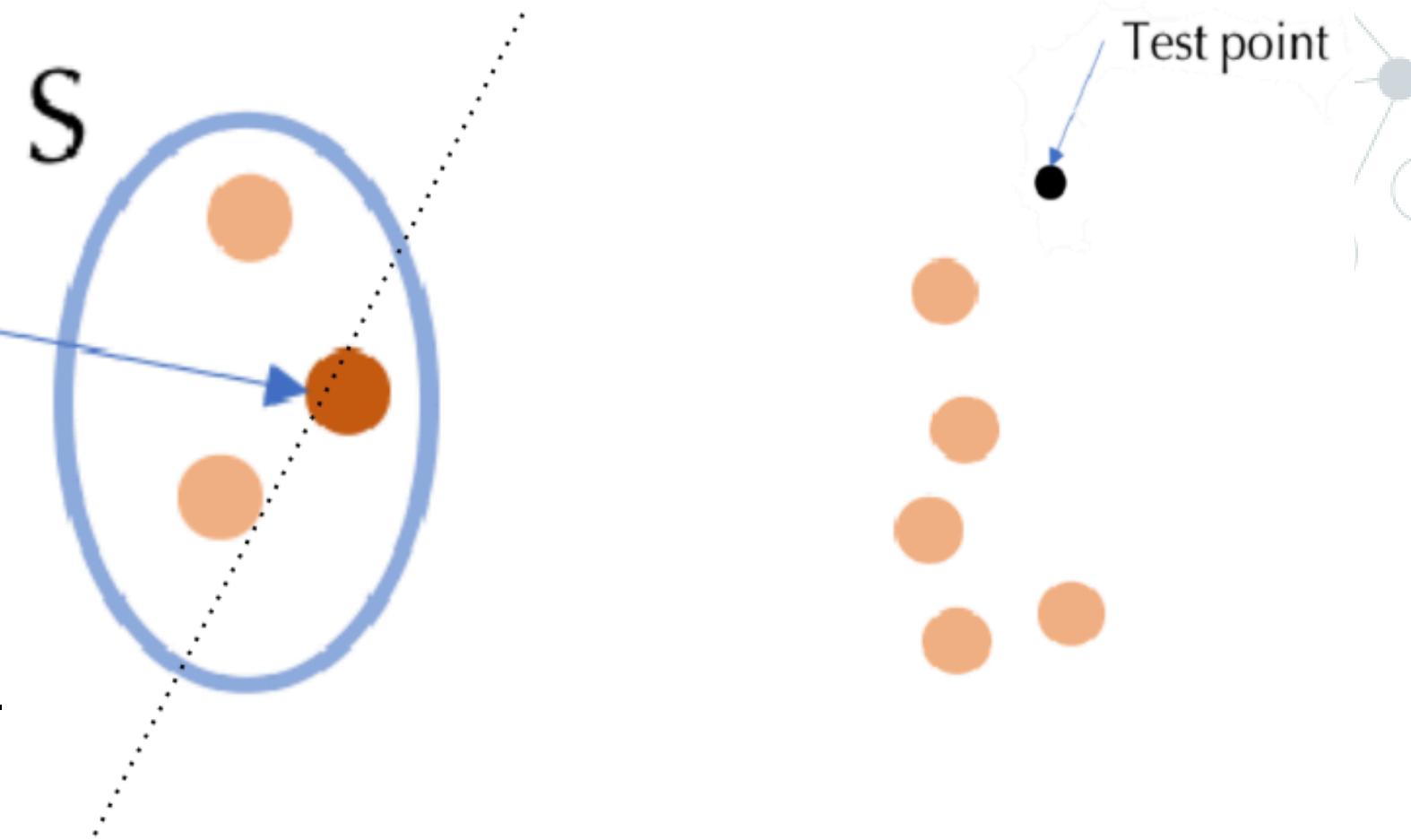
# Đề xuất cho KNN - Shapley

**Thiết kế thuật toán:**

$$\phi_{a_N} = \phi_{a_{N-1}} = \dots = \phi_{a_{N-T+1}} = 0$$

$$\phi_{a_{N-T}} = \frac{1[y_{N-T} = y_{\text{test}}]}{N - T}$$

$$\phi_{a_i} = \phi_{a_{i+1}} + \frac{1[y_i = y_{\text{test}}] - 1[y_{i+1} = y_{\text{test}}]}{K} \frac{\min(K, i)}{i}$$

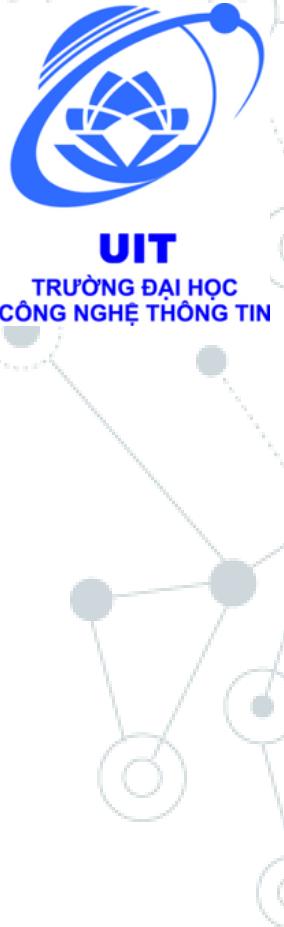


Như vậy ta thêm một biến  $T$  (threshold) nhằm  
cắt bỏ những tập con không cần thiết.

Dùng grid search trên:  $T = 2K, N/2, N - 2K$



## Đề xuất cho LAVA



**Thay đổi trong hàm tính chi phí vận chuyển:**

$$C((x_t, y_t), (x_v, y_v)) := d(x_t, x_v) + cOT_d(\mu_t(\cdot|y_t), \mu_v(\cdot|y_v))$$

- *Tinh chỉnh khoảng cách bằng tham số:*

$$\text{cost\_matrix} = \text{feature\_distance} * \alpha + \text{label\_distance} * \beta$$

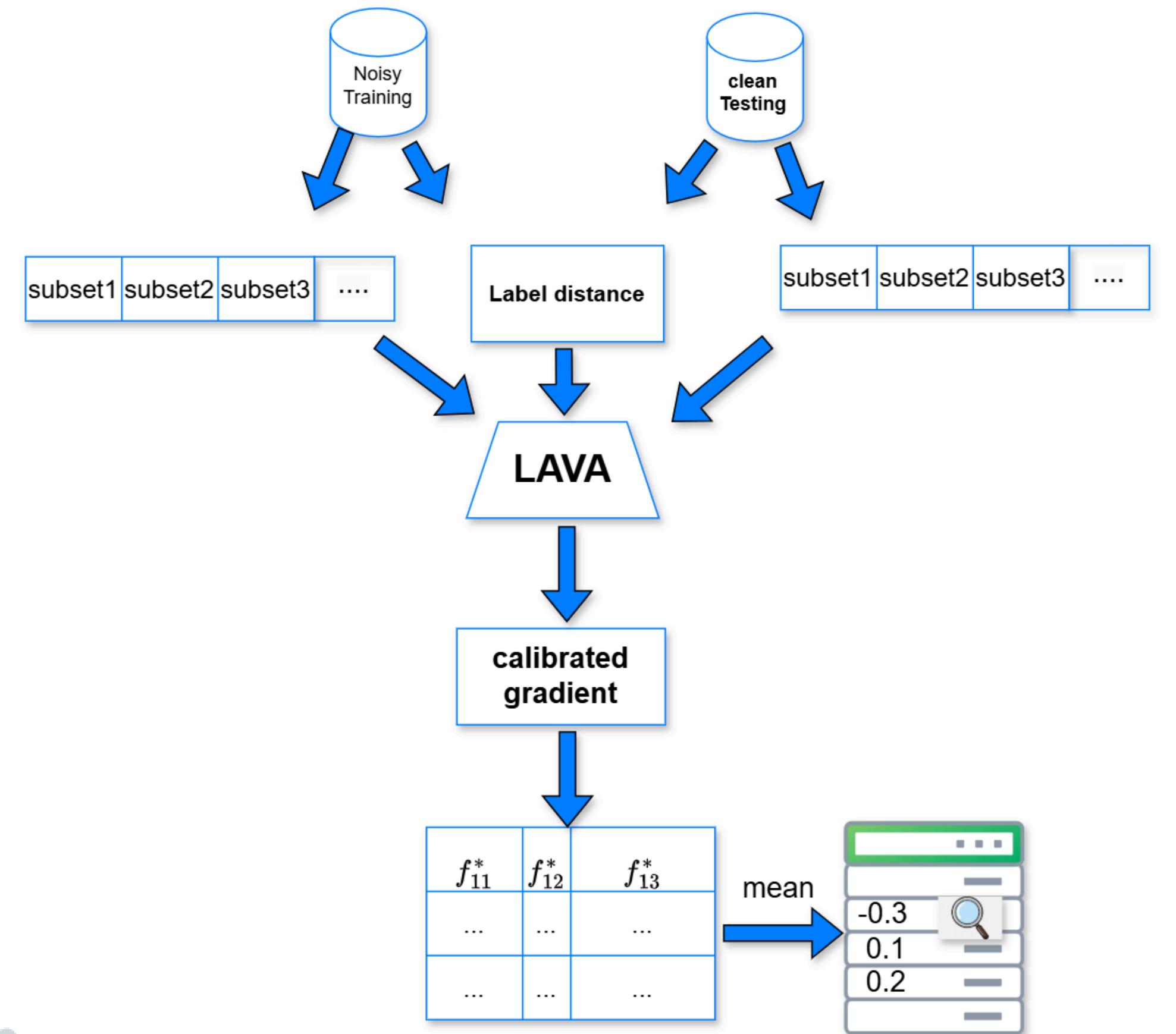
- *Chia nhỏ tập dữ liệu thành nhiều phần:*

Tính LAVA trên nhiều batch nhỏ kết hợp với việc tính khoảng cách nhãn trước.

Việc chọn Batch size có thể là:  $\text{Batch\_size} = 32, 64, 128\dots$



# Đề xuất cho LAVA





04

# Thực nghiệm



# Thực nghiệm

**Mục tiêu:** So sánh các phương pháp trên các tác vụ:

## 1. Phát hiện nhiễu:

- Tạo ra 20% nhiễu ngẫu nhiên.
- Nhiễu nhãn: Lật nhãn ban đầu thành nhãn khác.
- Nhiễu đặc trưng: Chuẩn hóa và thêm lõi ngẫu nhiên từ phân phối Gaussian với trung bình bằng 0 và độ lệch chuẩn là 2.

## 2. Đánh giá hiệu suất:

- Sau khi thêm dữ liệu chất lượng thấp/cao.
- Sau khi xóa dữ liệu chất lượng thấp/cao.



# DATASETS

**Bài toán giải quyết:**

**Bài toán phân lớp:**

- Dạng bảng: 2dplan, credit, digits, iris, pol. (OpenML)
- Dữ liệu hình ảnh: cifar10 (10 lớp), mnist, stl10, SVHN, FashionMnist (torchvision.datasets).
- Dữ liệu văn bản: BBC (5 topics), IMDB, SST-2.
- Dữ liệu y tế (ảnh X-ray) .
- Dữ liệu chuỗi thời gian (lưu lượng giao thông).

**Bài toán hồi quy (Dữ liệu dạng bảng):**

- Creditcard, diabetes, echomonths, mv, stock, wave (openML)



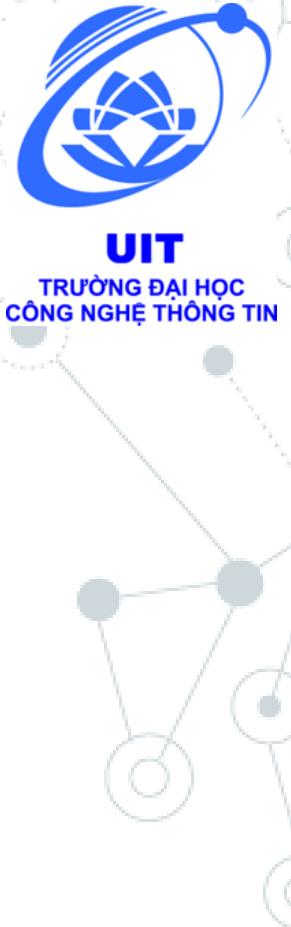
# DATASETS

Sử dụng Embedding để trích xuất trước đặc trưng.

- Trên ảnh: Resnet50
- Trên ngôn ngữ tự nhiên: DistilBertModel



# Metric



## Phát hiện nhiễu (Corrupted detection):

**Precision:** Tỉ lệ tìm thấy nhiễu trên cho tỉ lệ các điểm được chọn ra để đánh giá.

**Recall:** Tỉ lệ tìm thấy nhiễu trên tất cả các nhiễu.

$$\text{Precision} = \frac{|S_{\phi}^{(\text{low})} \cap S^{(\text{mislabeled})}|}{|S_{\phi}^{(\text{low})}|}, \quad \text{Recall} = \frac{|S_{\phi}^{(\text{low})} \cap S^{(\text{mislabeled})}|}{|S^{(\text{mislabeled})}|}.$$

$$F_1 = \frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}}.$$



# Weighted accuracy drop (WAD)

**Đo độ chính xác thứ hạng dữ liệu (Point removal):**

$$WAD_D = \sum_{j=1}^n \left( \frac{1}{j} \sum_{i=1}^j [V - V_{D \setminus \{1:i\}}] \right)$$

Tập D được sắp xếp từ giá trị cao nhất đến giá trị thấp nhất dựa trên data value.

Thuật toán học để kiểm tra là Logistic Regression (CrossEntropyLoss)



# Kết quả

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Baseline Performance = 0.9699</i>				
KNN shapley	0.9400	0.9400	0.9400	0.9642
KNN Shapley (Best Threshold)	0.9450	0.9450	<b>0.9450</b>	0.9642
LAVA	0.6550	0.6550	0.6550	0.9638
LAVA (Best Batch Size)	0.6450	0.6450	0.6450	0.9637
LAVA (label-to-label)	0.6600	0.6600	<b>0.6600</b>	<b>0.9660</b>

Bảng 4.12: Experiments on the BBC dataset with noisy label 0.2

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Baseline Performance = 0.9799</i>				
KNN shapley	0.8050	0.8050	0.8050	0.9795
KNN Shapley (Best Threshold)	0.8600	0.8600	<b>0.8600</b>	<b>0.9825</b>
LAVA	1.0000	1.0000	1.0000	0.9799
LAVA (Best Batch Size)	1.0000	1.0000	1.0000	0.9799
LAVA (label-to-label)	1.0000	1.0000	1.0000	0.9799

Bảng 4.13: Experiments on the BBC dataset with noisy feature 0.2

## Kết quả dạng số

Thực nghiệm F1 - score, hiệu suất thay đổi khi loại bỏ nhiễu trên BBC (bộ phân loại thể loại văn bản gồm 6 lớp)



# Kết quả

Evaluator	Precision	Recall	F1-Score	WAD	Improvement
<i>Baseline Performance = 0.8840</i>					
KNN Shapley	0.6523	0.9785	0.7828	0.01931	0.8873
KNN Shapley ( $T = N/2$ )	0.6513	0.9770	0.7816	0.01925	0.8888
LAVA	0.4220	0.6330	0.5064	0.02074	0.8859
LAVA (Best Batch Size)	0.4250	0.6375	0.5100	0.01726	0.8823
LAVA (OT Library Implementation)	0.4626	0.6940	<b>0.5551</b>	0.02485	0.8757

Bảng 4.5: Experiments on the CIFAR dataset with noisy labels 0.3.

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Baseline Performance = 0.8927</i>				
KNN Shapley	0.5165	0.5165	0.5165	0.8919
KNN Shapley ( $T = N/2$ )	0.6513	0.9770	<b>0.7816</b>	<b>0.8923</b>
LAVA	1.0000	1.0000	1.0000	0.8924
LAVA (Best Batch Size)	1.0000	1.0000	1.0000	0.8924
LAVA (OT Library Implementation)	1.0000	1.0000	1.0000	0.8923

Bảng 4.6: Experiments on the CIFAR dataset with noisy feature 0.2

## Kết quả dạng số

Thực nghiệm F1-score, WAD, hiệu suất thay đổi khi loại bỏ nhiễu trên cifar10 (bộ gồm 10 lớp)



## Time series

Bộ dữ liệu: Lưu lượng giao thông gồm

- Tiền xử lý: Lấy ra lượng giao thông tháng 7,8,9.
- Chuyển thành bài toán phân loại nhị phân. Chia tập tỉ lệ 7:3.

**Thực nghiệm 1:** Dùng data valuation đánh giá trên dữ liệu tháng 7+8.

Sau đó thêm lần lượt dữ liệu tháng 8 vào tháng 7 và đánh giá trên 7+8 validation.

**Thực nghiệm 2:** Chọn ra 25% điểm tháng 8 (làm hiệu suất đạt kết quả tốt nhất). Kết hợp với tháng 7 vào tiến hành đánh giá trên tháng 9.

Thêm lần lượt điểm tháng 9 vào và xem hiệu suất đánh giá trên tập <sup>33</sup> 7+8+9 validation.



# Time series

Evaluator	Peak Performance	25% of Data Points Remain
<i>Baseline Performance = 0.640</i>		
KNN Shapley	0.67	0.67
KNN Shapley (Best Threshold)	<b>0.68</b>	<b>0.68</b>
LAVA	0.720	0.730
LAVA (Best Batch Size)	0.728	0.725
LAVA+label-to-label	<b>0.735</b>	<b>0.730</b>
LAVA (OT Library Implementation)	0.740	0.730

## Kết quả thực nghiệm 1

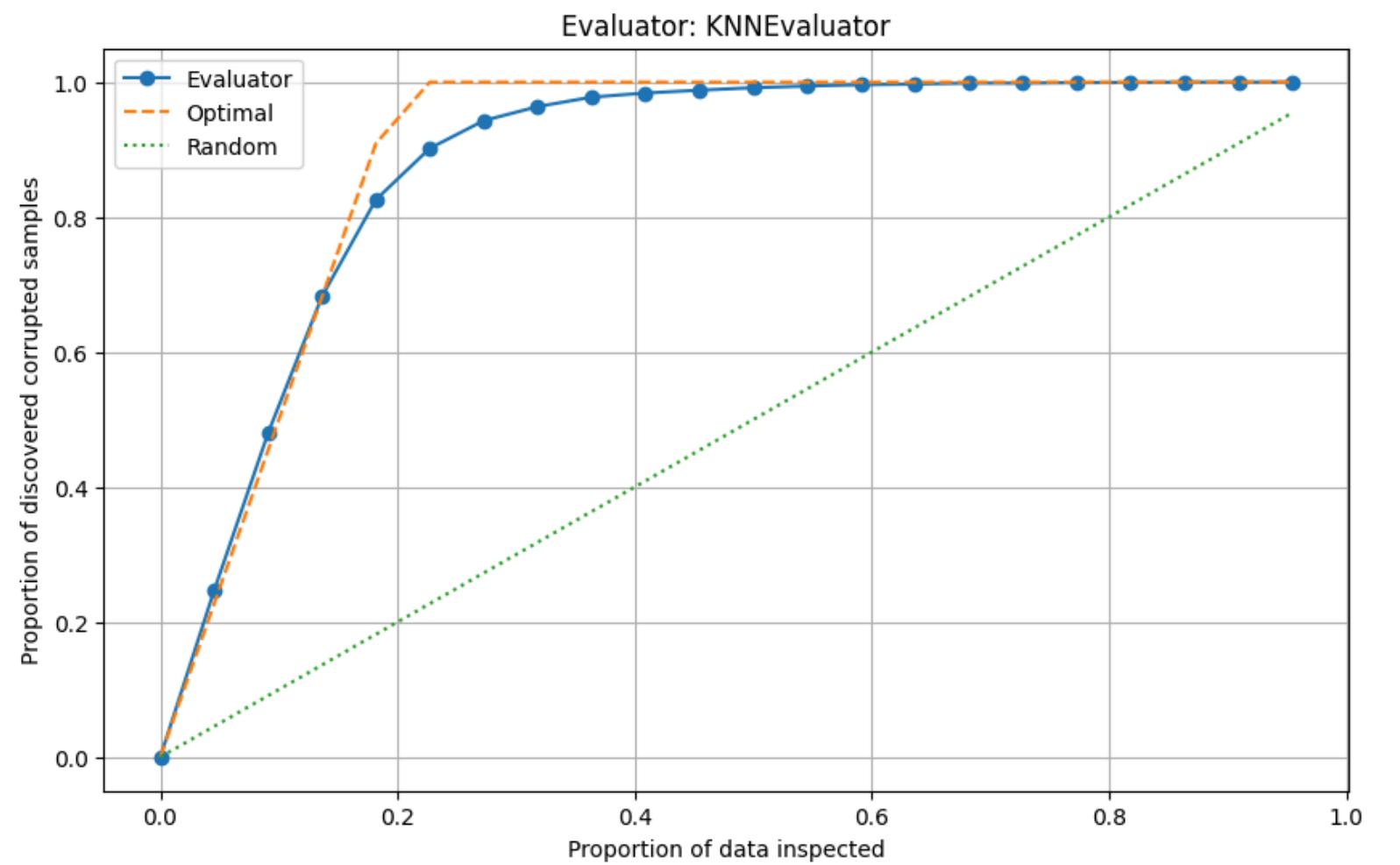
Evaluator	Peak Performance	25% of Data Points Remain
<i>Baseline Performance = 0.650</i>		
KNN Shapley	0.643	0.615
KNN Shapley (Best Threshold)	<b>0.65</b>	<b>0.62</b>
LAVA	0.715	0.705
LAVA+label-to-label	0.715	0.705
LAVA (OT Library Implementation)	0.715	0.64

## Kết quả thực nghiệm 2

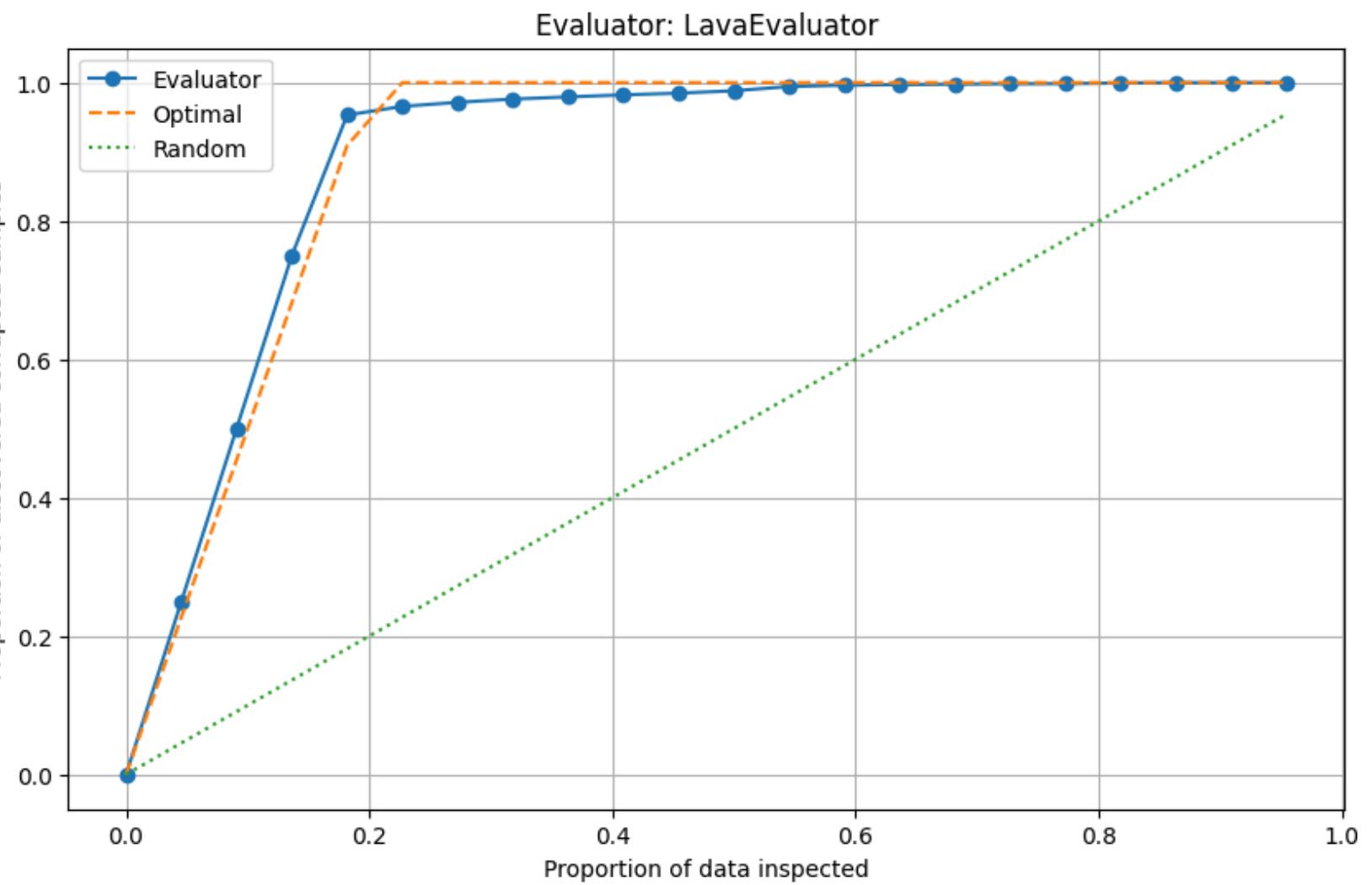


# Minh họa

## 2dplan-nhiều nhãn-knn-shapley



## 2dplan-nhiều đặc trưng-lava



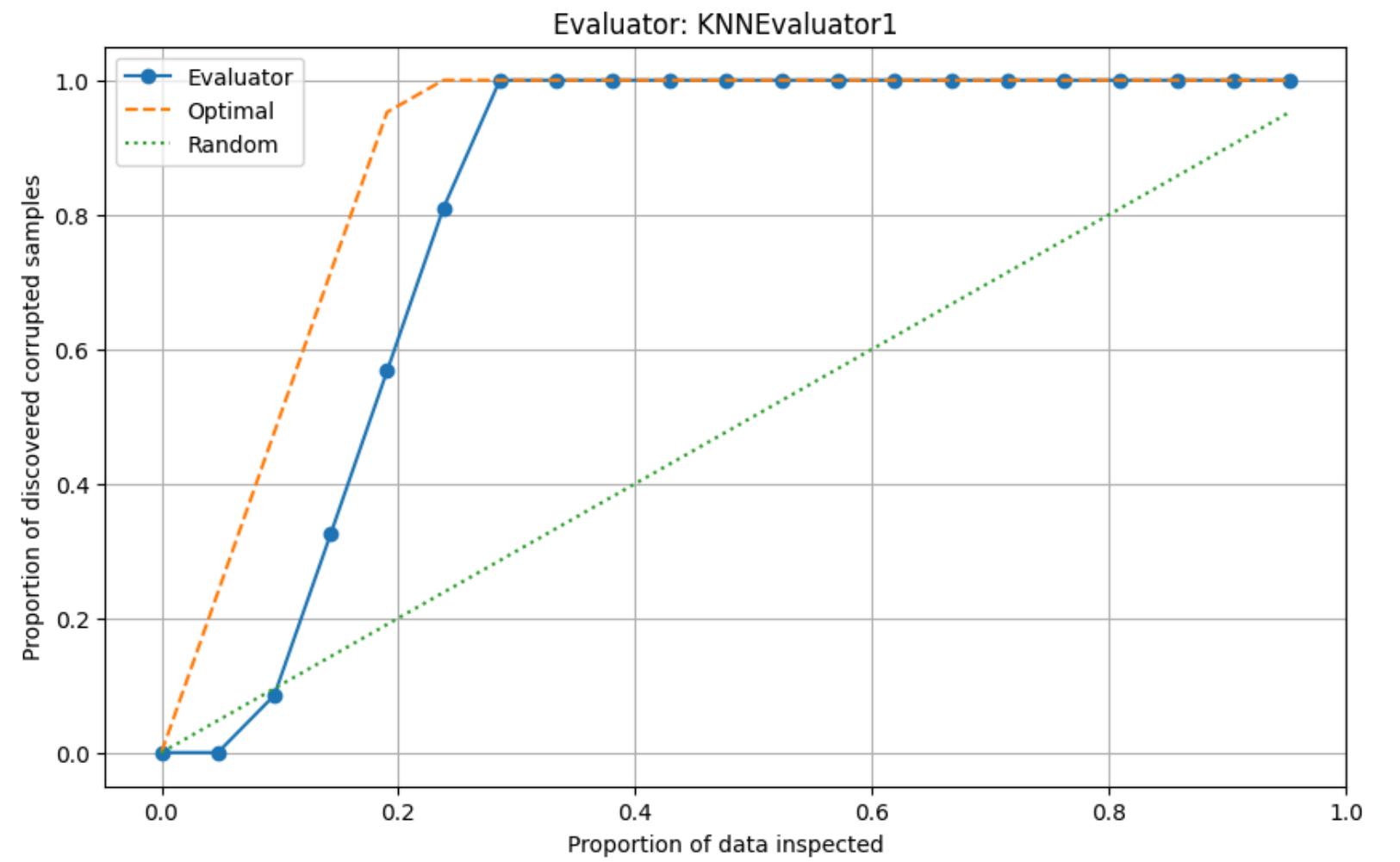
## Thực nghiệm trên bộ dữ liệu dạng bảng 2dplan

Trên một biểu đồ ta có ba đường lần lượt là: **màu xanh nước biển** là **tỉ lệ phát hiện nhiễu** trên cho lượng data point chúng ta lấy ra để xét. Hình **màu vàng** và **xanh lá cây** nét đứt lần lượt là **kết quả tối ưu** và **kết quả random**.

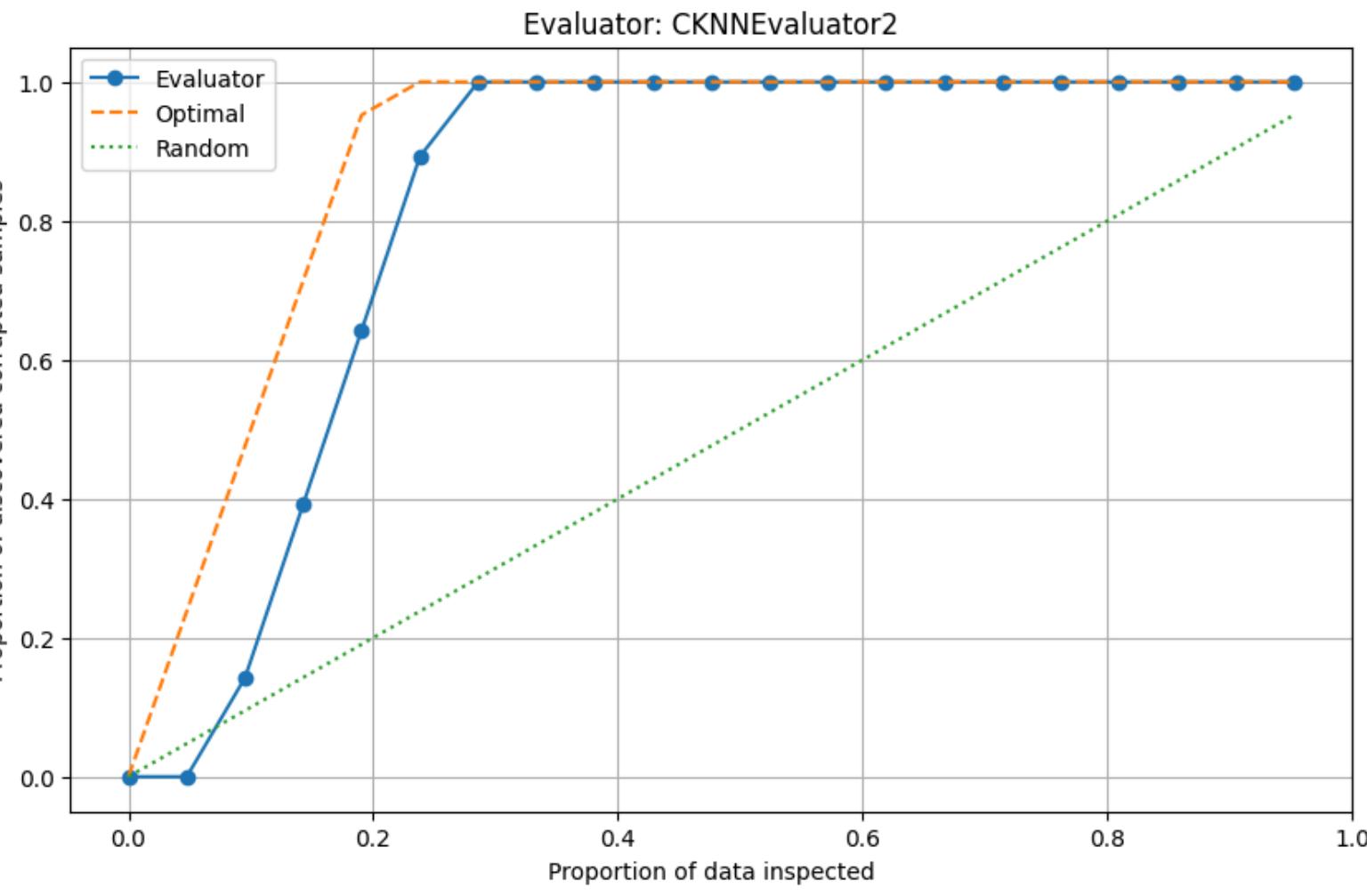


# Minh họa

## 2dplan-knn-shapley



## 2dplan-threshold KNN shapley

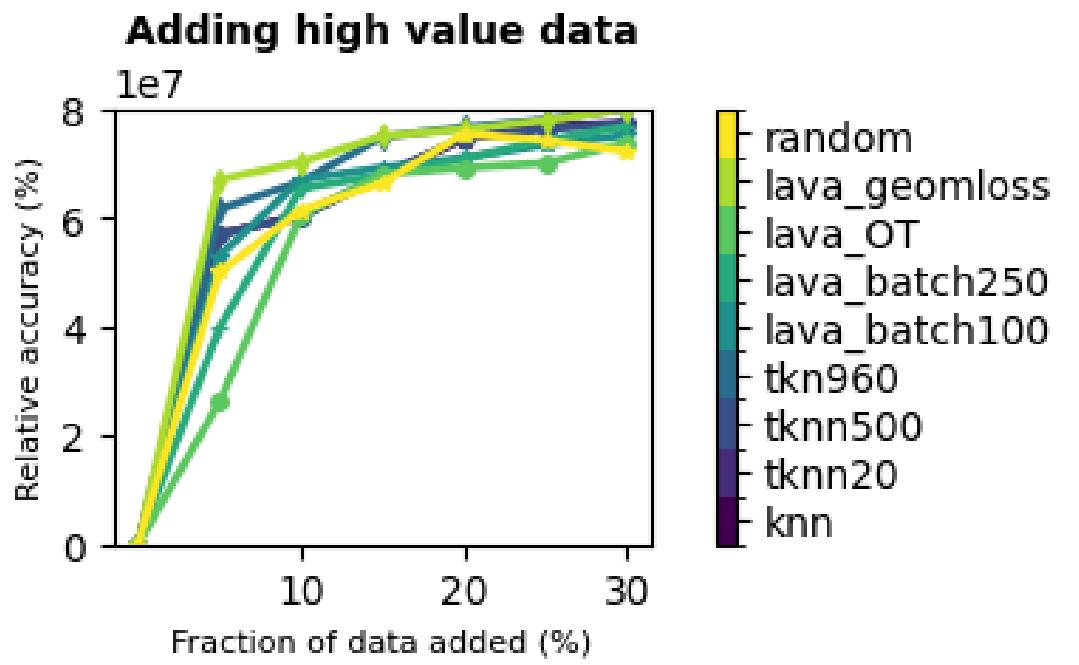
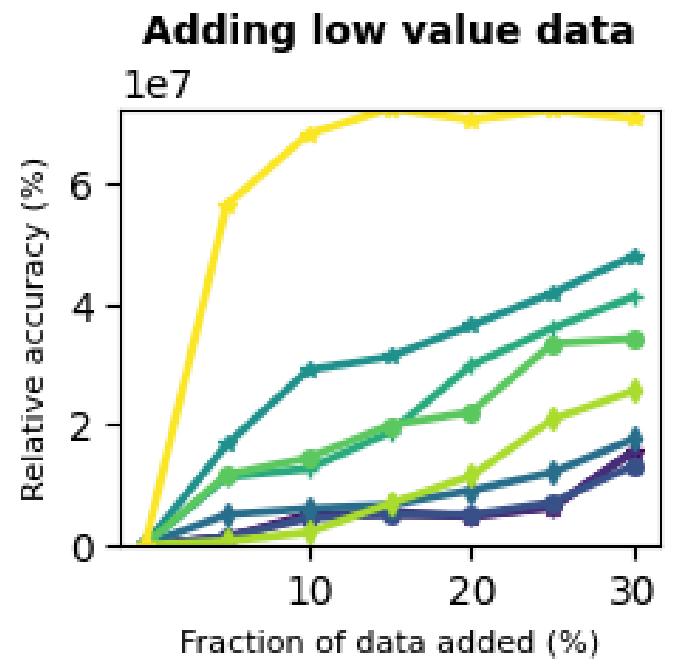
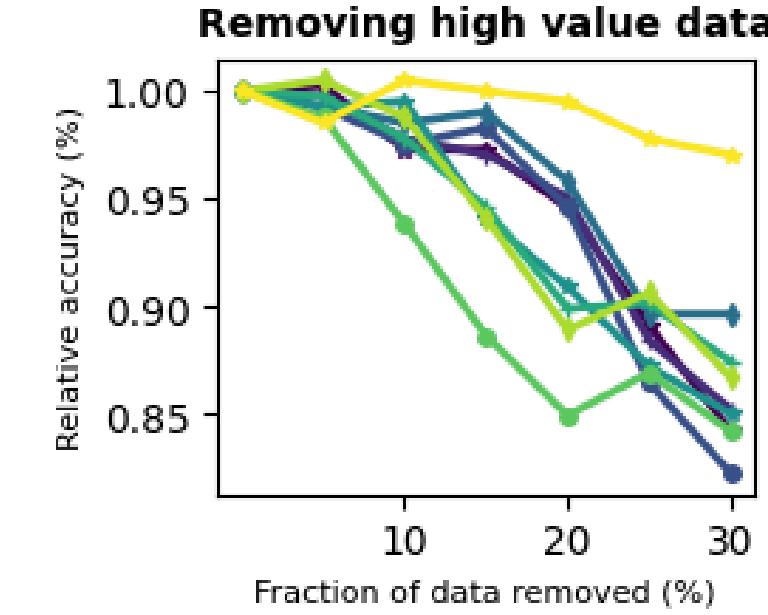
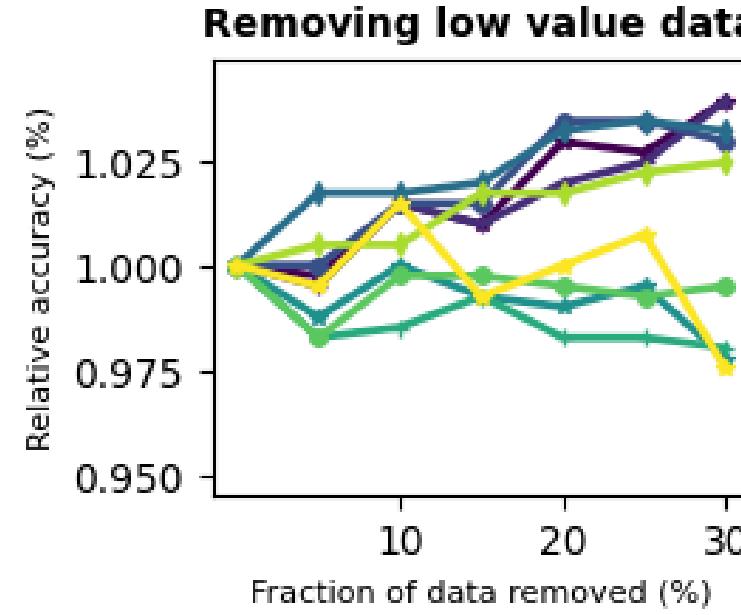


Thực nghiệm trên bộ 2dplan khi thêm nhiễu đặc trưng.

Điểm khác biệt giữa KNN-shapley đề xuất với KNN-shapley thông thường.



# Minh họa



- Sự thay đổi hiệu suất của các thuật toán khi ta chỉ xem xét 30% việc thao tác trên dữ liệu
  - Thực hiện thí nghiệm trên cifar
  - kích thước train:val: 10000:5000
  - Nhiều trên label tỉ lệ p=0.2
  - Sử dụng thuật toán để đánh giá: Logistic Regression



# Nhận xét

## 1. Phát Hiện Nhiễu Nhãn (Label Noise):

- KNN-Shapley vượt trội trong phát hiện nhiễu nhãn nhờ vào mô hình KNN.
- Khi nhãn bị nhiễu, khoảng cách đến các điểm test trong cùng lớp tăng, giúp KNN dễ nhận diện bất thường.

## 2. Phát Hiện Nhiễu Đặc Trưng (Feature Noise):

- LAVA tốt hơn trong phát hiện nhiễu đặc trưng nhờ nhạy cảm với sự khác biệt trong không gian đặc trưng.
- Khi đặc trưng thay đổi hoặc thêm nhiễu, LAVA dễ nhận ra điểm không nhất quán, phù hợp với dữ liệu biến thiên cao.



## Nhận xét

### 3. Hiệu Suất Mô Hình Khi Thêm/Xóa Dữ Liệu:

- **Xóa dữ liệu kém chất lượng:** Độ chính xác tăng nhẹ trước khi giảm dần.  
→ Cải thiện hiệu suất và giảm kích thước tập huấn luyện mà không mất nhiều độ chính xác.
- **Xóa dữ liệu chất lượng tốt:** Độ chính xác giảm nhanh xuống đáy.
- **Thêm dữ liệu kém chất lượng:** Hiệu suất mô hình chỉ tăng nhẹ hoặc đi ngang.
- **Thêm dữ liệu chất lượng tốt:** Mô hình nhanh chóng đạt đỉnh hiệu suất.



## Nhận xét

### 4. Nhận xét phương pháp đề xuất:

- **Threshold KNN-shapley:** Cho kết quả tốt hơn KNN-shapley ở nhiệm vụ nhiễu đặc trưng. Tương đối ngang ở nhiệm vụ nhiễu nhãn.
- **Batch LAVA:** Cho kết quả kém hơn LAVA lý do vì thực hiện trên dữ liệu nhỏ, các batch làm cho mô hình học không được tổng quát.
- **Batch LAVA (label-to-label):** Nhờ tính sẵn khoảng cách nhãn nên mô hình có kết quả ổn định so với LAVA gốc.
- **Ưu điểm của việc chia batch:** Giúp mô hình tối ưu thời gian và bộ nhớ khi thực hiện tính toán trên GPU.



**05**

# Thảo luận



# Hướng nghiên cứu tiếp theo



01

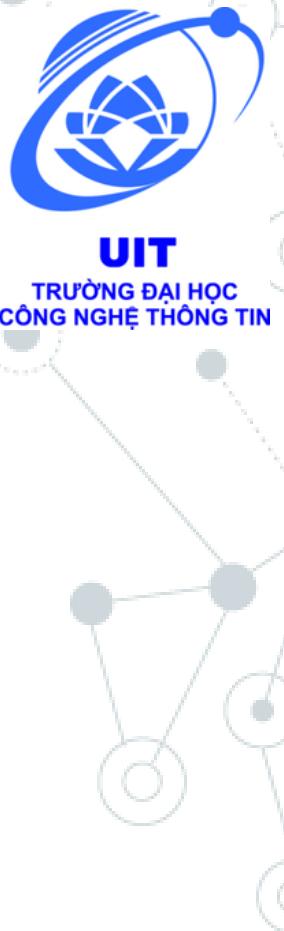
- Áp dụng thêm ứng dụng vào bộ dữ liệu NLP, time series.

02

- Áp dụng thẳng các thuật toán KNN-shapley, LAVA vào quy trình huấn luyện mô hình có dữ liệu nhiễu. Bằng cách gán lại nhãn có chất lượng thấp (khả năng cao bị nhiễu)



## Thảo luận



- **Ứng dụng:** Đánh giá dữ liệu (data valuation) là hướng tiếp cận tiềm năng, áp dụng trên nhiều loại tập dữ liệu khác nhau.
- **Cải tiến:** Thuật toán nhóm sử dụng:
  1. Tốc độ tính toán nhanh.
  2. Không yêu cầu mô hình học sẵn có.
  3. Hỗ trợ tiền xử lý, giải thích dữ liệu, và áp dụng real-time.
  4. Qua thực nghiệm hiệu quả phương pháp đề xuất không thua kém thuật toán truyền thống.



# Tài liệu tham khảo

- [1]. Amirata Ghorbani, James Y. Zou:  
Data Shapley: Equitable Valuation of Data for Machine Learning.  
ICML 2019: 2242-2251
- [2]. Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gürel, Bo Li, Ce Zhang, Costas J. Spanos, Dawn Song:  
Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms. CoRR abs/1908.08619 (2019).
- [3]. Hoang Anh Just, Feiyang Kang, Tianhao Wang, Yi Zeng, Myeongseob Ko, Ming Jin, Ruoxi Jia:  
LAVA: Data Valuation without Pre-Specified Learning Algorithms.  
ICLR 2023



06

# Appendix



# Thực nghiệm thêm

**Độ phức tạp thuật toán:**

*KNNshapley* :  $\mathcal{O}((N \log N) N_{\text{test}})$

*LAVA* =  $\mathcal{O}\left(\frac{N^2}{\varepsilon^2}\right)$



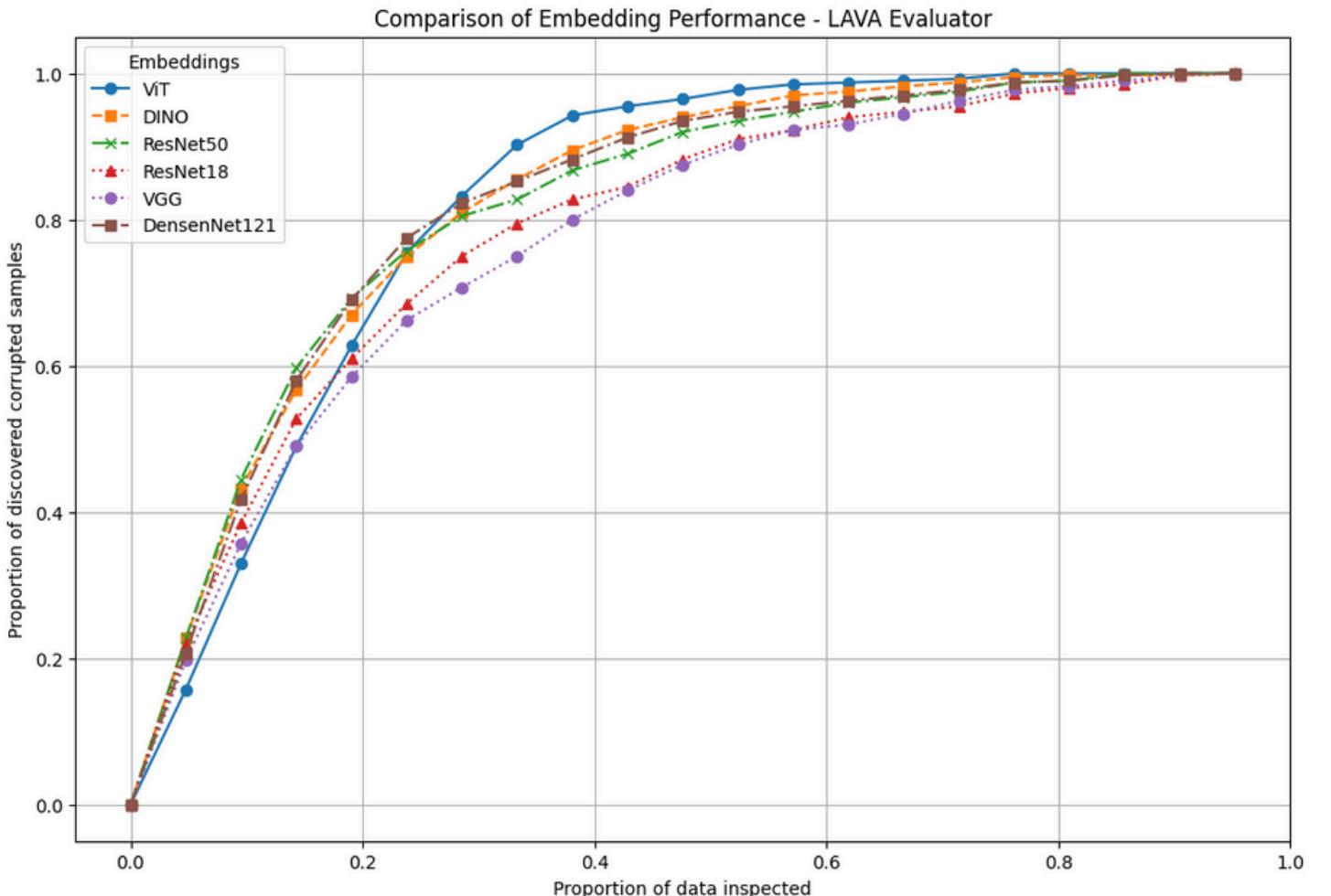
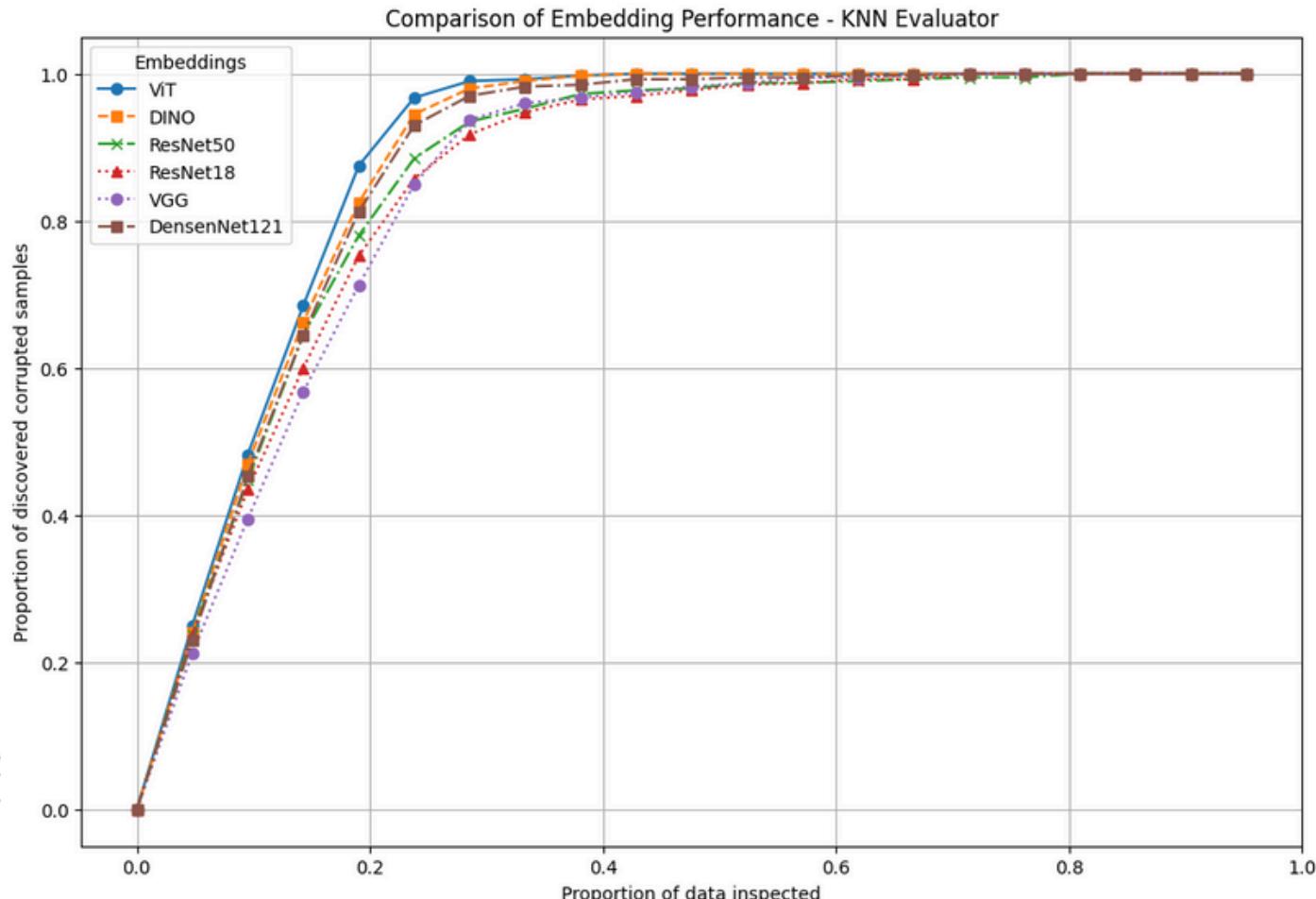
# Thực nghiệm thêm

## Chất lượng Embedding tác động thế nào đến các thuật toán:

Data valuation có thể dùng để kiểm tra trước chất lượng embedding trước khi cho vào mô hình học máy.

Evaluator	f1-score
resnet50	0.4650
resnet18	0.4650
vit	0.9158
vgg-16	0.7736
densenet121	0.8201
dino-small	0.8426

Huấn luyện mô hình trên cifar với 20% nhiễu bằng Logistic regression

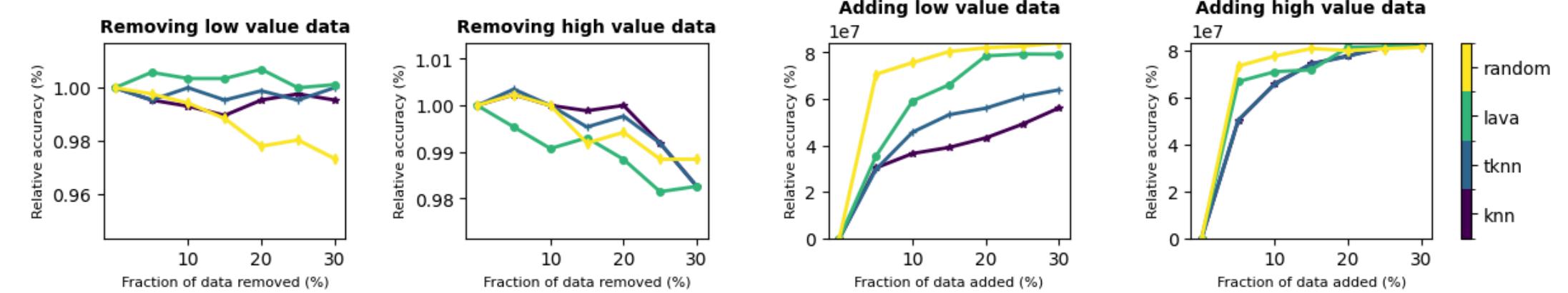




# Thực nghiệm thêm

## Dữ liệu mất cân bằng:

Trên bộ dữ liệu mất cân bằng, các thuật toán không chỉ làm giảm số lượng điểm dữ liệu mà còn làm tăng hiệu suất mô hình.



**Chọn ra 10 lớp trong cifar, trong đó số lượng lớp frog gấp đôi số lớp còn lại tiến hành loại bỏ và đánh giá bằng logistic regression**



# KNN Shapley

B1) Thiết kế hàm đánh giá (Utility) dựa vào bộ phân loại KNN:

$$V(S) = \frac{1}{\min(K, |S|)} \sum_{k=1}^{\min(K, |S|)} 1_{y_{\alpha_k(S)} = y_{\text{test}}}$$

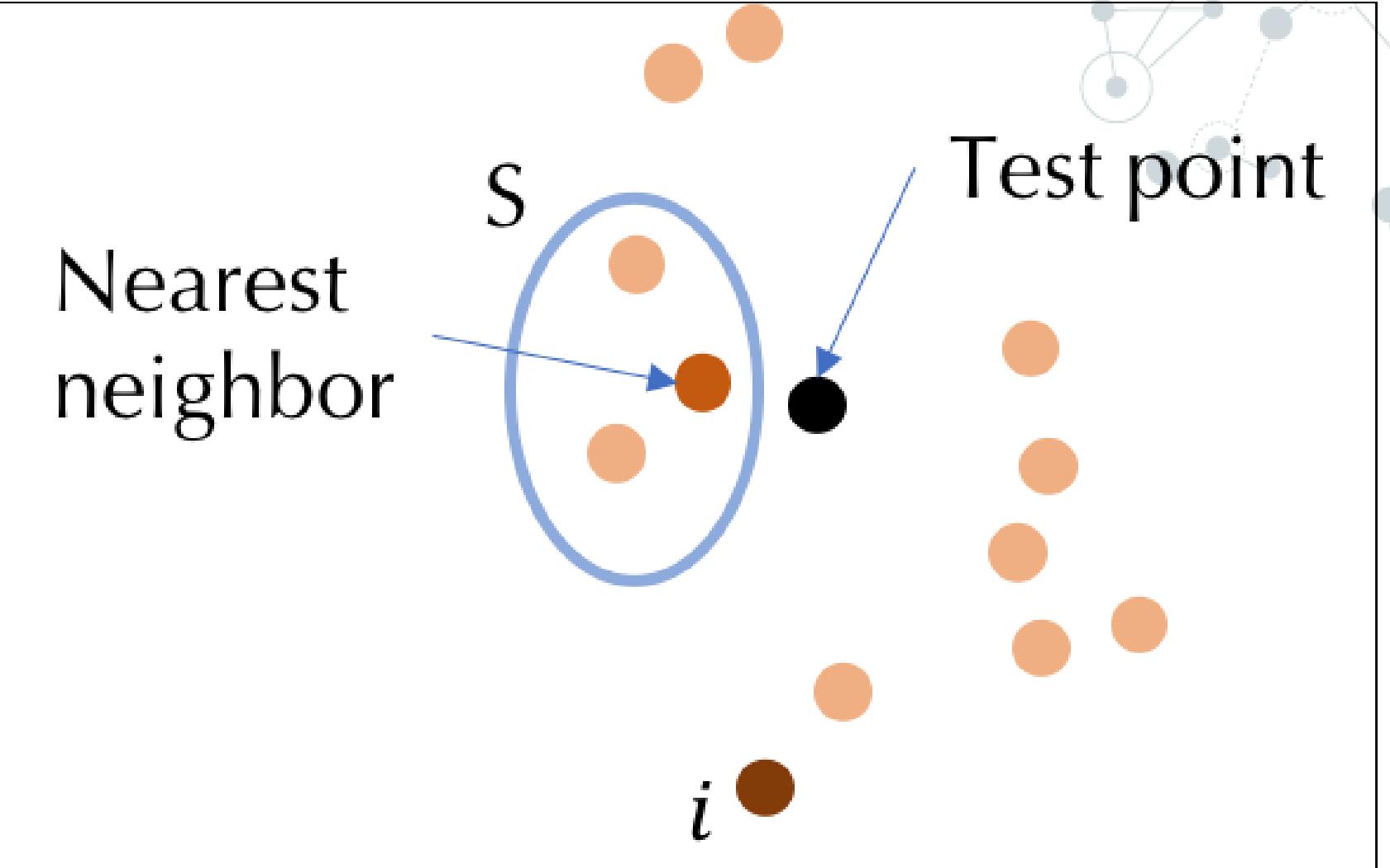
Công thức trên 1 điểm test:

$$\phi_{\alpha_N} = \frac{1_{y_{\alpha_N} = y_{\text{test}}}}{N}$$

$$\phi_{\alpha_i} = \phi_{\alpha_{i+1}} + \frac{1_{y_{\alpha_i} = y_{\text{test}}} - 1_{y_{\alpha_{i+1}} = y_{\text{test}}}}{K} \frac{\min(K, i)}{i}$$

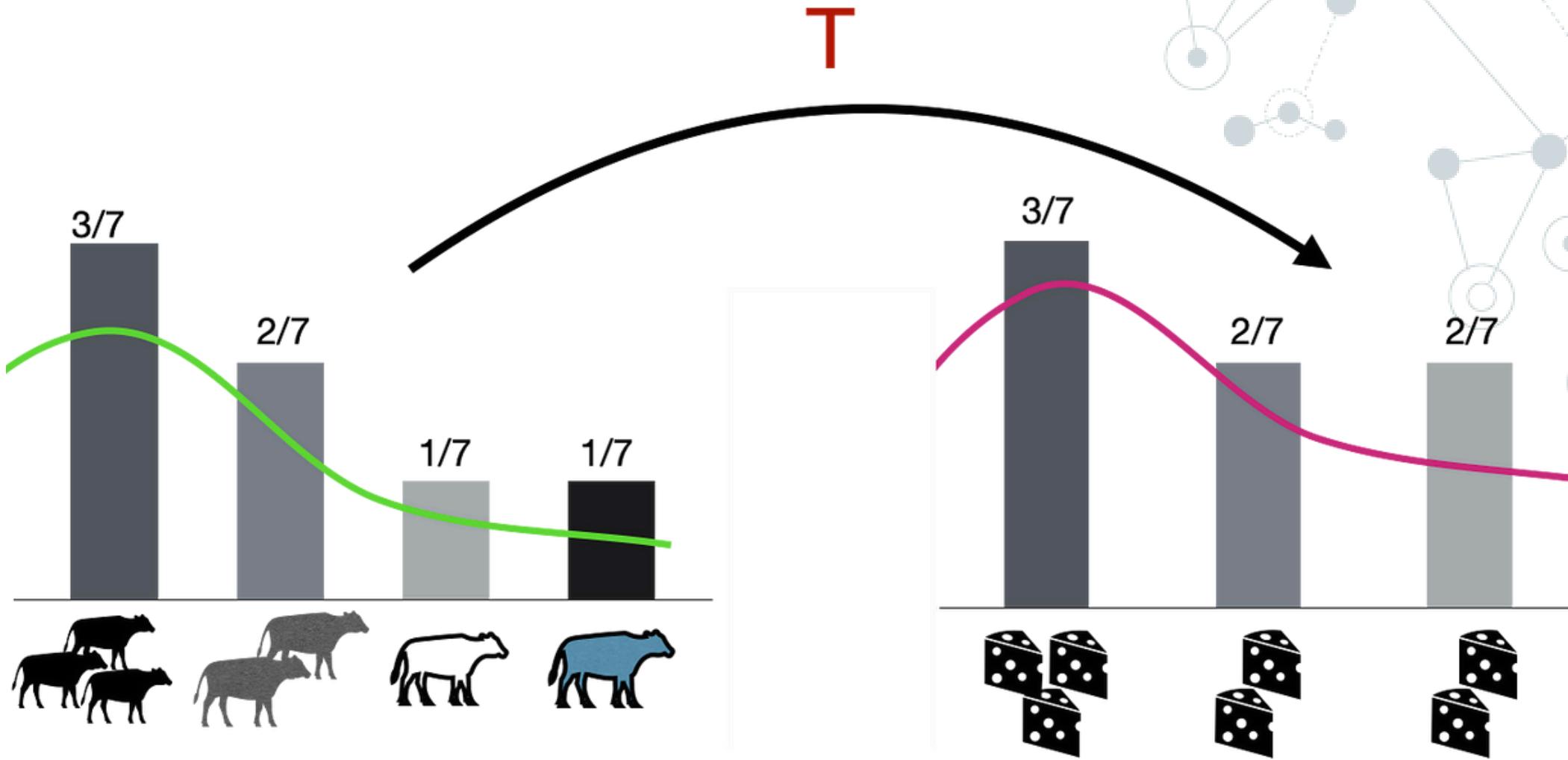
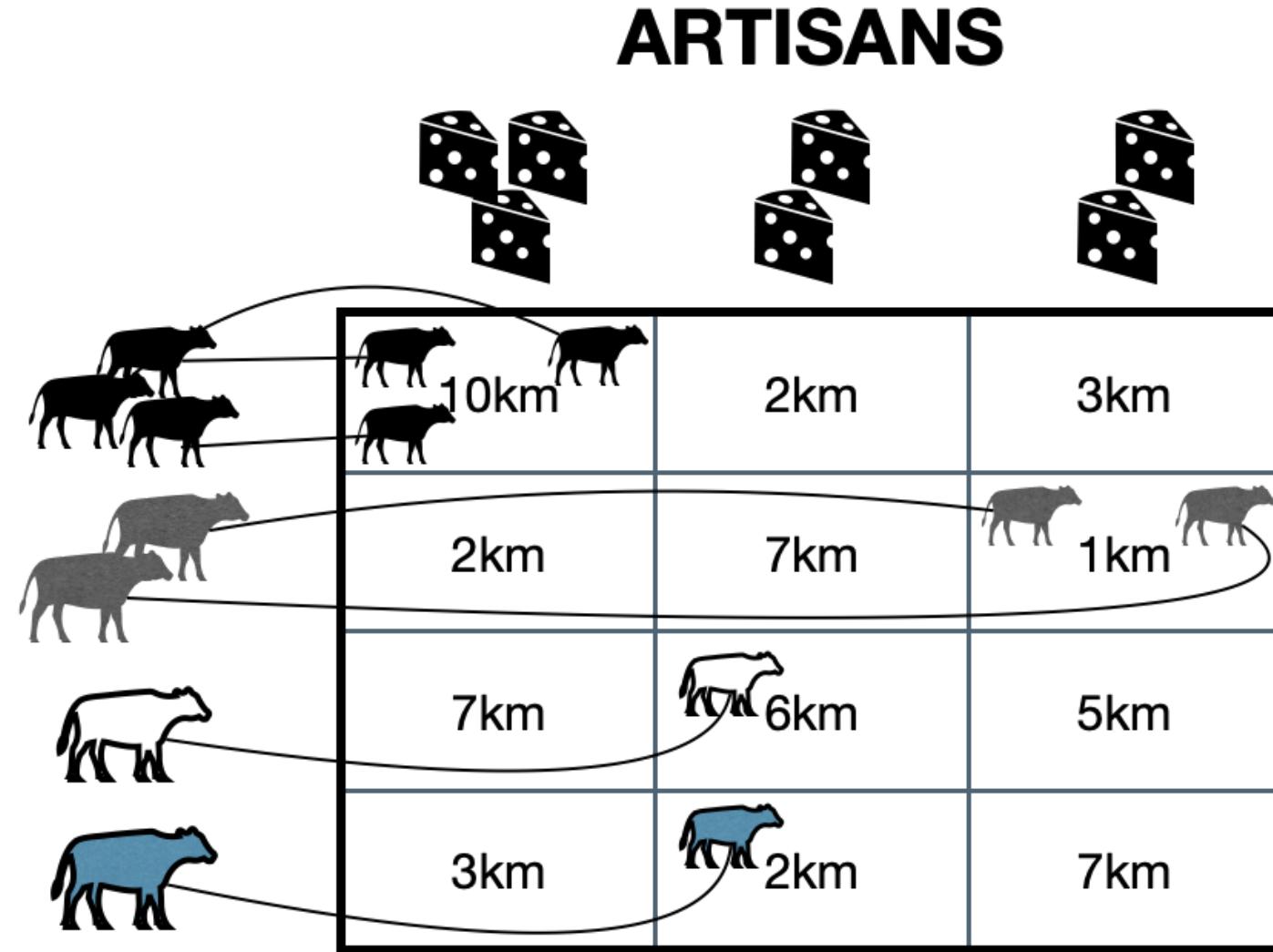
Công thức trên nhiều điểm test:

$$\phi_i^{all} = \frac{1}{N_{\text{test}}} \sum_{j=1}^{N_{\text{test}}} \phi_{ji}$$





# Bài toán vận chuyển



## Bài toán gồm:

- Nguồn cung và nhu cầu.
- Chi phí vận chuyển.  
=> Tối thiểu hóa tổng chi phí vận chuyển.

## Đảm bảo:

- Tất cả nguồn cung được sử dụng hết.
- Tất cả nhu cầu được đáp ứng hết.



# Định nghĩa

Giả sử ta có hai phân phối rời rạc  $p = (p_1, p_2, \dots, p_n)$ ,  $q = (q_1, q_2, \dots, q_m)$

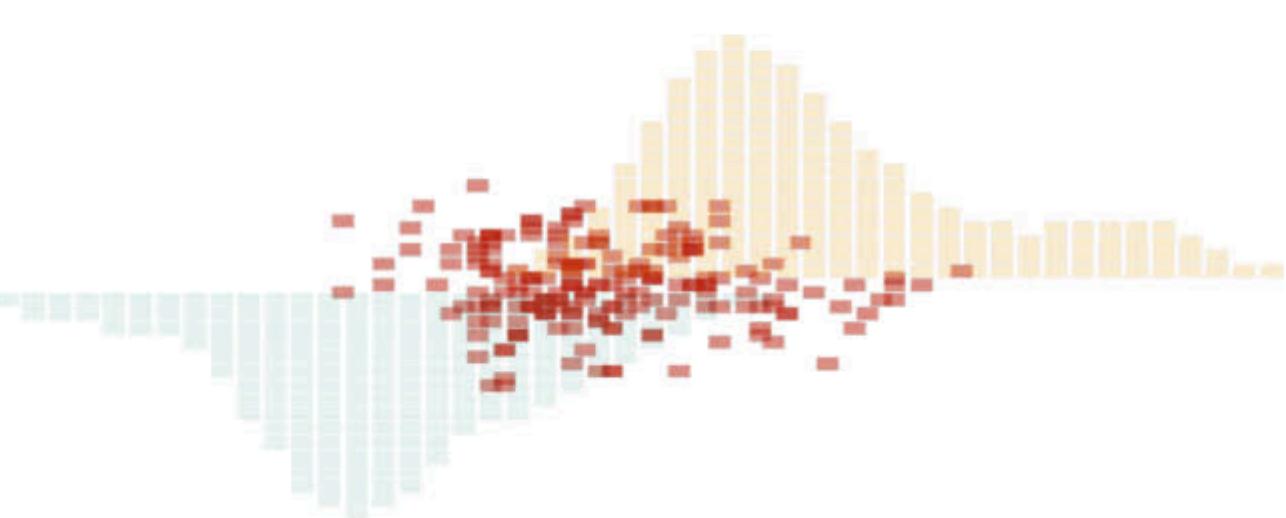
đại diện cho phân phối xác suất trên các điểm rời rạc  $(x_1, x_2, \dots, x_n)$ ,  $(y_1, y_2, \dots, y_m)$

Mục tiêu là tìm phân phối chung  $\gamma$  các phép vận chuyển từ  $x_i$  sang  $y_j$  sao cho chi phí vận chuyển là tối thiểu.

$$\mathcal{L}(\gamma) = \min_{\gamma} \sum_{i=1}^n \sum_{j=1}^m c(x_i, y_j) \gamma_{ij}$$

$$\sum_{j=1}^m \gamma_{ij} = p_i \quad , \quad \sum_{i=1}^n \gamma_{ij} = q_j$$

“Optimal Transport” (a side note / a cultural interlude)



Trong đó:

Monge (1781), “Mémoire sur la théorie des déblais et des remblais”

$\gamma_{ij}$  là lượng khối lượng vận chuyển từ điểm  $x_i$  đến điểm  $y_j$

$c(x_i, y_j)$  là chi phí vận chuyển từ  $x_i$  đến điểm  $y_j$

Phân phối nguồn p và phân phối đích q: Cho biết tổng khối lượng tại các điểm bên nguồn và bên đích.



## Optimal transport bao gồm:

$$\mathcal{L}(\gamma) = \min_{\gamma} \sum_{i=1}^n \sum_{j=1}^m c(x_i, y_j) \gamma_{ij}$$

$$\sum_{j=1}^m \gamma_{ij} = p_i \quad , \quad \sum_{i=1}^n \gamma_{ij} = q_j$$

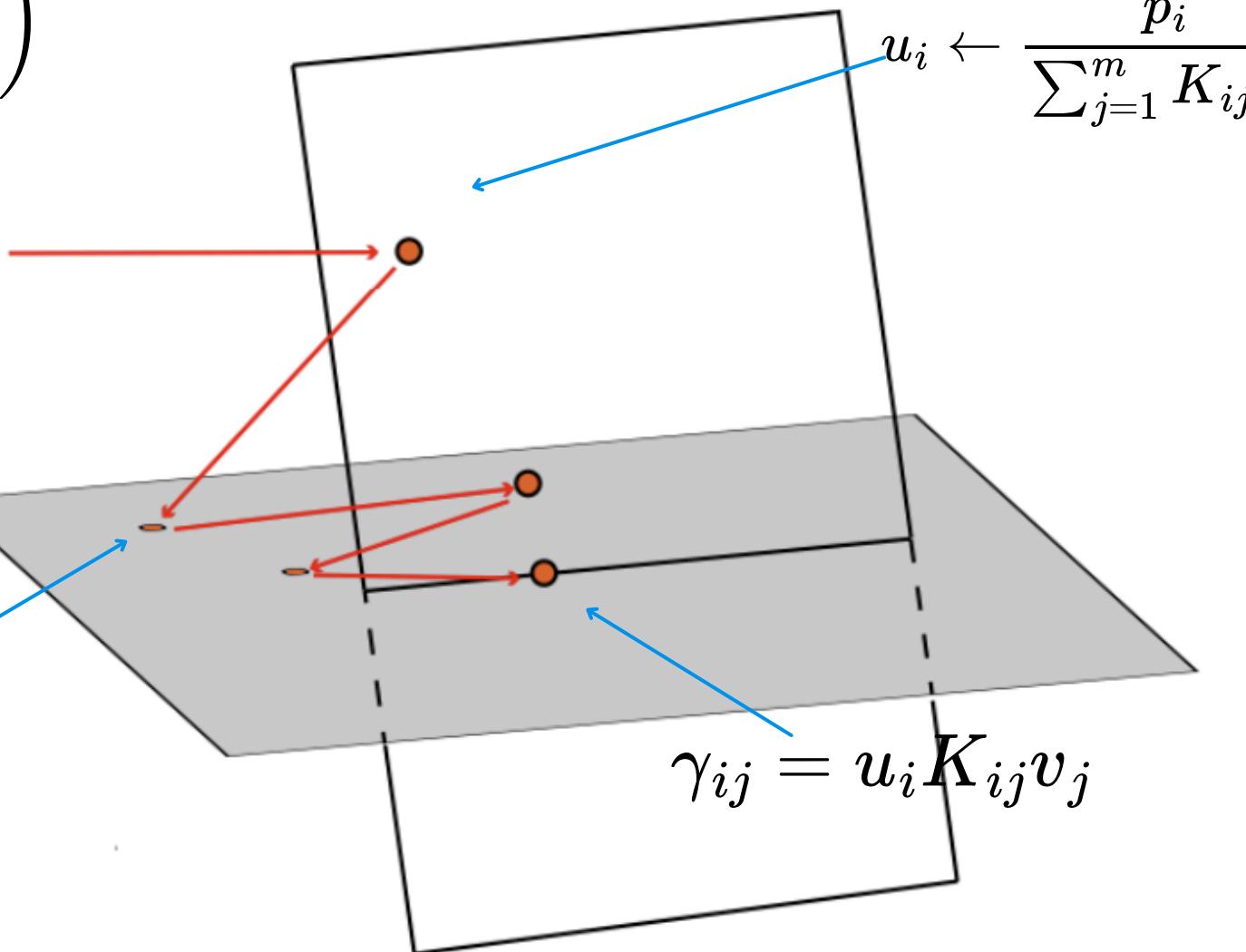
Trong đó:

$\gamma_{ij}$  là lượng khối lượng vận chuyển từ điểm  $x_i$  đến điểm  $y_j$

$c(x_i, y_j)$  là chi phí vận chuyển từ  $x_i$  đến điểm  $y_j$

Phân phối nguồn  $p$  và phân phối đích  $q$ : Cho biết tổng khối lượng tại các điểm bên nguồn và bên đích.

$$K_{ij} = \exp\left(-\frac{c(x_i, y_j)}{\epsilon}\right)$$
$$v_j \leftarrow \frac{q_j}{\sum_{i=1}^n K_{ij} u_i}$$



## Minh họa thuật toán sinkhorn



# Sinkhorn

**Thay vì tối ưu hóa trên  $\gamma$  ta đi tối ưu hóa  $\alpha$  và  $\beta$  sử dụng chuyển hóa lagrange như sau:**

$$\mathcal{L}(\gamma, \alpha, \beta) = \sum_{i,j} \gamma_{ij} c_{ij} + \varepsilon \sum_{i,j} \gamma_{ij} (\ln \gamma_{ij}) - \sum_i \alpha_i \left( \sum_j \gamma_{ij} - p_i \right) - \sum_j \beta_j \left( \sum_i \gamma_{ij} - q_j \right) - \varepsilon \left( \sum_{ij} \gamma_{ij} - 1 \right)$$

**Đạo hàm theo  $\gamma$ :**

$$\frac{\partial \mathcal{L}}{\partial \gamma_{ij}} = c_{ij} + \varepsilon(\ln \gamma_{ij} + 1) - \alpha_i - \beta_j - \varepsilon = 0$$

$$\ln \gamma_{ij}^* = - \left( \frac{c_{ij} - \alpha_i - \beta_j}{\varepsilon} \right)$$

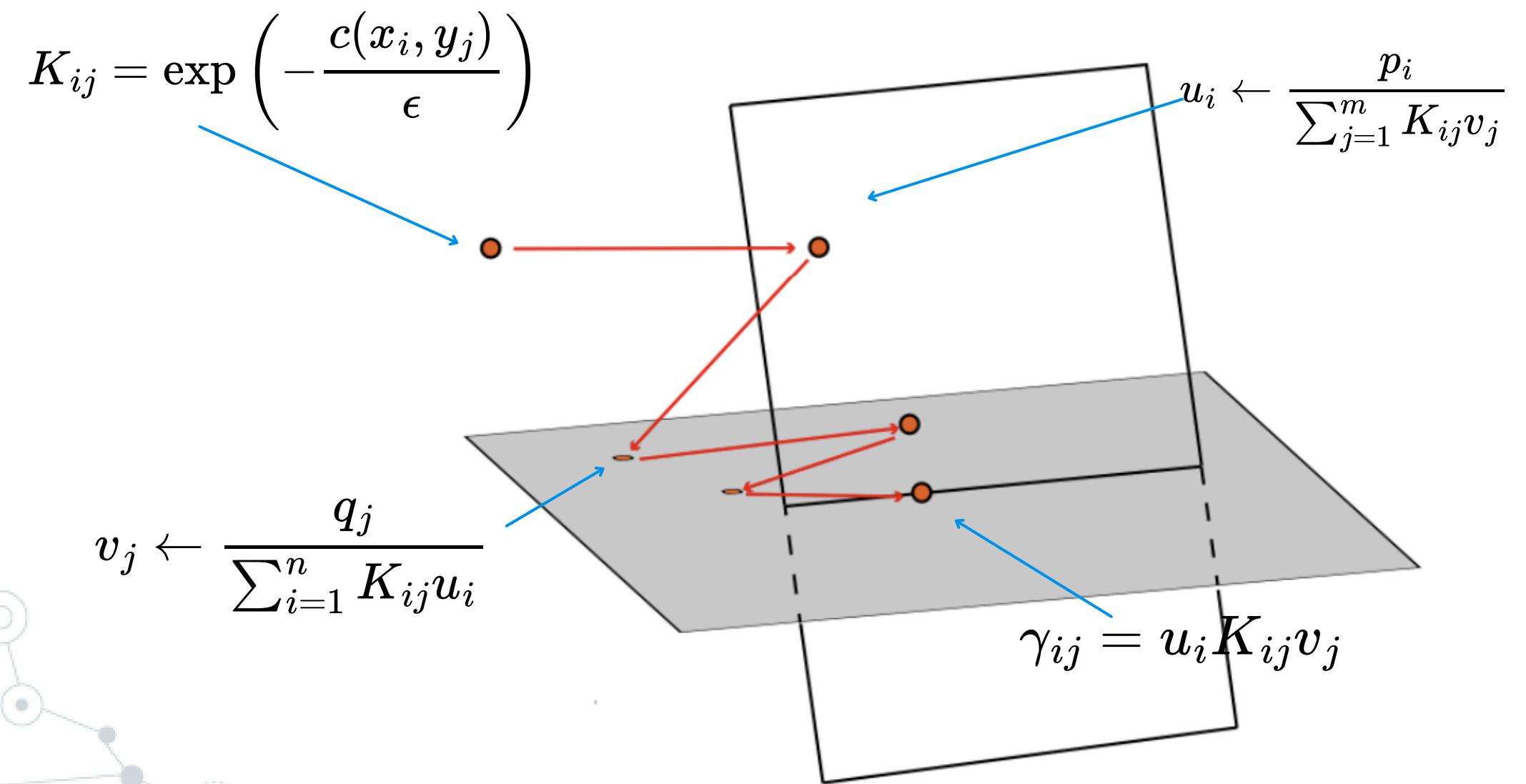


# Sinkhorn



$$\gamma_{ij}^* = \exp\left(\frac{\alpha_i}{\varepsilon}\right) \exp\left(-\frac{c_{ij}}{\varepsilon}\right) \exp\left(\frac{\beta_j}{\varepsilon}\right)$$

Đặt  $K_{ij} = \exp\left(-\frac{c_{ij}}{\varepsilon}\right)$ ,  $u_i = \exp\left(\frac{\alpha_i}{\varepsilon}\right)$ ,  $v_j = \exp\left(\frac{\beta_j}{\varepsilon}\right)$  :  $\gamma_{ij} = u_i K_{ij} v_j$   
 $\gamma 1 = p, \gamma^T 1 = q$





# Calibrated Gradients



Trong đó:

$$\text{OT}(\gamma^*(\mu_t, \mu_\nu)) = \text{OT}(f^*, g^*)$$

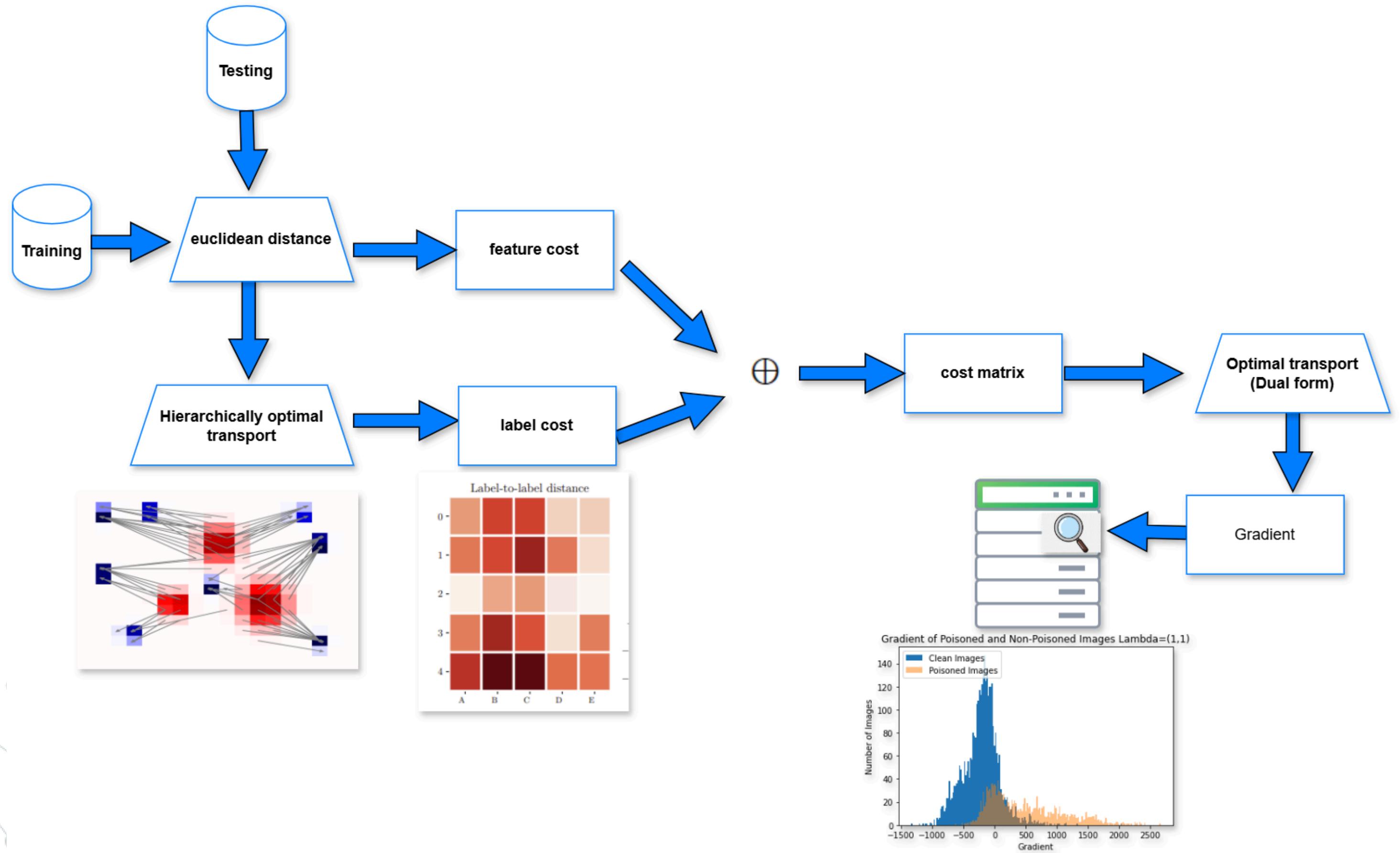
**Đạo hàm của khoảng cách OT trên phân phối xác suất của mỗi điểm dữ liệu:**

$$\frac{\partial \text{OT}}{\partial \mu_t(z_i)} = f_i^*, \quad \frac{\partial \text{OT}}{\partial \mu_\nu(z'_j)} = g_j^*.$$

$$\frac{\partial \text{OT}(\mu_t, \mu_\nu)}{\partial \mu_t(x_i)} = f_i^* - \sum_{j \in \{1, \dots, N\} \setminus i} \frac{f_j^*}{N-1}, \quad \frac{\partial \text{OT}(\mu_t, \mu_\nu)}{\partial \mu_\nu(x_j)} = g_j^* - \sum_{i \in \{1, \dots, M\} \setminus j} \frac{g_i^*}{M-1}. \quad 21$$

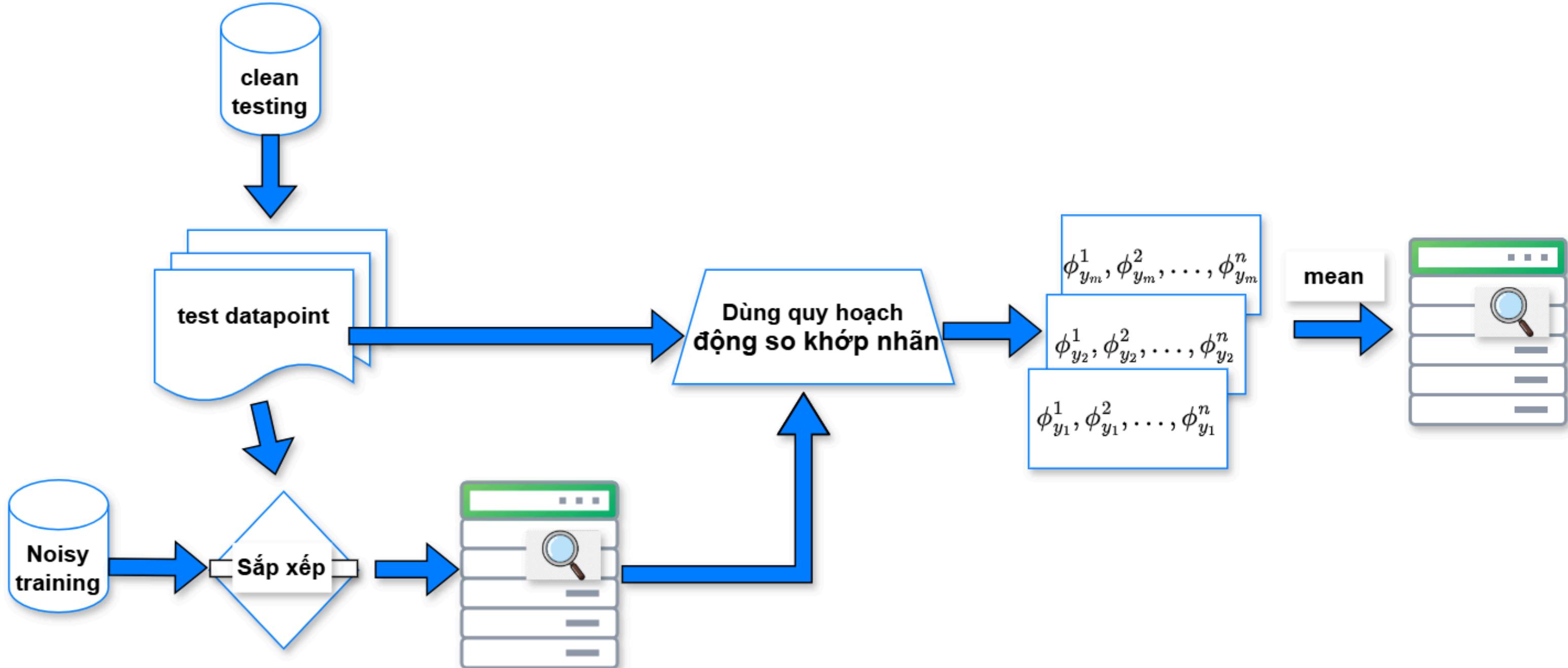


# LAVA





# KNN Shapley





# Khó khăn

## Phụ thuộc vào thuật toán học (A)

Giá trị của dữ liệu bị ảnh hưởng mạnh bởi lựa chọn thuật toán học.

## Định giá dữ liệu trước khi chọn thuật toán

- Trong các quy trình thu thập dữ liệu, ưu tiên các nguồn dữ liệu khác nhau.
- Định giá trong thị trường dữ liệu.
- Yêu cầu dữ liệu được định giá trước khi lựa chọn thuật toán học cụ thể.

## Gánh nặng tính toán

- Phải chạy lại thuật toán nhiều lần (có và không có từng điểm dữ liệu) để đánh giá giá trị.
- Không khả thi với các tập dữ liệu lớn do chi phí tính toán quá cao.