Learned Data-aware Image Representations of Line Charts for Similarity Search

Yuyu Luo[1], Yihui Zhou[1], Nan Tang[2,3], Guoliang Li[1], Chengliang Chai[4], Leixian Shen[1]

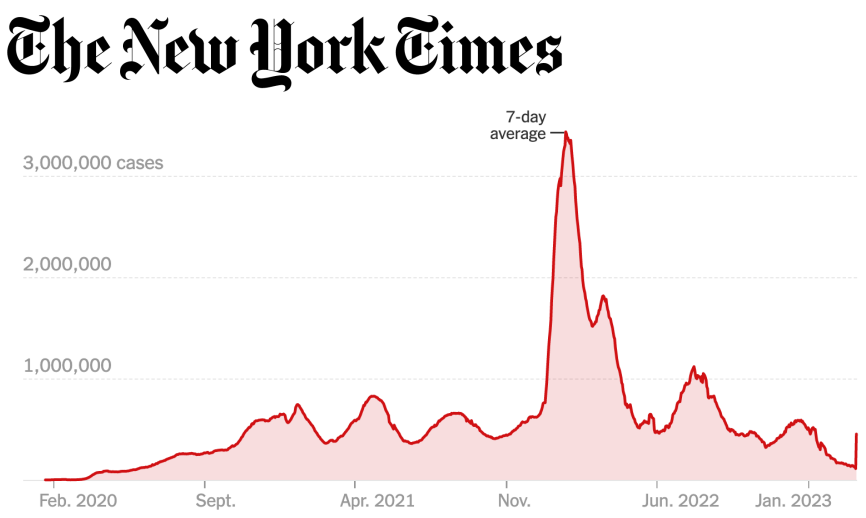
1 Tsinghua University

2 QCRI معهد قطر لبحوث الحوسبة Qatar Computing Research Institute جامعة حمد بن خليفة HAMAD BIN KHALIFA UNIVERSITY

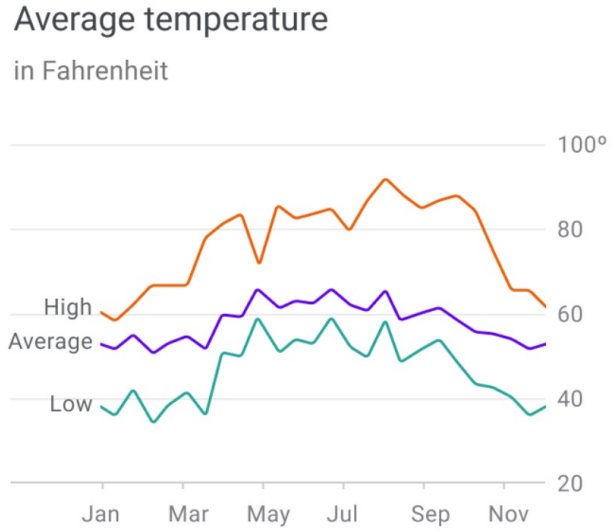
3 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

4 BEIJING INSTITUTE OF TECHNOLOGY

Line charts are everywhere

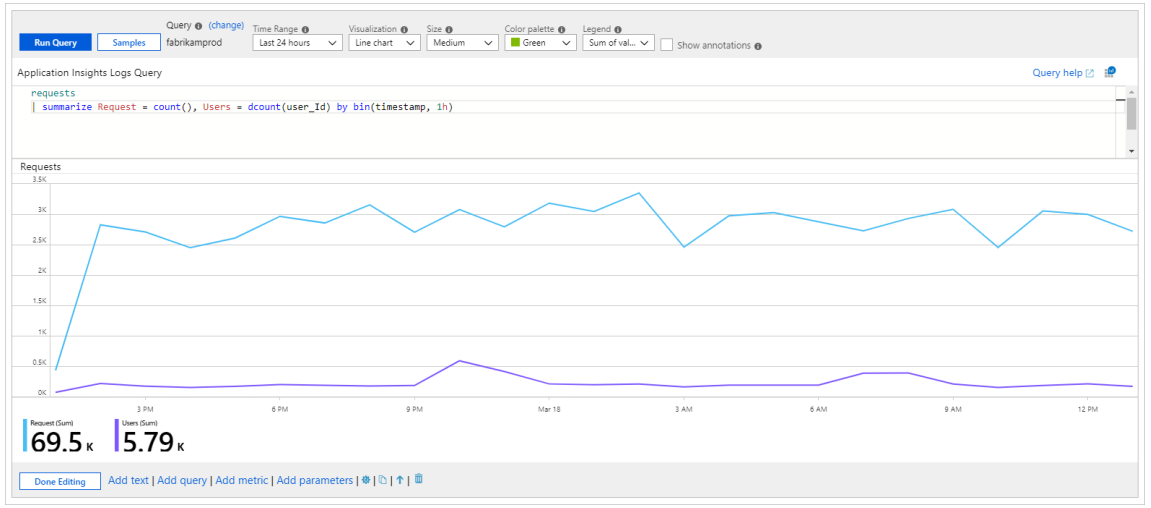


Digital News



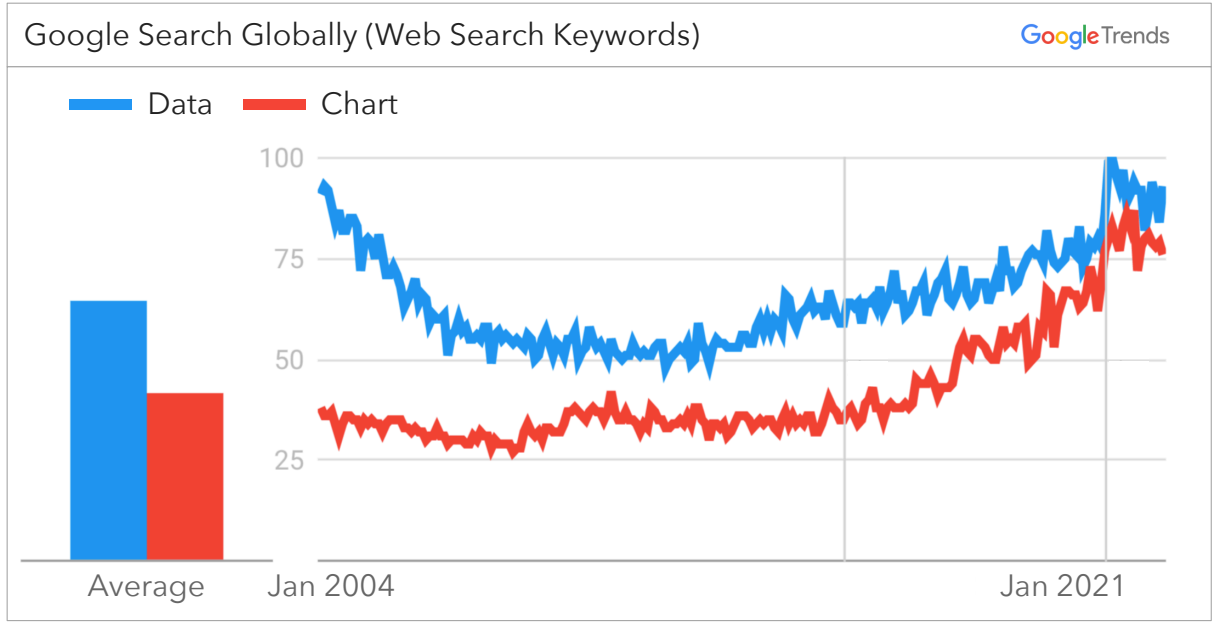
Temperature Tracking

Stock Market

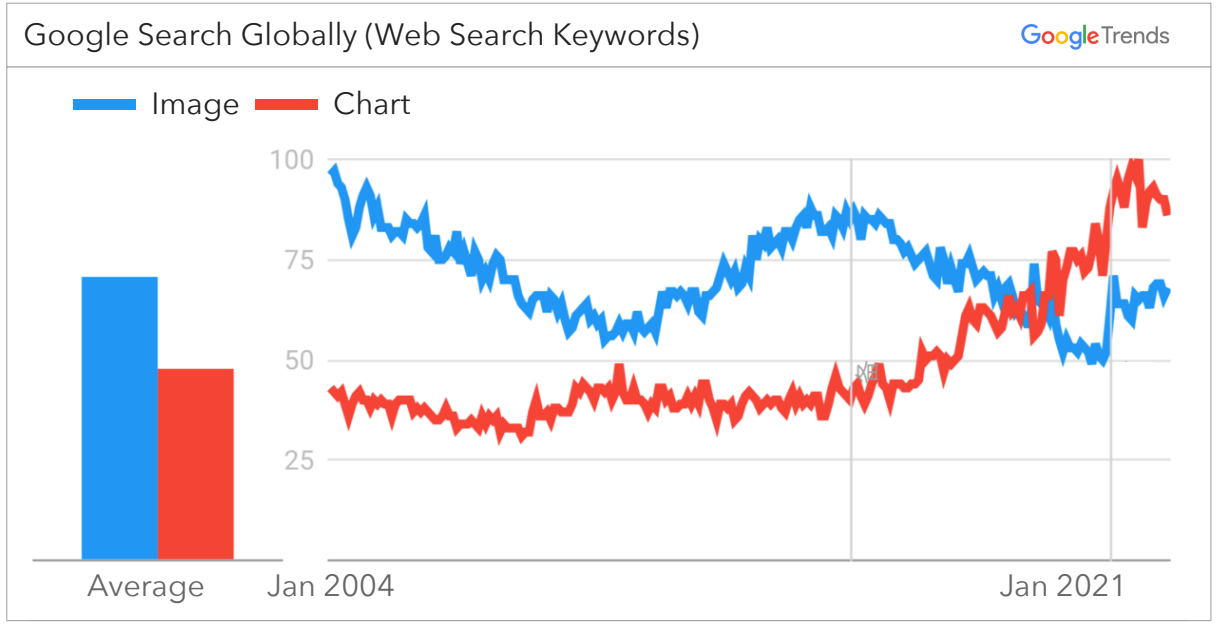


Server Monitoring

Visualization charts have gained increasing importance



Google Search Trends: ***Data*** vs. ***Chart***



Google Search Trends: ***Image*** vs. ***Chart***

Visualization charts have become a new type of data![1]

[1] Aoyu Wu et al. AI4VIS: Survey on Artificial Intelligence Approaches for Data Visualization. IEEE TVCG 2021.

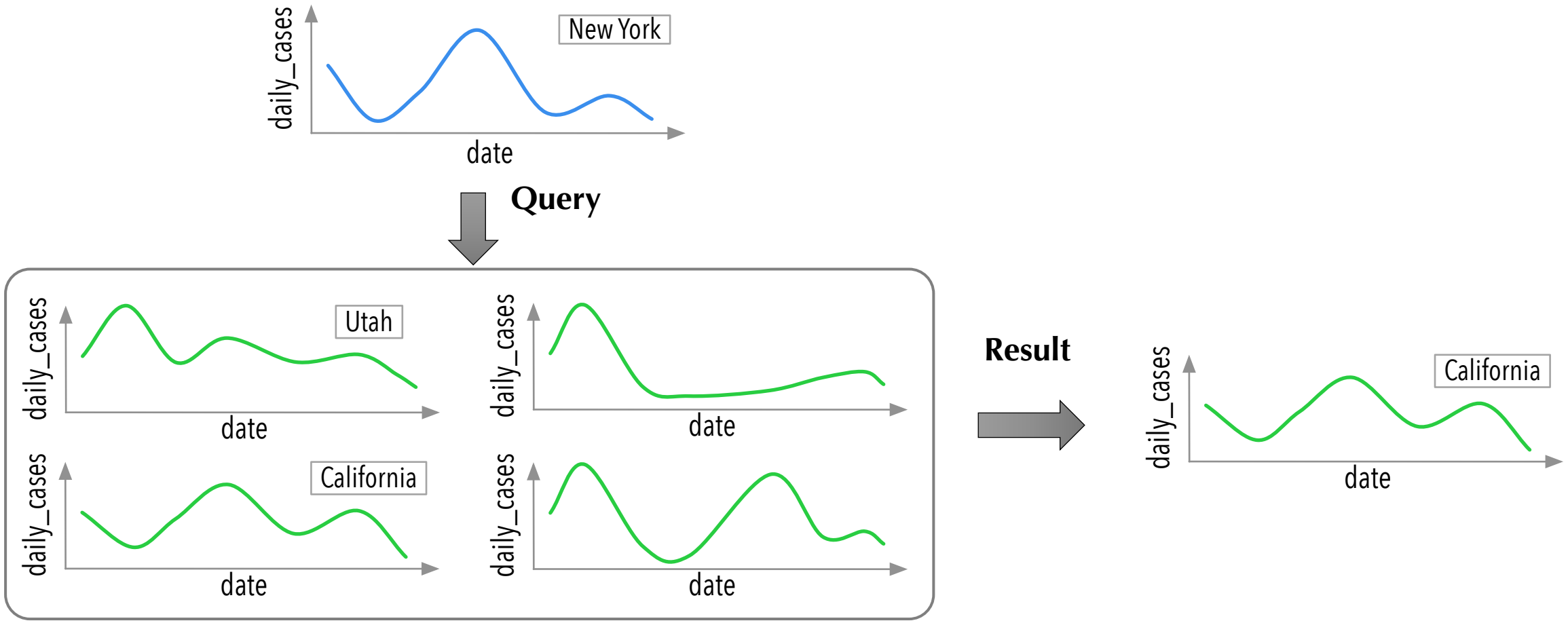
Problem: How to search similar line charts ?

How to search similar line charts has attracted extensive attention from academia and industry, it can support various applications:

- visualization recommendation
- visual query systems
- visualization corpus construction
- web mining
- computational journalism

[1] Miro Mannino at al. Qetch: Time Series Querying with Expressive Sketches. SIGMOD 2018.
[2] Tarique Siddiqui, et al. Parameswaran. 2020. ShapeSearch: A Flexible and Efficient System for Shape-based Exploration of Trendlines. SIGMOD 2022
[3] Enamul Hoque and Maneesh Agrawala. 2020. Searching the Visual Style and Structure of D3 Visualizations. IEEE Trans. Vis. Comput. Graph. 26, 1 (2020).
[4] Haotian Li, Yong Wang, Aoyu Wu, Huan Wei, and Huamin Qu. Structure-Aware Visualization Retrieval. CHI Conference 2022.

Similarity Search of Line Charts



The key premise of this problem **is how to measure the similarity** between line charts.

Similarity Measure of Line Charts

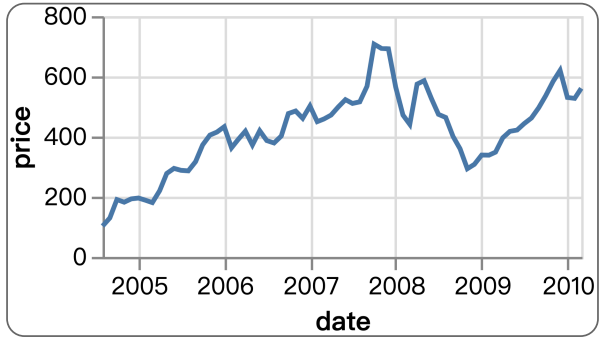
Underlying Data ($\mathbf{D}$)

date	price
Jan 1 2000	102.37
Feb 1 2000	129.6
Mar 1 2000	190.64
…	…
Jan 1 2010	593.23

Mark: line
X-axis: date
Y-axis: price
…

Visual Encoding

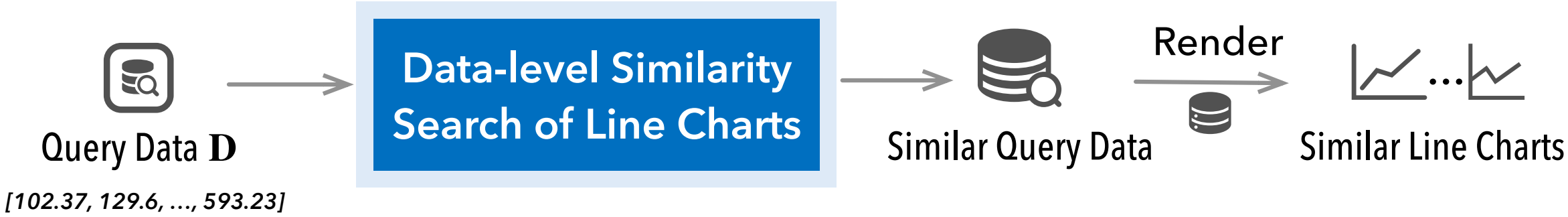
Rendered Line Chart ($\mathbf{V}$)



(*In the form of an* ***image***)

- Measured by Underlying Data — **Data-level Similarity**
- Measured by Line-Chart Images — **Image-level Similarity**

Data-level Similarity Search of Line Charts



$$\mathbf{dist}(\mathrm{L}_i, \mathrm{L}_j) = \mathbf{dist}(\mathrm{D}_i, \mathrm{D}_j)$$

e.g., Euclidean distance

Pros:

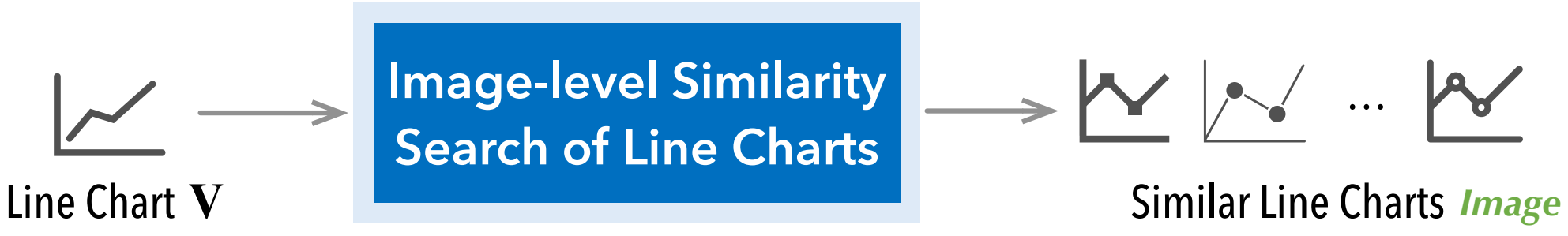
- Similarity measure based on data distribution, simple yet effective

Cons:

- Fail in capturing the perceptual similarity
- Fail in searching similar line charts when underlying data are unavailable

[1] Miro Mannino at al. **Qetch**: Time Series Querying with Expressive Sketches. SIGMOD 2018.
[2] Tarique Siddiqui et al. Fast-Forwarding to Desired Visualizations with **Zenvisage**. CIDR 2017.

Image-level Similarity Search of Line Charts



$$\mathbf{dist}(\mathbf{L}_i, \mathbf{L}_j) = \mathbf{dist}(\mathcal{F}(\mathbf{V}_i), \mathcal{F}(\mathbf{V}_j))$$

Embedding vector

Pros:

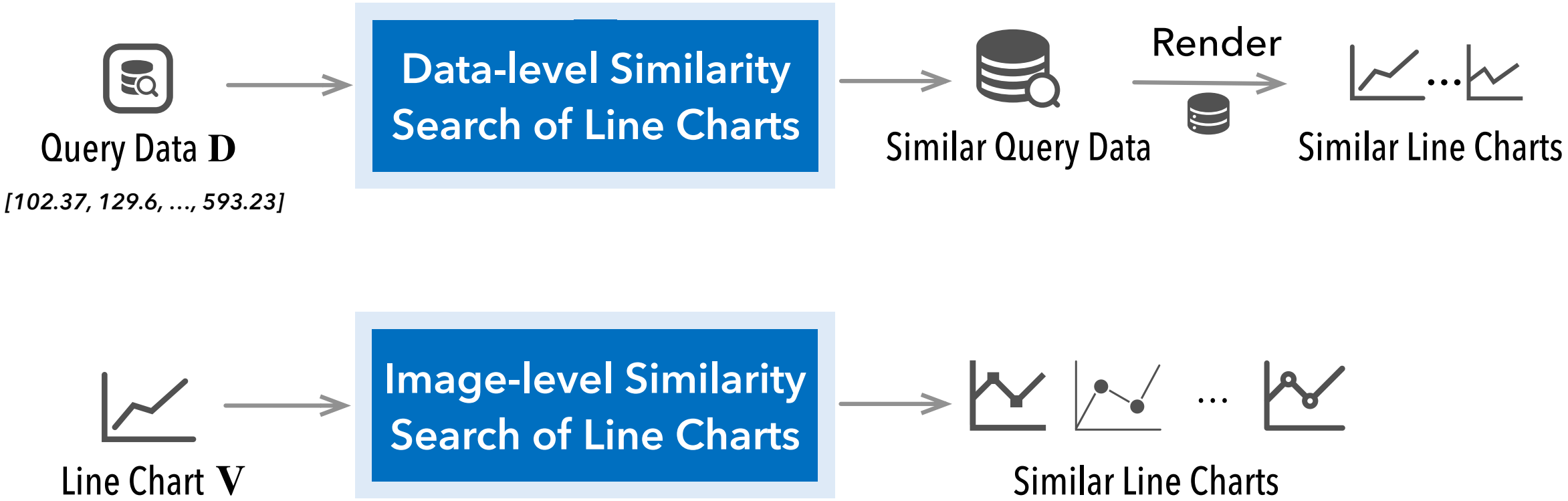
- Can capture perceptual similarity
- Work when underlying data are unavailable

Cons:

- Fully rely on low-level image features
- Hard to accurately reflect the data distribution

[4] Haotian Li, Yong Wang, Aoyu Wu, Huan Wei, and Huamin Qu. Structure-Aware Visualization Retrieval. CHI Conference 2022.

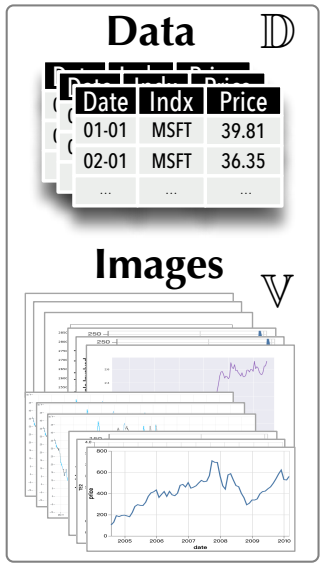
Research Question



Can we make the best of both worlds?

Learned Data-aware Image Representations of Line Charts

Training



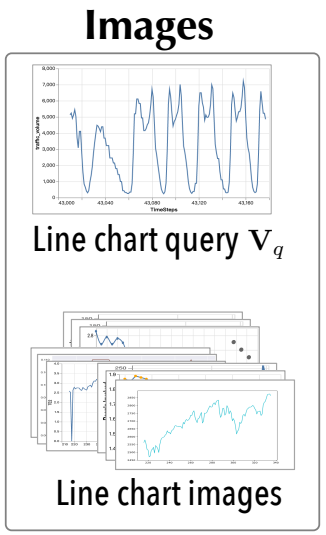
LineNet

Learned Data-aware Image Representations

LineNet aims to capture data-aware and image-level similarity more accurately

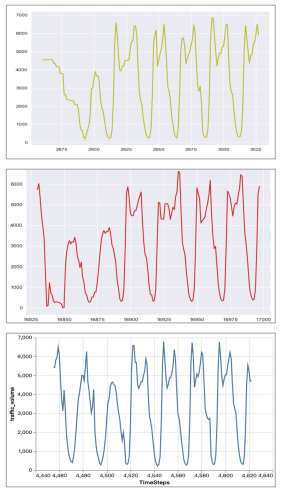
---

Deployment



LineNet

Similarity search by line chart images



Results

Learned Data-aware Image Representations of Line Charts

Problem Definition

- **Input for Training**: a set $\mathbb{L}$ of line charts $\{L_1, L_2, ..., L_n\}$, and the associated underlying data $\mathbb{D} = \{D_1, D_2, ..., D_n\}$ and rendered images $\mathbb{V} = \{V_1, V_2, ..., V_n\}$
- **Goal**:
  - It aims to **learn a** neural-network based **embedding function** $\mathcal{F}(\cdot)$ to map every line chart image $V_i \in \mathbb{V}$ to a low-dimensional embedding vector E_i ($E_i = \mathcal{F}(V_i)$)
  - For any pair of line chart images V_i and V_j, **dist**$(\mathcal{F}(V_i), \mathcal{F}(V_j))$ is close to **dist**(D_i, D_j)

$$\arg\min_{\theta} |\mathbf{dist}(\mathcal{F}(\mathbf{V}_i), \mathcal{F}(\mathbf{V}_j)) - \mathbf{dist}(\mathbf{D}_i, \mathbf{D}_j)|$$

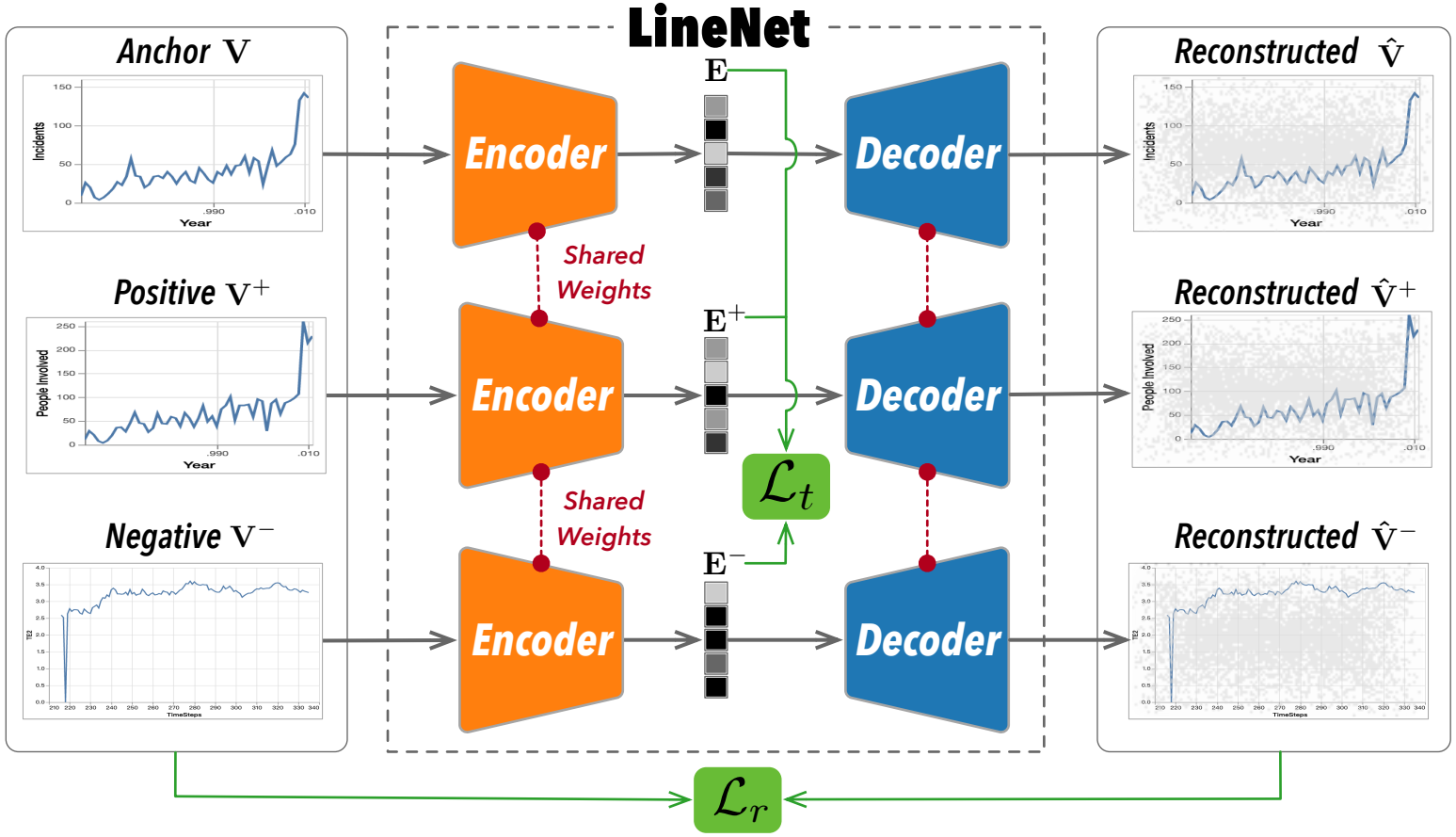
where θ denotes the neural network parameters, i.e., the $\mathcal{F}(\cdot)$ is parameterized by θ.

How to Learn Data-aware Image Representations?

- **Challenge** 1: Given the rendered images V and the underlying data D of line charts, how can we design a model that can **simultaneously learn better representations by capturing both** inherent **image-level** characteristics and the **data distribution** of **line chart images**?

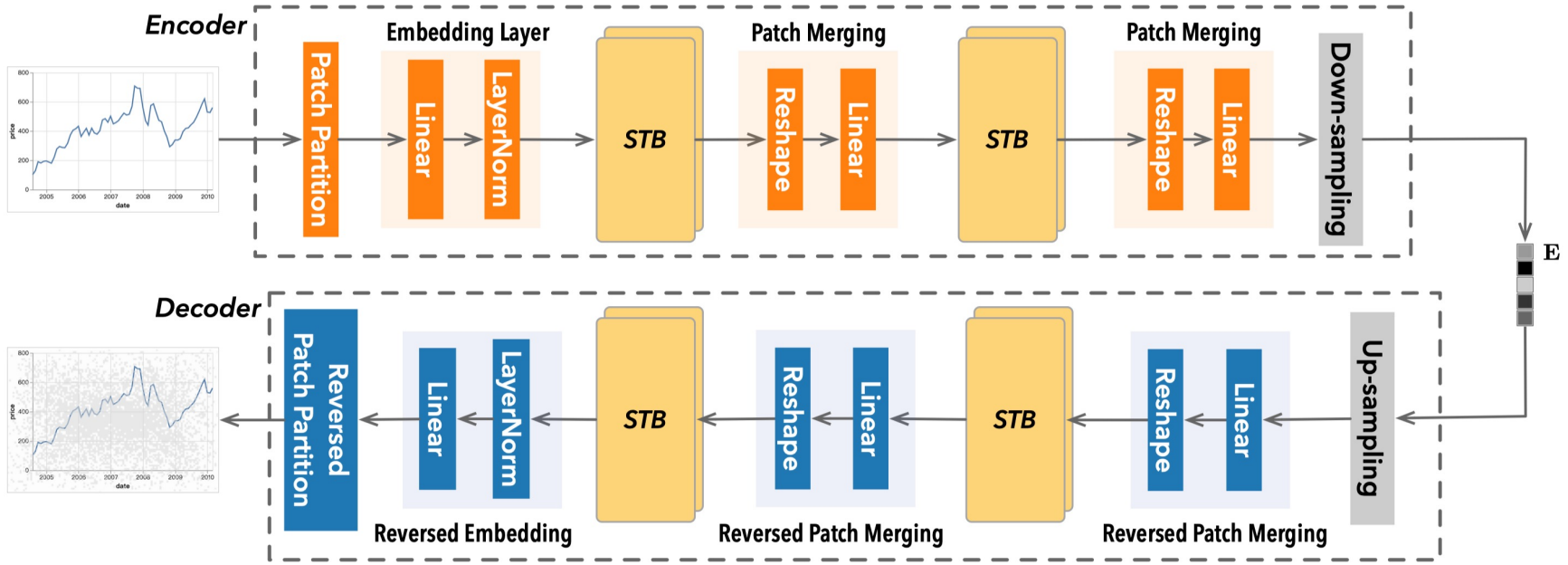
Details of LineNet

- **LineNet**: A Triplet Autoencoder architecture
- **Goal**: Learn data and visualization similarity simultaneously
- **Input**: A Triplet of Line-Chart Images

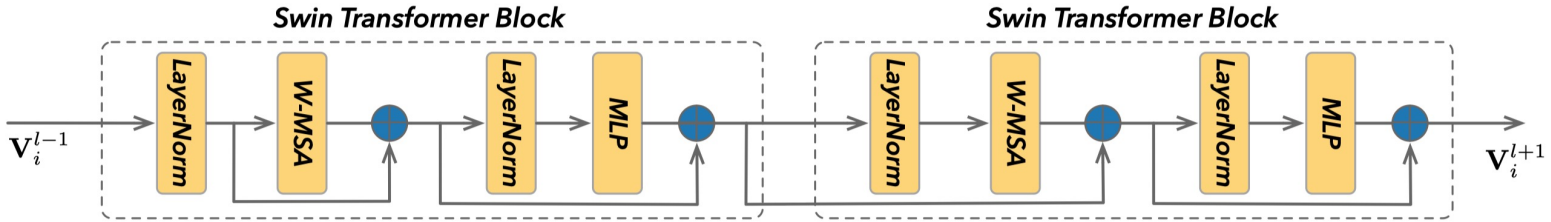


Details of LineNet

- **LineNet**: A **Vision Transformer**-based Triplet Autoencoder architecture



(a) Design Details of LineNet



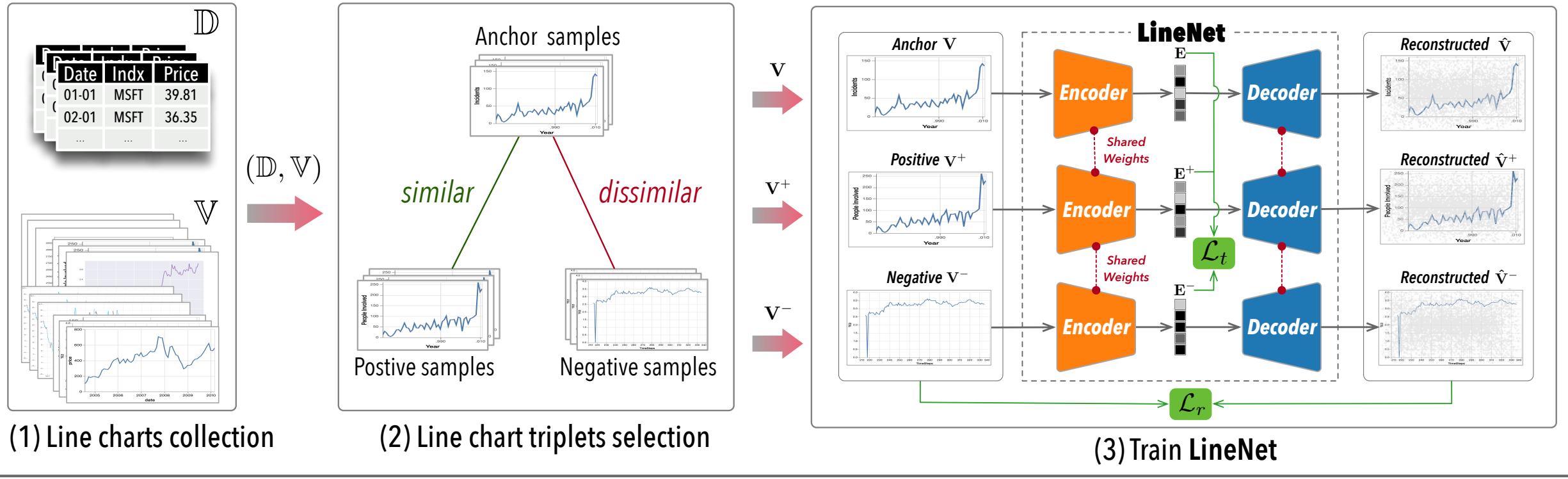
(b) Swin Transformer Blocks (STB)

Fig. 4. The architecture of LineNet.

[1] Ze Liu et al. Swin transformer: Hierarchical vision transformer using shifted windows. IEEE/CVF ICCV 2021.

Overview of LineNet: Training Step

Learned Data-aware Image Representations of Line Charts

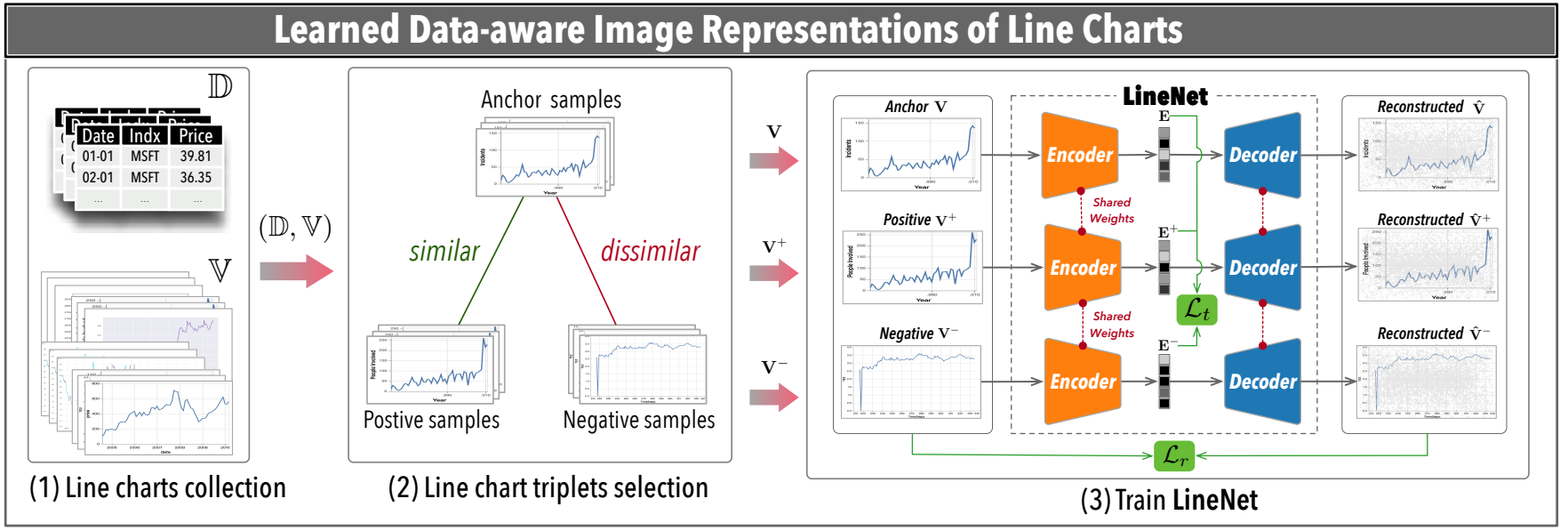


(1) Line charts collection

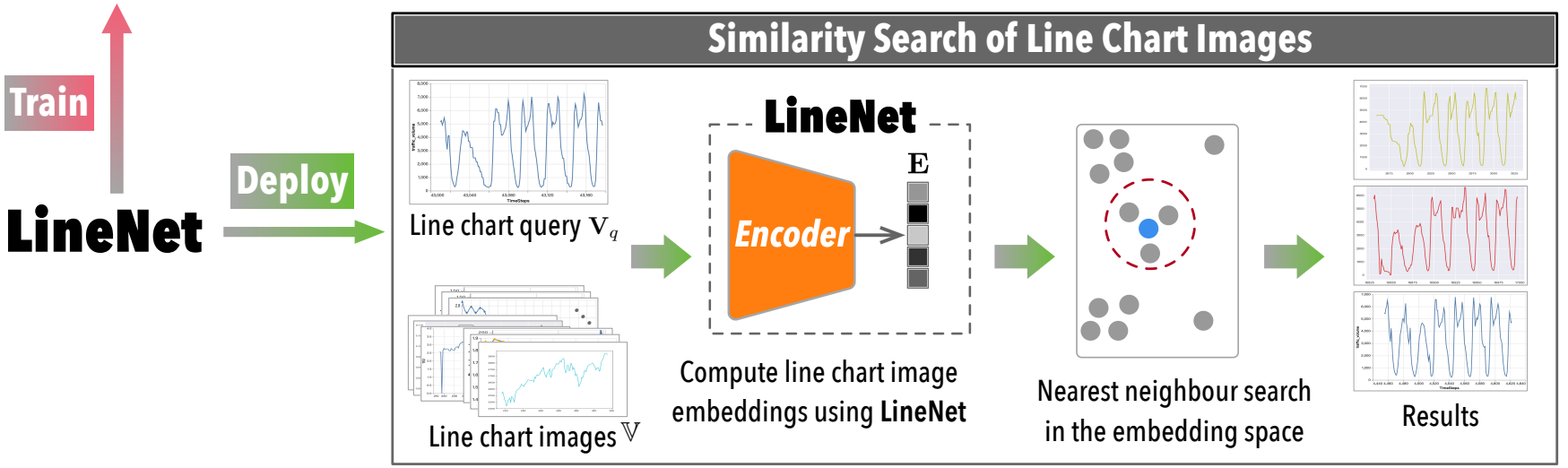
(2) Line chart triplets selection

(3) Train **LineNet**

LineNet Overview



(a) Inputs for model training: line chart images $\mathbb{V}$ and underlying data $\mathbb{D}$



(b) Inputs for similarity search: line chart images $\mathbb{V}$

How to train the LineNet ?

- Challenge 2: ***There is currently no off-the-peg corpus*** containing a large number of line charts and associated underlying data and rendered images **to drive the model training** and **evaluate the performance**.

- Basic idea:
  - Develop a Web Crawler to harvest line chart visualizations

    Scale ↑ Cost ↓ Quality ↓ (e.g., missing metadata of line charts)

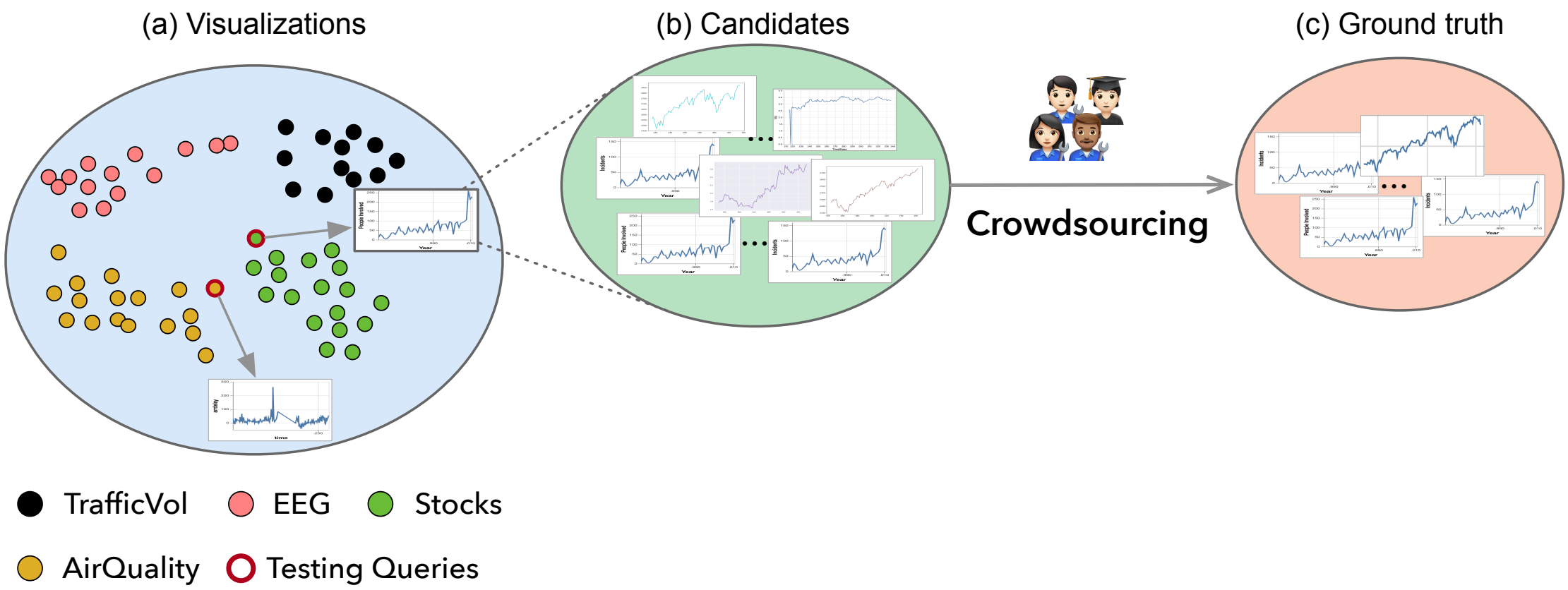
  - Collect the source datasets and curate a line chart corpus

    Scale ↑ Cost ↑ Quality ↑

Line Chart Corpus Construction

Datasets Collection → **Line Chart Generation** → **Similarity Labeling**

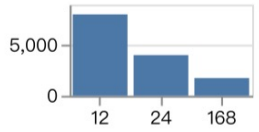
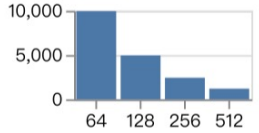
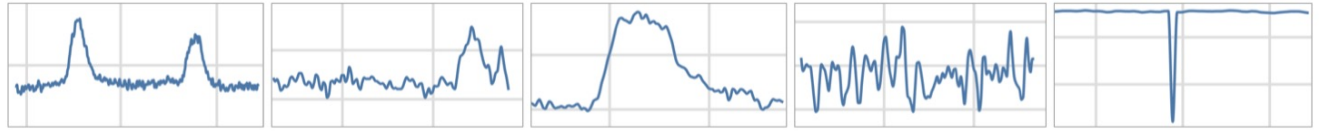
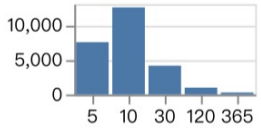
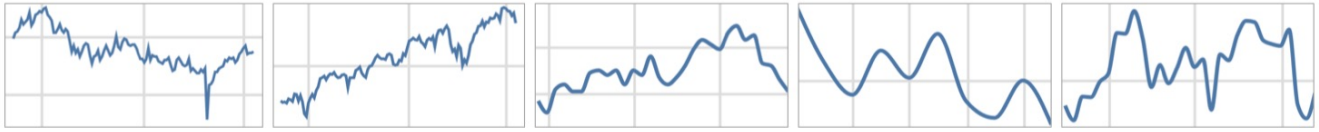
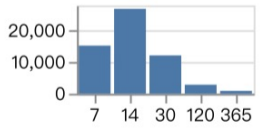
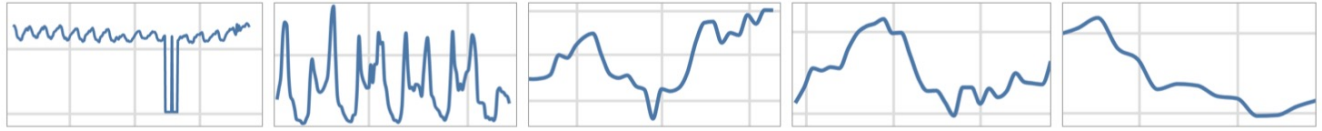
UCI Repository | **Automatic Visualization Technique** | **Crowdsourcing**



LineBench: a large-scale line chart visualizations corpus

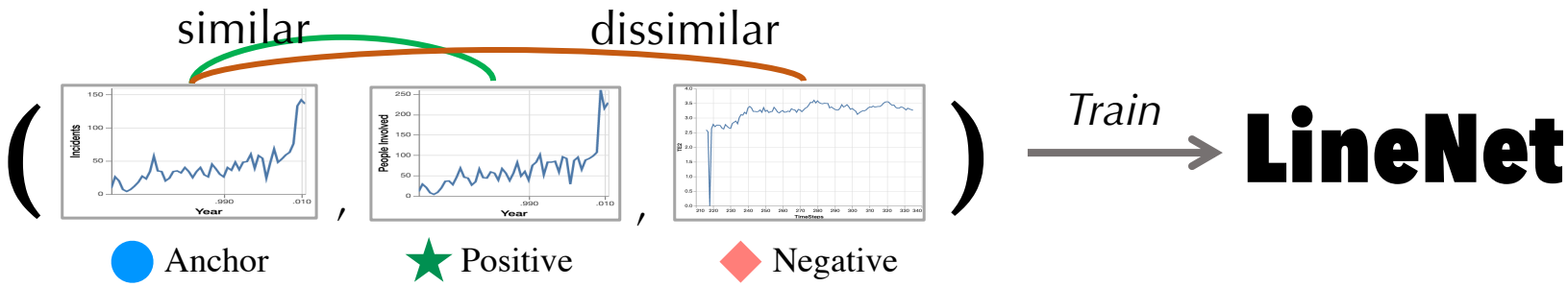
- **LineBench**
  - **115K line charts with data and images** for **four domains**: Traffic, EEG, Stocks, Air Quality
  - **400** line chart similarity search **queries** with ground truth

Table 2. Statistics of LineBench. The column–Dist. of Len. shows the distribution of lengths of line charts.

Datasets	#-Rows	#-Cols	#-Charts	Dist. of Len.	Sample Patterns of Line Charts
TrafficVol	48,205	10	13,798		
EEG	14,981	15	18,551		
Stocks	1,985	85	25,460		
AirQuality	9,358	15	58,127		

How to make the learning process more effective ?

- Challenge 3: How can we **judiciously select a set of representative training samples** to train the model **effectively and efficiently**?



Similarity can be measured by their underlying data, i.e., **dist**$(D_{Anchor}, D_{positive})$

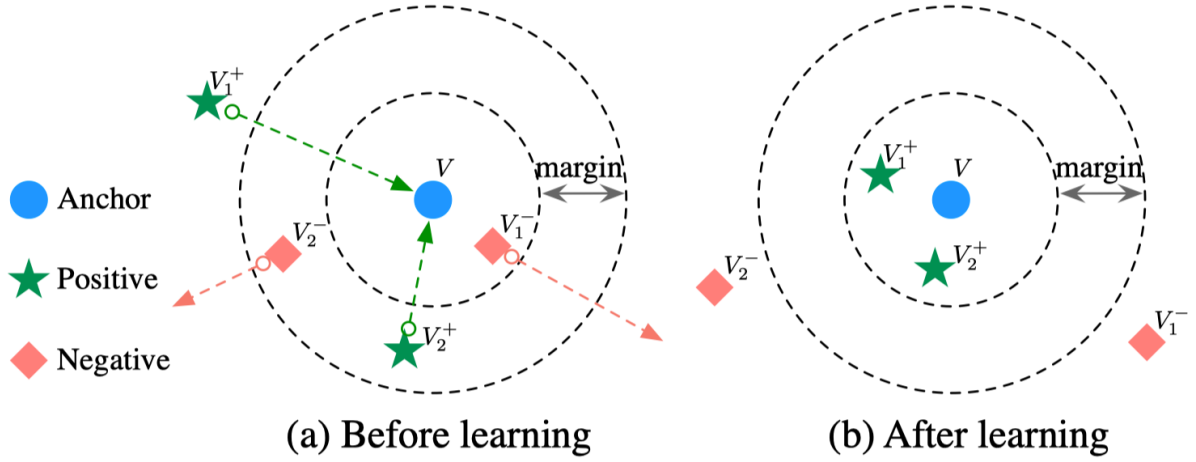
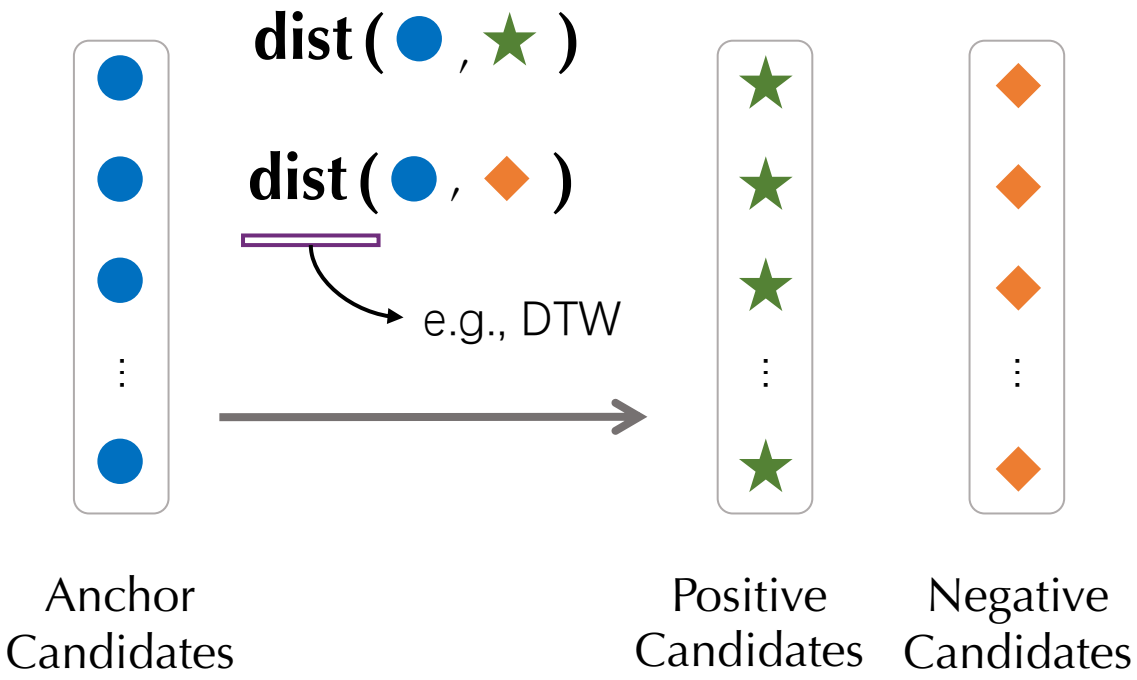


Fig. 3. An example of triplet selection for learning.

Semi-hard Triplet Selection Strategy

Step 1: Compute data-level similarity, as a proxy for the "image-level similarity"

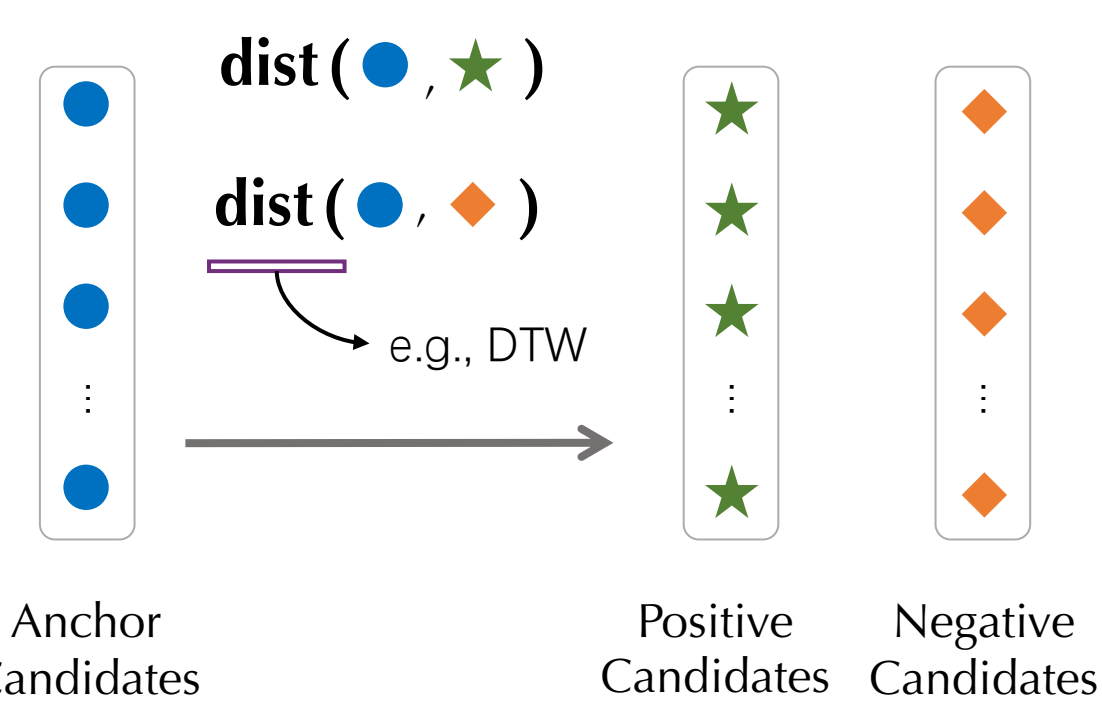
$$R(\mathbf{V}_i, \mathbf{V}_j) = 1 - \mathbf{dist}(\mathbf{D}_i, \mathbf{D}_j)$$

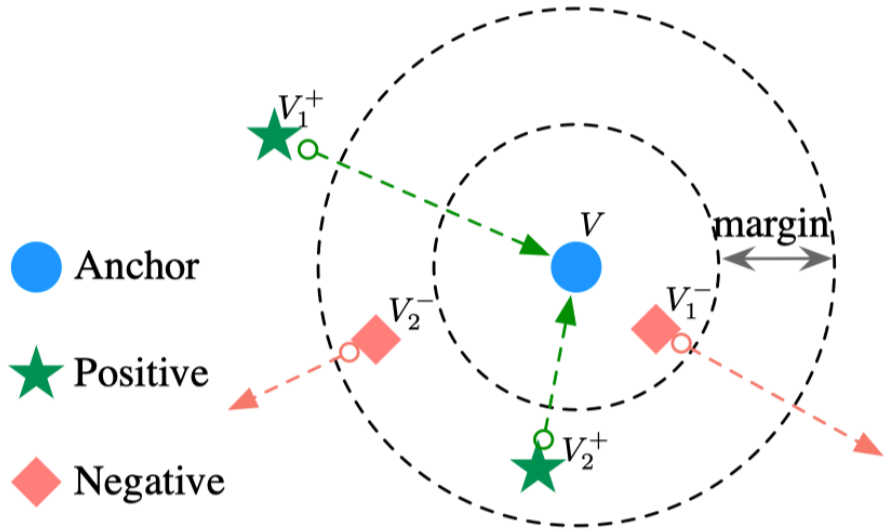


Semi-hard Triplet Selection Strategy

Step 2: Identify the informative positive and negative samples in the embedding space

$$R(\mathbf{V}_i, \mathbf{V}_j) = 1 - \mathbf{dist}(\mathbf{D}_i, \mathbf{D}_j)$$





triplet $(\mathbf{V}, \mathbf{V}_1^+, \mathbf{V}_1^-)$

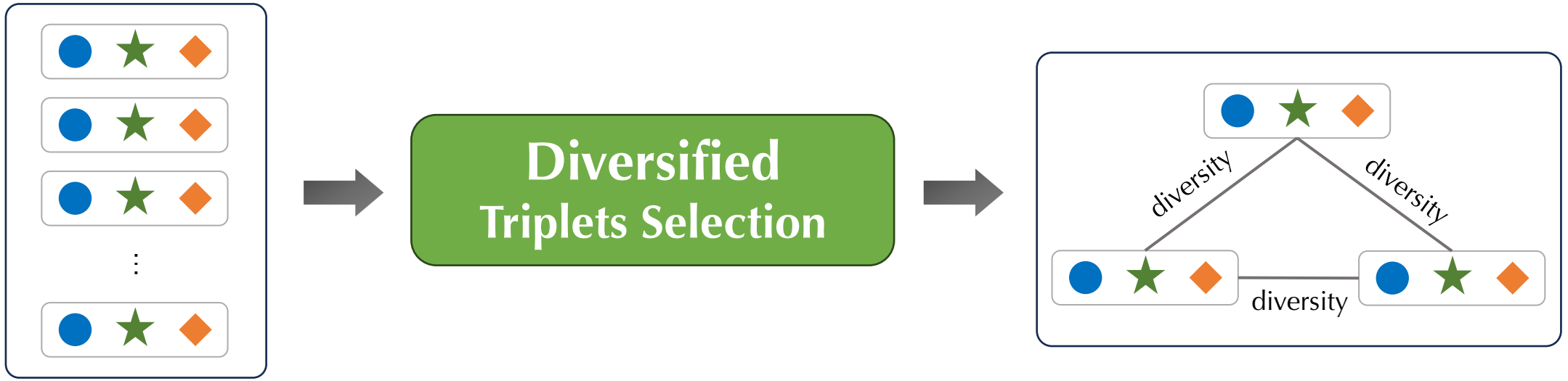
How to make the learning process more effective ?

- We employ the semi-hard triplets selection strategy[1] to generate a set of training data.
- However, ***this method may not guarantee that the selected triplets are discriminative and meaningful, which may lead to slow convergence and lower performance***.

- To alleviate this problem, we exploit a **diversified triplets selection** strategy to **carefully pick a small set of representative** and discriminative **triplets** from current mini-batch of training data.

[1] Florian Schroff et al. FaceNet: A unified embedding for face recognition and clustering. CVPR 2015.

Diversified Triplets Selection Algorithm



- The diversified triplets selection problem can be reduced to the **NP-hard** max-sum dispersion problem

Evaluation

- Can LineNet retrieve similar line charts?
- How does LineNet perform compared to SOTA?

Table 3. Experimental datasets – LineBench

Datasets	Training		Testing	
	Training	Validation	#-Queries	Repository
TrafficVol	6,899	1,380	100	5,519
EEG	9,274	1,855	100	7,421
Stocks	12,730	2,546	100	10,184
AirQuality	29,063	5,813	100	23,251

Methods

- Our methods:
  - LineNet
  - LineNet+ (training with the diversified triplet selection strategy)

- Data-level similarity
  - Zenvisage (CIDR 2017)
  - Qetch (SIGMOD 2018)
  - PEAX (EuroVIS 2020 Best Paper)
- Image-level similarity
  - V-CNN (CHI 2022 Best Paper Candidate)
  - Resnet-based baseline

Quantitative Results

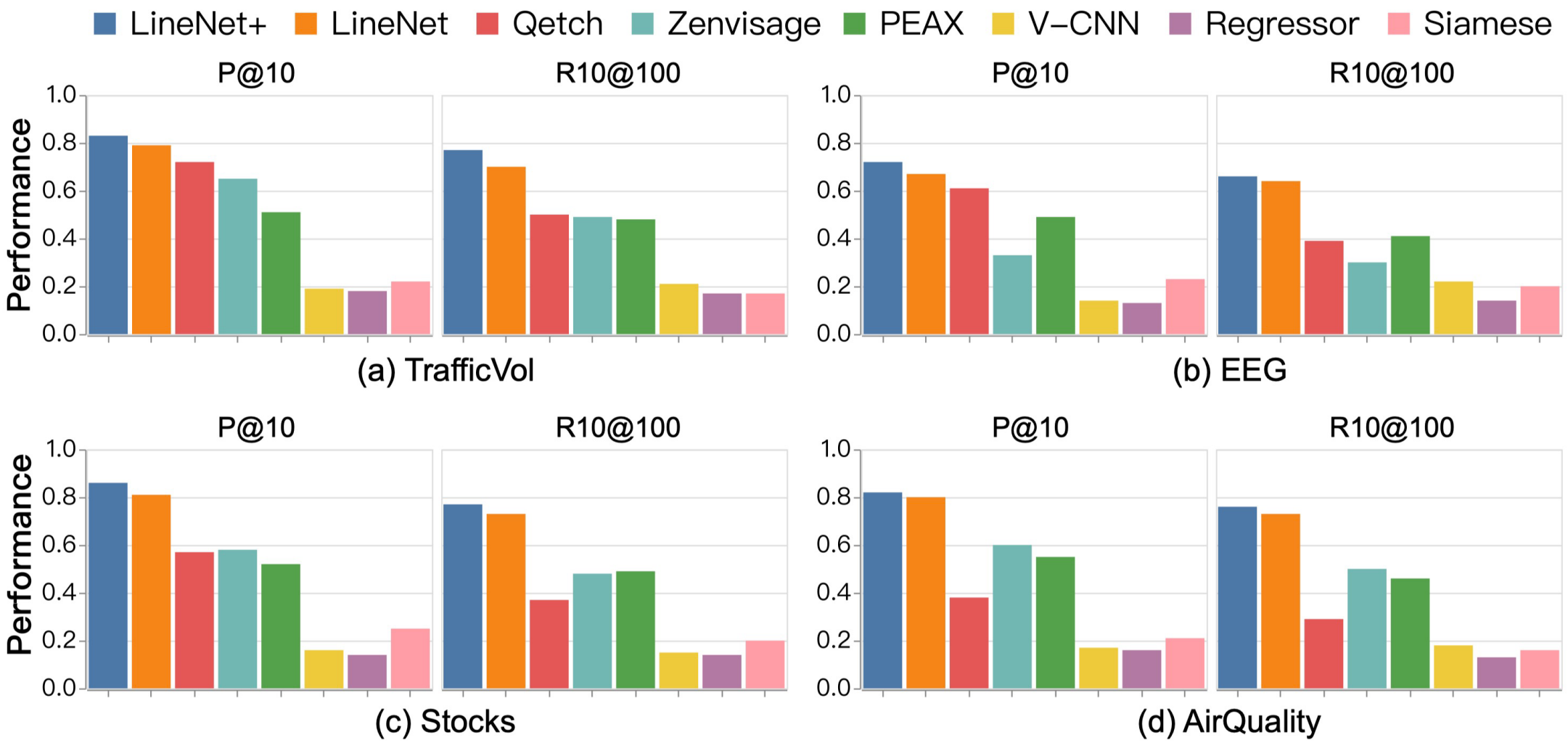
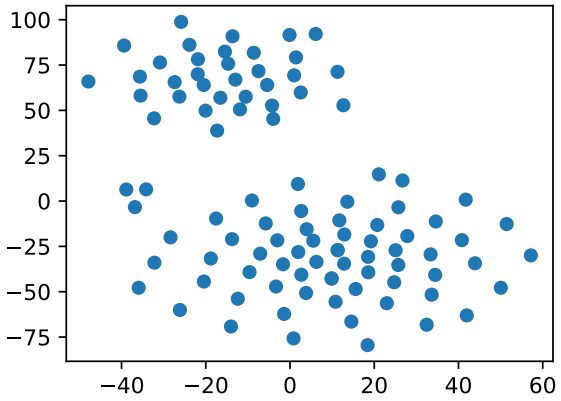
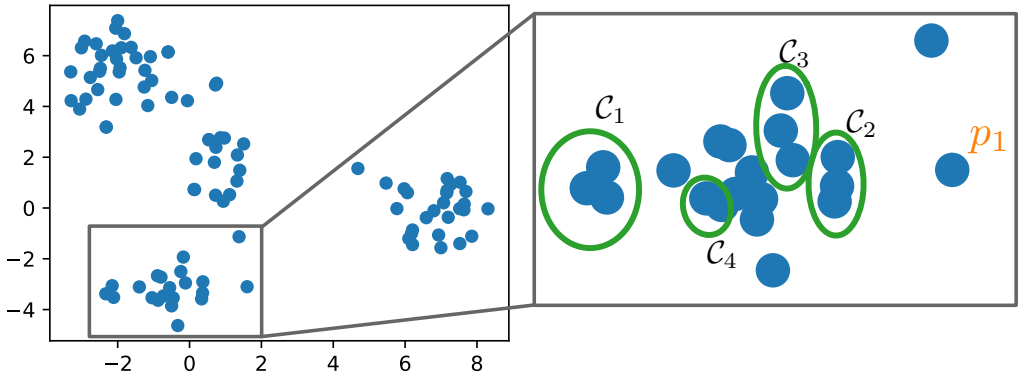


Fig. 5. Comparison with the state of the art.

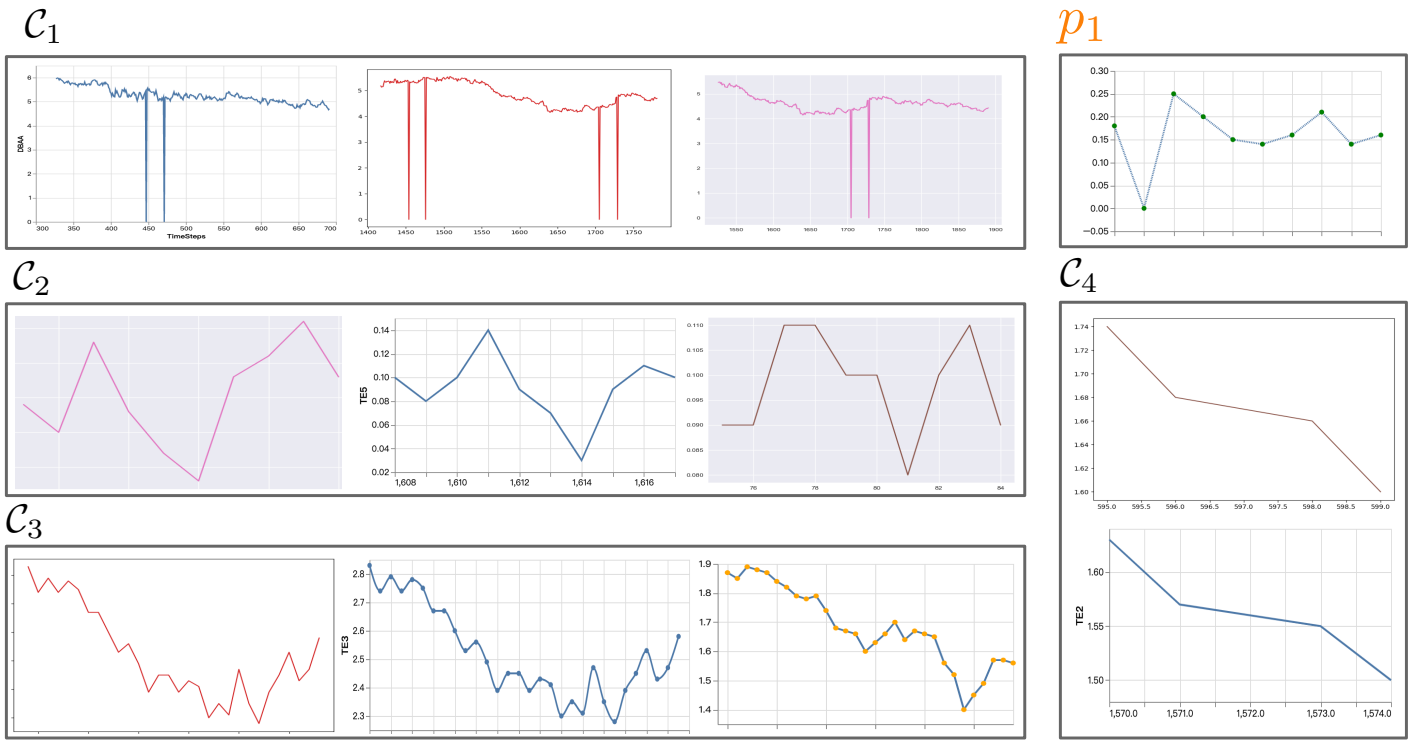
Learned Representations of LineNet



(a) Initialized embedding space

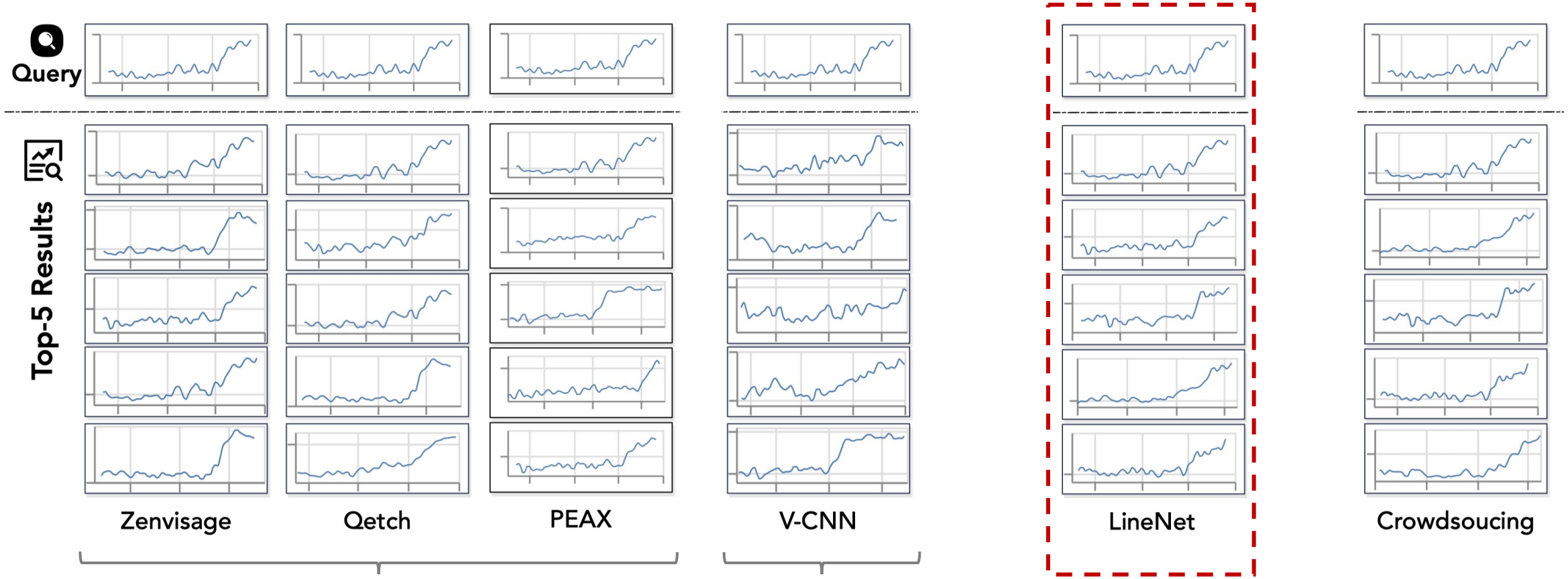


(b) Learned embedding space



(c) Line chart visualizations of the corresponding clusters (point)

Case study on LineBench



Data-level similarity

Image-level similarity

Learned data-aware image-level similarity

Ground Truth

Case study on real-world application scenarios

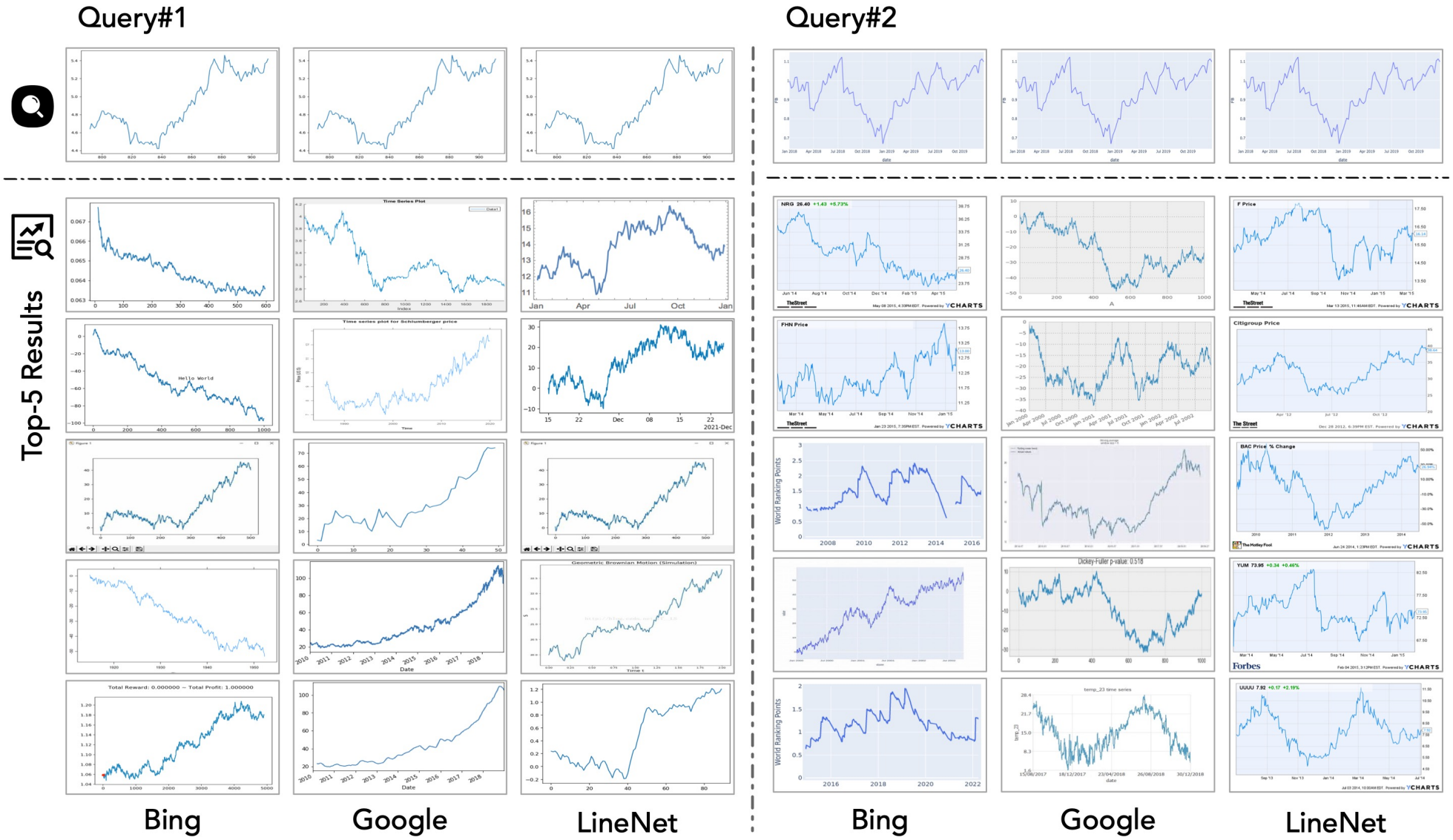
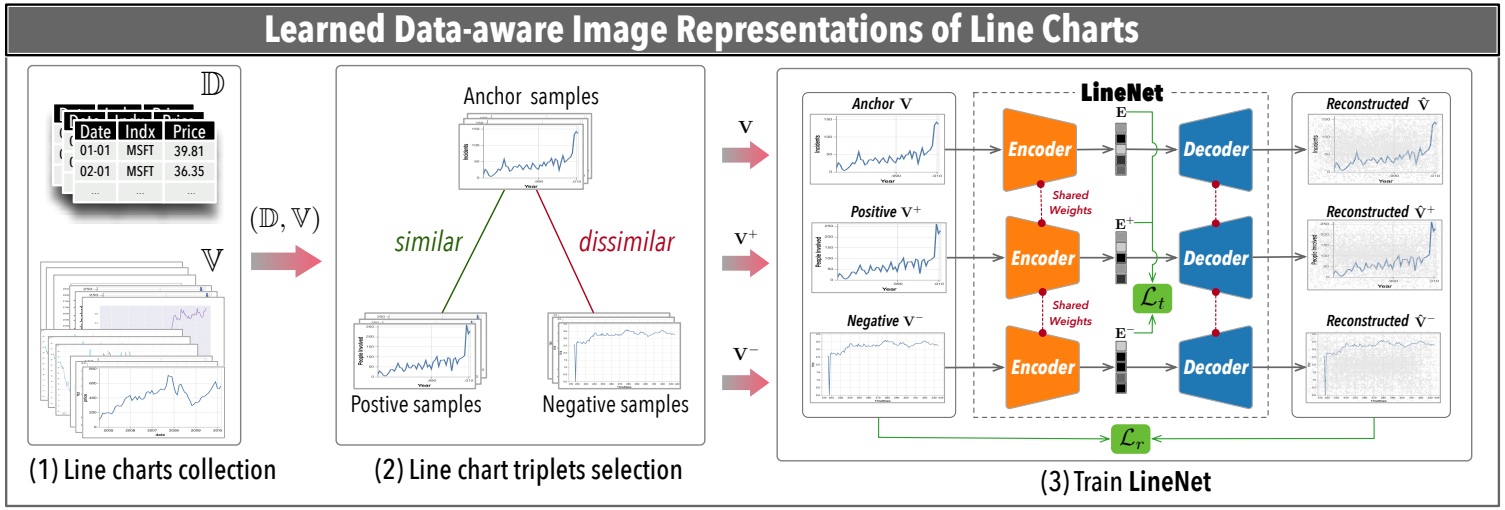


Fig. 13. Case study on real cases.

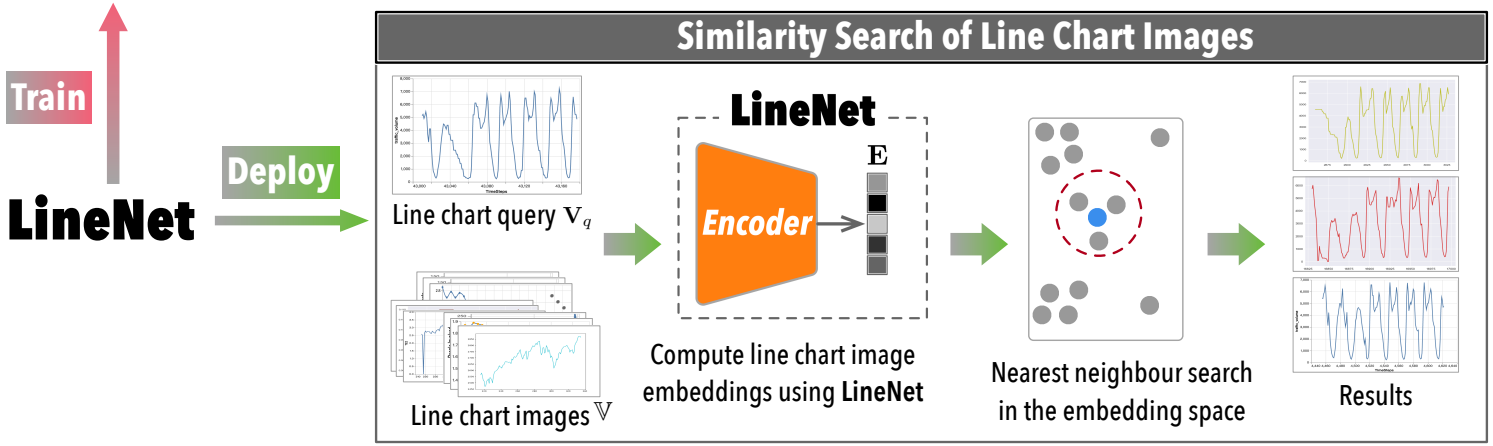
Conclusion

- **Problem Statement:**
  - We formally define the problem of **learned data-aware image representations** of line charts for similarity search.
- **LineNet:**
  - We propose **LineNet**, a novel **Vision Transformer-based Triplet Autoencoder model** to learn data-aware image representations of line charts by leveraging both the underlying data and rendered images of line charts.
  - We further design a **diversified training samples selection algorithm** to make the learning process more effective.
- **LineBench:**
  - We develop **a large-scale line chart corpus**, namely **LineBench**, which has more than **115K** line charts and the metadata from four real-world datasets.

Thanks!



(a) Inputs for model training: line chart images $\mathbb{V}$ and underlying data $\mathbb{D}$



(b) Inputs for similarity search: line chart images $\mathbb{V}$

Code & Datasets

Code & Datasets: https://github.com/Thanksyy/LineNet-and-LineBench-SIGMOD2023

Future Work

- **Towards Universal Visualization Representations**
  - ➢ Develop large models to "understand" visualizations
  - ➢ Enable semantics search (similarity search), visualization recommendation

Sourced Data (relational data)

Origin	Year	Cylinders	…
USA	1970	8	…
USA	1970	8	…
Europe	1970	8	…
…	…	…	…

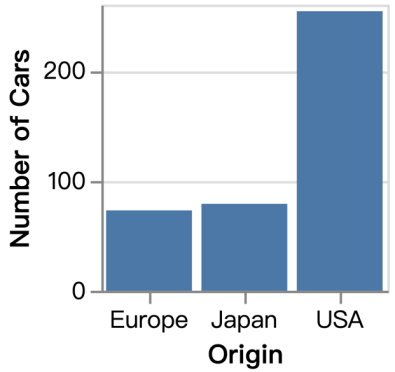
Transformed Data (relational data)

Origin	COUNT(Origin)
USA	254
Europe	73
Japan	39

Visual Format Data (relational data)

Elements	X	Width	Y	…
Europe	7.5	100	307.5	…
Japan	157.5	100	…	…
USA	307.5	100	…	…

Visualization (graphic data)



Visualization Query (Program data)

```
{
"$schema": "https://vega.github.io/schema/vega-lite/v5.json",
"width": "container",
"height": 130,
"width": 120,
"data": {"url": "data/cars.json"},
"mark": "bar",
"encoding": {
"x": {"field": "Origin"},
"y": {"aggregate": "count", "title": "Number of Cars"}
}
}
```

=

SQL-like Visualization Query (Program data)

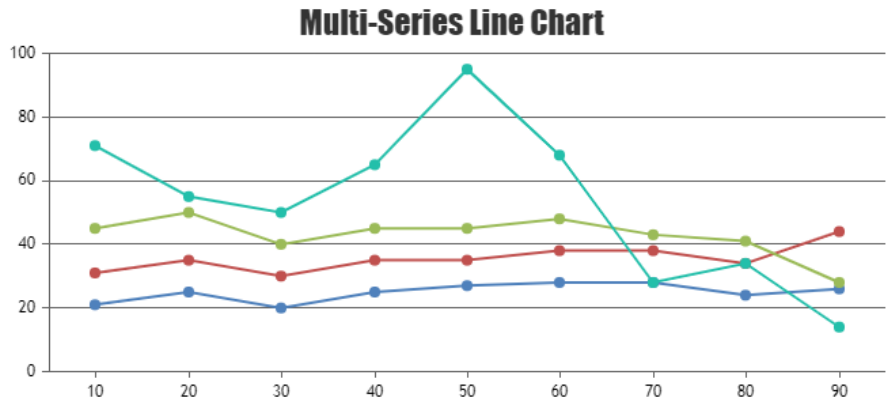
mark bar
data Cars
encoding x Origin y aggregate count Origin
transform group x sort y desc

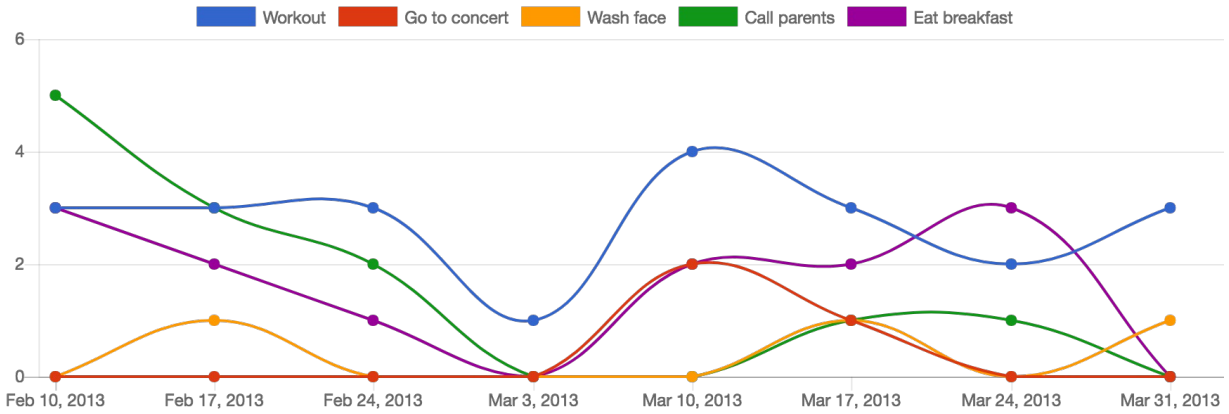
Generalize to Other Visualization Types

- Although we take the line charts as the exemplar to investigate the problem of learned data-aware visualization representations for similarity search, our framework can be extended to support other visualization types such as bar charts, pie charts, and scatter plots. The key to this extension is measuring data-level similarity between visualization charts based on their underlying data.

Similarity Search of Multi-series Line Charts

- Although the multi-series line chart is popular in the data visualization community, how to measure the similarity between two multi-series line charts is still a challenging task. The reason is that it is difficult for us to weigh the contribution of the similarity between each series to the overall similarity of the multi-series line charts. Thus, we leave this interesting topic for future work.





Details of LineNet

- Learn Inherent Image Similarity.

Reconstruction Loss: $\mathcal{L}_r = \frac{1}{n}\sum_{i=1}^{n}(||\mathbf{V}_i - \hat{\mathbf{V}}_i||_2^2 + ||\mathbf{V}_i^+ - \hat{\mathbf{V}}_i^+||_2^2 + ||\mathbf{V}_i^- - \hat{\mathbf{V}}_i^-||_2^2)$

- Capture Data Similarity.

Triplet Loss: $\mathcal{L}_t = \frac{1}{n}\sum_{i=1}^{n} max\{0,\ \beta + ||\mathbf{E}_i - \mathbf{E}_i^+||_2^2 - ||\mathbf{E}_i - \mathbf{E}_i^-||_2^2\}$

- Overall Objective and Optimization.

$$\mathcal{L} = (1-\alpha)\mathcal{L}_r + \alpha\mathcal{L}_t$$