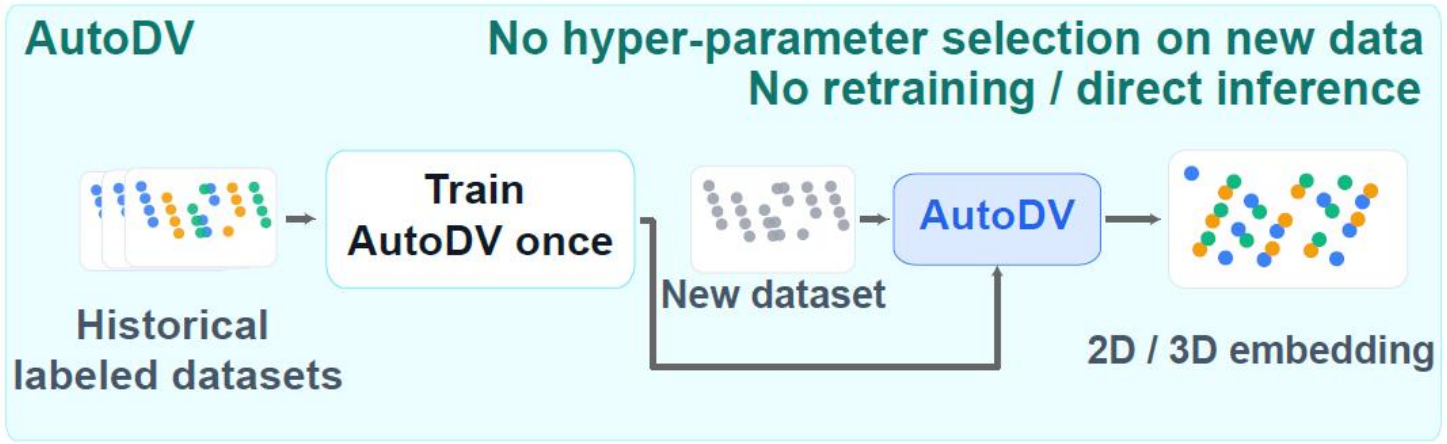
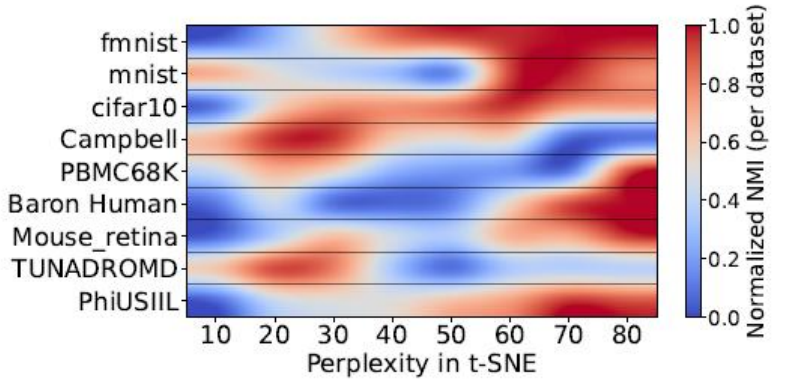
AutoDV: An End-to-End Deep Learning Model for High-Dimensional Data Visualization

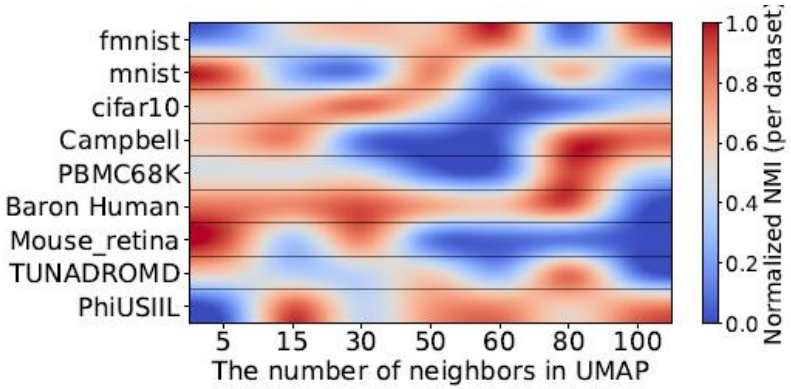
Wei Dai[1,2], Jicong Fan[2]*
[1]Zhejiang University of Technology, Hangzhou, China
[2]The Chinese University of Hong Kong, Shenzhen, China
weidai@zjut.edu.cn, fanjicong@cuhk.edu.cn

A single trained model visualizes unseen datasets directly — no hyper-parameter tuning, no retraining.

Motivation







Visualization quality is sensitive to perplexity / n_neighbors.

Problem Formulation

Definition 1 (AutoDV) *Suppose we have L labeled historical datasets $(\boldsymbol{X}_1, \boldsymbol{y}_1)$, $(\boldsymbol{X}_2, \boldsymbol{y}_2)$, ..., $(\boldsymbol{X}_L, \boldsymbol{y}_L)$, where $\boldsymbol{X}_i \in \mathbb{R}^{N_i \times d_i}$ is the data matrix with N_i samples and d_i features, and $\boldsymbol{y}_i \in \mathbb{R}^{N_i}$ are the labels, $i = 1, \ldots, L$. Let $\mathcal{A}_\theta$ be an effective data visualization algorithm, with hyper-parameters θ. For each $\boldsymbol{X}_i$, an optimal low-dimensional embedding $\boldsymbol{Z}_i^* \in \mathbb{R}^{N_i \times d'}$ is obtained by $\boldsymbol{Z}_i^* = \mathcal{A}_{\theta_i^*}(\boldsymbol{X}_i)$, where θ_i^* denotes the optimal hyper-parameters selected using $\boldsymbol{y}_i$ with respect to some metric $\mathcal{M}$. The goal of AutoDV is to train an end-to-end model $f_\phi : \mathbb{R}^{N_i \times d_i} \to \mathbb{R}^{N_i \times d'}$ from $\{(\boldsymbol{X}_i, \boldsymbol{Z}_i^*)\}_{i=1}^L$, such that, for any unseen dataset $\boldsymbol{X}_{new} \in \mathbb{R}^{N_{i'} \times d_{i'}}$, with $\boldsymbol{Z}_{new}^* = \mathcal{A}_{\theta_{i'}^*}(\boldsymbol{X}_{new})$, it holds that*

$$dist(f_\phi(\boldsymbol{X}_{new}), \boldsymbol{Z}_{new}^*) \leq \varepsilon \tag{1}$$

where $dist(\cdot, \cdot)$ denotes some distance metric and $\varepsilon > 0$ is a small constant.

C1 Designing the end-to-end model f_ϕ is non-trivial, especially when the model is expected to handle input datasets with different sizes and dimensions, i,e., $N_i \neq N_j$ and $d_i \neq d_j$ for some i and j. **Cross-Dimension Challenge**

C2 There exist multiple optimal low-dimension embeddings $\boldsymbol{Z}_i^*$ for the same input $\boldsymbol{X}_i$ if we apply translation, rotation, and scaling to $\boldsymbol{Z}_i^*$, e.g., $\boldsymbol{Z}_i^*$ and $\boldsymbol{Z}_i^*\boldsymbol{Q}$ with any orthonormal matrix $\boldsymbol{Q}$ are equivalent in terms of visualization. This will result in a one-to-many problem and will be difficult to converge (Arridge et al., 2019). **Translation, Rotation, and Scaling Ambiguity**

Method

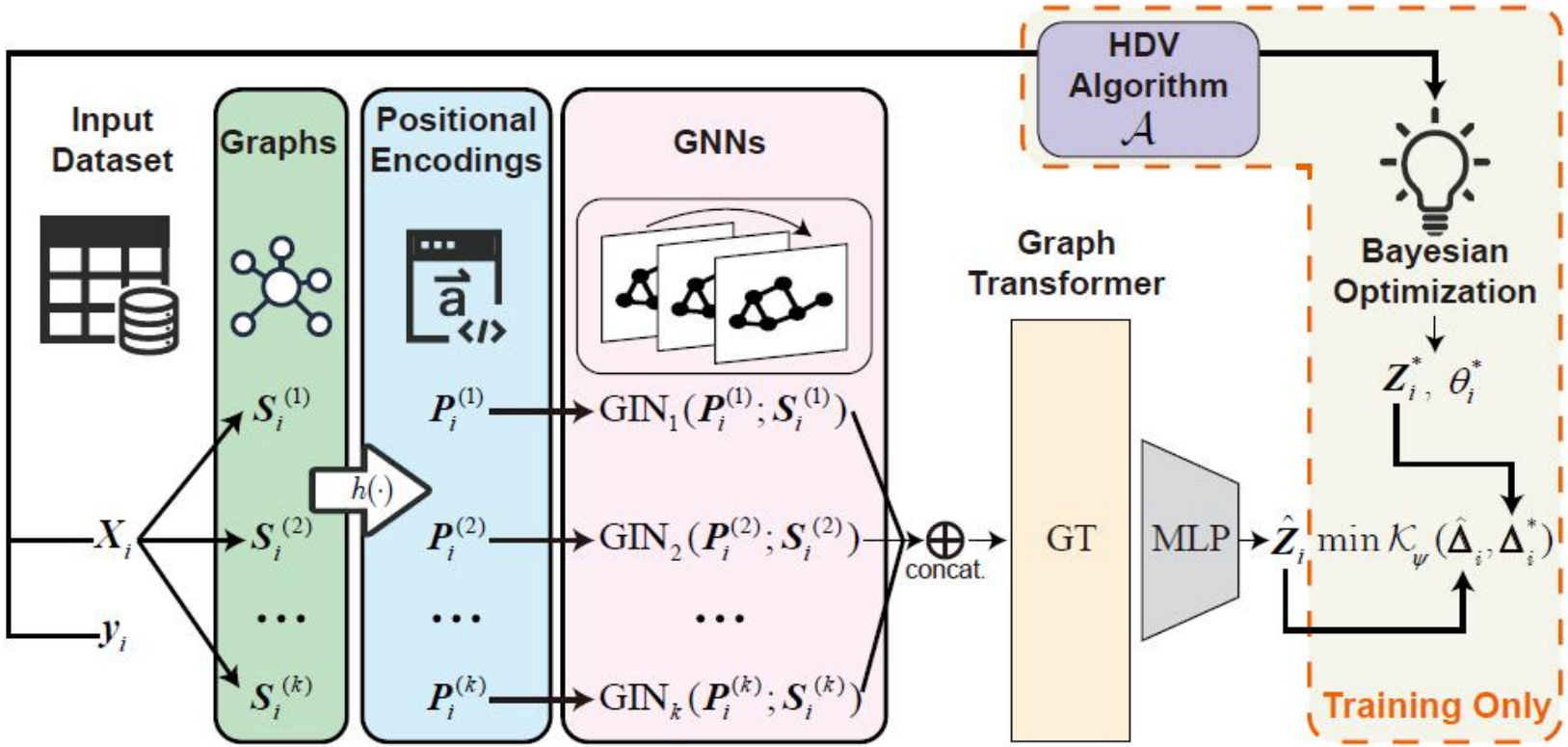


Figure 2: AutoDV framework. The input datasets will first be converted into k graphs. Then, positional encodings are extracted from the graph. Each graph is processed by an independent GIN network. The output node features of GINs are concatenated and sent to a graph transformer. Finally, an MLP head is responsible for outputting the low-dimensional embedding. Ground truth label y_i, HDV algorithm $\mathcal{A}$, and Bayesian optimization are only used at the training stage to produce Z_i^*.

Main Results

Table 2: Comparative data visualization results on image datasets. Average NMI, NMI precision, SC, and SC precision are reported on the test set. 500 subsets of MNIST and FMNIST are sampled for training. 50 subsets of CIFAR-10 are sampled for testing. The best performance on test sets is **bold**. Selected visualization results are plotted in Figure 3.

Methods	t-SNE*	UMAP*	D. t-SNE	D. UMAP	PCA	AE	p-UMAP	i-tSNE	i-UMAP	AutoDV-tSNE	AutoDV-UMAP
Test NMI	$77.04_{\pm 8.8}$	$80.45_{\pm 6.6}$	$71.71_{\pm 13.5}$	$79.72_{\pm 6.7}$	$15.97_{\pm 6.9}$	$6.08_{\pm 3.8}$	$15.58_{\pm 7.6}$	$5.38_{\pm 2.1}$	$3.63_{\pm 1.7}$	$68.70_{\pm 9.0}$	$\mathbf{73.28}_{\pm 7.6}$
Test NMI Prec.	100	100	-	-	-	-	$18.54_{\pm 5.4}$	$10.41_{\pm 25.4}$	$4.40_{\pm 2.1}$	$89.37_{\pm 7.8}$	$\mathbf{91.05}_{\pm 5.3}$
Test SC	$63.32_{\pm 9.5}$	$68.54_{\pm 6.6}$	$47.24_{\pm 9.9}$	$67.61_{\pm 6.79}$	$34.51_{\pm 2.5}$	$42.85_{\pm 8.7}$	$43.84_{\pm 8.5}$	$39.34_{\pm 5.2}$	$40.50_{\pm 6.5}$	$55.15_{\pm 6.6}$	$\mathbf{70.41}_{\pm 7.1}$
Test SC Prec.	100	100	-	-	-	-	$62.63_{\pm 7.2}$	$62.65_{\pm 8.6}$	$58.10_{\pm 11.1}$	$88.60_{\pm 14.4}$	$\mathbf{103.0}_{\pm 8.6}$

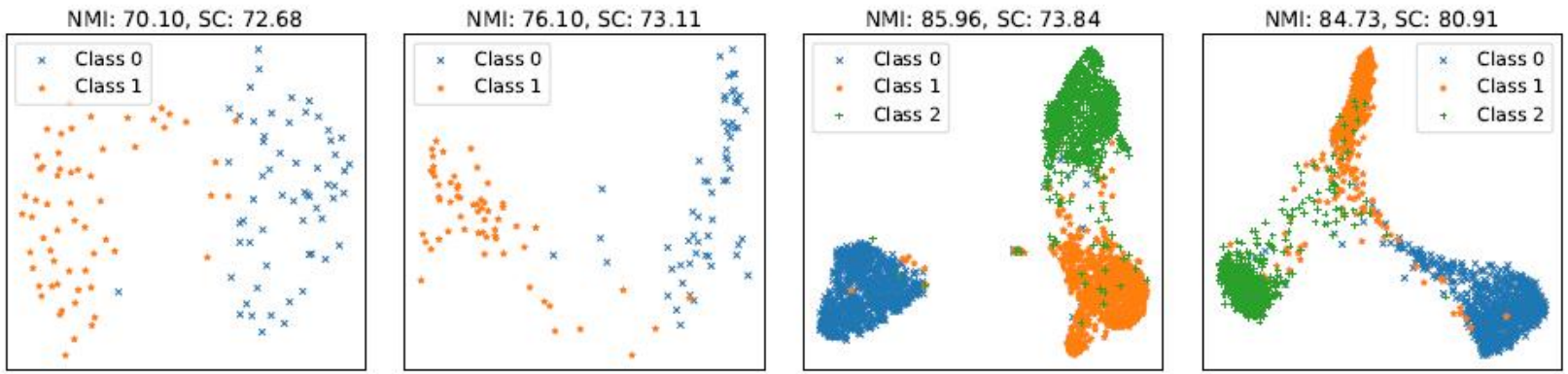
Table 3: Data visualization results on gene datasets. Average NMI, NMI precision, SC, and SC precision are reported on the test set. 2918 subsets of Baron Human, Campbel, and PBMC68k are sampled for training. 113 subsets of the Mouse Retina dataset are sampled for testing. The best performance on test sets is **bold**.

Methods	t-SNE*	UMAP*	D. t-SNE	D. UMAP	PCA	AE	p-UMAP	i-tSNE	i-UMAP	AutoDV-tSNE	AutoDV-UMAP
Test NMI	$32.73_{\pm 30.3}$	$28.85_{\pm 34.7}$	$15.67_{\pm 21.9}$	$22.45_{\pm 33.2}$	$9.27_{\pm 5.0}$	$5.21_{\pm 5.9}$	$24.78_{\pm 15.1}$	$4.80_{\pm 3.5}$	$13.98_{\pm 7.9}$	$\mathbf{33.22}_{\pm 28.7}$	$33.03_{\pm 25.0}$
Test NMI Prec.	100	100	-	-	-	-	$93.38_{\pm 72.9}$	$35.42_{\pm 29.7}$	$52.50_{\pm 61.1}$	$102.7_{\pm 36.7}$	$\mathbf{111.9}_{\pm 60.2}$
Test SC	$34.43_{\pm 4.6}$	$47.98_{\pm 20.1}$	$35.84_{\pm 13.3}$	$57.00_{\pm 24.9}$	$34.10_{\pm 7.6}$	$\mathbf{73.18}_{\pm 27.3}$	$65.58_{\pm 15.3}$	$54.70_{\pm 1.6}$	$61.51_{\pm 6.0}$	$47.11_{\pm 10.8}$	$47.98_{\pm 12.2}$
Test SC Prec.	100	100	-	-	-	-	$151.5_{\pm 53.8}$	$\mathbf{161.1}_{\pm 17.8}$	$160.3_{\pm 26.2}$	$110.4_{\pm 30.1}$	$112.5_{\pm 22.1}$

Main Results

Table 4: Data visualization results on UCI tabular datasets. Average NMI, NMI precision, SC, and SC precision are reported on the test set. 519 subsets are sampled from 26 datasets, of which 156 subsets are split for testing. The best performance on test sets is **bold**.

Methods	t-SNE*	UMAP*	D. t-SNE	D. UMAP	PCA	AE	p-UMAP	i-tSNE	i-UMAP	AutoDV-tSNE	AutoDV-UMAP
Test NMI	$30.92_{\pm 12.2}$	$24.80_{\pm 16.2}$	$23.53_{\pm 11.2}$	$15.13_{\pm 12.1}$	$10.96_{\pm 11.56}$	$5.17_{\pm 11.0}$	$14.11_{\pm 11.8}$	$8.00_{\pm 10.6}$	$10.56_{\pm 12.6}$	$33.45_{\pm 21.6}$	$\mathbf{35.15}_{\pm 34.5}$
Test NMI Prec.	100	100	-	-	-	-	$57.08_{\pm 30.8}$	$27.96_{\pm 32.2}$	$50.89_{\pm 51.7}$	$121.3_{\pm 40.3}$	$\mathbf{129.0}_{\pm 93.2}$
Test SC	$44.42_{\pm 10.0}$	$64.46_{\pm 15.9}$	$43.87_{\pm 10.6}$	$61.81_{\pm 19.1}$	$63.87_{\pm 25.46}$	$75.4_{\pm 6.6}$	$75.26_{\pm 9.7}$	$\mathbf{77.11}_{\pm 16.9}$	$76.68_{\pm 16.5}$	$48.25_{\pm 10.7}$	$64.28_{\pm 9.7}$
Test SC Prec.	100	100	-	-	-	-	$171.2_{\pm 22.0}$	$178.1_{\pm 44.2}$	$\mathbf{179.7}_{\pm 40.9}$	$110.0_{\pm 12.5}$	$106.4_{\pm 32.3}$



(a) UMAP* on CIFAR10 with 120 points. (b) AutoDV-UMAP on CIFAR10 with 120 points. (c) UMAP* on CIFAR10 with 2759 points. (d) AutoDV-UMAP on CIFAR10 with 2759 points.

Figure 3: (a) and (b) are results of UMAP* and AutoDV-UMAP on the same small test subset of CIFAR10, where AutoDV is better. (c) and (d) are results of UMAP* and AutoDV-UMAP on the same large test subset of CIFAR10.

Transferability, Efficiency, and Theory

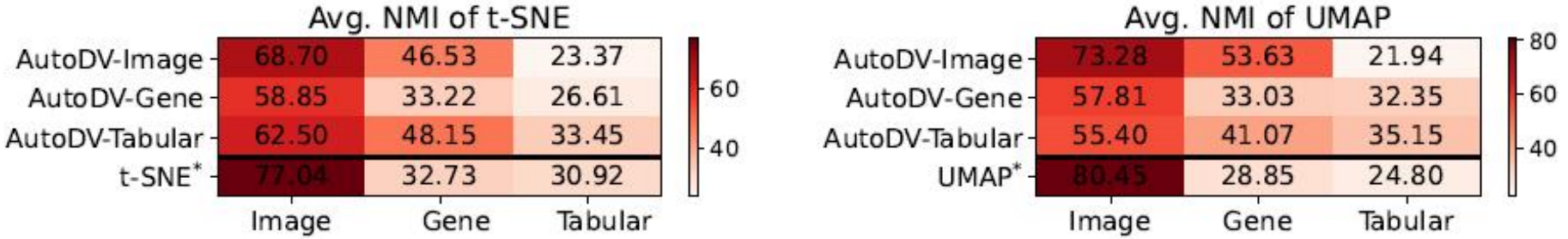


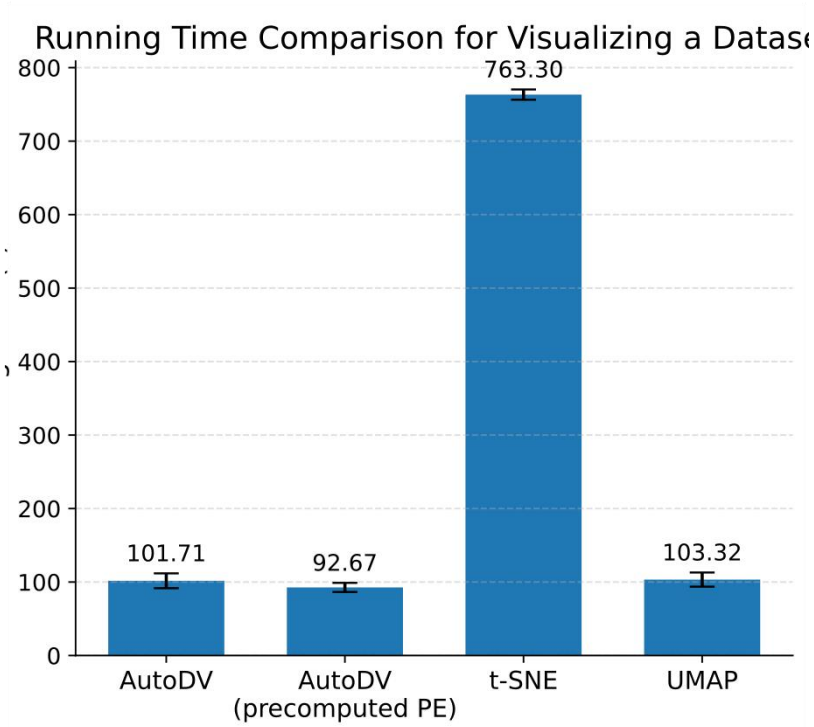
Figure 4: Cross-domain transferability performance using t-SNE and UMAP. Each heatmap shows average NMI results for (left) t-SNE and (right) UMAP. Within each map, columns denote the source domain and rows the target domain for the first three rows; the fourth row ("TSNE*" or "UMAP*") reports the within-domain performance of the optimal low-dimensional embedding. Test sets for each domain are the same as in Tables 2, 3, and 4.

Transferability, Efficiency, and Theory

Theorem 1 *Given a dataset $\boldsymbol{X}$ with n samples, let $\{\boldsymbol{P}^{(j)}, \boldsymbol{S}^{(j)}\}_{j=1}^{k}$ with $\boldsymbol{P}^{(j)} \in \mathbb{R}^{n \times d_e}$ be the input of AutoDV model f_ϕ, where the output is $\boldsymbol{Z} = f_\phi(\{\boldsymbol{P}^{(j)}, \boldsymbol{S}^{(j)}\}_{j=1}^{k})$. Denote $\tilde{\boldsymbol{X}}$ as a perturbed counterpart of $\mathbf{X}$, where the output is $\tilde{\boldsymbol{Z}} = f_\phi(\{\tilde{\boldsymbol{P}}^{(j)}, \tilde{\boldsymbol{S}}^{(j)}\}_{j=1}^{k})$. Suppose f_ϕ is L_ϕ-Lipschitz continuous, then*

$$\|\boldsymbol{Z} - \tilde{\boldsymbol{Z}}\|_F \leq 2kL_\phi \sqrt{\frac{2n}{e}} \max_j \left(\frac{c^{(j)} + 1}{\sqrt{\gamma^{(j)}}} \right) \|\boldsymbol{X} - \tilde{\boldsymbol{X}}\|_F, \tag{9}$$

where $c^{(j)} = \frac{2\sqrt{2\lambda_{\max}^{(j)}}}{\delta^{(j)}} + \frac{1}{2}\sqrt{\frac{d_e}{\lambda_{\min}^{(j)}}}$, $\delta^{(j)}$ is the eigen gap between $\boldsymbol{S}^{(j)}$ and $\tilde{\boldsymbol{S}}^{(j)}$, $\lambda_{\max}^{(j)}$ is the maximum eigenvalue of $\boldsymbol{S}^{(j)}$, and $\lambda_{\min}^{(j)}$ is the minimum the d_e-th eigenvalue between $\boldsymbol{S}^{(j)}$ and $\tilde{\boldsymbol{S}}^{(j)}$.



Conclusion

- Takeaways:
  - Train once, visualize directly
  - Works across varying dimensions and domains
  - Affine-invariant supervision stabilizes learning
- Core Contributions:
  - End-to-end HDV across varying feature dimensions and domains by converting each dataset into multi-scale similarity graphs.
  - Affine-invariant supervision aligns pairwise geometry instead of raw coordinates, resolving rotation / translation / scale ambiguity.
  - Strong generalization on unseen datasets with competitive or even better results than optimized t-SNE / UMAP on gene and tabular data.