Content

- Motivation
- Architecture
  Components
- Experiment Design
  Experiment Result
  Experiment Evaluation
- Discussion and Prospect

Motivation

Self-Supervised Learning?

Supervised Learning

Using labels

Classification

Unsupervised Learning

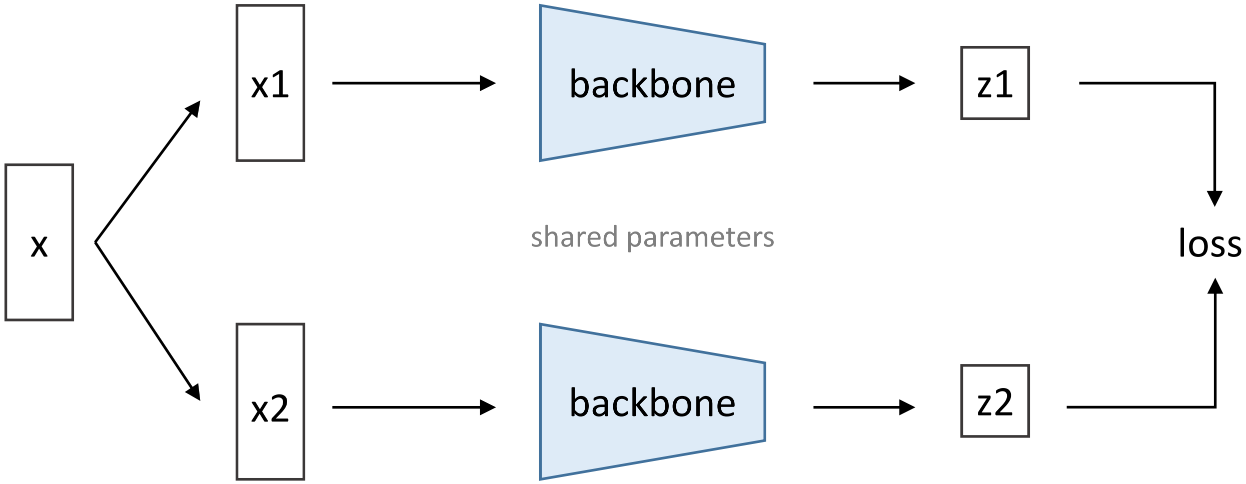
No labels

Clustering,
Pattern Mining

Motivation

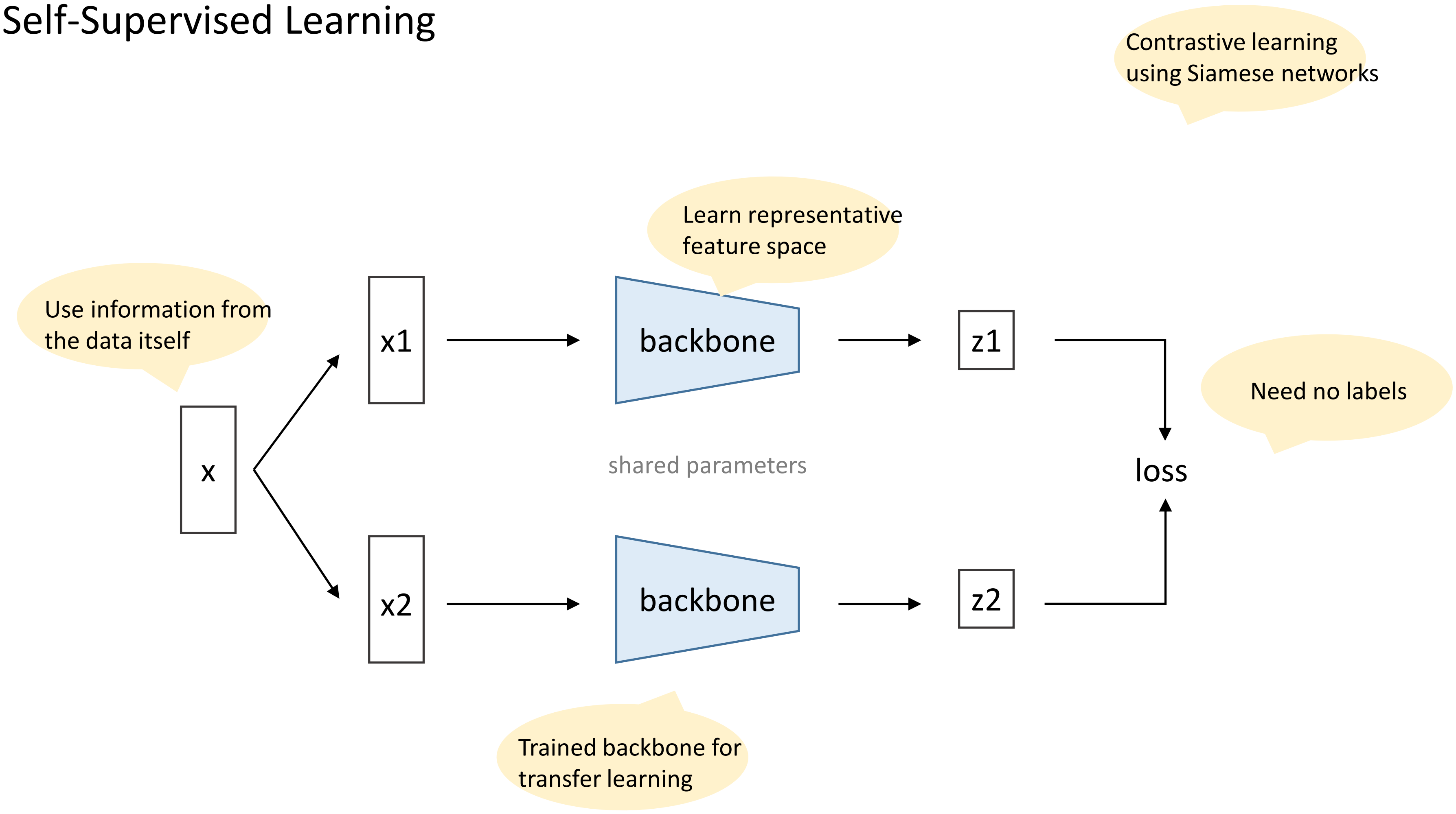
Self-Supervised Learning

Contrastive learning
using Siamese networks



Motivation

Self-Supervised Learning



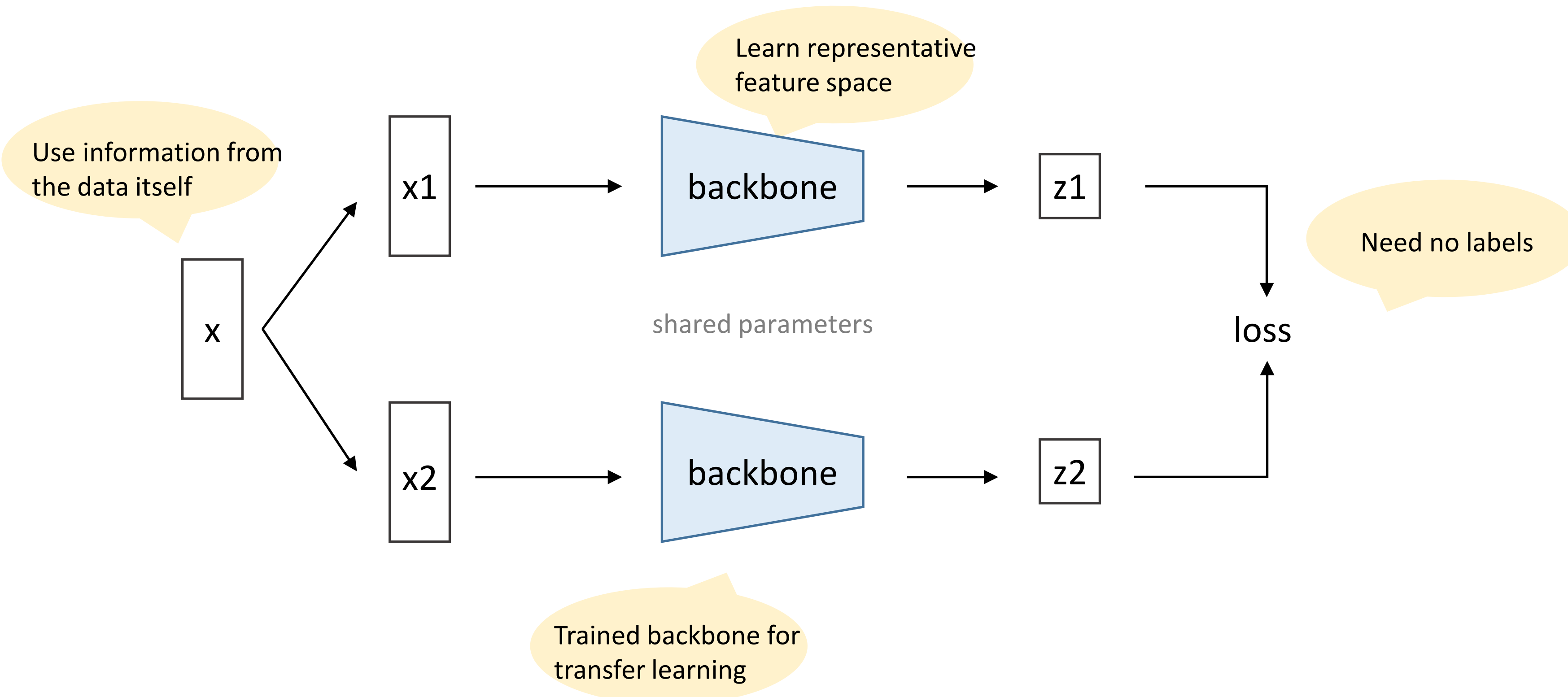
Motivation

Self-Supervised Learning

As upstream task in 2-stage model

Downstream task varies

Contrastive learning using Siamese networks



Motivation

Self-Supervised Learning

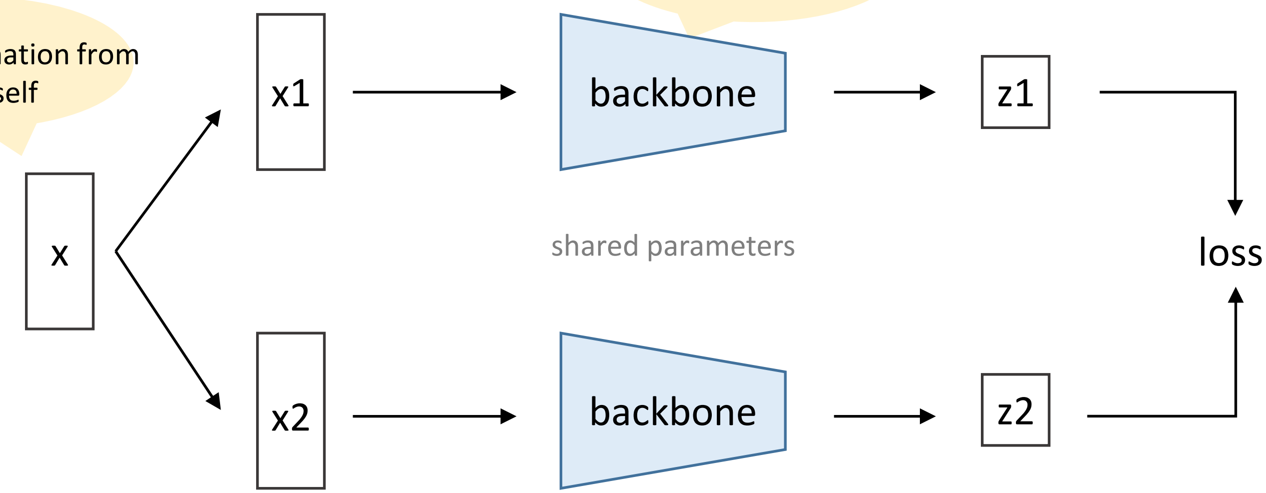
As upstream task in 2-stage model

Downstream task varies

Contrastive learning using Siamese networks

Learn representative feature space

Use information from the data itself



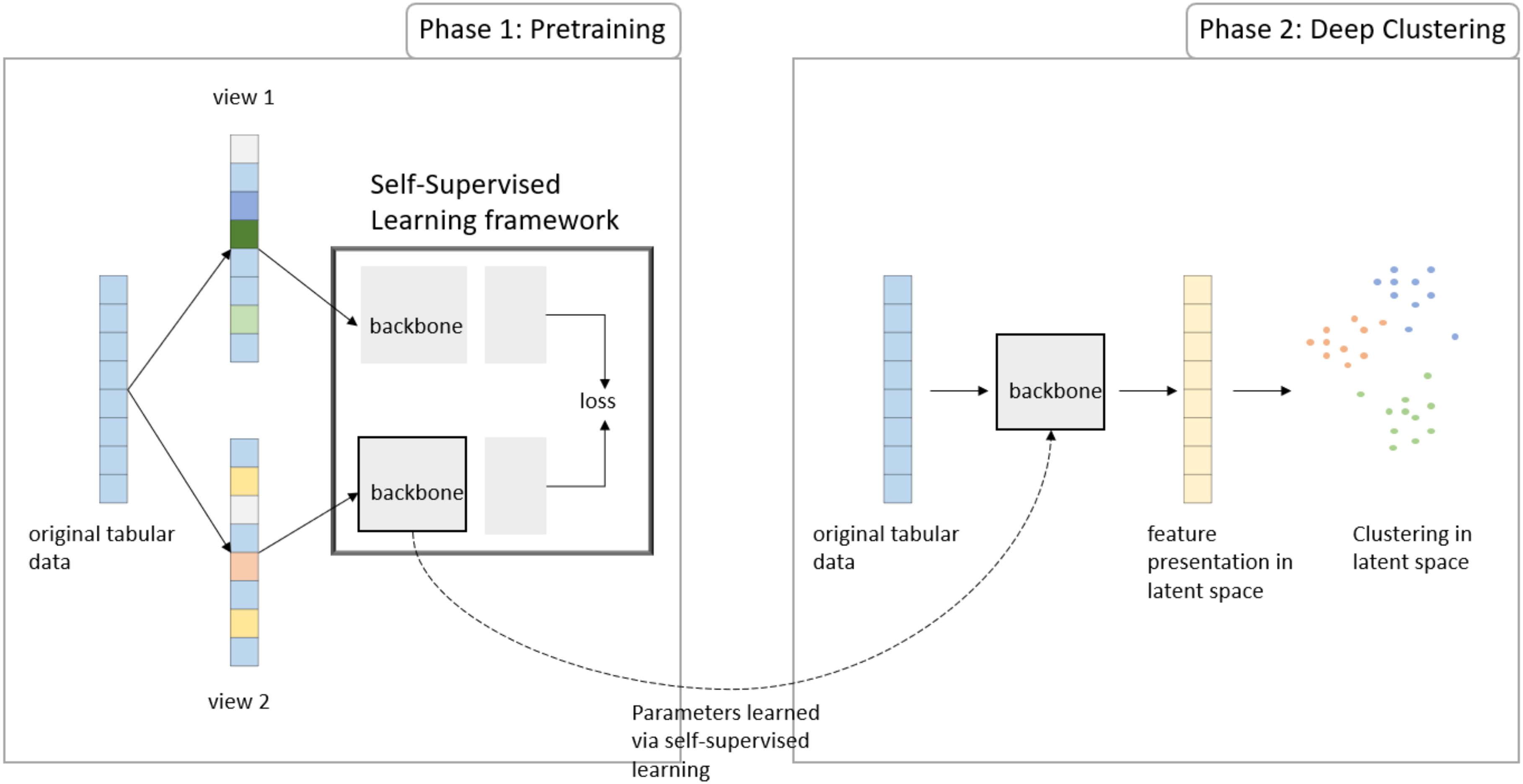
Need no labels

Trained backbone for transfer learning

Successful in Computer Vision and NLP

-> for tabular data?

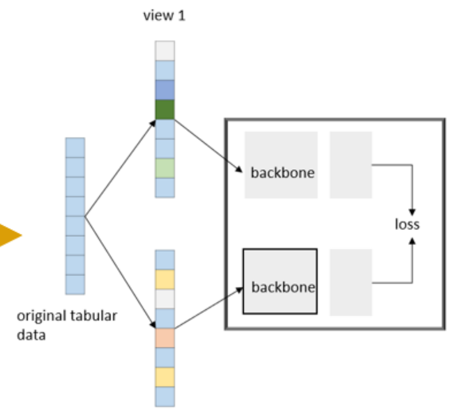
Architecture



Components

Comparison: Data Sets

- Synthetic Data Set
- MNIST
- Pendigits
- ...
- Wine
- Mobile



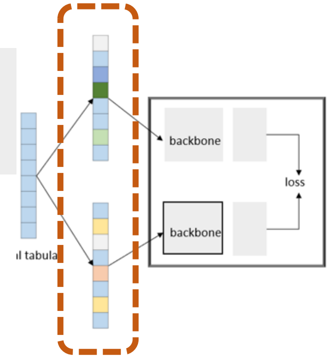
- SimCLR
- SimSiam
- Barlow Twins

Comparison: Self-Supervised Learning Frameworks

- Masked with mead
- Masked with random
- ...
- MixCut
- Mix with Random Neighbor
- Random Neighbor

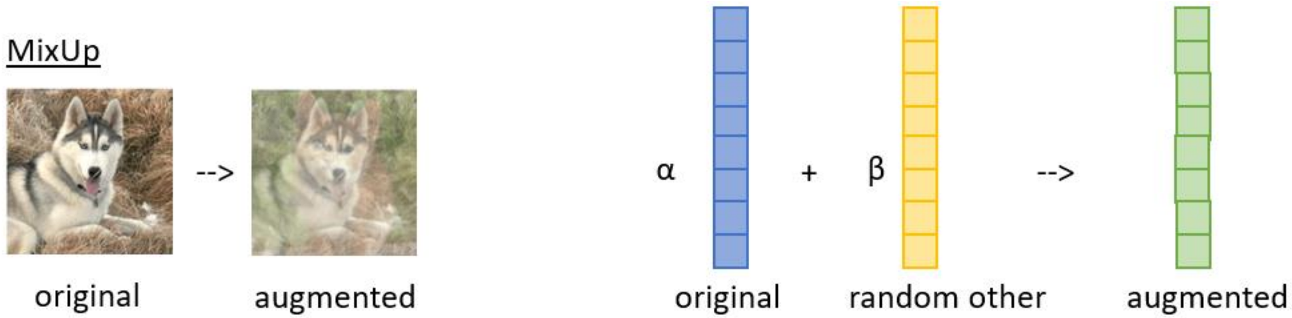
Comparison: Tabular Data Augmentation Methods

Component – Tabular Data Augmentation

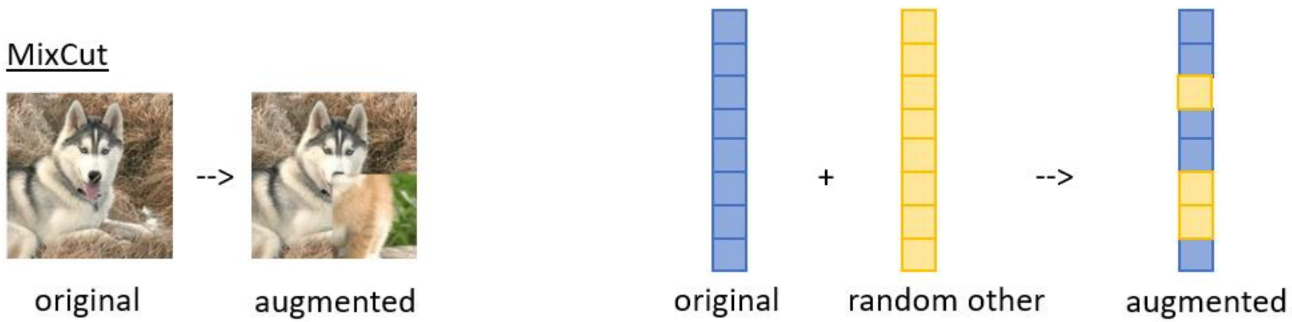


Approach I: Inspiration from Image Augmentation

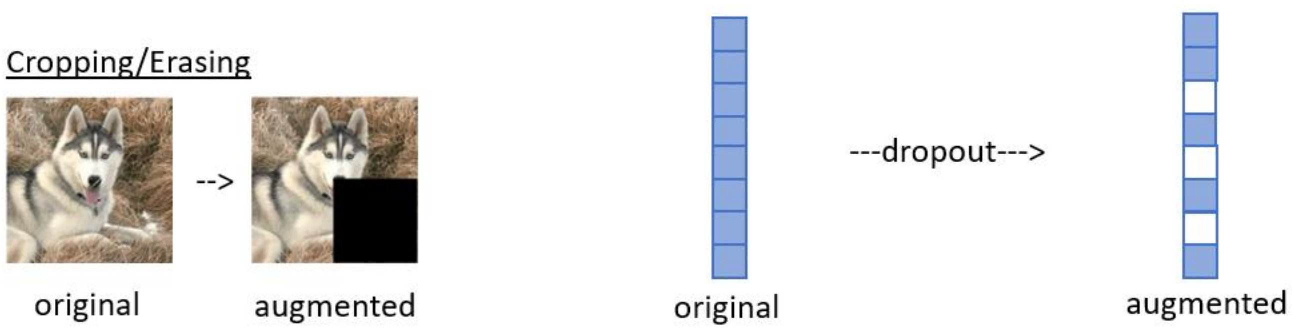
MixUp



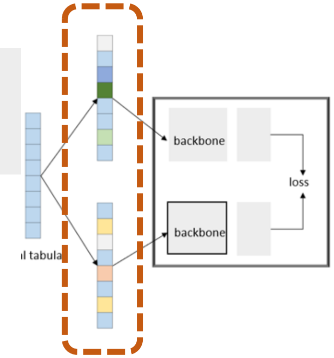
MixCut



Cropping/Erasing

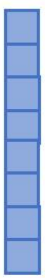


Component – Tabular Data Augmentation



Approach II: Inspiration from Masked Language Modeling

Masked Language Modeling



original data

---masked with --->

masked with mean of the feature or,

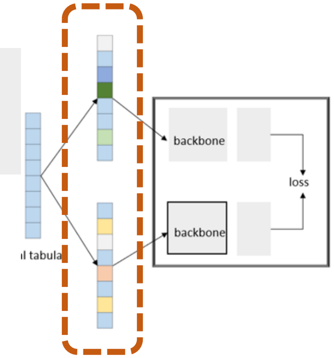
masked with random value in the domain range of the feature or,

masked with margin distribution of the feature

--->

augmented

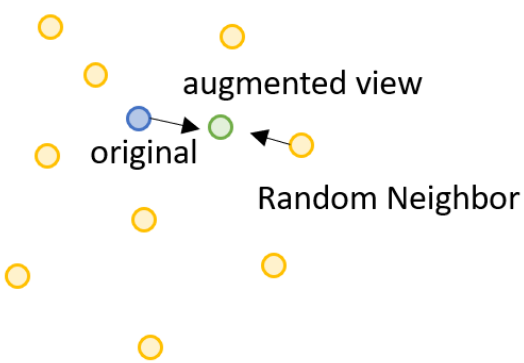
Component – Tabular Data Augmentation



Approach III: Inspiration from Upsampling

K nearest neighbor

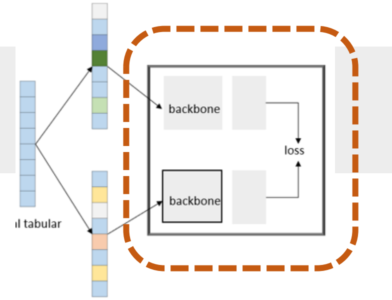
Data points in euclidean space



Mix With Random Neighbor

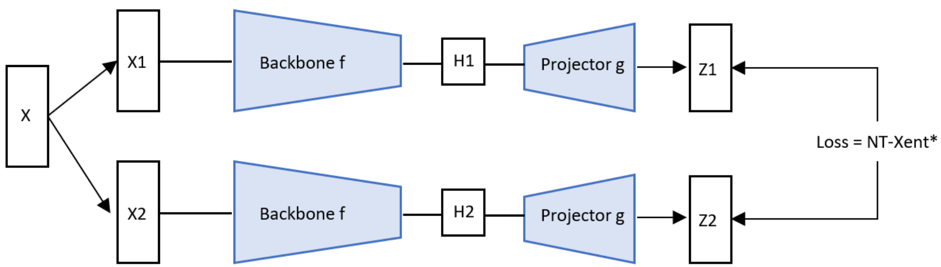
Random Neighbor

Component – Self-Supervised Learning Frameworks



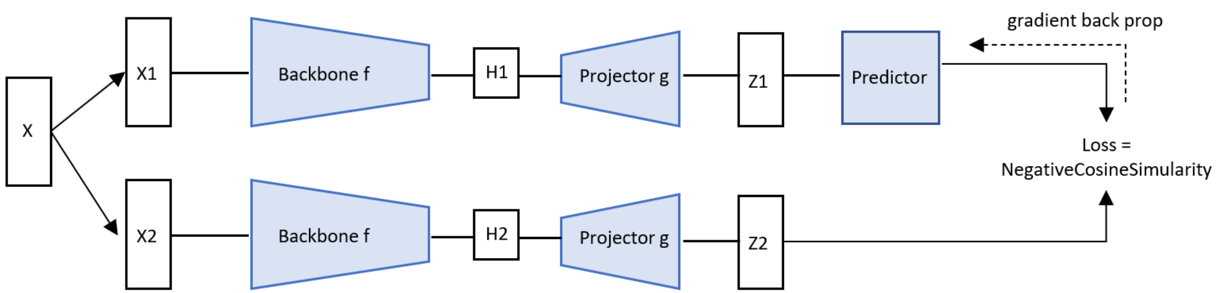
* normalized temperature-scaled cross entropy loss

SimCLR

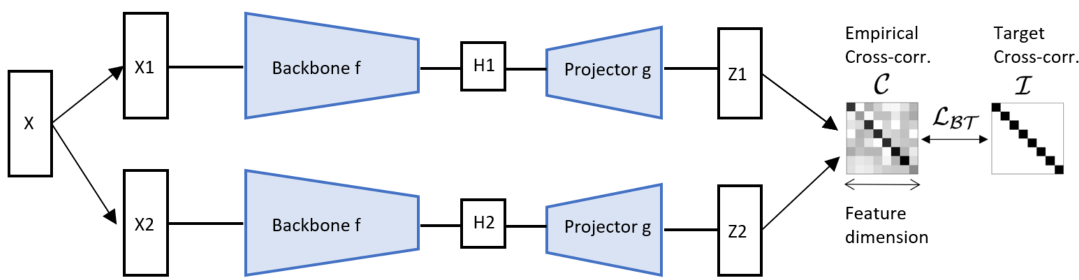


$$\mathbb{l}_{i,j} = -\log \frac{\exp(\mathrm{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}$$

SimSiam



Barlow Twins

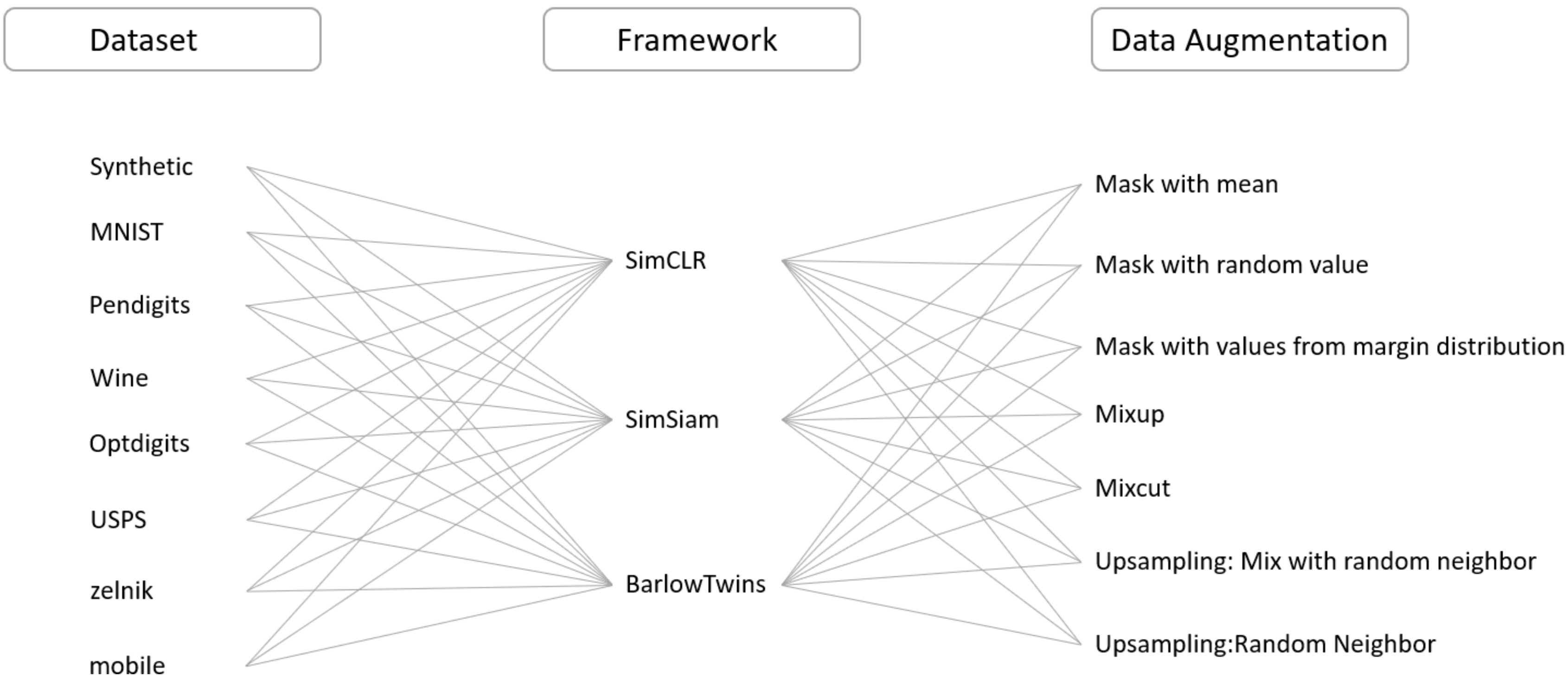


$$\mathcal{L_{BT}} \triangleq \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} {\mathcal{C}_{ij}}^2}_{\text{redundancy reduction term}}$$

Experiment Design

Dataset	Number of Instances	Number of Features	Number of Clusters
synthetic	10000	200	3
MNIST	69999	784	10
Pendigits	10992	16	10
Wine	6497	12	2
Optdigits	5620	64	10
USPS	9298	256	10
zelnik	512	2	4
mobile	2000	20	4

Ablation Study



Experiment Results

Dataset	Clustering Method	Self-supervised Framework	Tab-data Augmentation	NMI
MNIST	Kmeans (baseline)	none	none	0.49 (baseline)
	Latent Space Kmeans	SimCLR	masked with mean	0.54
			masked with random	0.51
			masked with margin distribution	0.52
			mixup	0.45
			mixcut	0.52
			input with dropout	0.65
			upsampling: mix with neighbors	0.7
			upsampling: neighbors as augmentation	**0.77**
		SimSiam	masked with mean	0.49
			masked with random	0.5
			masked with margin distribution	0.51
			mixup	0.49
			mixcut	0.53
			input with dropout	0.53
			upsampling: mix with neighbors	0.66
			upsampling: neighbors as augmentation	0.67
		Barlow Twins	masked with mean	0.47
			masked with random	0.47
			masked with margin distribution	0.5
			mixup	0.44
			mixcut	0.5
			input with dropout	0.51
			upsampling: mix with neighbors	0.65
			upsampling: neighbors as augmentation	0.69

Dataset	Clustering Method	Self-supervised Framework	Tab-data Augmentation	NMI
2zelnik_synthetic	Kmeans (baseline)	none	none	0.67 (baseline)
	Latent Space Kmeans	SimCLR	masked with mean	0.37
			masked with random	0.45
			masked with margin distribution	0.43
			mixup	0.42
			mixcut	0.44
			input with dropout	0.82
			upsampling: mix with neighbors	0.86
			upsampling: neighbors as augmentation	**0.99**
		SimSiam	masked with mean	0.6
			masked with random	0.67
			masked with margin distribution	0.62
			mixup	0.67
			mixcut	0.39
			input with dropout	0.81
			upsampling: mix with neighbors	0.64
			upsampling: neighbors as augmentation	0.79
		Barlow Twins	masked with mean	0.42
			masked with random	0.33
			masked with margin distribution	0.48
			mixup	0.48
			mixcut	0.46
			input with dropout	0.65
			upsampling: mix with neighbors	0.55
			upsampling: neighbors as augmentation	0.5

Dataset	Clustering Method	Self-supervised Framework	Tab-data Augmentation	NMI
Wine	Kmeans (baseline)	none	none	0.84 (baseline)
	Latent Space Kmeans	SimCLR	masked with mean	0.44
			masked with random	0.39
			masked with margin distribution	0.46
			mixup	0.57
			mixcut	0.31
			input with dropout	0.2
			upsampling: mix with neighbors	0.55
			upsampling: neighbors as augmentation	**0.88**
		SimSiam	masked with mean	0.6
			masked with random	0.42
			masked with margin distribution	0.44
			mixup	0.27
			mixcut	0.2
			input with dropout	0.42
			upsampling: mix with neighbors	0.75
			upsampling: neighbors as augmentation	**0.88**
		Barlow Twins	masked with mean	0.57
			masked with random	0.45
			masked with margin distribution	0.41
			mixup	0.2
			mixcut	0.22
			input with dropout	0.21
			upsampling: mix with neighbors	0.39
			upsampling: neighbors as augmentation	0.45

Dataset	Clustering Method	Self-supervised Framework	Tab-data Augmentation	NMI
USPS	Kmeans (baseline)	none	none	0.61 (baseline)
	Latent Space Kmeans	SimCLR	masked with mean	0.5
			masked with random	0.48
			masked with margin distribution	0.54
			mixup	0.52
			mixcut	0.48
			input with dropout	0.54
			upsampling: mix with neighbors	0.73
			upsampling: neighbors as augmentation	**0.74**
		SimSiam	masked with mean	0.59
			masked with random	0.45
			masked with margin distribution	0.52
			mixup	0.53
			mixcut	0.55
			input with dropout	0.47
			upsampling: mix with neighbors	0.71
			upsampling: neighbors as augmentation	0.72
		Barlow Twins	masked with mean	0.51
			masked with random	0.44
			masked with margin distribution	0.49
			mixup	0.49
			mixcut	0.46
			input with dropout	0.51
			upsampling: mix with neighbors	0.67
			upsampling: neighbors as augmentation	0.66

Dataset	Clustering Method	Self-supervised Framework	Tab-data Augmentation	NMI
Pendigits	Kmeans (baseline)	none	none	0.69 (baseline)
	Latent Space Kmeans	SimCLR	masked with mean	0.72
			masked with random	0.62
			masked with margin distribution	0.62
			mixup	0.51
			mixcut	0.66
			input with dropout	0.38
			upsampling: mix with neighbors	0.72
			upsampling: neighbors as augmentation	**0.77**
		SimSiam	masked with mean	0.64
			masked with random	0.61
			masked with margin distribution	0.63
			mixup	0.62
			mixcut	0.61
			input with dropout	0.56
			upsampling: mix with neighbors	0.72
			upsampling: neighbors as augmentation	0.74
		Barlow Twins	masked with mean	0.53
			masked with random	0.66
			masked with margin distribution	0.63
			mixup	0.64
			mixcut	0.54
			input with dropout	0.58
			upsampling: mix with neighbors	0.62
			upsampling: neighbors as augmentation	0.65

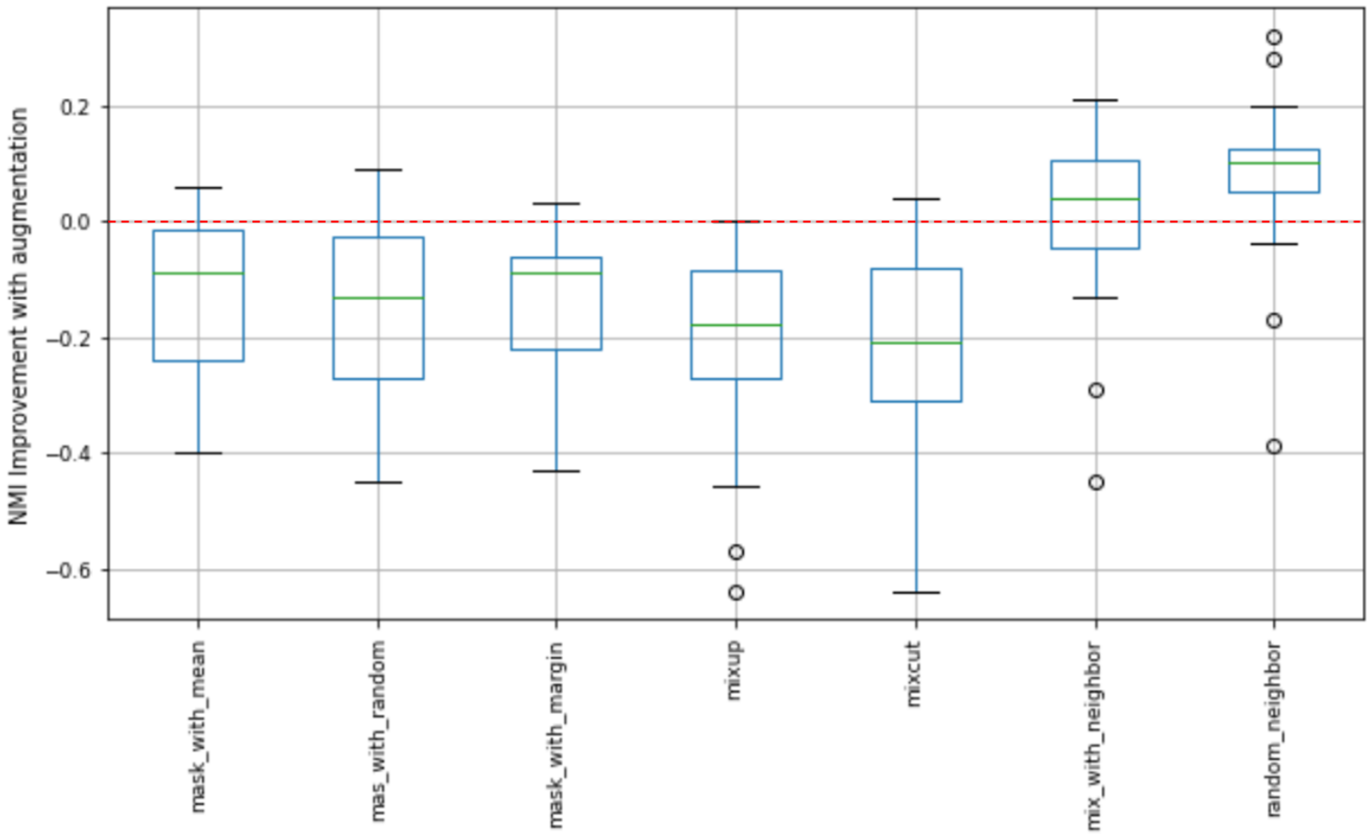
Dataset	Clustering Method	Self-supervised Framework	Tab-data Augmentation	NMI
Optdigits	Kmeans (baseline)	none	none	0.75 (baseline)
	Latent Space Kmeans	SimCLR	masked with mean	0.72
			masked with random	0.68
			masked with margin distribution	0.69
			mixup	0.64
			mixcut	0.64
			input with dropout	0.72
			upsampling: mix with neighbors	0.82
			upsampling: neighbors as augmentation	**0.85**
		SimSiam	masked with mean	0.72
			masked with random	0.66
			masked with margin distribution	0.7
			mixup	0.53
			mixcut	0.59
			input with dropout	0.67
			upsampling: mix with neighbors	0.79
			upsampling: neighbors as augmentation	0.81
		Barlow Twins	masked with mean	0.64
			masked with random	0.65
			masked with margin distribution	0.67
			mixup	0.51
			mixcut	0.49
			input with dropout	0.69
			upsampling: mix with neighbors	0.73
			upsampling: neighbors as augmentation	0.78

Dataset	Clustering Method	Self-supervised Framework	Tab-data Augmentation	NMI
Synthetic Dataset	Kmeans (baseline)	none	none	0.69 (baseline)
	Latent Space Kmeans	SimCLR	masked with mean	0.72
			masked with random	0.51
			masked with margin distribution	0.6
			mixup	0.39
			mixcut	0.21
			input with dropout	0.83
			upsampling: mix with neighbors	**0.86**
			upsampling: neighbors as augmentation	0.85
		SimSiam	masked with mean	0.6
			masked with random	0.4
			masked with margin distribution	0.55
			mixup	0.34
			mixcut	0.43
			input with dropout	0.61
			upsampling: mix with neighbors	0.77
			upsampling: neighbors as augmentation	0.8
		Barlow Twins	masked with mean	0.3
			masked with random	0.32
			masked with margin distribution	0.4
			mixup	0.23
			mixcut	0.19
			input with dropout	0.6
			upsampling: mix with neighbors	0.75
			upsampling: neighbors as augmentation	0.79

Dataset	Clustering Method	Self-supervised Framework	Tab-data Augmentation	NMI
mobile	Kmeans (baseline)	none	none	0.59 (baseline)
	Latent Space Kmeans	SimCLR	masked with mean	0.48
			masked with random	0.34
			masked with margin distribution	0.31
			mixup	0.29
			mixcut	0.21
			input with dropout	0.33
			upsampling: mix with neighbors	0.65
			upsampling: neighbors as augmentation	**0.75**
		SimSiam	masked with mean	0.35
			masked with random	0.3
			masked with margin distribution	0.39
			mixup	0.44
			mixcut	0.33
			input with dropout	0.47
			upsampling: mix with neighbors	0.54
			upsampling: neighbors as augmentation	0.65
		Barlow Twins	masked with mean	0.34
			masked with random	0.67
			masked with margin distribution	0.33
			mixup	0.32
			mixcut	0.25
			input with dropout	0.63
			upsampling: mix with neighbors	0.62
			upsampling: neighbors as augmentation	0.64

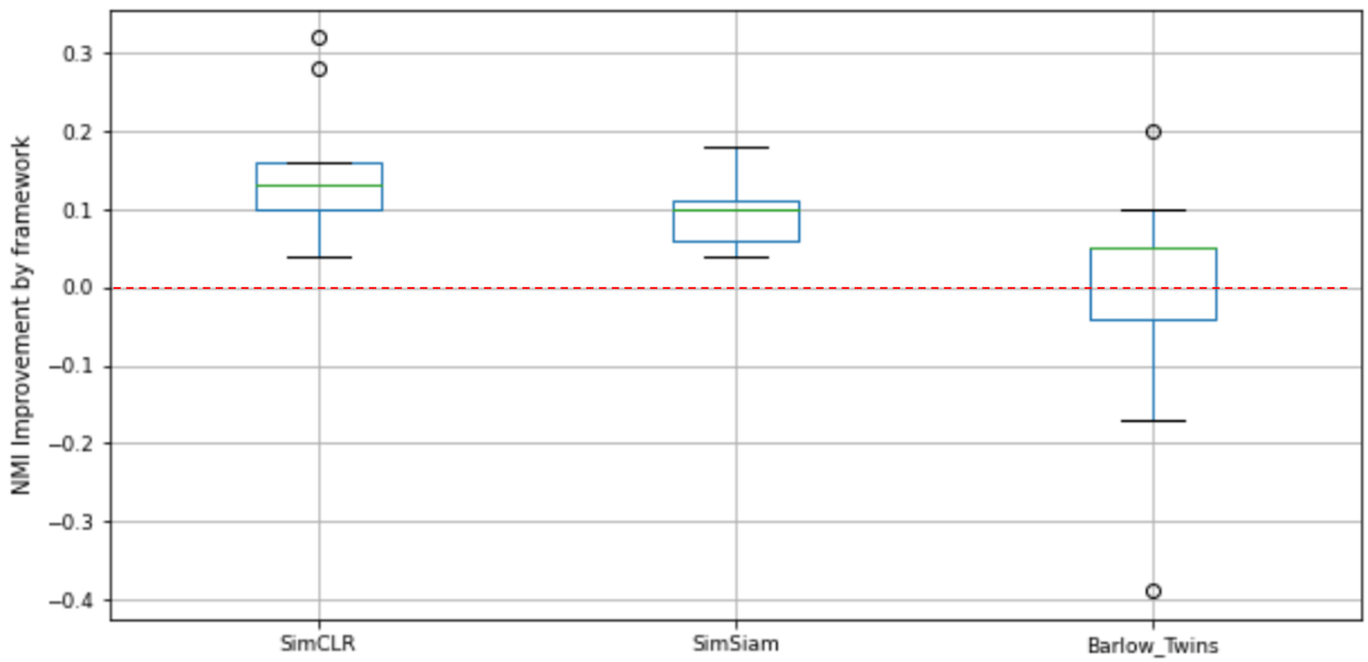
Experiment Evaluation

Comparison Augmentation Methods

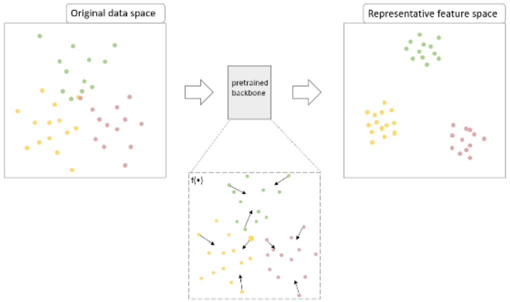


Experiment Evaluation

Comparison Self-Supervised Learning Frameworks

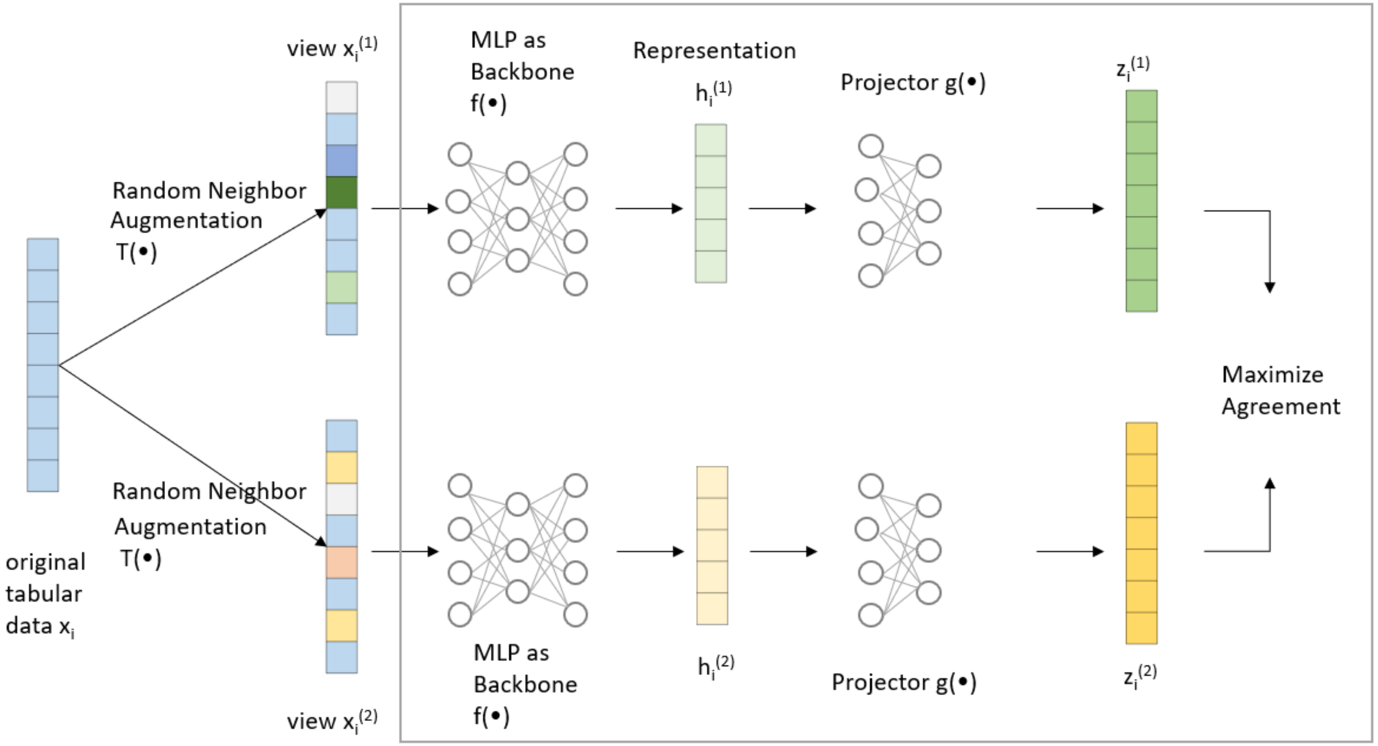


Proposed Model

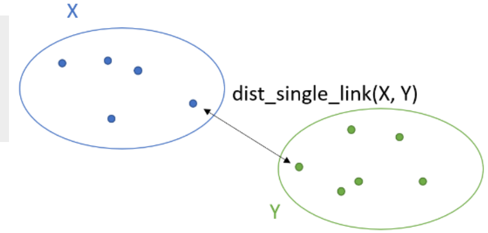


Phase 1: Pretraining

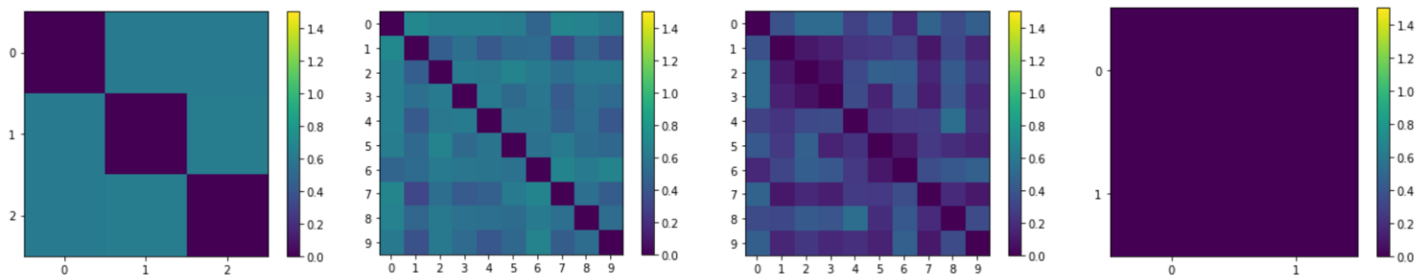
SimCLR framework + 'Random Neighbor' Augmentation



Experiment Evaluation



Performance Difference Between Data Sets

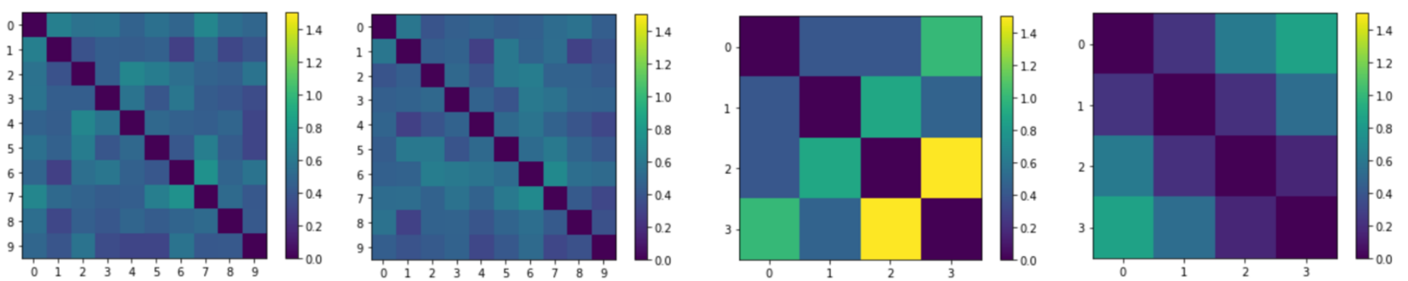


Synthetic ΔNMI =0.15

MNIST ΔNMI =0.28

Pendigits ΔNMI =0.08

Wine ΔNMI =0.04



Optdigits ΔNMI =0.10

USPS ΔNMI =0.13

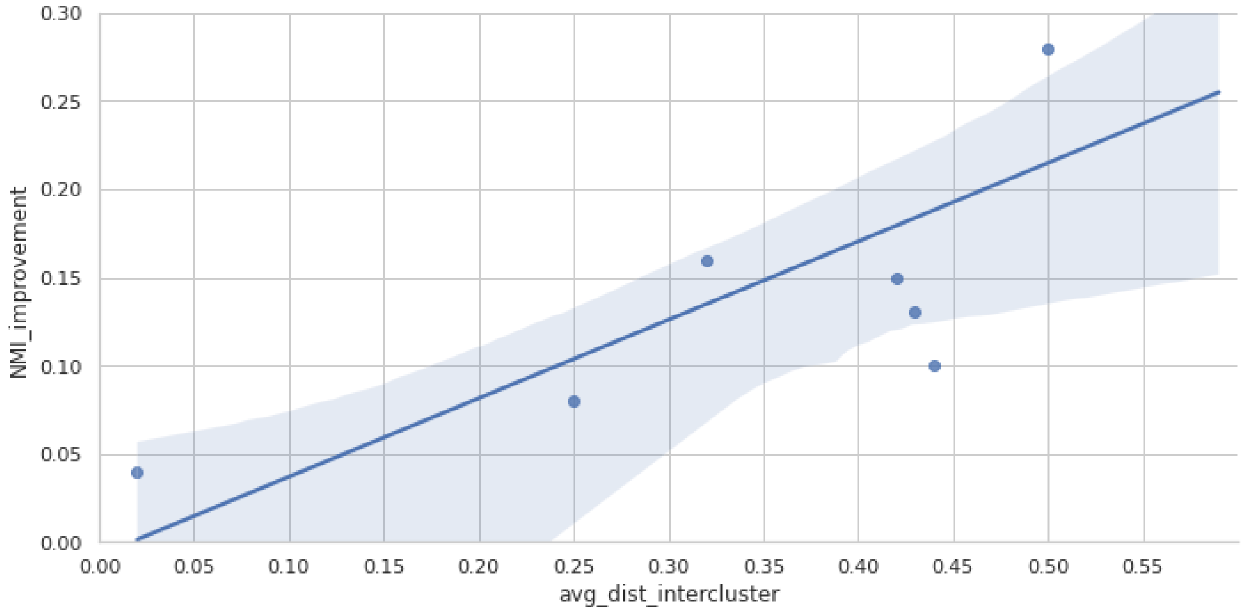
Zelnik ΔNMI =0.32

Mobile ΔNMI =0.16

Separation of Clusters and NMI Improvement

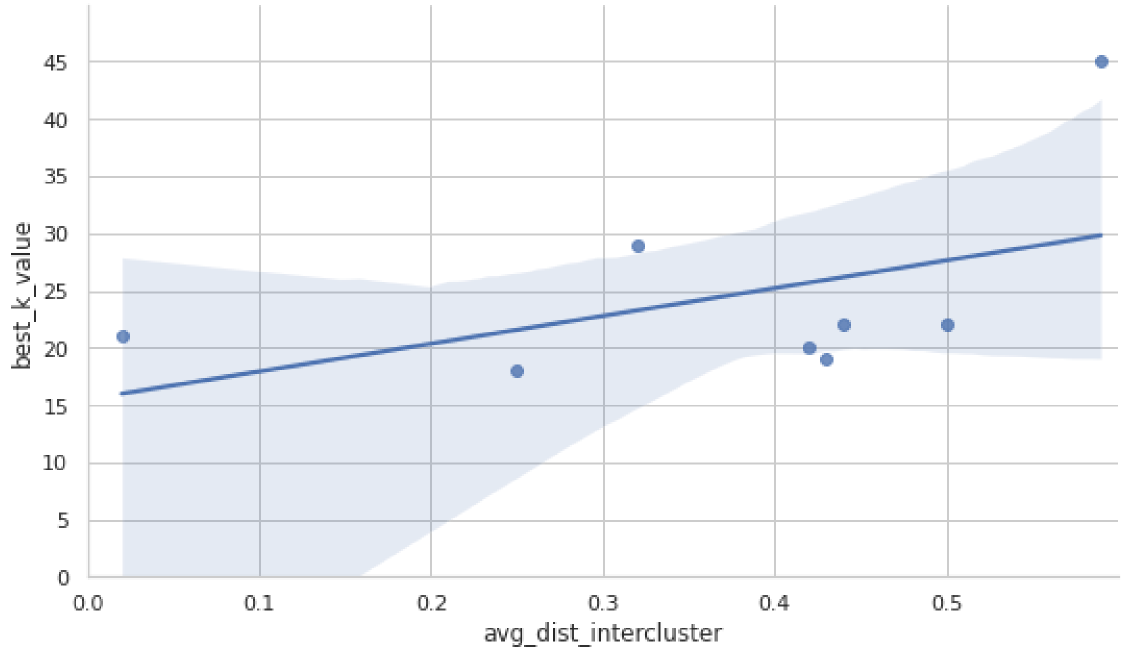
Experiment Evaluation

Performance Difference Between Data Sets

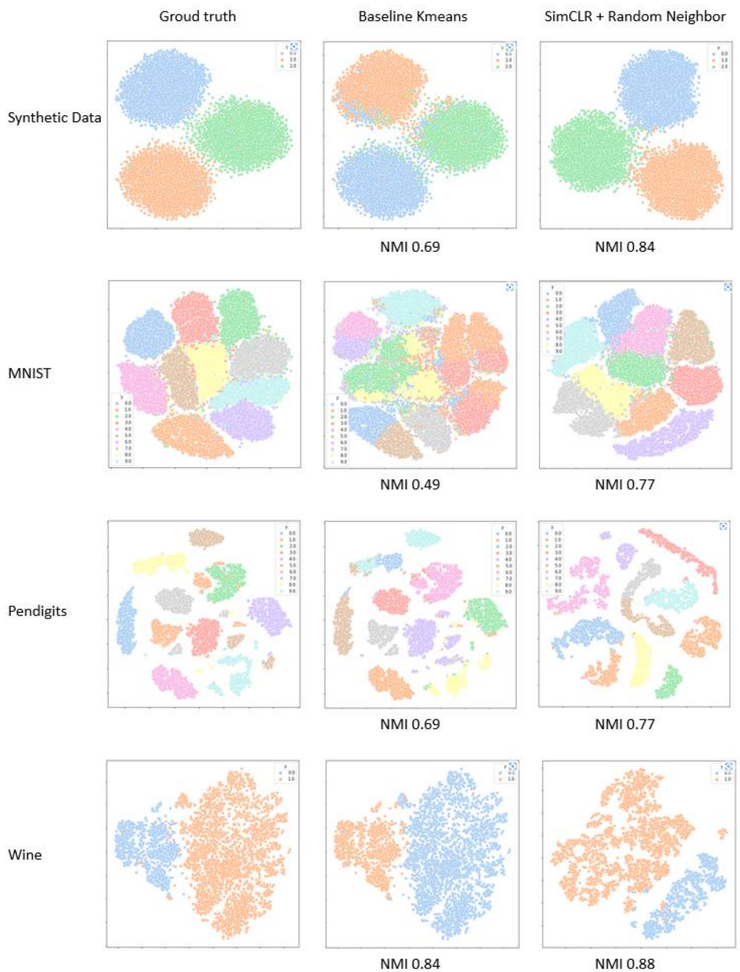


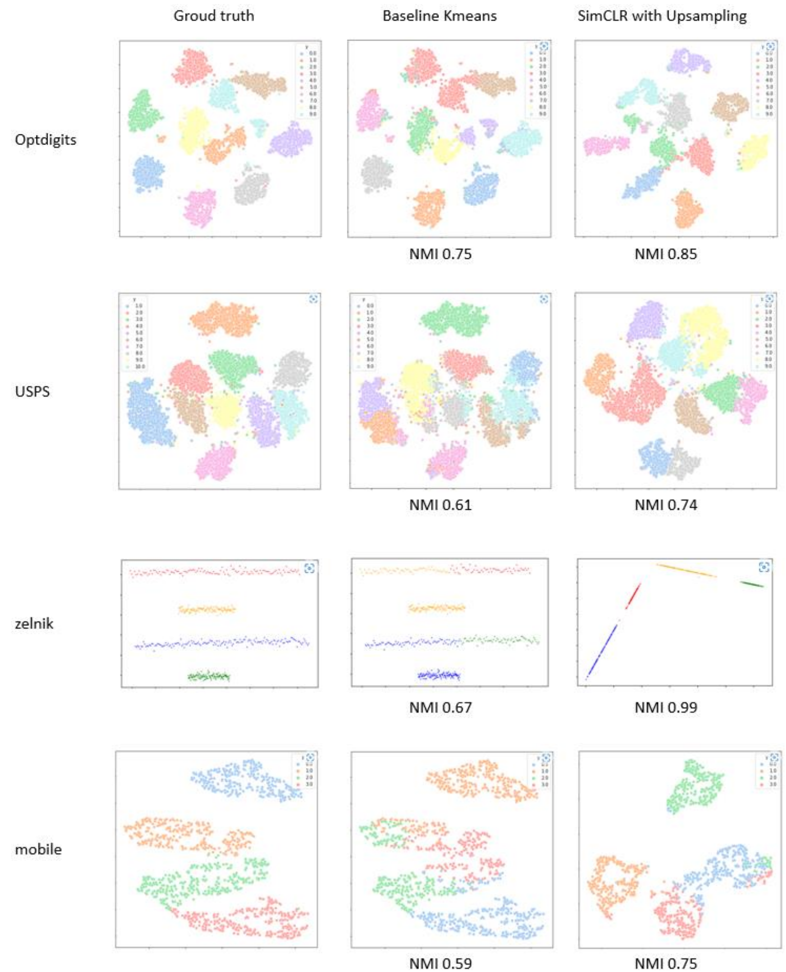
Experiment Evaluation

Hyper-parameter K for Random Neighbor in the K-nearest neighbors



Visualization of Clustering Improvement



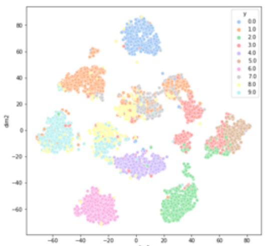


Visualization of Clustering Improvement

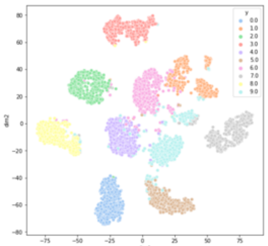
Other Simple Clustering Methods

Optdigits | original data space | embedding space from trained backbone

GMM

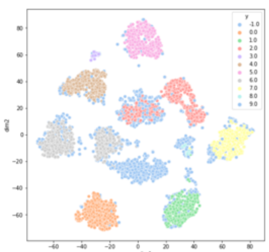


NMI = 0.67

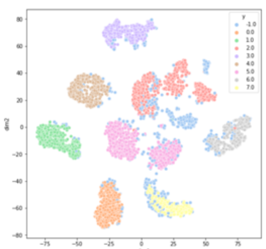


NMI = 0.89

DBSCAN

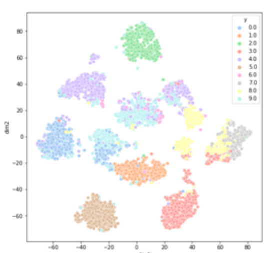


NMI = 0.75

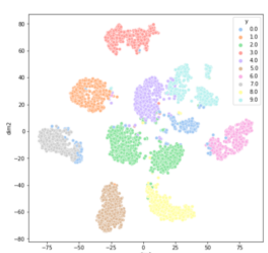


NMI = 0.77

Agglomerative Clustering



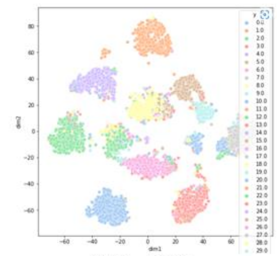
NMI = 0.67



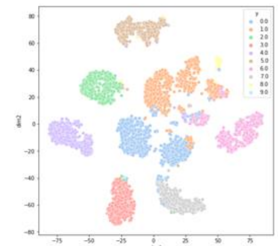
NMI = 0.87

Optdigits | original data space | embedding space from trained backbone

Mean Shift

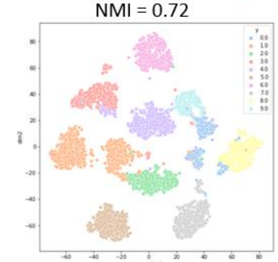


NMI = 0.72

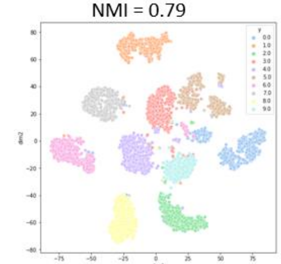


NMI = 0.79

BIRCH

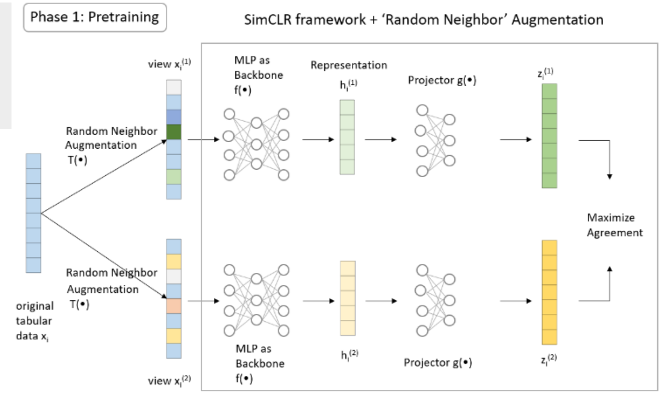


NMI = 0.84

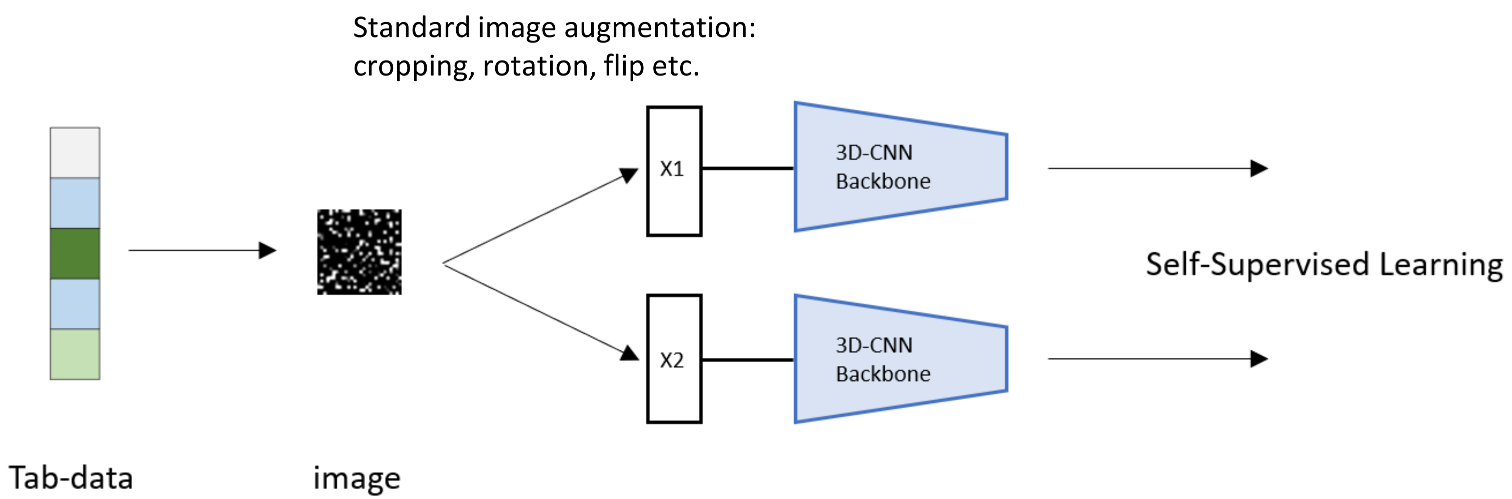


NMI = 0.88

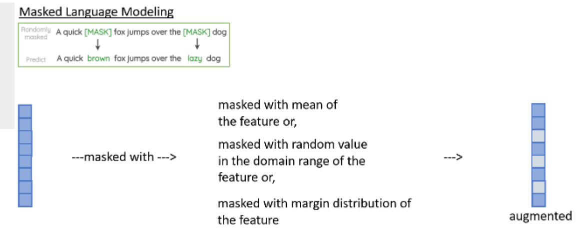
Discussion and Prospect



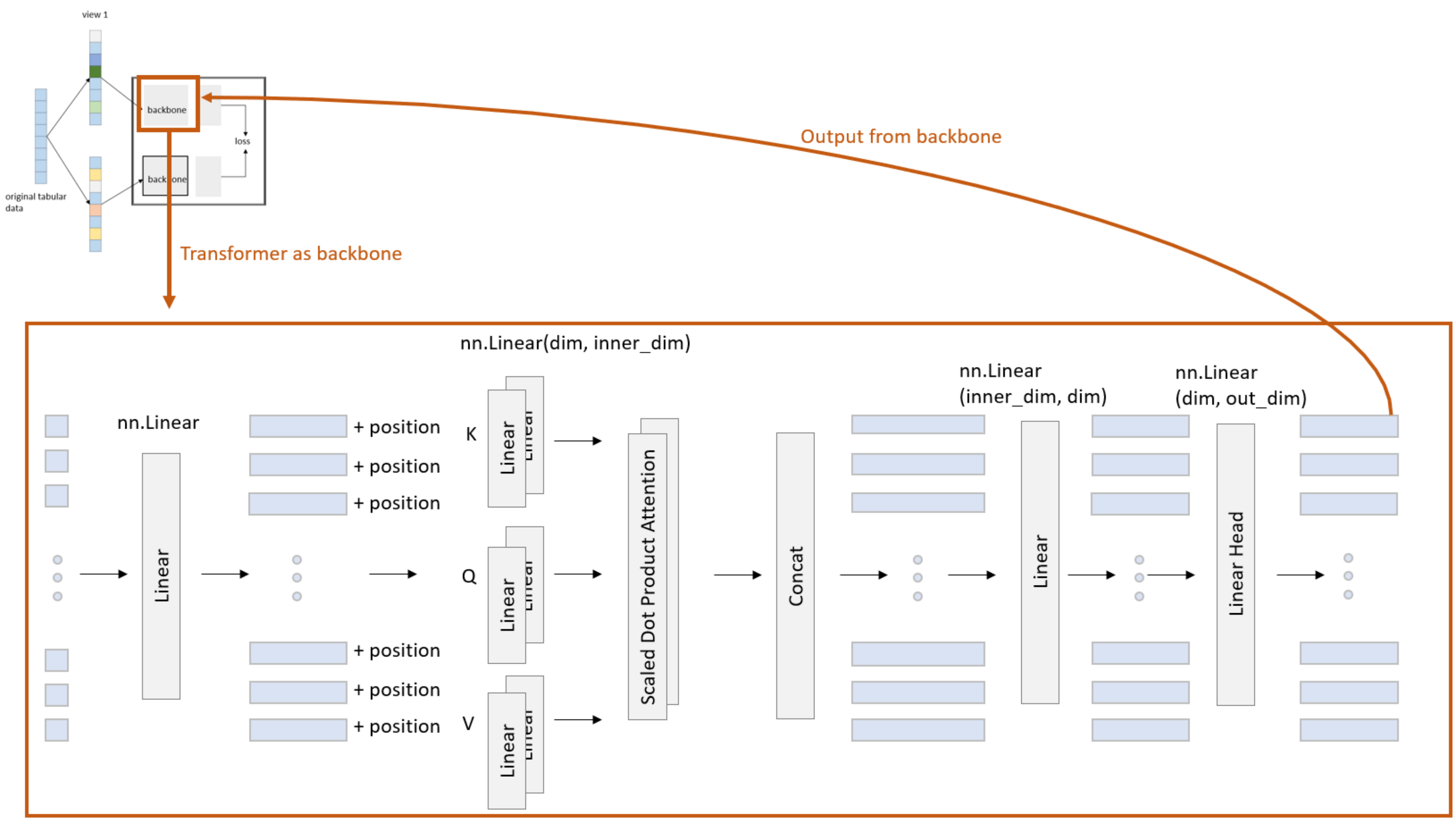
Convolutional Neural Network as Backbone of the Model



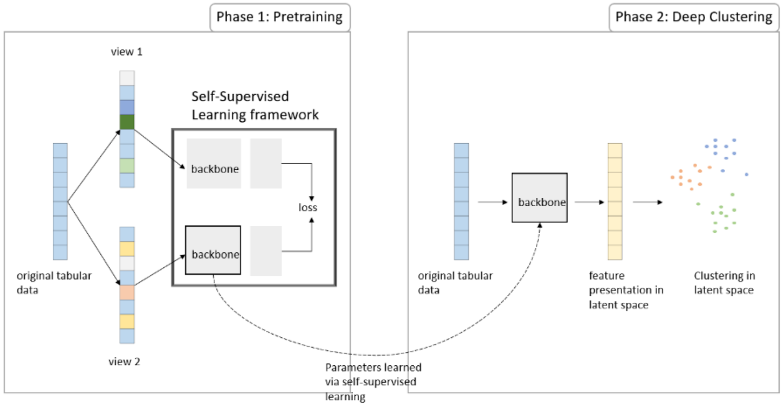
Discussion and Prospect



Transformer as Backbone of the Model



Conclusion



- **two-stage deep learning model**
  -> 1. stage: pretraining with self-supervised learning framework
  -> 2. stage: deep clustering in embedding space
- **SimCLR as self-supervised learning framework**
  -> most stable
- **Random Neighbor as augmentation method for tabular data**
  -> most effective
- **clustering in embedding space outperforms clustering in original data space**
- **discussion and prospect**
  -> CNN as backbone, image augmentation methods may work
  -> self-attention-block as backbone, masked value methods may also work

Thanks for your attention!

References

[1] Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. In arXiv preprint arXiv:2106.15147, 2021.
[2] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard S¨ackinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In Advances in neural information processing systems, 1993.
[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Proceedings of the European conference on computer vision (ECCV), pages 132–149, 20018.
[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Proceedings of the European conference on computer vision (ECCV), 2018.
[5] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. In Journal of artificial intelligence research, pages 321–357, 2002.
[6] Jiaao Chen, Zichao Yang, and Diyi Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In arXiv preprint arXiv:2004.12239, 2020.
[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607, 2020.
[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15750–15758, 2021.
[9] Van der Maaten, Laurens, and Geoffrey Hinton. Visualizing data using t-sne. In Journal of machine learning research, 2008.
[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.Bert: Pre-training of deep bidirectional transformers for language understanding. In arXiv preprint arXiv:1810.04805, 2018.
[11] William Falcon and Kyunghyun Cho. A framework for contrastive self-supervised learning and designing a new approach. In arXiv preprint arXiv:2009.00104, 2020.
[12] Jean-Bastien Grill, Florian Strub, Florent Altch´e, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In Advances in neural information processing systems, pages 21271–21284, 2020.
[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9729–9738, 2020.
[14] Kaiming He, Georgia Gkioxari, Piotr Doll´ar, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.
[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
[16] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. In The handbook of brain theory and neural networks, page 1995, 1995.
[17] Andrzej Rusiecki. Standard dropout as remedy for training deep neural networks with label noise. In International Conference on Dependability and Complex Systems, pages 534–542, 2020.
[18] A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, L Kaiser, and I Polosukhin. Attention is all you need. In Advances in neural information processing systems, 2017.
[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.
[20] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, pages 1096–1103, 2008.
[21] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In International conference on machine learning, pages 478–487, 2016.
[22] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In international conference on machine learning, pages 3861–3870, 2017.
[23] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. Vime: Extending the success of self-and semi-supervised learning to tabular domain. In Advances in Neural Information Processing Systems, pages 11033–11043, 2020.
[24] W Zaremba, I Sutskever, and O Vinyals. Recurrent neural network regularization. In arXiv preprint arXiv:1409.2329, 2014.
[25] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and St´ephane Deny. Barlow twins: Self-supervised learning via redundancy reductio. In International Conference on Machine Learning, pages 12310–12320, 2021.
[26] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In Advances in neural information processing systems, 2015.