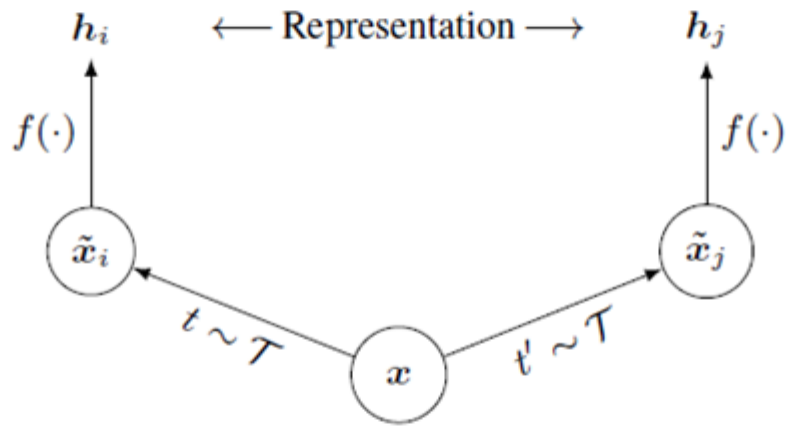
Contrastive Learning Is Spectral Clustering On Similarity Graph

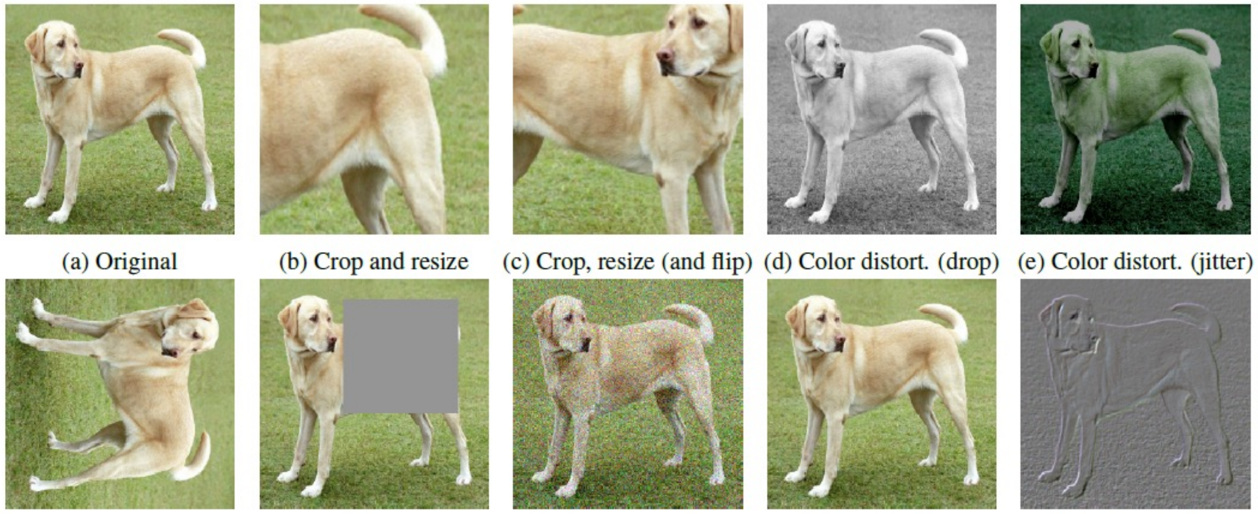
Zhiquan Tan* Yifan Zhang* Jingqin Yang* Yang Yuan

Tsinghua University

Contrastive learning: SimCLR



- Given x, sample two augmentations of x
  - Dog → (cropped dog, flipped dog)
- Given $2N$ augmentation pairs, what do we hope?
  - InfoNCE loss: $\mathrm{L}\left(q, p_1, \{p_i\}_{i=2}^{N}\right) = -\log \frac{\exp(-\|f(q)-f(p_1)\|^2/2\tau)}{\sum_{i=1}^{N} \exp(-\|f(q)-f(p_i)\|^2/2\tau)}$
  - Similar images are mapped together, different images are far apart



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)
(f) Rotate {90°, 180°, 270°} (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Why does it work?

- Haochen et al. 2021 proved that, replace InfoNCE with spectral loss, contrastive learning is approximately spectral clustering：
  - $\hat{F} = F^* \cdot diag([\sqrt{\gamma_1}, \cdots, \sqrt{\gamma_k}])R$
  - Adds additional linear transformations to F^*

$$\mathcal{L}(f) = -2 \cdot \mathbb{E}_{x,x^+}\left[f(x)^\top f(x^+)\right] + \mathbb{E}_{x,x^-}\left[\left(f(x)^\top f(x^-)\right)^2\right],$$

- We prove:
  - The standard InfoNCE (not spectral loss), does exactly spectral clustering (no additional transformation) on the similarity graph
  - This equivalence is exact!

Synthetic Experiments

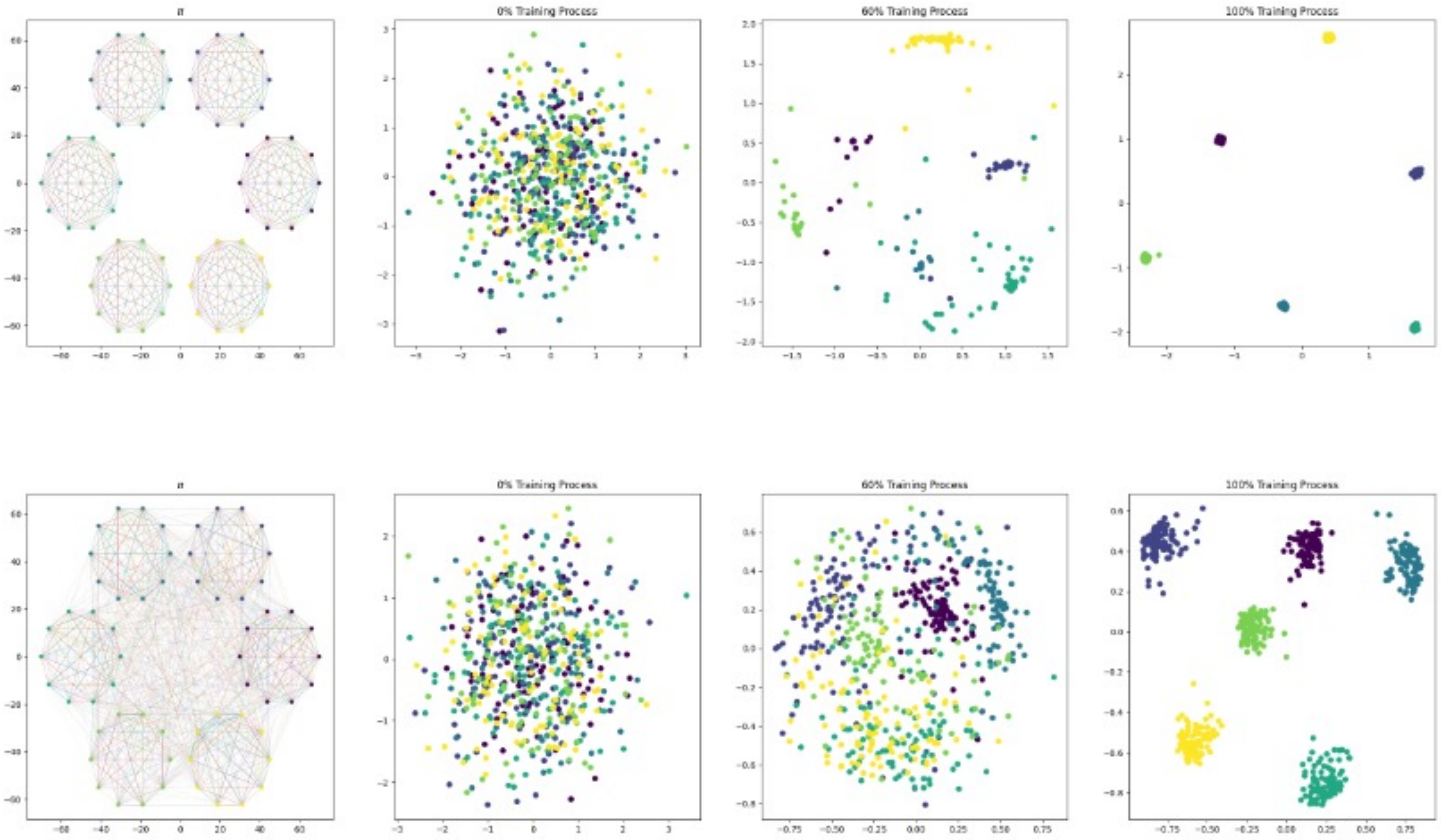
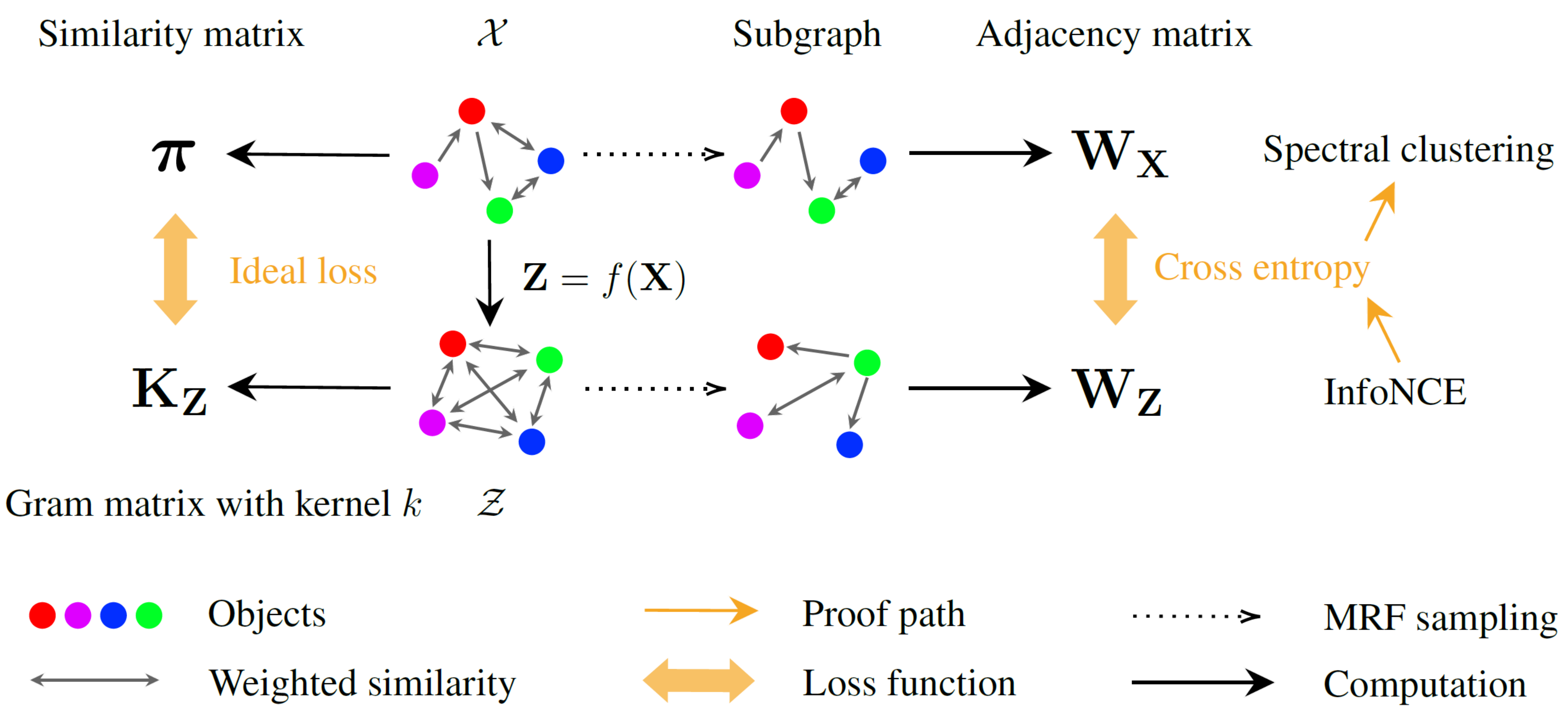


Illustration of our analysis

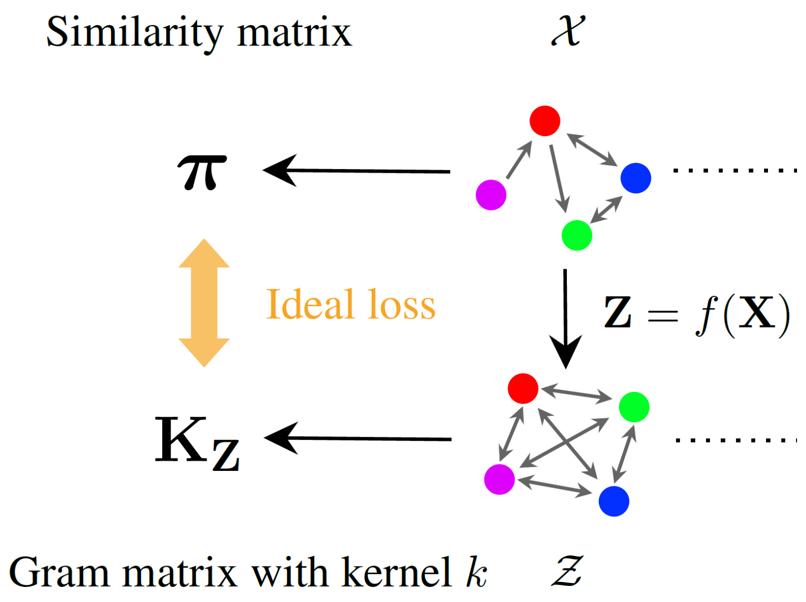


Proof sketch

- Given n objects $X_1, \cdots, X_n \in \chi$
  - Including all augmented images, finite
- Augmentation pair (X_i, X_j) defines a similarity edge in π
  - $\pi_{i,j}$ =Prob(X_i, X_j sampled together)
  - X_i, X_j are similar **semantically**
  - However, X_i and X_j are not similar in pixel space (large ℓ_p distance)
- Question:
  - Can we find an ideal space, such that semantic similarity is captured naturally?
    - $Z = f(X)$
  - Various solutions! Today: Reproducing Kernel Hilbert Space.

Reproducing kernel Hilbert space

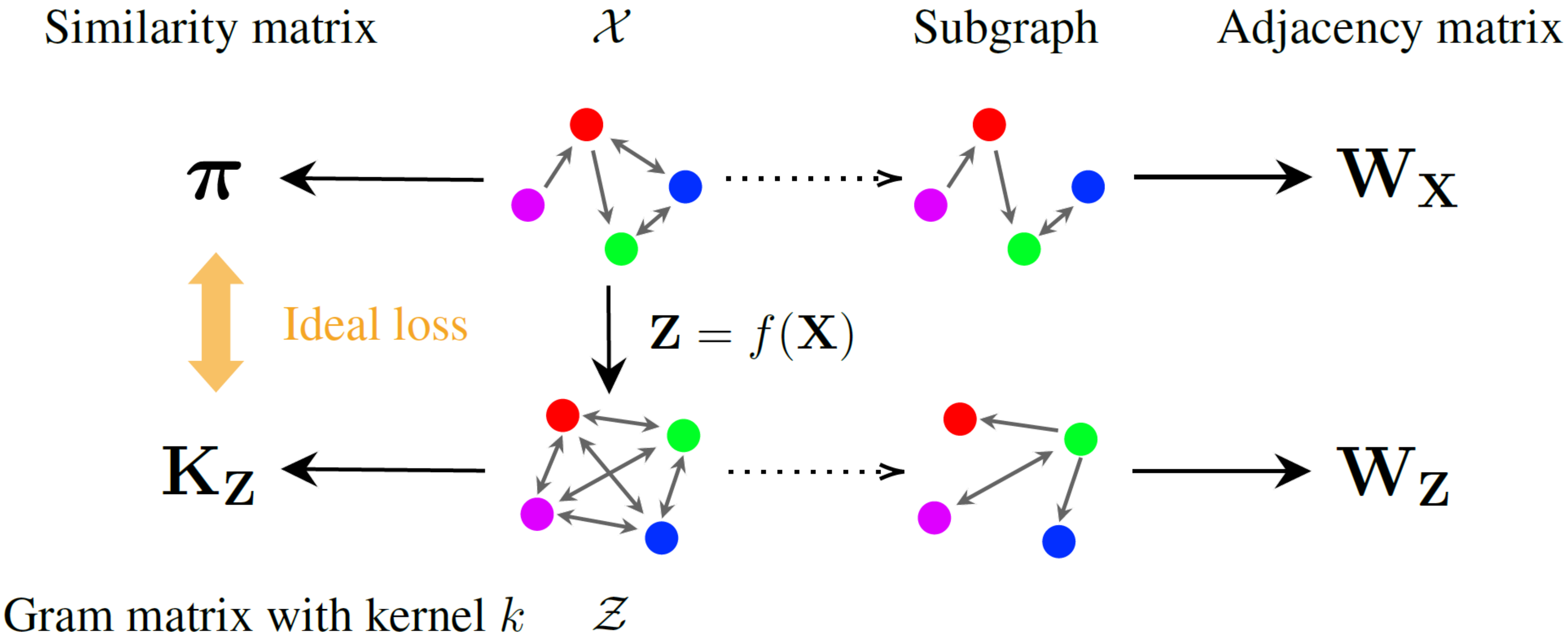
- Given Z_i, Z_j, consider $\phi: Z \to H$, such that
  - $k(Z_i, Z_j) =< \phi(Z_i), \phi(Z_j) >_H$
  - Inner product in RKHS H, is the kernel function in Z
  - H can have infinite dimension, we do not need to compute ϕ explicitly
- Similarity between Z_i, Z_j defined in H
  - Well defined, well shaped
  - Denote similarity matrix as K_Z
- Question: how to learn f?
  - When n is huge, hard to design the loss
  - Too many edges in between



Markov random fields (MRF)

- Given graph π, we may sample unweighted subgraphs from π
  - $W_{i,j} \in \{0,1\}$
- The score of W: $s(\mathrm{W},\pi) = \Pi_{(i,j)\in[n]^2} \pi_{i,j}^{W_{i,j}}$
  - Given π, which is the score of W?
    - Multiple score of each edge together
- Add restriction: $\Omega(W)\Pi_{(i,j)\in[n]^2} \pi_{i,j}^{W_{i,j}}$
  - For example, $\Omega(W) = 1$, if and only if each node in W has out-deg=1
- $P(W;\pi) \propto \Omega(W)\Pi_{(i,j)\in[n]^2} \pi_{i,j}^{W_{i,j}}$
  - Each W is sampled with probability proportional to its score

Sampling subgraphs for both π and K_Z



How to compare W_X and W_Z?

- W_X and W_Z are random variables based on π and K_Z
- Cross entropy loss
  - $H_\pi^k(Z) = -E_{W_X \sim P(\cdot;\pi)}[\log P(W_Z = W_X; K_Z)]$
  - Sample W_X from $P(\cdot;\pi)$, and check the probability that $W_Z = W_X$
- $H_\pi^k(Z)$ is equivalent to
  - InfoNCE loss
  - running spectral clustering (Van Assel et al. 2022)
- Therefore
  - Optimizing InfoNCE loss = running spectral clustering

$H_\pi^k(Z)$ is equivalent to InfoNCE

- $H_\pi^k(Z) = -E_{W_X \sim P(\cdot;\pi)}[\log P(W_Z = W_X; K_Z)]$
- $W_X \sim P(\cdot;\pi)$ means we sample each node with its similarity neighbor in π $\Rightarrow$ Data augmentation step
- For unitary out-deg W, $W_i \sim M\left(1, \frac{\pi_i}{\sum_j \pi_{i,j}}\right)$.
  - Every row i is independent!
  - $\Rightarrow H_\pi^k(Z) = -\sum_i E_{W_{X,i}}\left[\log P\left(W_{Z,i} = W_{X,i}; K_Z\right)\right]$
  - $W_{X,i}$ is i-th row of W_x with single 1 (to j), other entries are 0. Same for $W_{Z,i}$

$H_\pi^k(Z)$ is equivalent to InfoNCE

- $\text{InfoNCE} = -\sum_{i=1}^N \log \frac{\exp(-\|f(X_i)-f(X_{i'})\|^2/2\tau)}{\sum_{j=1}^N \exp\left(-\|f(X_i)-f(X_j)\|^2/2\tau\right)}$
- $\log \frac{\exp(-\|f(X_i)-f(X_{i'})\|^2/2\tau)}{\sum_{j=1}^N \exp\left(-\|f(X_i)-f(X_j)\|^2/2\tau\right)} = -\log \frac{k(Z_i, Z_{i'})}{\|K_{Z,i}\|_1}$
- Let $Q_i = \frac{K_{Z,i}}{\|K_{Z,i}\|_1}$, the distribution of $P(\cdot; K_Z)$
- $\text{InfoNCE} = -\sum_{i=1}^N \log Q_{i,i'}$
- i, i' are sampled in data augmentation, so we are optimizing

$$-\sum_{i=1}^N \sum_{i'=1}^N \Pr\left(W_{x,i,i'} = 1\right) \log Q_{i,i'}$$

$$= -\sum_i E_{W_{X,i}}\left[\log P\left(W_{Z,i} = W_{X,i}; K_Z\right)\right] = H_\pi^k(Z)$$

$H_\pi^k(Z)$ is equivalent to spectral clustering (Van Assel et al. 2022)

- $H_\pi^k(Z) = \min_Z - \sum_{(i,j)\in[n]^2} \overline{W_{i,j}} \log k\left(Z_i - Z_j\right) + \log S(Z)$
  - $\bar{W} = E_{W_X \sim P(\cdot;\pi)}[W_X]$
  - $S(Z) = \sum_W s(Z, W)$, punish solutions like $Z = \mathbf{0}$, as $\mathbf{0}$ is valid for all W, which gives larger $P(Z)$
- Since k is Gaussian, this becomes

$$\min_Z tr(Z^T L(\pi) Z) + \log S(Z)$$

  - Since $E_{W_X \sim P(\cdot;\pi)}[L(W_X)] = L(\pi)$
  - Role of projection head?

Can we replace Gaussian kernel?

- ψ_i is the similarity between the query and the contrastive sample

Minimize the worst case assignment diversity

$$\begin{aligned} \text{(P1)} \quad \max_{\boldsymbol{\alpha}} \quad & H(\boldsymbol{\alpha}) \\ \text{s.t.} \quad & \boldsymbol{\alpha}^{\top}\mathbf{1}_n = 1, \alpha_1, \ldots, \alpha_n \geq 0 \\ & \psi_1 - \sum_{i=1}^{n} \alpha_i \psi_i \leq 0 \end{aligned}$$

Introducing τ as Lagrangian dual variable will give the following

$$: -\tau \log \frac{\exp\left(\frac{1}{\tau}\psi_1\right)}{\sum_{i=1}^{n} \exp\left(\frac{1}{\tau}\psi_i\right)}$$

New Losses with Our Anlysis

- The kernel used in representation space can be changed. We use kernel in expoential family and construct new ones.

Simple Sum Kernel:

$$K(x_i, x_j) := \exp(-\|\boldsymbol{f}(\mathbf{x}_i) - \boldsymbol{f}(\mathbf{x}_j)\|_2^2/\tau_2) + \exp(-\|\boldsymbol{f}(\mathbf{x}_i) - \boldsymbol{f}(\mathbf{x}_j)\|_2^1/\tau_1)$$

Concatenation Sum Kernel:

$$K(x_i, x_j) := \exp(-\|\boldsymbol{f}(\mathbf{x}_i)[0:n] - \boldsymbol{f}(\mathbf{x}_j)[0:n]\|_2^2/\tau_2) + \exp(-\|\boldsymbol{f}(\mathbf{x}_i)[n:2n] - \boldsymbol{f}(\mathbf{x}_j)[n:2n]\|_2^1/\tau_1)$$

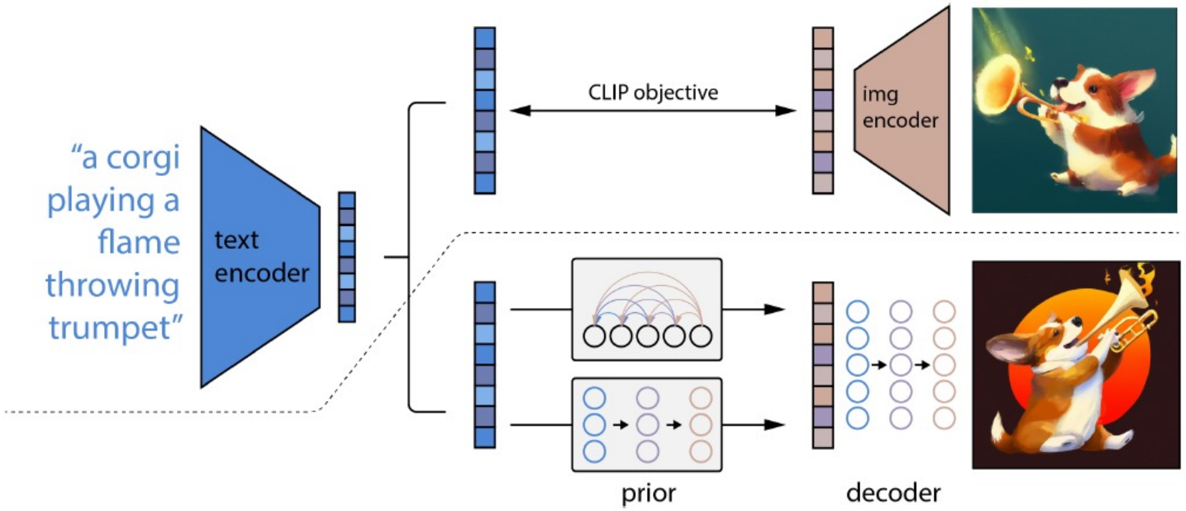
Table 1: Results on CIFAR-10, CIFAR-100, and TinyImageNet datasets.

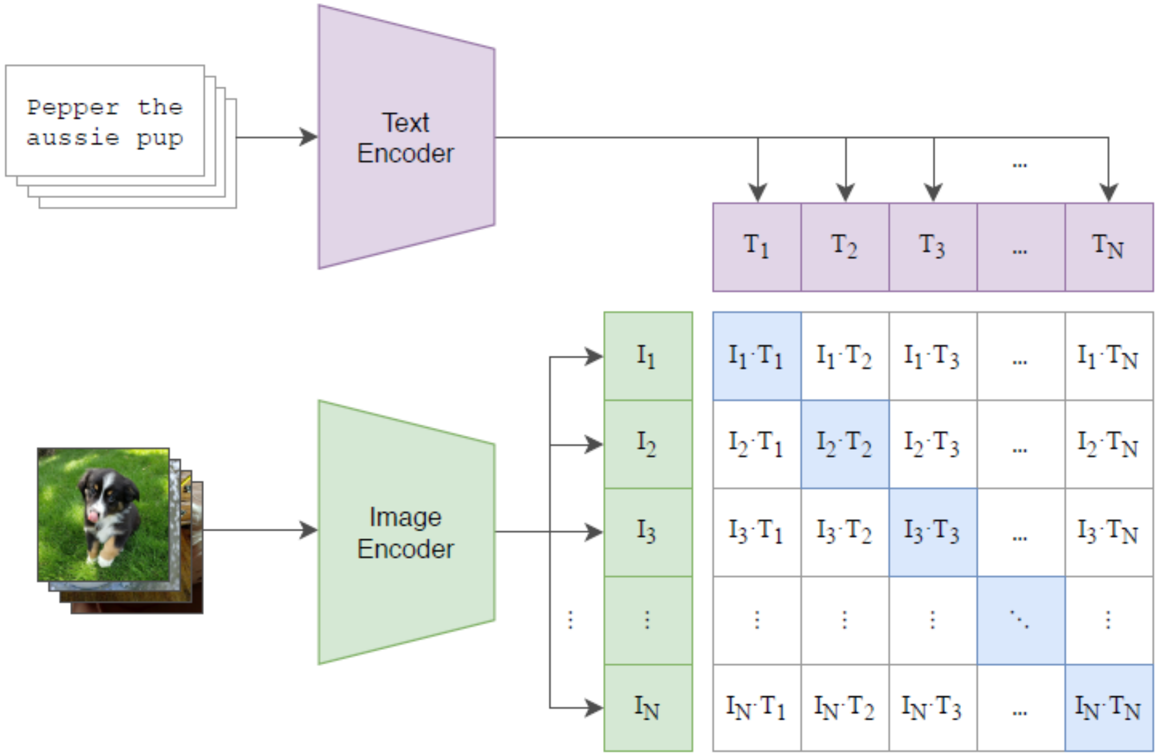
Method	CIFAR-10		CIFAR-100		TinyImageNet	
	200 epochs	400 epochs	200 epochs	400 epochs	200 epochs	400 epochs
SimCLR (repro.)	88.13	90.59	62.67	66.23	34.03	37.86
Laplacian Kernel	89.31	91.05	63.17	66.06	35.92	38.76
$\gamma = 0.5$ Exponential Kernel	89.00	91.23	63.47	65.71	34.21	38.70
Simple Sum Kernel	89.80	**91.76**	**66.73**	**68.62**	**36.60**	**39.38**
Concatenation Sum Kernel	**89.89**	91.28	66.09	68.53	35.92	38.76

Extension to CLIP

- CLIP samples N image-text pairs, and maps every image with its matched text (and vice versa)
  - Using InfoNCE loss
- We prove:
  - CLIP runs spectral clustering on this bipartite graph
- Extension:
  - Explaining LaCLIP

$$\mathcal{L}_I := -\sum_{i=1}^{N} \log \frac{\exp\left(\mathrm{sim}\left(\boldsymbol{f}_I\left(\mathrm{aug}_I\left(x_I^i\right)\right), \boldsymbol{f}_T\left(\mathrm{aug}_T\left(x_T^i\right)\right)\right)/\tau\right)}{\sum_{k=1}^{N} \exp\left(\mathrm{sim}\left(\boldsymbol{f}_I\left(\mathrm{aug}_I\left(x_I^i\right)\right), \boldsymbol{f}_T\left(\mathrm{aug}_T\left(x_T^k\right)\right)\right)/\tau\right)},$$





Thank you!