Your contrastive learning problem is secretly a distribution alignment problem

Zihao Chen[*], **Chi-Heng Lin, Ran Liu, Jingyun Xiao, Eva L. Dyer**[*]
School of Electrical & Computer Engineering
Georgia Tech, Atlanta, GA

Abstract

Despite the success of contrastive learning (CL) in vision and language, its theoretical foundations and mechanisms for building representations remain poorly understood. In this work, we build connections between noise contrastive estimation losses widely used in CL and distribution alignment with entropic optimal transport (OT). This connection allows us to develop a family of different losses and multistep iterative variants for existing CL methods. Intuitively, by using more information from the distribution of latents, our approach allows a more distribution-aware manipulation of the relationships within augmented sample sets. We provide theoretical insights and experimental evidence demonstrating the benefits of our approach for *generalized contrastive alignment*. Through this framework, it is possible to leverage tools in OT to build unbalanced losses to handle noisy views and customize the representation space by changing the constraints on alignment. By reframing contrastive learning as an alignment problem and leveraging existing optimization tools for OT, our work provides new insights and connections between different self-supervised learning models in addition to new tools that can be more easily adapted to incorporate domain knowledge into learning.

1 Introduction

In machine learning, the availability of vast amounts of unlabeled data has created an opportunity to learn meaningful representations without relying on costly labeled datasets [26, 52, 27]. Self-supervised learning has emerged as a powerful solution to this problem, allowing models to leverage the inherent structure in data to build useful representations. Among self-supervised methods, contrastive learning (CL) is widely adopted for its ability to create robust representations by distinguishing between similar (positive) and dissimilar (negative) data pairs. With success in fields like image and language processing [8, 46], contrastive learning now also shows promise in domains where cross-modal, noisy, or structurally complex data make labeling especially challenging [34, 56, 10].

Traditional contrastive learning methods primarily aim to bring positive pairs—often augmentations of the same sample—closer together in representation space. While effective, this approach often struggles with real-world challenges such as noise in views, variations in data quality, or shifts introduced by complex transformations, where positive pairs may not perfectly align. Additionally, in tasks requiring domain generalization, aligning representations across diverse domains (e.g., variations in style or sensor type) is critical but difficult to achieve with standard contrastive learning, which typically lacks mechanisms for incorporating domain-specific relationships. These limitations highlight the need for a more flexible approach that can adapt alignment strategies based on the data structure, allowing for finer control over similarity and dissimilarity among samples.

[*] Contact: {zchen959, evadyer}@gatech.edu

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

To address this challenge, we introduce a novel *generalized contrastive alignment* (GCA) framework, which reinterprets contrastive learning as a distributional alignment problem. Our method allows flexible control over the alignment of samples by defining a target transport plan, $\mathbf{P}_{tgt}$, that serves as a customizable alignment guide. For example, setting $\mathbf{P}_{tgt}$ to resemble a diagonal matrix encourages each positive to align primarily with itself or its augmentations, thereby reducing the effect of noise between views. Alternatively, we can incorporate more complex constraints, such as weighting alignments based on view quality or enforcing partial alignment structures where noise or data heterogeneity is prevalent. This flexibility enables GCA to adapt effectively to a wide range of tasks, from simple twin view alignments to scenarios with noisy or variably aligned data.

Our approach also bridges connections between GCA and established methods, such as InfoNCE (INCE) [38], Robust InfoNCE (RINCE) [12], and BYOL [22], demonstrating that these can be viewed as iterative alignment objectives with Bregman projections [6, 21]. This perspective allows us to systematically analyze and improve uniformity within the latent space, a property that enhances representation quality and ultimately boosts downstream classification performance.

We validate our method through extensive experiments on both image classification and noisy data tasks, demonstrating that GCA's unbalanced OT (UOT) formulations improve classification performance by relaxing our constraints on alignment. Our results show that GCA offers a robust and versatile framework for contrastive learning, providing flexibility and performance gains over existing methods and presenting a promising approach to addressing different sources of variability in self-supervised learning.

The contributions of this work include:

- A new framework called *generalized contrastive alignment* (GCA), which reinterprets standard contrastive learning as a distributional alignment problem, using optimal transport to provide flexible control over alignment objectives. This approach allows us to derive a novel class of contrastive losses and algorithms that adapt effectively to varied data structures and build customizable transport plans.
- We present GCA-UOT, a contrastive learning method that achieves strong performance on standard augmentation regimes and excels in scenarios with more extreme augmentations or data corrupted by transformations. GCA-UOT leverages unbalanced transport to adaptively weight positive alignments, enhancing robustness against view noise and cross-domain variations.
- We provide theoretical guarantees for the convergence of our GCA-based methods and show that our alignment objectives improve representation quality by enhancing the uniformity of negatives and strengthening alignment within positive pairs. This leads to more discriminative and resilient representations, even in challenging data conditions.
- Empirically, we demonstrate the effectiveness of GCA in both image classification and domain generalization tasks. Through flexible, unbalanced OT-based losses, GCA achieves superior classification performance and adapts alignment to include domain-specific information where relevant, without compromising classification accuracy in domain generalization.

2 Background

2.1 Contrastive learning

Contrastive learning (CL) is a representation learning methodology that uses positive and negative pairs to define similarity in the latent space. Let $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ denote our dataset. For each sample $\mathbf{x}_i$ in a batch of training data with size B, we create two augmented copies $\mathbf{x}'_i$ and $\mathbf{x}''_i$ independently, i.e., $\mathbf{x}'_i = \psi(\mathbf{x}_i)$ where ψ is a randomly drawn augmentation function from some augmentation class $\mathcal{A}$ and likewise for $\mathbf{x}''_i$. The $(\mathbf{x}'_i, \mathbf{x}''_i)$ is called a positive pair of $\mathbf{x}_i$ while $(\mathbf{x}'_i, \mathbf{x}''_j)$ is treated as a negative pair for any $j \neq i$. One of the most widely used formulations of the CL problem, InfoNCE (INCE) [8], seeks to maximize the negative log probability that a sample is correctly classified as

$$\mathcal{L}_{\text{INCE}} = -\log\left(\frac{e^{s_{ii}}}{e^{s_{ii}} + \sum_{i \neq j} e^{s_{ij}}}\right), \tag{1}$$

where $s_{ij} = \varepsilon^{-1} f_\theta(\mathbf{x}'_i)^\top f_\theta(\mathbf{x}''_j) / \|f_\theta(\mathbf{x}'_i)\| \|f_\theta(\mathbf{x}''_j)\|$ is the score between augmented samples.

Building upon the principles of INCE, SimCLR [8] and MoCo [24] are two representative works that form the foundation of contrastive learning methods for visual representation tasks. Alternatively, BYOL [22] and SimSiam [9] discard the use of negative samples to avoid large batch size and instead use exponential moving average-based updates to avoid representational collapse. Recent contrastive methods have focused on improving the tolerance to noise in samples to enhance robustness in diverse scenarios [13]. Among them, Robust INCE (RINCE) is a robust contrastive loss function characterized by its symmetric properties and theoretical resistance to noisy labels [47, 12]. Specifically, RINCE provides robustness to noisy views by introducing adjustable parameters λ and q [12] which rebalance the cost of positive and negative views, resulting in the following loss:

$$\mathcal{L}^{\lambda,q}_{\text{RINCE}} = \frac{1}{q}\big(- e^{q s_{ii}} + \lambda^q (e^{s_{ii}} + \sum\nolimits_{i \neq j} e^{s_{ij}})^q \big) \tag{2}$$

By optimizing the above loss functions, the encoder f is trained to construct a semantically coherent representation space where positive pairs of samples are positioned nearby, while those negative pairs with divergent semantic attributes are separated [57].

2.2 Proximal Operators and Projections

To make the connections between different CL losses clearer later, we use the notion of proximal operators. In words, the proximal operator will provide a way to find the closest point in some closed convex set. Formally, we can define the proximal operator as follows.

Definition 1 (Proximal Operator). *Let $d_\Gamma(\mathbf{x}, \mathbf{v}) = \Gamma(\mathbf{x}) - \Gamma(\mathbf{v}) - \langle \nabla\Gamma(\mathbf{v}), \mathbf{x} - \mathbf{v}\rangle$ be a Bregman divergence with a convex function Γ. The proximal operator of $h : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ is defined for a point $\mathbf{v} \in \mathcal{X}$ with a closed convex set $\mathcal{B} \subseteq \mathcal{X}$:*

$$\mathit{Prox}^{d_\Gamma}_{h,\mathcal{B}}(\mathbf{v}) = \arg\min_{\mathbf{x}\in\mathcal{B}} \{h(\mathbf{x}) + d_\Gamma(\mathbf{x}, \mathbf{v})\} .$$

Moreover, we can define the concept of a projection as a special case of the proximal operator when we let $h(\mathbf{x})$ be an indicator function $h_{\mathcal{B}}(x) = \{0, \text{if } x \in \mathcal{B}; \infty, \text{if } x \notin \mathcal{B}\}$ on constraint set $\mathcal{B}$. See Appendix A.2 for more details.

2.3 Solving Optimal Transport Through Proximal Point Methods

Optimal transport (OT) is widely used in characterizing the distance between two collections of samples $\{\mathbf{x}_i\}_{i=1}^B$ and $\{\mathbf{y}_j\}_{j=1}^B$ with associated measures $\mu = \sum_{i=1}^B \delta_{\mathbf{x}_i} p_i$ and $\nu = \sum_{j=1}^B \delta_{\mathbf{y}_j} q_j$ with Dirac delta function $\delta_{\mathbf{x}}$ and $\delta_{\mathbf{y}}$ on finite support [43]. Here, p and q are vertices of the $\mathbb{R}^B$ simplex defined as $\Delta_B := \{v \in \mathbb{R}^B : v_i \geq 0, \sum_{i=1}^B v_i = 1\}$. OT aims to learn a joint coupling matrix, or transport plan $\mathbf{P} \in \mathbb{R}_+^{B\times B}$ that minimizes the cost of transporting mass encoded by cost matrix $\mathbf{C} \in \mathbb{R}_+^{B\times B}$, from one distribution to another. In practice, entropy regularization is used to solve the OT objective, resulting in the following entropy-regularized OT (EOT) objective:

$$\min_{\mathbf{P}\in\mathcal{B}} \langle \mathbf{P}, \mathbf{C}\rangle - \varepsilon H(\mathbf{P}), \quad \text{where } H(\mathbf{P}) = -\sum_{ij} \mathbf{P}_{ij} \log(\mathbf{P}_{ij}), \tag{3}$$

where ε is a user specified parameter that controls the amount of smoothing in the transport plan, and $\mathbf{C}(\mathbf{x}, \mathbf{y}) = 1 - \langle \mathbf{x}, \mathbf{y}\rangle / \|\mathbf{x}\|\|\mathbf{y}\|$ is often set to encode the cosine similarity between pairs of samples.

The Sinkhorn Algorithm and its Interpretation as a Bregman Projection. Solving Equation (3) could be interpreted as iterative alignment problem on a Hilbert space generated from the kernel $\mathbf{K}_{ij} = \exp(-\mathbf{C}_{i,j}/\varepsilon)$. This alignment problem can be solved through iterative Bregman projections onto the two constraints sets that encode the marginals along the rows and columns [3, 5, 43]:

$$C_1^\mu := \{\mathbf{P} : \mathbf{P}\mathbb{1}_B = \mu\}, C_2^\nu := \{\mathbf{P} : \mathbf{P}^\top \mathbb{1}_B = \nu\} \tag{4}$$

The first step of Bregman projection is to find the minimizer $\mathbf{P}^{(1)} = \arg\min\{\varepsilon \text{KL}(\mathbf{P}\|\mathbf{K}) : \mathbf{P}\mathbb{1}_B = \mu\}$ by the proximal operator $\text{Prox}^{\text{KL}}_{C_1^\mu}(\mathbf{K})$ with Lagrange multiplier f on the row constraint set $\mathcal{C}_1^\mu$, and compute its derivatives with respect to $\mathbf{P}$ with $\mathbf{u} = e^{f/\varepsilon} > 0$:

$$\varepsilon \log(\mathbf{P}^{(1)}/\mathbf{K}) - f\mathbb{1} = 0 \Rightarrow \mathbf{P}^{(1)} = \mathbf{u}\mathbf{K}, \quad \langle \mathbf{P}^{(1)}, \mathbb{1}\rangle = \mu \Rightarrow \langle \mathbf{u}\mathbf{K}, \mathbb{1}\rangle = \mu, \mathbf{u} = \frac{\mu}{\mathbf{K}\mathbb{1}} \tag{5}$$

Next, we project $\mathbf{P}^{(1)}$ onto the column constraint set C_2^{ν}, resulting in $\mathbf{P}^{(2)} := \text{Prox}^{\text{KL}}_{C_2^{\nu}}(\mathbf{P}^{(1)}) = \mathbf{P}^{(1)}\text{diag}(\frac{\nu}{\mathbf{P}^{(1)\top}\mathbb{1}_B})$. The iterative updates can be succinctly expressed as the Sinkhorn iterations:

$$\mathbf{P}^{(2t+1)} = \text{diag}(\mathbf{u}^{(t+1)})\mathbf{K}\,\text{diag}(\mathbf{v}^{(t)}), \quad \mathbf{P}^{(2t+2)} = \text{diag}(\mathbf{u}^{(t+1)})\mathbf{K}\,\text{diag}(\mathbf{v}^{(t+1)}), \tag{6}$$

with the scaling vectors $\mathbf{u}^{(t)}$ and $\mathbf{v}^{(t)}$ updated according to:

$$\mathbf{u}^{(t+1)} \overset{\text{def}}{=} \frac{\mu}{\mathbf{K}\mathbf{v}^{(t)}}, \quad \mathbf{v}^{(t+1)} \overset{\text{def}}{=} \frac{\nu}{\mathbf{K}^{\text{T}}\mathbf{u}^{(t)}}. \tag{7}$$

Here, iterations converge to a stable transport plan $\mathbf{P}^{(\infty)}$as the optimal solution of Equation (3), which provides the minimum cost matching between two distributions. The convergence and dynamics of OT and its dual formulation have been studied extensively in [4, 43, 19, 1]. Thus, these results guarantee that the iterates will converge to the optimal solution of the EOT objective, or that $\mathbf{P}^{(t)} \to \mathbf{P}^{(\infty)}$ with $t \to \infty$. See Appendix A.3 for more details on both the continuous and discrete formulations of OT.

2.4 Wasserstein Dependency Measure

The Wasserstein Dependency Measure (WDM) is a measure of deviation between two probability measures. We will use this later and thus provide the formal definition here [39].

Definition 2 (Wasserstein Dependency Measure)**.** *Define the WDM as the Wasserstein distance (W_1) between the joint distribution $\pi(x, y)$ and the product of marginal distributions $\mu \otimes \nu(x, y)$ of two random variables x and y. $W_1(\pi, \mu \otimes \nu) = \sup_{f \in \mathcal{C}(\mathcal{X}\times\mathcal{Y})} \left(\mathbb{E}_{\pi(x,y)}[f(x, y)] - \mathbb{E}_{\mu\otimes\nu(x,y)}[f(x, y)]\right)$, where $\mathcal{C}(\mathcal{X} \times \mathcal{Y})$ denotes the set of all 1-Lipschitz functions from $\mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$.*

2.5 Optimal Transport and Alignment in Representation Learning

Distribution alignment and OT have been widely used for domain adaptation [33, 14, 30, 59], and in generative modeling [2, 55, 49, 58]. The connections between distribution alignment and contrastive learning, however, are still nascent. In [51], the authors explore the connection between inverse OT (IOT) [32, 53, 18] and INCE. Our work builds on this connection to OT to build robust divergences (RINCE) and to build a novel unbalanced optimal transport (UOT) method (Section 3.3). Additionally, we show how our framework can be used to build flexible methods for encouraging contrast at multiple levels. We use this concept of hierarchical contrast and show that it can be used in domain generalization settings (Section 6.2). It is of note that GCA-UOT focuses on relaxing the hard constraints on the row and columns into the soft penalties, which is different with the idea of "unbalanced matching" in [51] which considers the case where the encoders may not have the same weights.

3 Generalized Contrastive Alignment (GCA)

In this section, we will introduce a new framework for *generalized contrastive alignment* and demonstrate the connections between contrastive learning and optimal transport.

3.1 Problem Formulation

Traditional contrastive learning methods focus on bringing positive examples, such as augmentations of the same sample, closer together in representation space. In contrast, our approach reframes contrastive learning as a distributional alignment problem, allowing flexible control over how pairs are matched by imposing specific constraints on the target transport plan, $\mathbf{P}_{tgt}$.

Our objective is to learn an encoder f_θ that minimizes the *transport cost* between positive samples. By defining $\mathbf{P}_{tgt}$ with specific alignment rules, such as domain-specific or hierarchical constraints, we can influence how samples are organized in the latent space. For instance, setting $\mathbf{P}_{tgt}$ to resemble a diagonal matrix encourages each positive to align primarily with itself or its augmentations, minimizing $\text{div}(\mathbf{I}||\mathbf{P}) \approx 0$, where div measures the deviation from an identity matrix (e.g., KL-divergence).

This flexibility allows us to encode more nuanced forms of similarity, adapting to tasks where alignment structure varies based on domain, class, or other high-level constraints. By expanding contrastive learning in this way, our method enhances separation of negatives while addressing complex relational patterns, making it suitable for a wider range of learning tasks.

Defining the Kernel Space. Before formally stating our objective, we first need to define the concept of an augmentation kernel for our positive and negative examples.

Definition 3 (Augmentation Kernel)**.** *Let f_θ denote an encoder with parameters θ and let $(\mathbf{x}'_i, \mathbf{x}''_j) \sim \mathcal{A}$ be two views drawn from the family of augmentations $\mathcal{A}$. The augmentation kernel for the encoder θ is defined as $\mathbf{K}_\theta(\mathbf{x}'_i, \mathbf{x}''_j) = \exp(-dist(\widetilde{f}_\theta(\mathbf{x}'_i), \widetilde{f}_\theta(\mathbf{x}''_j))/\varepsilon)$, where $dist(\cdot)$ can be an arbitrary distance metric, and $\widetilde{f}_\theta(\mathbf{x}'_i)$ is the normalized output of f_θ, and ε is the regularization parameter.*

Main Objective. With this definition in hand, we can now formalize our objective as follows:

$$\min_{\theta} \ d_M\big(\mathbf{P}_{\text{tgt}}||\mathbf{P}_\theta\big), \text{ with } \ \ \mathbf{P}_\theta = \arg\min_{\mathbf{P}\in\mathcal{B}}\{h(\mathbf{P}) + d_\Gamma(\mathbf{P}||\mathbf{K}_\theta)\}, \tag{8}$$

where $\mathbf{K}_\theta$ is the augmentation kernel defined in Definition (3), $h(x)$ is a convex function (typically an indicator function), $\mathcal{B}$ is a closed convex constraint set (i.e. Birkhoff polytope) that defines the constraints of proximal operators, d_Γ is a Bregman divergence that is used to find the nearest points $\mathbf{P}_\theta$ on the constraint set $\mathcal{B}$ of $\mathbf{K}_\theta$, d_M is a convex function (e.g., KL-divergence) that measures divergence between $\mathbf{P}_\theta$ and the target coupling plan $\mathbf{P}_{\text{tgt}}$.

Our objective is a bi-level optimization problem which aims to learn a representation that minimizes the divergence between the transport plan $\mathbf{P}_\theta$ with the target alignment plan $\mathbf{P}_{\text{tgt}}$ that encodes the matching constraints. When we consider a standard contrastive learning setup where we have pairs of positive examples the source and target distribution, then the target $\mathbf{P}_{\text{tgt}}$ is the identity matrix $\mathbf{I}$. However, we will show later that other alignment constraints can be considered. Moreover, when $\mathcal{B}$ is the intersection of more constraint sets like $C_1^\mu \cap C_2^\nu$ in Equation (4), a nature way to get the approximation of the nearest points $\mathbf{P}_\theta$ of $\mathbf{K}_\theta$ is to run iterative projections algorithm [3], which could be extended into the intersection of several constraint sets like $\{\cap_{i=1}^n C_i\}$, resulting in a multi-marginal problem [41].

3.2 A Proximal Point Algorithm for GCA

In practice, we can solve the alignment problem above by iteratively updating the two main components in our bi-level objective. First, for a fixed encoder parameters θ, we obtain the transport coupling $\mathbf{P}_\theta$ through our corresponding proximal operator. Second, we measure the deviation between the transport plan $\mathbf{P}_\theta$ with the target $\mathbf{P}_{\text{tgt}}$ that encodes our matching constraints, which denotes the ideal alignment plan on the intersection of the constraint sets. We provide pseudocode for this iterative approach in Algorithm 1, which we refer to as generalized contrastive alignment or GCA. The implementation of our methods is in https://github.com/nerdslab/gca.

Algorithm 1 Proximal-Point Algorithm for Generalized Contrastive Alignment (GCA)

```
1: Initialization: Initial encoder parameters $\theta$, target transport plan $\mathbf{P}_{\text{tgt}}$, kernel function $\mathbf{K}_\theta$, the
   function $h(x)$, divergences $d_\Gamma$ and $d_M$ (KL or $W_1$). Initialize transport plan $\mathbf{P}_\theta$ based on $\theta$.
2: Compute the transport coupling $\mathbf{P}_\theta$: Update $\mathbf{P}_\theta$ using the proximal operator scaling for fixed
   $\theta$ as described in Eq. (8):
       $\mathbf{P}_\theta = \arg\min_{\mathbf{P}\in\mathcal{B}}\{h(\mathbf{P}) + d_\Gamma(\mathbf{P}||\mathbf{K}_\theta)\}.$
3: Calculate the loss: Calculate deviation between the target and current transport plans
       $\mathcal{L}_{GCA} = d_M(\mathbf{P}_\theta, \mathbf{P}_{\text{tgt}}).$
   Update networks $f_\theta$ (encoder) and $g_\theta$ (projector) to minimize $\mathcal{L}_{GCA}$.
4: Repeat until convergence: Repeat steps 2 and 3 until convergence.
```

Computing the transport coupling $\mathbf{P}_\theta$ [2] (forward-pass) in GCA algorithms could be treated as a specific type of Dykstra's projection algorithms [5], which computes the **iterative projection** on the intersection of affine convex sets [3, 42]. The proofs of convergence are provided in Appendix B.1.

[2] With a single constraint set like C_1^μ in Equation (4), computing the proximal point only involves a single projection. However, if there are intersecting constraint sets like $C_1^\mu \cap C_2^\nu$, solving for the proximal point requires multiple projections before we approach the nearest point on their intersection.

3.3 GCA-UOT Method

We can also benefit from the rich literature on optimal transport to build different relaxations of our objective [43, 7, 54, 33, 35]. In particular, we choose to leverage a formulation of *unbalanced optimal transport* (UOT) to further relax the marginal constraints [11] in our objective.

In this case, we can add the dual form of d_Γ to the Equation (8) and reformulate our objective as:

$$\min_{\theta} \; d_M(\mathbf{P}_{\text{tgt}}||\mathbf{P}_\theta) + \lambda_\mathbf{1}\mathbf{h}_{\mathcal{F}}(\mathbf{P}_\theta \mathbb{1}||\mu) + \lambda_\mathbf{2}\mathbf{h}_{\mathcal{G}}(\mathbf{P}_\theta^\top \mathbb{1}||\nu) + \varepsilon \mathbf{H}(\mathbf{P}_\theta). \tag{9}$$

Here $h_{\mathcal{F}}$ and $h_{\mathcal{G}}$ can be different divergence measures (e.g., KL divergence) that penalize deviations from the desired marginals μ and ν, and λ_1 and λ_2 are regularization parameters that control the trade-off between the transport cost and the divergence penalties. This relaxation leads to different types of proximal operators which we outline in Appendix B.2. The impact of the entropy regularization parameter ε on the coupling matrix is studied in Figure A5, along with the number of iterations and corresponding sensitivity is provided in Figure A6.

3.4 Modifying the Target Transport Plan to Encode Matching Constraints

Contrastive learning objectives can be cast as a minimization of the deviations between the transport plan $\mathbf{P}_\theta$ and the identity matrix, i.e., $\mathbf{P}_{tgt} = \mathbf{I}$. However, our GCA formulation enables learning representations that extend beyond this one-to-one matching constraint. This flexibility allows us to incorporate additional matching constraints informed by domain-specific knowledge. For example, in domain generalization scenarios [23, 28], where each batch contains samples from multiple domains, the target alignment plan can be structured as:

$$\mathbf{P}_{\text{tgt}}[i,j] = \mathbf{I}[i,j] + \alpha \cdot \mathbb{I}(D_i = D_j,\, i \neq j) + \beta \cdot \mathbb{I}(D_i \neq D_j,\, i \neq j),$$

Where $\mathbb{I}(\cdot)$ is the indicator function, which equals 1 if the condition inside is true and 0 otherwise. D_i represents the domain of sample i, where $\alpha \geq 0$ and $\beta \geq 0$. In this case, we can improve the representation by building the block constraints which encode either class information (in supervised setting) or domain information (in across domain generalization, visualized in Figure 1).

3.5 Computational Complexity

The forward-pass only involves the scaling operations in Equation (7) and doesn't affect the complexity of the backward-pass. Therefore, GCA methods can be thought of as a form of batch normalization operations with adaptive scaling. An analysis of the complexity is provided along with experiments in Appendix B.1. Our results show that GCA iterations only slightly increase the computational complexity when compared with their single step equivalent (GCA-INCE vs. INCE). However, we found that GCA-UOT is faster than INCE due to the improved symmetry and smoothness of the loss. Moreover, we record the floating point operations per second (Flops) of running GCA methods. We find that GCA-INCE (6.65 MFlops) has 5% more Flops than INCE (6.31 MFlops), while GCA-UOT saves 30% Flops (4.54 MFlops). These results show that our GCA-UOT method is not only superior in terms of accuracy but also in speed.

4 Building Connections to Different CL Objectives

In this section, we show how the modification of the different parts of our main objective ($d_\Gamma, d_M, \mathcal{B}, \mathbf{K}_\theta$) in Equation (8) can be connected to different contrastive losses. See Table 1 for a summary of how different losses can be mapped back to our formulation.

Table 1: *Comparison of different contrastive alignment objectives.* Here we have C_1^μ and C_2^ν as constraint sets (denoted as $\mathcal{B}$) defined in Equation (4) with their corresponding indicator function. "Iter" refers to iterative methods.

Methods	d_M	d_Γ	$\mathcal{B}$	Iter
INCE	KL	KL	C_1^μ	
GCA-INCE	KL	KL	$C_1^\mu \cap C_2^\nu$	✓
RINCE (q=1)	W1	KL	C_1^μ	
GCA-RINCE (q=1)	W1	KL	$C_1^\mu \cap C_2^\nu$	✓
BYOL	KL	L2	$R^{B\times B}$	

4.1 Connection to INCE

An interesting connection that we can make between GCA main objective and contrastive learning is that we can interpret INCE as a **single step** in a iterative GCA objective [51]. This connection can be further summarized through the following theorem.

Theorem 1 (INCE Equivalence)**.** *Let $\mathbf{K}_\theta$ denote the augmentation kernel as in Definition (3) with cosine similarity, d_Γ and d_M equal to KL-divergence, and constraint set as C_1^μ in Equation (4). The INCE objective in Equation (1) can be re-expressed as a GCA problem in Equation (8) as follows:*

$$\min_\theta KL\big(\mathbf{I}||Prox^{KL}_{C_1^\mu}(\mathbf{K}_\theta)\big). \tag{10}$$

The proof is contained in Appendix B.3. Theorem (1) shows that the INCE loss can be viewed as solving the matching problems in Equation (3) with row normalization constraints C_1^μ. This connection between GCA and INCE allows us to derive the iterative algorithm for GCA-INCE by running Bregman projection iteratively on both row and column normalization sets.

4.2 Connection to RINCE

We introduce the following result to build the connection between our framework and RINCE [12].

Theorem 2 (RINCE Equivalence)**.** *Let $\mathbf{K}_\theta$ denote the augmentation kernel as in Definition (3). Set target plan $\mathbf{P}_{tgt} = \mathbf{I}$, d_Γ equal to the KL-divergence, $d_M(\mathbf{I}\|\mathbf{P}) = -\frac{1}{q}(\frac{\mathrm{diag}(\mathbf{P}_\theta)}{\mathbf{u}})^q + \big(\frac{\lambda \mathbf{I}}{\mathbf{u}}\big)^q$ with λ, q, and $\mathbf{u} = \mathrm{diag}\left(\frac{\mu}{\mathbf{P}^{(0)}\mathbb{1}}\right)$, and constraint set C_1^μ defined in Equation (4). The RINCE objective in Equation (2) can be re-expressed as a GCA problem as follows:*

$$\min_\theta d_M(\mathbf{I}\|\mathbf{P}_\theta), \quad with \;\; \mathbf{P}_\theta = Prox^{KL}_{C_1^\mu}(\mathbf{K}_\theta), \tag{11}$$

The proof is provided in Appendix B.4.1. As we can see, RINCE introduces adjustable parameters q and λ, with λ controlling the weight of negative samples, while $q \in (0, 1]$ serves to switch between KL divergence and Wasserstein discrepancy. When $q = 1$, we have the following theorem:

Theorem 3 (W1 Equivalence)**.** *Let $\mathbf{K}_\theta$ denote the augmentation kernel as in Definition (3) with cosine similarity. Set target plan $\mathbf{P}_{tgt} = \mathbf{I}$, d_Γ equal to the KL-divergence, d_M equal to the 1-Wasserstein distance (W_1) in Definition (2), and the constraint set as C_1^μ defined in Equation (4). The RINCE object in Equation (2) with $q = 1$ can be re-expressed as a GCA problem as follows:*

$$\min_\theta W_1\big(\mathbf{P}_{tgt}||Prox^{KL}_{C_1^\mu}(\mathbf{K}_\theta)\big). \tag{12}$$

See Appendix B.5 for the proof. This connection to RINCE suggests an extended iterative formulation to calculate the coupling plan as the projection point $\mathbf{P}^{(\infty)} = \mathrm{Prox}^{\mathrm{KL}}_{C_1^\mu \cap C_2^\nu}(\mathbf{K}_\theta)$ of $\mathbf{K}_\theta$ on the constraint set $C_1^\mu \cap C_2^\nu$. In this case, we can write an iterative algorithm for robust alignment called GCA-RINCE as follows:

$$L^{\lambda,q}_{\text{GCA-RINCE}} = \min_\theta -q^{-1}(\mathrm{diag}(\mathbf{P}_\theta^{(2t-1)})/\mathbf{u}^{(t)})^q + q^{-1}(\lambda \mathbf{P}_{tgt}/\mathbf{u}^{(t)})^q, \tag{13}$$

where λ and q are hyperparameters, $\mathbf{P}^{(1)} := \mathrm{diag}(\mathbf{u}^{(1)})\mathbf{K}_\theta\,\mathrm{diag}(\mathbf{v}^{(0)})$, and t is the number of iterations.

4.3 Connection to BYOL

Our framework also allows us to make connections to BYOL [22]. BYOL learns by encouraging similarity between positive image pairs, without explicitly conditioning on negative examples. To build this connection, recall that BYOL has the online network parameterized by θ and target network parameterized by ξ, where $\mathbf{z}'_\theta = \widetilde{f}_\theta(\mathbf{x}')$ and $\mathbf{z}''_\xi = \widetilde{f}_\xi(\mathbf{x}'')$ are the normalized outputs of the online and target networks, respectively. A simplified version of the BYOL loss can be written as: $L_{\mathrm{BYOL}} = \|\widetilde{q}_\theta(\mathbf{z}'_\theta) - \mathbf{z}''_\xi\|_2^2$, where $\widetilde{q}_\theta(\mathbf{z}'_\theta)$ is the normalized output after online network and q_θ is the predictor.[3] In this case, we can provide the following connection between GCAand BYOL as follows.

Theorem 4 (BYOL Equivalence)**.** *Let $\mathbf{S}_\theta(\mathbf{x}'_i, \mathbf{x}''_j) = \exp(-\|\widetilde{q}_\theta(\mathbf{z}'_i) - \mathbf{z}''_j\|)$ denote the augmentation kernel. Set the target plan $\mathbf{P}_{tgt} = \mathbf{I}$, d_Γ equal to the L2-distance, d_M equal to the KL-divergence, and constraint set as $R^{B\times B}$. The BYOL objective can be re-expressed as a GCA problem as follows:*

$$\min_\theta KL\big(\mathbf{I}||\mathbf{S}_\theta\big), \quad with \;\; \mathbf{S}_\theta = Prox^{\|\cdot\|}_{R^{B\times B}}(\mathbf{S}_\theta). \tag{14}$$

See the proof in Appendix B.6.

[3] In practice, BYOL also switches the order of views to symmetrize the loss. For ease of discussion, we consider just one pair of views but the same could be argued for the full symmetric version.

5 Theoretical Analysis

In this section, we aim to show how the GCA-methods can improve alignment and uniformity in the latent space [57]. Here, *alignment* means that the features of the positive samples are as close as possible, while *uniformity* means that the features of negative samples are uniformly distributed on latent space (see Appendix C.1 for formal definitions). These quantities have been studied in a number of related works [57, 45], where one can show that improved alignment and uniformity can lead to different benefits in representation learning.

5.1 Improved alignment with GCA

Contrastive learning minimizes the deviation between the target alignment plan with the transport plan in Definition 3 through empirical risk minimization (ERM). Therefore, a tighter bound on the empirical risk corresponds to a smaller difference between the ideal alignment with the coupling matrix. We show that this in turn leads to better alignment of the positive views.

Analysis of INCE vs GCA-INCE. GCA-INCE ensures that the final transport plan $\mathbf{P}^{(\infty)}$ is closer to the ideal identity matrix compared to the INCE, as we show in the following theorem.

Theorem 5 (Improved Alignment with INCE)**.** *Let $\mathbf{K}_\theta$ denote the augmentation kernel as in Definition (3). Set d_M and d_Γ to the KL-divergence, and $\mathbf{P}_{tgt} = \mathbf{I}$. The GCA-INCE loss with converged plan $\mathbf{P}_\theta^{(\infty)}$ is lower than the GCA-INCE loss with $\mathbf{P}_\theta^{(t)}$ in Equation (6) for all t.*

The full proof is provided in Appendix C.1.1. The above theorem tells us that solving Equation (8) with iterative projection will converge to a transport plans $\mathbf{P}_\theta^{(\infty)}$ with lower KL divergence than the one-step solution provided by INCE. We can establish the convergence of the $\mathbf{P}^{(t)} \to \mathbf{P}^{(\infty)}$, based on the convergence of Bregman projection.

Analysis of RINCE vs GCA-RINCE.

GCA also benefits from other Bregman divergences, like the WDM in RINCE, which provides robustness against distribution shift compared to the KL-divergence in INCE. GCA-RINCE provides a lower bound on the RINCE loss in Equation (2), which allows us to develop a tighter bound with $\mathbf{P}^{(\infty)}$ obtained by several proximal steps with GCA.

Theorem 6 (Improved Alignment with RINCE)**.** *GCA-RINCE loss with $\mathbf{P}_\theta^{(t)}$ in Equation (13) is lower than the loss in the Theorem (2) as $L_{GCA\text{-}RINCE}^{\lambda,q=1}(\mathbf{P}_\theta^{(t)}) \leq L_{RINCE}^{\lambda,q=1}(\mathbf{P}_\theta^{(1)})$.*

See Appendix C.1.1 for the full proof and an analysis of GCA methods for different choices of d_M.

5.2 Improved Uniformity of Representations Through GCA

The improved alignment of GCA-methods comes from maximization of the uniformity under the constraint of intersection $C_1^\mu \cap C_2^\nu$ in Equation (4), rather than the constraint set C_1^μ in INCE (see Table 1). Finding the projection of $\mathbf{K}_\theta$ on set of $C_1^\mu \cap C_2^\nu$ through proximal steps is equivalent to solving the dual problem of EOT, which can be summarized through the following theorem.

Theorem 7 (Improved Uniformity)**.** *Given the constraint sets in Equation (4), the optimal transport coupling upon convergence of Equation (6), denoted as $\mathbf{P}^{(\infty)}$, achieves a higher uniformity loss compared to the single-step transport plan $\mathbf{P}^{(1)}$ obtained by INCE.*

The proof is provided in the Appendix C.2. Through loss propagation, we show that the alignment plan offered by $\mathbf{P}^{(\infty)}$ will guide the subsequent iterations towards more uniform representations.

5.3 Impacts of GCA on a downstream classification task

We take this one step further and examine the impact of GCA on a downstream classification task. For a classification task, using a labeled dataset $\mathcal{D} = \{(\bar{\mathbf{x}}_i, \mathbf{y}_i)\} \in \bar{\mathcal{X}} \times \mathcal{Y}$ where $\mathcal{Y} = [1, \ldots, M]$ with M classes, we consider a fixed, pre-trained encoder $f_\theta \in \mathcal{F} : \mathcal{X} \to \mathcal{S}$. Assume that positive and negative views of n original samples $(\bar{\mathbf{x}}_i)_{i \in [1..n]} \subset \bar{\mathcal{X}}$ are sampled from the data distribution $p(\bar{\mathbf{x}})$.

In this case, the uniformity loss is equivalent to optimizing the downstream supervised classification tasks with cross-entropy (CE) loss when the following two assumptions are satisfied [16].

	Standard Setting				Noisy Setting			
Method	CIFAR-10	CIFAR-100	SVHN	ImageNet100	CIFAR-10 (Ex)	CIFAR-100 (Ex)	CIFAR-10C	CIFAR-10C (Ex)
INCE	92.01 ± 0.40	71.09 ± 0.31	92.42 ± 0.24	73.01 ± 0.61	82.03 ± 0.32	62.54 ± 0.20	81.52 ± 1.04	83.28 ± 0.25
GCA-INCE	93.02 ± 0.19	71.55 ± 0.12	92.64 ± 0.26	73.04 ± 0.76	82.18 ± 0.69	62.65 ± 0.17	82.63 ± 0.28	82.72 ± 0.27
Δ	+1.01	+0.46	+0.22	+0.03	+0.15	+0.11	+1.11	-0.56
RINCE	93.27 ± 0.20	71.63 ± 0.36	93.26 ± 0.15	71.91 ± 0.43	82.60 ± 0.63	63.55 ± 0.14	82.86 ± 0.21	83.64 ± 0.26
GCA-RINCE	93.47 ± 0.30	71.95 ± 0.48	93.57 ± 0.26	73.44 ± 0.55	82.76 ± 0.49	63.14 ± 0.41	82.87 ± 0.11	**83.69 ± 0.16**
Δ	+0.20	+0.32	+0.31	+1.53	+0.16	-0.47	+0.01	+0.05
SimCLR	92.07 ± 0.90	70.85 ± 0.50	92.13 ± 0.34	72.20 ± 0.78	81.87 ± 0.53	62.94 ± 0.13	81.74 ± 1.54	83.25 ± 0.18
BYOL	90.56 ± 0.59	69.75 ± 0.37	89.50 ± 0.46	69.75 ± 0.83	81.55 ± 0.50	62.11 ± 0.25	82.43 ± 0.06	70.09 ± 0.34
IOT [51]	92.09 ± 0.22	68.37 ± 0.42	92.25 ± 0.19	72.27 ± 0.53	80.59 ± 0.64	62.69 ± 0.34	82.01 ± 0.80	81.57 ± 0.83
IOT-uni [51]	91.49 ± 0.11	68.62 ± 0.35	92.33 ± 0.11	72.88 ± 0.71	80.79 ± 0.24	62.56 ± 0.22	81.19 ± 1.12	81.82 ± 0.51
GCA-UOT	**93.50 ± 0.31**	**72.16 ± 0.38**	**93.82 ± 0.17**	**74.09 ± 0.40**	**83.18 ± 0.44**	**63.62 ± 0.27**	**82.90 ± 0.50**	83.64 ± 0.19

Table 2: *Test accuracy (%) on a downstream classification task after pretraining.* Results are shown for CIFAR-10 (ResNet18), CIFAR-100 (ResNet18), SVHN (ResNet18), and ImageNet100 (ResNet50) under standard and extreme (Ex) augmentation conditions (averaged over 5 seeds). The **CIFAR-100** column is now updated with the latest values from Table A3. The top model is in bold and the second-place model is underlined. For INCE and RINCE, we also provide the improvement Δ by adding GCA to each method.

Assumption 1 (Expressivity of the Encoder)**.** *Let us define $\mathcal{H}_{\bar{\mathcal{X}}}$ is the RKHS associated with the kernel $\mathbf{K}_{\bar{\mathcal{X}}}$ defined on $\bar{\mathcal{X}}$, and $(\mathcal{H}_{f_\theta}, \mathbf{K}_\theta)$ defined on $\mathcal{X}$ with augmentation kernel $\mathbf{K}_\theta = \langle f_\theta(\cdot), f_\theta(\cdot)\rangle_{\mathbb{R}^d}$ in Definition 3. And we assume that $\forall g \in \mathcal{H}_{f_\theta},\ \mathbb{E}_{\mathcal{A}(x|\cdot)} g(x) \in \mathcal{H}_{\bar{\mathcal{X}}}$.*

Assumption 2 (Small Intra-Class Variance)**.** *For $y \neq y'$, the intra-class variance δ_i, δ_j are negligible compared to the distance among different class centroids, $\mu_y, \mu_{y'}$ as $\|\mu_y - \mu_{y'}\| \gg \|\delta_i - \delta_j\|$.*

Claim 1. *If Assumption 1 and Assumption 2 hold, then maximizing the uniformity is equivalent to minimizing the downstream CE loss.*

The proof is provided in Appendix C.2. Optimizing the self-supervised loss under ideal conditions improves downstream CE tasks and helps to explain why maximizing uniformity aids classification.

Remark.. Maximizing uniformity can enhance downstream classification but risks "feature suppression" by encouraging shortcut features that harm generalization [48]. In GCA-UOT, adding penalties modifies the transport plan from that of a pure uniformity loss, helping to avoid feature suppression. We find empirical evidence that UOT provides a more robust transport plan which appears to circumvent some of these shortcut features from being learned (Figure A4 in Appendix C.3).

6 Experiments

In this section, we conduct empirical evaluations to study the performance of our approach in both handling noisy and corrupted views and in domain generalization tasks.

6.1 Comparison with CL Baselines

Experiment setup. To examine the robustness of our framework, we trained INCE and RINCE as baselines, and developed their GCA-based alternatives (+GCA). In addition, we also compared with our novel GCA-UOT method, two variants of IOT established in [51], and other CL baselines, including BYOL and SimCLR. For experiments with SVHN [36] and ImageNet100 [15] we use the ResNet-50 encoder as the backbone and use a ResNet-18 encoder as the backbone for CIFAR-10, CIFAR-100 [29] and a corrupted version of CIFAR called CIFAR-10C [25].

In all of these cases, we follow the standard self-supervised learning evaluation protocol [8], where we train the encoder on the training set in an unsupervised manner and then train a linear layer on top of the frozen representations to obtain the final accuracy on the test set. In addition to standard data augmentation policies commonly used [12], we also apply three different extreme augmentation policies to examine the robustness of GCA towards noisy views (details in Appendix D.2). Learning rates and other training details for CIFAR-10, CIFAR-100, SVHN, and ImageNet100 are provided in Appendix D.1, while specific training details for CIFAR-10C are included in Appendix D.2.

Results on Standard Augmentations. First, we performed experiments on CIFAR-10, CIFAR-100, SVHN, and ImageNet100 using standard sets of augmentations that are applied to achieve state-of-the-art performance (Table 2, Standard Setting). We found the +GCA versions of INCE and RINCE exhibit performance gains in almost all settings except for SVHN, with bigger gains observed when adding GCA to RINCE. Additionally, we find that our unbalanced OT method, GCA-UOT, achieves the top performance across the board, on all four datasets tested. The transport plans obtained by each methods are provided in Figure A4 along with a study of the sensitivity of the methods to hyperparameters (Appendix A7).

Results on Corrupted Data and Extreme Augmentations. Next, we tested the methods in two noisy settings. In the first set of experiments, we apply extreme augmentations to CIFAR-10 (Ex) and CIFAR-100 (Ex) (see Appendix D.2) to introduce noisy views during training. In the second set of experiments, we used the CIFAR-10C to further test the ability of our method to work in noisy settings.

Our experimental results demonstrate that the GCA-based strategy effectively enhances the model's generalization ability and adaptability to aggressive data augmentations. In addition to improving classification accuracy, the GCA-based methods also improve the representational alignment and uniformity, as shown in Appendix E.2. This observation is in line with our theoretical analysis in Section 5.2, where we show that the obtained representations provide better overall alignment of positive views and better spread in terms of uniformity [57].

6.2 Block Diagonal Transport in Domain Generalization

In a final experiment, we aimed to demonstrate the flexibility and robustness of our framework by applying it to a domain generalization task, where samples originate from different domains (e.g., Photo, Cartoon, Sketch, Art). We explored the effects of introducing domain-specific alignment constraints in our transport plan, hypothesizing that this could enhance the latent space organization to capture more nuanced domain similarities.

Our approach enables additional contextual information to be seamlessly integrated into the transport process. In this case, domain information was incorporated to distinguish the alignment of samples from the same versus different domains. To achieve this, we adjusted the target transport plan $\mathbf{P}_{tgt}$, selectively modifying parameters (α, β) to vary the influence of domain-based alignment constraints as shown in Figure 1(A). Specifically, we set $\{\alpha = 0, \beta > 0\}$ to prioritize cross-domain alignment and $\{\alpha > 0, \beta = 0\}$ to focus on intra-domain alignment.

The training was conducted on the PACS dataset [31] using a ResNet-18 encoder with the GCA-INCE objective. After training the encoder in an unsupervised manner, we freeze the encoder and then train a linear readout layer to predict either the sample's class or the domain it belonged to. This setup allowed us to isolate the effect of our transport adjustments on the latent space's capacity to encode both class and domain information.

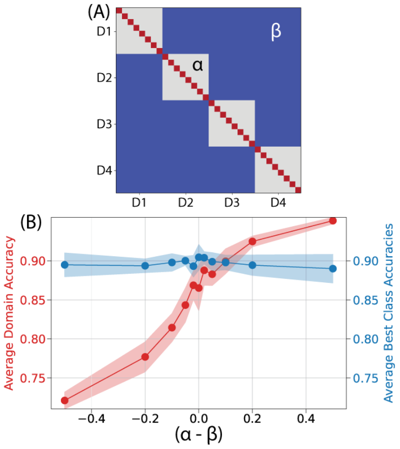

Figure 1: *Incorporating different priors into learning across multiple domains.* (A) Example target alignment plan $\mathbf{P}_{tgt}$, where the target over all samples from the same domain are set to α, the diagonal values are set to 1, and across-domain samples are set to β. (B) The domain classification accuracy (red) and overall class accuracy (blue) with $(\alpha - \beta)$ increases.

The results, displayed in Figure 1(B), revealed that increasing the domain alignment weight enhances the accuracy of domain classification (from 72.11% to 95.16%) without diminishing classification performance. This outcome suggests that GCA can effectively encode both domain and class information in a single latent representation. The ability to adjust alignment constraints provides a powerful tool for domain generalization tasks, enabling multiple types of similarity to be jointly encoded. This flexibility can potentially alleviate issues related to information loss from data augmentation, especially in fine-grained classification settings, by retaining essential domain-specific characteristics across transformations.

7 Conclusion

In this work, we introduced *generalized contrastive alignment* (GCA), a flexible framework that redefines contrastive learning as a distributional alignment problem using optimal transport to control alignment. By allowing targeted control over alignment objectives, GCA demonstrates strong performance across both standard and challenging settings, such as noisy views and domain generalization tasks. This work opens up broader possibilities for learning robust representations in real-world scenarios, where data is often diverse, noisy, or comes from multiple domains.

Future work includes applications of GCA to graphs and time series data, as well as multi-modal settings where our approach can integrate various forms of similarity. As alignment strategies become integral to contrastive learning, GCA offers a promising foundation for more adaptive and expressive self-supervised models.

Acknowledgements

We would like to thank Mehdi Azabou, Divyansha, Vinam Arora, Shivashriganesh Mahato, and Ian Knight for their valuable feedback on the work. This work was funded through NSF IIS-2212182, NSF IIS-2039741, and the support from the Canadian Institute for Advanced Research (CIFAR). We would also like to acknowledge the use of ChatGPT for providing useful feedback and suggestions on the writing of the paper.

References

[1] Dongsheng An, Na Lei, Xiaoyin Xu, and Xianfeng Gu. Efficient optimal transport algorithm by accelerated gradient descent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10119–10128, 2022.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[3] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[4] Robert J Berman. The sinkhorn algorithm, parabolic optimal transport and geometric monge–ampère equations. *Numerische Mathematik*, 145(4):771–836, 2020.

[5] Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

[6] Xing-Ju Cai, Ke Guo, Fan Jiang, Kai Wang, Zhong-Ming Wu, and De-Ren Han. The developments of proximal point algorithms. *Journal of the Operations Research Society of China*, 10(2):197–239, 2022.

[7] Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin. Wasserstein contrastive representation distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16296–16305, 2021.

[8] 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. PMLR, 2020.

[9] 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.

[10] Zining Chen, Weiqiu Wang, Zhicheng Zhao, Fei Su, Aidong Men, and Yuan Dong. Instance paradigm contrastive learning for domain generalization. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(2):1032–1042, 2023.

[11] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018.

[12] Ching-Yao Chuang, R Devon Hjelm, Xin Wang, Vibhav Vineet, Neel Joshi, Antonio Torralba, Stefanie Jegelka, and Yale Song. Robust contrastive learning against noisy views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16670–16681, 2022.

[13] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

[14] Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*, pages 274–289. Springer, 2014.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[16] Benoit Dufumier, Carlo Alberto Barbano, Robin Louiset, Edouard Duchesnay, and Pietro Gori. Integrating prior knowledge in contrastive learning with kernel. In *International Conference on Machine Learning*, pages 8851–8878. PMLR, 2023.

[17] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, Florian Bernard, and Daniel Cremers. A unified framework for implicit sinkhorn differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 509–518, 2022.

[18] Zhenghan Fang, Sam Buchanan, and Jeremias Sulam. What's in a prior? learned proximal networks for inverse problems. *arXiv preprint arXiv:2310.14344*, 2023.

[19] Promit Ghosal and Marcel Nutz. On the convergence rate of sinkhorn's algorithm. *arXiv preprint arXiv:2212.06000*, 2022.

[20] Aritra Ghosh, Naresh Manwani, and PS Sastry. Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107, 2015.

[21] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, 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. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[23] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[24] 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.

[25] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[26] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[27] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

[28] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9619–9628, 2021.

[29] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[30] John Lee, Max Dabagia, Eva Dyer, and Christopher Rozell. Hierarchical optimal transport for multimodal distribution alignment. *Advances in neural information processing systems*, 32, 2019.

[31] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[32] Ruilin Li, Xiaojing Ye, Haomin Zhou, and Hongyuan Zha. Learning to match via inverse optimal transport. *Journal of machine learning research*, 20(80):1–37, 2019.

[33] Chi-Heng Lin, Mehdi Azabou, and Eva L Dyer. Making transport more robust and interpretable by moving data through a small number of anchor points. *Proceedings of machine learning research*, 139:6631, 2021.

[34] Ran Liu, Mehdi Azabou, Max Dabagia, Chi-Heng Lin, Mohammad Gheshlaghi Azar, Keith Hengen, Michal Valko, and Eva Dyer. Drop, swap, and generate: A self-supervised approach for generating neural activity. *Advances in neural information processing systems*, 34:10587–10599, 2021.

[35] Eduardo Fernandes Montesuma, Fred Ngole Mboula, and Antoine Souloumiac. Recent advances in optimal transport for machine learning. *arXiv preprint arXiv:2306.16156*, 2023.

[36] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, volume 2011, 2011.

[37] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[38] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[39] Sherjil Ozair, Corey Lynch, Yoshua Bengio, Aaron Van den Oord, Sergey Levine, and Pierre Sermanet. Wasserstein dependency measure for representation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[40] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.

[41] Brendan Pass. Multi-marginal optimal transport: theory and applications. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1771–1790, 2015.

[42] Gabriel Peyré. Entropic approximation of wasserstein gradient flows. *SIAM Journal on Imaging Sciences*, 8(4):2323–2351, 2015.

[43] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[44] Khiem Pham, Khang Le, Nhat Ho, Tung Pham, and Hung Bui. On unbalanced optimal transport: An analysis of sinkhorn algorithm. In *International Conference on Machine Learning*, pages 7673–7682. PMLR, 2020.

[45] Shi Pu, Kaili Zhao, and Mao Zheng. Alignment-uniformity aware representation learning for zero-shot video classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19968–19977, 2022.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[47] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

[48] Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? *Advances in neural information processing systems*, 34:4974–4986, 2021.

[49] Litu Rout, Alexander Korotin, and Evgeny Burnaev. Generative modeling with optimal transport maps. *arXiv preprint arXiv:2110.02999*, 2021.

[50] Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. In *International Conference on Machine Learning*, pages 19250–19286. PMLR, 2022.

[51] Liangliang Shi, Gu Zhang, Haoyu Zhen, Jintao Fan, and Junchi Yan. Understanding and generalizing contrastive learning from the inverse optimal transport perspective. In *International Conference on Machine Learning*, pages 31408–31421. PMLR, 2023.

[52] Saeed Shurrab and Rehab Duwairi. Self-supervised learning methods and applications in medical imaging analysis: A survey. *PeerJ Computer Science*, 8:e1045, 2022.

[53] Andrew M Stuart and Marie-Therese Wolfram. Inverse optimal transport. *SIAM Journal on Applied Mathematics*, 80(1):599–619, 2020.

[54] Fariborz Taherkhani, Ali Dabouei, Sobhan Soleymani, Jeremy Dawson, and Nasser M Nasrabadi. Self-supervised wasserstein pseudo-labeling for semi-supervised image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12267–12277, 2021.

[55] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[56] Ankit Vishnubhotla, Charlotte Loh, Akash Srivastava, Liam Paninski, and Cole Hurwitz. Towards robust and generalizable representations of extracellular data using contrastive learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[57] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[58] Yule Wang, Chengrui Li, Weihan Li, and Anqi Wu. Exploring behavior-relevant and disentangled neural dynamics with generative diffusion models. *arXiv preprint arXiv:2410.09614*, 2024.

[59] Yule Wang, Zijing Wu, Chengrui Li, and Anqi Wu. Extraction and recovery of spatio-temporal structure in latent dynamics alignment with diffusion model. *Advances in Neural Information Processing Systems*, 36, 2024.

Appendix

A Background and Notation

A.1 Notation

Datasets and contrastive pairs: Let $\mathbf{x}$ denotes a vector and $\mathbf{X}$ denotes a matrix, with right subscript $\mathbf{X}_b$ denote the batch of the input samples, $\mathbf{X}_b := [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_B]$, here B is equal to batch size. For each sample $\mathbf{x}_i$ in the input batch matrix $\mathbf{X}_b$, $\mathbf{x}'_i$ means augmented view 1 of $\mathbf{x}'_i$, $\mathbf{x}''_i$ means augmented view 2 of $\mathbf{x}_i$, the positive pairs in input data denoted as ($\mathbf{x}_i$, $\mathbf{x}''_i$), negative pairs in input data denoted as ($\mathbf{x}'_i$, $\mathbf{x}''_j$), $i \neq j$. Give a weights (θ) parametrized representation function (artificial neural network) f_θ with adjustable adjustable temperature ε, which project the the positive pairs in latent space denoted as $s^+ = \langle \varepsilon^{-1} \widetilde{f}_\theta(\mathbf{x}'_i), \widetilde{f}_\theta(\mathbf{x}''_i) \rangle$, and negative pairs in latent space denoted as $s^- = \langle \varepsilon^{-1} \widetilde{f}_\theta(\mathbf{x}'_i), \widetilde{f}_\theta(\mathbf{x}''_j) \rangle, i \neq j$. Here, $\langle \cdot, \cdot \rangle$ is the inner product, which means $\langle \widetilde{f}_\theta(\mathbf{x}'_i), \widetilde{f}_\theta(\mathbf{x}''_i) \rangle = f_\theta(\mathbf{x}'_i)^\top f_\theta(\mathbf{x}''_j) / \|f_\theta(\mathbf{x}'_i)\| \|f_\theta(\mathbf{x}''_j)\|$ is the normalized form.

Continuous settings for optimal transport $\mathcal{X}$ and $\mathcal{Y}$ are topological spaces, $\mathcal{X} \times \mathcal{Y}$ is the product space, or Torus. $C(\mathcal{X})$ is the compact topological space which contains all of continuous functions on $\mathcal{X}$ endowed with the sup-norm. On Torus $\mathcal{X}$ and $\mathcal{Y}$ we define M as a compact n-dimensional manifold in product space $\mathcal{X} \times \mathcal{Y}$ ($\mathcal{X} = \mathcal{Y} := \mathbb{R}^n / \mathbb{Z}^n$) endowed with a cost function $c(x, y) := d_M(x, y)^2/2$ (Euclidean dsitance function) on $\mathbb{R}^n$. Transport plan $\pi(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is an element in $\mathcal{P}(\mathcal{X} \times \mathcal{Y})$. $\mathcal{P}(\mathcal{X} \times \mathcal{Y})$ means the collections of the joint distributions of the two marginal distributions $\mu \in \mathcal{P}(\mathcal{X})$ and $\nu \in \mathcal{P}(\mathcal{Y})$. $\mathcal{P}(\mathcal{X})$ the space of all (Borel) probability measures on $\mathcal{X}$, $\mathcal{P}(\mathcal{Y})$ means the same to $\mathcal{Y}$. To find a joint distribution (or plan) $\pi(x, y)$ in collections $U(\mu, \nu)$ with marginals μ and ν in the product space $\mathcal{X} \times \mathcal{Y}$, we can formulate as:

$$\min_{\pi \in U(\mu,\nu)} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \pi(x, y) c(x, y) \quad \text{s.t.} \sum_{y \in \mathcal{Y}} \pi(x, y) = \mu(x), \ \sum_{x \in \mathcal{X}} \pi(x, y) = \nu(y)$$

Discrete settings of optimal transport : μ and ν can be discrete probability measures whose supports are finite sets $X := \{f_\theta(\mathbf{x}'_i)\}_{i=1}^N, Y := \{f_\theta(\mathbf{x}''_i)\}_{i=1}^B$. And we define $\mu = \sum_{i=1}^B \delta_{f_\theta(\mathbf{x}'_i)} p_i, \nu = \sum_{i=1}^B \delta_{f_\theta(\mathbf{x}''_i)} q_i$, with vectors p and q in a simplex Δ_B in $\mathbb{R}^B$ defined by $\Delta_B := \left\{ v \in \mathbb{R}^B : v_i \geq 0, \sum_{i=1}^B v_i = 1 \right\}$, which we identify with $\mathcal{P}(\{1, \dots, B\})$. $\mathbf{C}$ is a $B \times B$ cost matrix calculated by $c(x, y)$, whose sampled from finite sets X and Y defined previously, and N is the batch size. $\mathbf{u}$ and $\mathbf{v}$ are $B \times 1$ scale factors matrix, $\mathbf{u}^{(t)}$ and $\mathbf{v}^{(t)}$ mean the scale factor matrix after t iterations of sinkhorn algorithms. $\mathbf{P}$ is a $B \times B$ joint distribution matrix of μ and ν, which represents the transport plan $\pi(x, y)$ that corresponds to minimize the cost. $\mathbf{P}^{(2t-1)}$ means we use t iterations $\mathbf{u}^{(t)}$ and $\mathbf{v}^{(t-1)}$ to calculate $\mathbf{P}^{(t)}$. When $t = 1$ means we use $\mathbf{u}^{(1)}$ and $\mathbf{v}^{(0)}$ to calculate $\mathbf{P}^{(1)}$, which is called half-step OT or one step Bregman projection. When $t = 2$ means we use $\mathbf{u}^{(1)}$ and $\mathbf{v}^{(1)}$ to calculate $\mathbf{P}^{(2)}$, which is called half-step OT or one step Bregman projection.

A.2 Proximal operator setup

In this section, we are going to provide the detailed illustration about the proximal operators. How the proximal operator would convert to the projection. And how to solve the Bregman projection with KL divergence.

A.2.1 Explanation of Definition 1

Let $h : \mathcal{X} \to [-\infty, +\infty]$ be a proper, lower semi-continuous convex function on Hilbert space $\mathcal{X}$. The **proximal operator** of h at point $\mathbf{v} \in \mathcal{X}$ is defined as a unique minimizer of the function $\mathbf{x} \mapsto h(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|_2^2$ [40]. For an instance, h is convex function relative to the constraint set $\mathcal{B}$, like the indicator function $h_{\mathcal{B}}$. Given the Euclidean norm on $\mathcal{X}$, we can write the proximal operator $\text{Prox}_{h,\mathcal{B}}^{\|\cdot\|^2}(\mathbf{v})$ as the most common way:

$$\text{Prox}_{h,\mathcal{B}}^{\|\cdot\|^2}(\mathbf{v}) = \arg\min_{\mathbf{x} \in \mathcal{B}} \left\{ h(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|_2^2 \right\}$$

The solution to the proximal operator exists and is unique due to the strong convexity of the above function.

A.2.2 Connection to the projection

Here, if we treat $h(x)$ as the indicator functions of the constraint set $\mathcal{B}$ as:

$$h(x) = \begin{cases} 0, & \text{if } x \in \mathcal{B} \\ \infty, & \text{if } x \notin \mathcal{B}. \end{cases}$$

Then the proximal operators problem could be understood intuitively as finding the "shortest distance" between the point v with the constraint set $\mathcal{B}$, which means the projection. And the following lemma holds:

Lemma 1. *If the constraint set $\mathcal{B}$ is closed and convex, then the projection of point v is unique on $\mathcal{B}$.*

Proof of the Lemma 1: This lemma can be proved by the strict convexity of the proximal operator.

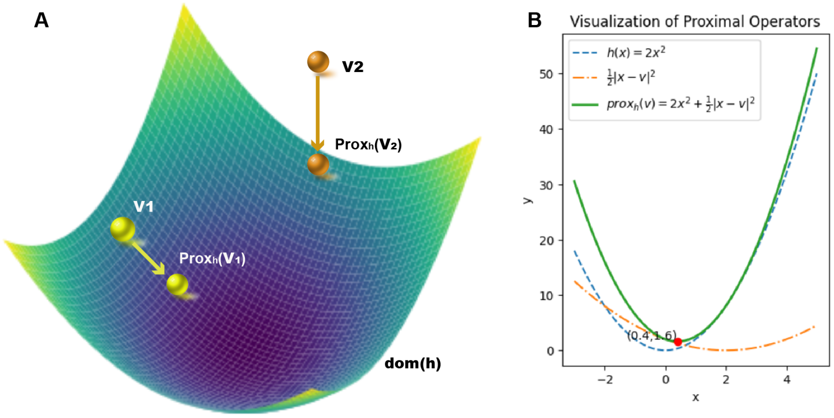

Figure A1: *Illustration of the proximal operators* A. Visualization of proximal operators in $\mathbb{R}^3$. On the surface defined by $h(x, y) = x^2 + y^2$ within the domain constraints $-1.2 < x < 1.2$ and $-1.2 < y < 1.2$. If $v = v_1 = (0.76, 0.76, 1.16)$, it lies within the domain of h, represented on the surface at the exact location matching its third coordinate with $h(x, y)$. If $v = v_2 = (1.5, 1.5, 6)$, which is outside the feasible region defined by h, the proximal operator projects it to the closest point within the domain, resulting in v_2's projection to approximately $(0.85, 0.85, 1.45)$. B. Visualization of proximal operators in $\mathbb{R}^2$. The blue dashed line represents the function $h(x) = x^2$. The orange dash-dotted line illustrates the penalty term $\frac{1}{2}\|x - v\|^2$ with $v = (2, 0)$, indicating the squared distance from any x to v. The green solid line is the proximal operator $2x^2 + \frac{1}{2}\|x - v\|^2$, which gets close to the minimization point of h(x) from v. The red point marks the $\text{Prox}_h(v)$ in this space.

A.2.3 Connection to the Bregman divergence

First, we define d_Γ as a generic Bregman divergence on some convex set $\mathcal{B}$, and the proximal map of a convex function d_ϕ according to this divergence is:

$$\text{Prox}^{d_\Gamma}_{d_\phi, \mathcal{B}}(\mathbf{K}) := \arg\min_{\mathbf{P} \in \mathcal{B}} d_\Gamma(\mathbf{P} \| \mathbf{K}) + d_\phi(\mathbf{P}). \tag{15}$$

Γ is a strictly convex function smooth on int($\mathcal{B}$), and $\text{Prox}^{d_\Gamma}_{d_\phi}(\mathbf{K}) \in \text{int}(\mathcal{B})$ is always uniquely defined by strict convexity. (Note that this theory is general and does not need to parametrize the $\mathbf{K}$ and $\mathbf{P}$ as models with θ). As $\mathcal{B} = \text{dom}(\Gamma)$,

$$\forall (\mathbf{P}, \mathbf{K}) \in \mathcal{B} \times \text{int}(\mathcal{B}), d_\Gamma(\mathbf{P} \| \mathbf{K}) = \Gamma(\mathbf{P}) - \Gamma(\mathbf{K}) - \langle \nabla \Gamma(\mathbf{K}), \mathbf{P} - \mathbf{K} \rangle,$$

which has its Legendre transform is also smooth and strictly convex:

$$\Gamma^*(\rho) = \max_{\mathbf{P} \in \mathcal{B}} \langle \mathbf{P}, \rho \rangle - \Gamma(\mathbf{P})$$

The Bregman divergence for a convex function Γ between points $\mathbf{x}$ and $\mathbf{y}$ is defined as:

$$d_\Gamma(\mathbf{x}, \mathbf{y}) = \Gamma(\mathbf{x}) - \Gamma(\mathbf{y}) - \langle \nabla\Gamma(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$$

where $\nabla\Gamma(\mathbf{y})$ is the gradient of Γ at $\mathbf{y}$. Giving the squared L2 distance can be viewed as a Bregman divergence derived from the convex function $\Gamma(\mathbf{x}) = \|\mathbf{x}\|^2$. For this function, the Bregman divergence between two points $\mathbf{x}$ and $\mathbf{y}$ becomes:

$$d_\Gamma(\mathbf{x}, \mathbf{y}) = \|\mathbf{x}\|^2 - \|\mathbf{y}\|^2 - 2\mathbf{y}^\top(\mathbf{x} - \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$$

Table A1: *Examples of functions Γ and their corresponding divergences d_Γ.*

Γ	d_Γ	**Description**
$\|\mathbf{x}\|^2$	$\|\mathbf{x} - \mathbf{y}\|^2$	squared Euclidean distance
$\mathbf{x} \ln \mathbf{x}$	$\mathbf{y} \ln \frac{\mathbf{y}}{\mathbf{x}} - (\mathbf{y} - \mathbf{x})$	Kullback–Leibler (KL) divergence
$-H(p) = \sum_j p_j \ln p_j$	$KL(q\|p) = \sum_j q_j \ln \frac{q_j}{p_j}$ $\sum p_j = \sum q_j = 1$	KL divergence between distributions p, q

A.2.4 Connection to the Bregman projection

Bregman projections solve the alignment problem onto the two constraints sets that encode the marginals along the rows and columns [3, 5, 43].

$$C_1^\mu := \{\mathbf{P} : \mathbf{P}\mathbb{1}_B = \mu\}, C_2^\nu := \{\mathbf{P} : \mathbf{P}^\top\mathbb{1}_B = \nu\} \tag{16}$$

If we specify the constraint set $\mathcal{B}$ as some set $C_1^\mu := \{\mathbf{P} : \mathbf{P}\mathbb{1}_m = \mu\}$, and select the $h_\mathcal{F}(x)$ as some indicator function of C_1^μ, which satisfies:

$$h_\mathcal{F}(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \in \mathbf{C}_1^\mu \\ \infty, & \text{if } \mathbf{x} \notin \mathbf{C}_1^\mu. \end{cases} \tag{17}$$

For the first step Bregman projection $\text{Prox}_{C_1^\mu}^{KL}(\mathbf{K})$ onto the set $\mathcal{C}_1^\mu$ with indicator function in Equation (17):

$$\mathbf{P}^{(1)} = \text{Prox}_{C_1^\mu}^{\text{KL}}(\mathbf{K}_\theta) = \arg\min_{\mathbf{P}\in C_1^\mu}\{h_\mathcal{F}(\mathbf{P}) + \text{KL}(\mathbf{P}\|\mathbf{K})\} = \arg\min\{\text{KL}(\mathbf{P}\|\mathbf{K}) : \mathbf{P}\mathbb{1}_B = \mu\}. \tag{18}$$

We can minimize the function in Equation (18) with Lagrange multiplier f on C_1^μ:

$$\varepsilon\text{KL}(\mathbf{P}\|\mathbf{K}) - f(\mathbf{P}\mathbb{1}_B - \mu), \tag{19}$$

Then we get $\mathbf{P}^{(1)}$ as the minimizer through the derivatives with respect to $\mathbf{P}$, we have

$$\varepsilon \log\left(\mathbf{P}^{(1)}/\mathbf{K}\right) - f\mathbb{1} = 0 \Rightarrow \mathbf{P}^{(1)} = \mathbf{u}\mathbf{K}, \text{ as } \mathbf{u} = e^{f/\varepsilon} > 0, \tag{20}$$

and we can use these relationship into the constraints sets with $\mathbf{P}^{(0)} = \text{diag}(\mathbb{1})\mathbf{K}\,\text{diag}(\mathbb{1})$.

$$\langle \mathbf{P}^{(1)}, \mathbb{1} \rangle = \mu \Rightarrow \langle \mathbf{u}^{(1)}\mathbf{K}, \mathbb{1} \rangle = \mu, \mathbf{u}^{(1)} = \frac{\mu}{\sum_i \mathbf{K}_{ij}}, \mathbf{P}^{(1)} = \text{diag}\left(\frac{\mu}{\mathbf{P}^{(0)}\mathbb{1}_B}\right)\mathbf{P}^{(0)}, \tag{21}$$

If we repeat this progress for the set $C_2^\nu := \{\mathbf{P} : \mathbf{P}^T\mathbb{1}_n = \nu\}$, we will get the $\text{Prox}_{C_2^\nu}^{\text{KL}}(\mathbf{P}^{(t+1)})$. And in the second step, we project onto the second constraint set C_2^ν with indicator function $h_\mathcal{G}(x)$ defined on C_2^ν and get:

$$\mathbf{P}^{(2)} := \text{Prox}_{C_2^\nu}^{\text{KL}}(\mathbf{P}^{(1)}) = \mathbf{P}^{(1)}\text{diag}\left(\frac{\nu}{\mathbf{P}^{(1)\top}\mathbb{1}_B}\right). \tag{22}$$

Iterating over these two sets of projections $\mathbf{P}^{(t+1)} := \text{Prox}_{C_1^\mu}^{\text{KL}}(\mathbf{P}^{(t)})$ and $\mathbf{P}^{(t+2)} := \text{Prox}_{C_2^\nu}^{\text{KL}}(\mathbf{P}^{(t+1)})$ until convergence could be summarized as Sinkhorn algorithm with t via recursive form:

$$\mathbf{u}^{(t+1)} \overset{\text{def}}{=} \frac{\mu}{\mathbf{K}\mathbf{v}^{(t)}}, \quad \mathbf{v}^{(t+1)} \overset{\text{def}}{=} \frac{\nu}{\mathbf{K}^\text{T}\mathbf{u}^{(t+1)}}, \quad \mathbf{P}^{(2t+2)} = \text{diag}(\mathbf{u}^{(t+1)})\mathbf{K}\,\text{diag}(\mathbf{v}^{(t+1)}). \tag{23}$$

The Sinkhorn algorithm is composed with two steps Bregman projection, Similarly, we can write out this recursive relationship as: $\mathbf{P}^{t+1}$ can be updated with dual variables f, g and $\mathbf{u}^{(t)} = e^{f^{(t)}/\varepsilon}, \mathbf{v}^{(t)} = e^{g^{(t)}/\varepsilon}$. The set $U(\mu, \nu) = C_1^\mu \cap C_2^\nu$, representing the feasible transport plans with given marginals. It could be any random sets, i. e. $\mathcal{B} = C_1^\mathbb{1} \cap C_2^\mathbb{1}$ denote the Birkhoff polytope of doubly stochastic matrices where $\mu = \mathbb{1}$ and $\nu = \mathbb{1}$ are the uniform distributions with all one element.

A.3 Background on OT

This section defines discrete and continuous optimal transport. Since the section 2.3 lacks a discrete OT definition, we discuss it here and show the equivalence between solving Bregman projection and the entropy-regularized OT (EOT) problem.

To support convergence proofs later, we introduce definitions of continuous measures. Symbols μ and ν may represent both discrete and continuous measures for intuitive consistency, with precise definitions at the start of each subsection.

A.3.1 Background on discrete OT

In section 2.3, we provide a general definition of the discrete optimal transport. Here, we specifically define the optimal transport on the representations space after an encoder f_θ with two augmented views $(\mathbf{x}', \mathbf{x}'')$. As we mainly discussed the distribution on the representation space, so here we suppose there is an encoder f_θ will project the augmented views into the latent. Here we define $\mu = \sum_{i=1}^{N} \delta_{f_\theta(\mathbf{x}'_i)} p_i, \nu = \sum_{i=1}^{N} \delta_{f_\theta(\mathbf{x}''_i)} q_i$, with vectors p and q in a simplex Δ_B in $\mathbb{R}^B$ defined by $\Delta_B := \left\{ v \in \mathbb{R}^B : v_i \geq 0, \sum_{i=1}^{B} v_i = 1 \right\}$. Here, $\mathbf{P}$ is a $B \times B$ joint coupling matrix of the marginal distributions μ and ν, which describes how much mass is needed to convert one distribution to match another. $\mathbf{C}$ is a $B \times B$ cost matrix calculated by the cost function $c(x, y)$ i.e. cosine dissimilarity, and we can write the OT problem as the constrained linear programming problem:

$$\min_{\mathbf{P}} \langle \mathbf{P}, \mathbf{C} \rangle \text{ s.t. } \mathbf{P}\mathbb{1} = \mu, \ \mathbf{P}^\top \mathbb{1} = \nu. \tag{24}$$

Even though directly solving Equation (24) is high computational complexity $O(n^3)$, we introduce a common relaxation called entropic regularization to smooth the transport plan.

A.3.2 Entropy regularized OT and the Sinkhorn algorithm.

Solving the exact OT problem above can be very computationally intensive. In this case, we can add the Shannon entropy $H(\mathbf{P}) = -(\mathbf{P}_{ij}(\log(\mathbf{P}_{ij}))$ to our objective in Equation (24) and obtain an approximation of entropy-regularized optimal transport (EOT) plan as:

$$\min_{\mathbf{P} \in \mathcal{B}} \ \langle \mathbf{P}, \mathbf{C} \rangle - \varepsilon H(\mathbf{P}), \quad \text{where } H(\mathbf{P}) = -\sum \mathbf{P}_{ij} \log(\mathbf{P}_{ij}), \tag{25}$$

where ε is a user specified parameter that controls the amount of smoothing in the transport plan. The cost matrix $\mathbf{C}$ could be transformed into the Gibbs kernel matrix $\mathbf{K}$ on a Hilbert space with the given formula,

$$\mathbf{K}_{ij} = \exp\left(-\varepsilon^{-1} \mathbf{C}_{ij}\right) \tag{26}$$

To solve (25) under the kernel space induced by $\mathbf{K}$, we can use the iterative Sinkhorn algorithm with the initialization of $\mathbf{u}^{(0)}$ and $\mathbf{v}^{(0)}$ as all one vector divided by the batch size, and the update rules:

$$\mathbf{u}^{(t+1)} \overset{\text{def}}{=} \frac{\mu}{\mathbf{K}\mathbf{v}^{(t)}} \text{ and } \mathbf{v}^{(t+1)} \overset{\text{def}}{=} \frac{\nu}{\mathbf{K}^{\mathrm{T}} \mathbf{u}^{(t+1)}}, \tag{27}$$

Then, the output of plan after t iterations is

$$\mathbf{P}^{(t)} = \operatorname{diag}(\mathbf{u}^{(t)}) \mathbf{K} \operatorname{diag}(\mathbf{v}^{(t)}). \tag{28}$$

It also could be interpreted with dual variables f and g:

$$\mathbf{P}_{i,j}^{(t)} = e^{f_i^{(t)}/\varepsilon} e^{-\mathbf{C}_{i,j}/\varepsilon} e^{g_j^{(t)}/\varepsilon}, \quad \mathbf{u}^{(t)} = e^{f^{(t)}/\varepsilon}, \mathbf{v}^{(t)} = e^{g^{(t)}/\varepsilon} \tag{29}$$

After convergence, the resulting $\mathbf{P}$ will be the optimal solution to Equation (25). The convergence and dynamics of OT and the dual formulation have been studied extensively in [4, 43, 19, 1]. Here, iterations converge to a stable transport plan $\mathbf{P}^{(\infty)}$as the optimal solution of Equation (3), which provides the minimum cost matching between two distributions. The convergence and dynamics of OT and its dual formulation have been studied extensively in [4, 43, 19, 1]. Thus, these results guarantee that the iterates will converge to the optimal solution of the EOT objective, or that $\mathbf{P}^{(t)} \to \mathbf{P}^{(\infty)}$ with $t \to \infty$.

This allows us to state the following lemma:

Lemma 2. *Solving the entropy optimal transport in Equation (3) is consistent with iterative solving the Bregman projection.*

Proof of the Lemma 2: Giving that some points $\mathbf{K}$ and $\mathbf{P}$, their distance could be measured by KL divergence:

$$\mathrm{KL}(\mathbf{P}\|\mathbf{K}) = \sum_{ij} \mathbf{P}_{ij} \log\left(\frac{\mathbf{P}_{ij}}{\mathbf{K}_{ij}}\right) - \mathbf{P}_{ij} + \mathbf{K}_{ij}$$

As $\mathbf{C}_{ij} = -\varepsilon \log \mathbf{K}_{ij}$ in Sinkhorn, we can see find $\mathbf{P}$ to minimize the Equation (3) can be transformed into some formula about $\mathbf{K}_{ij}$:

$$\begin{aligned} \min_{\mathbf{P}} \langle \mathbf{P}, \mathbf{C}\rangle - \varepsilon H(\mathbf{P}) &= \min_{\mathbf{P}} \sum_{i,j} \mathbf{C}_{ij}\mathbf{P}_{ij} + \varepsilon \sum_{i,j} \mathbf{P}_{ij}(\log(\mathbf{P}_{ij})) && (30)\\ &= \min_{\mathbf{P}} \varepsilon \sum_{i,j} (-\mathbf{P}_{ij}\log \mathbf{K}_{ij} + \mathbf{P}_{ij}\log(\mathbf{P}_{ij})) && (31)\\ &= \min_{\mathbf{P}} \varepsilon \mathrm{KL}(\mathbf{P}\|\mathbf{K}) \;\text{ s.t. } \mathbf{P}\mathbb{1} = \mu,\; \mathbf{P}^\top \mathbb{1} = \nu, && (32) \end{aligned}$$

Consider the $\mathbf{K}$ is a point in Hilbert kernel space, and ε is the constant, we set the μ and ν form the $\mathcal{B}$, so here can have:

$$\mathbf{P} = \mathrm{Prox}_{\mathcal{B}}^{\mathrm{KL}}(\mathbf{K}) = \arg\min_{\mathbf{P}\in\mathcal{B}} \mathrm{KL}(\mathbf{P}||\mathbf{K}) = \arg\min_{\mathbf{P}} \{\langle \mathbf{P}, \mathbf{C}\rangle + \varepsilon H(\mathbf{P}) : \mathbf{P}\mathbb{1} = \mu,\; \mathbf{P}^\top \mathbb{1} = \nu\} \quad (33)$$

A.3.3 Background on continuous optimal transport

To show the convergence of the Bregman projection, here we define the optimal transport problem with the continuous measure. Inherit the definition of $\mathcal{X}$ and $\mathcal{Y}$ in the Appendix A.1, finding the optimal transport between two continuous measure μ and ν could be transformed into some problems with the minimization of Kantorovich functional.

Definition 4 (*Continuous optimal transport*)**.** *We redefine μ and ν be two probability measures on latent manifold $\mathcal{M}$ with Hölder continuous and strictly positive densities e^f and e^g, respectively: $\mu = e^f dM, \nu = e^g dM$, where dM is the Riemannian normalized volume form on $\mathcal{X}$. For each $x \in \mathcal{X}$ and $y \in \mathcal{Y}$:*

$$W(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\, d\pi(x,y). \quad (34)$$

Definition 5 (*Dual of continuous OT*)**.** *The dual of standard OT reads:*

$$W(\mu,\nu) = \sup_{f,g\in\mathcal{U}(c)} \int_{\mathcal{X}} f d\mu(x) + \int_{\mathcal{Y}} d\nu(y) \quad (35)$$

where the constraint set $\mathcal{U}(c)$ is defined by $\mathcal{U}(c) := \{(\mu,\nu) \in \mathcal{C}(\mathcal{X})\times\mathcal{C}(\mathcal{Y})\} | f(x)+g(y) \leq c(x,y)\}$.

Here, $C(\mathcal{X})$ is the space of all continuous functions on $\mathcal{X}$, the functions which measured using the supreme norm $||f||_\infty$, with the Legendre transform:

Definition 6 (*Legendre c-transforms*)**.** *For the dual variables, or so called potentials, there exists the Legendre c-transforms:*

$$f^c(y) := \sup_{x\in\mathcal{X}}(-c(x,y) + f(x)), \quad g^c(x) := \sup_{y\in\mathcal{Y}}(-c(x,y) + g(y)). \quad (36)$$

In which $g^c(x)$ and $f^c(y)$ are Legendre c-transforms of $g(y) \in \mathcal{C}(\mathcal{Y})$ and $f(x) \in \mathcal{C}(\mathcal{X})$ with cost function $c(x,y)$.

Definition 7 (*Pushforward measure*)**.** *The pushforward measure of μ under the map T, denoted as T_μ, is a measure on $\mathcal{X}$ defined by $T_\mu(B) = \mu(T^{-1}(B))$ for any Borel set B in $\mathcal{X}$. $T_\mu = \nu$ when T is an optimal transport map. Following the similar way we can define the push-forward measure T_μ and T_ν as:*

$$T_\mu : C(\mathcal{X}) \to C(\mathcal{Y}) := \log \int e^{-c(x,\cdot)+f(x)}\mu(x),\; T_\nu : C(\mathcal{Y}) \to C(\mathcal{X}) := \log \int e^{-c(\cdot,y)+g(y)}\nu(y) \quad (37)$$

Definition 8 (φ*-divergence regularized OT in continuous*)**.** *Given two dual variables (also called potentials) $f \in \mathbb{R}^n$ and $g \in \mathbb{R}^m$ for each marginal constraint, the entropy regularized optimal transport in Equation (3) could be transformed into some problems with the Kantorovich functional:*

$$W^{\varphi}_{\varepsilon,c}(\mu,\nu) = \inf_{\pi in \Pi(\mu,\nu)} (\int_{\mathcal{X}\times\mathcal{Y}} c(x,y)d\pi(x,y) + \varepsilon \int_{\mathcal{X}\times\mathcal{Y}} \varphi\left(\frac{d\pi(x,y)}{d\mu(x)d\nu(y)}\right) d\mu(x)d\nu(y)\biggr) \tag{38}$$

Proposition 1 (Dual of EOT)**.** *Consider OT between two probability measures μ and ν with a convex regularizer ϕ on $\mathbb{R}^+$ in Equation* (38)

$$W^{\varphi}_{c,\varepsilon}(\mu,\nu) = \sup_{f,g\in\mathcal{C}(\mathcal{X})\times\mathcal{C}(\mathcal{Y})} \int_{\mathcal{X}} f d\mu(x) + \int_{\mathcal{Y}} d\nu(y) - \varepsilon \int_{\mathcal{X}\times\mathcal{Y}} \varphi^*(\frac{f(x)+g(y)-c(x,y)}{\varepsilon})d\mu(x)d\nu(y) \tag{39}$$

where φ^* is the Legendre transform of φ defined by $\varphi^*(\mathbf{v}) := \sup_{\mathbf{x}} \mathbf{x}\mathbf{v} - \phi(\mathbf{x})$

A good choice for φ^* is that the $\varphi^*(\mathbf{v}) = e^{\mathbf{v}}$. The entropy regularization term ensures the problem is solvable, especially for computational schemes. If the $\varepsilon \to \infty$, the optimal primal plan π^* can be retrieved using, which corresponds to the mutual information formula:

$$\frac{d\pi^*}{d\mu d\nu}(x,y) = \exp\left(\frac{f^*(x)+g^*(y)-c(x,y)}{\varepsilon}\right)$$

In the discrete version in Equation (3), the optimal transport plan $\mathbf{P}$ can often be expressed in terms of the optimal transport map T^* when it exists, one can define the so-called barycentric projection map

$$T^* : \mathbf{x}_i \in \mathcal{X} \mapsto \frac{1}{\mu_i}\sum_j \mathbf{P}_{i,j}\mathbf{y}_j \in \mathbb{R}^d,$$

This link provides the connection between the mutual information with the optimal mapping:

$$T^* : x \in X \mapsto \int_Y y\frac{d\pi(x,y)}{d\mu(x)d\nu(y)}d\nu(y).$$

Note that the joint distribution π always has a density $\frac{d\pi(x,y)}{d\mu(x)d\nu(y)}$ with respect to $\mu\otimes\nu$, and the mutual information method will lead us to the optimal solution.

B Analysis of GCA

B.1 Convergence of GCA

In this section, we provide a proof of convergence in the forward pass for our GCA algorithm. To do this, we show the general form in Algorithms 1 for all Bregman divergence (d_Γ) in forward pass in GCA algorithms could be converged through Djkstra's projection algorithms. Finally, we show the uniformly convergence of the transport plan $\mathbf{P}$, and the convergence of its dual variables $f^{(t)}$ in each iteration.

B.1.1 Convergence of GCA-INCE

Corollary 1. *(Convergence of GCA-INCE [43]) Given $\mathbf{u}^* = e^{f^\infty}$, $\mathbf{v}^* = e^{g^{(\infty)}}$ and a kernel space H with the Hilbert-Birkhoff metric $d_H(\mathbf{u},\mathbf{u}^*) := \log\max_{i,j}\frac{\mathbf{u}_i\mathbf{u}^*_j}{\mathbf{u}_j\mathbf{u}^*_i}$, for all positive pairs $(\mathbf{u},\mathbf{u}^*)$, with $\mathbf{u}^{(t)} \to \mathbf{u}^*$ and $\mathbf{v}^{(t)} \to \mathbf{v}^*$, we can prove that:*

$$\|\log\mathbf{P}^{(t)} - \log\mathbf{P}^*\|_\infty \le d_H(\mathbf{u}^{(t)},\mathbf{u}^*) + d_H(\mathbf{v}^{(t)},\mathbf{v}^*), \tag{40}$$

Proof for Corollary 1: First, let's define the following Hilbert space: $\forall(\mathbf{u},\mathbf{u}') \in (\mathbb{R}^n_{+,*})^2$, $d_H(\mathbf{u},\mathbf{u}') := \log\max_{i,j}\frac{\mathbf{u}_i\mathbf{u}'_j}{\mathbf{u}_j\mathbf{u}'_i}$. For any pairs of vectors that $(\mathbf{v},\mathbf{v}') \in (\mathbb{R}^m_{+,*})^2$ holds:

$$d_H(\mathbf{v},\mathbf{v}') = d_H\left(\frac{\mathbf{v}}{\mathbf{v}'}, \mathbb{1}_m\right) = d_H\left(\mathbb{1}_m/\mathbf{v}, \mathbb{1}_m/\mathbf{v}'\right). \tag{41}$$

Algorithm A1 Generalized Contrastive Alignment (GCA)

```
**Input:** Encoder $f_\theta$, projector $g_\theta$, data $\{\mathbf{x}_k\}_{k=1}^N$, batch size $B$, cost function $c(x, y)$, entropy parameter $\varepsilon$, constant $\tau$, total iterations $T$, marginal constraints $\mu$ and $\nu$, relax items $d_1, d_2$ and constant $\delta_{\text{eps}}$, some divergence $d_M$ and $d_\Gamma$ (could be KL or WDM),
**for** sampled minibatch $\{\mathbf{x}_k\}_{k=1}^B$ **do**
    Generate two views $(\mathbf{z}_k', \mathbf{z}_k'')$ using $f_\theta$, $g_\theta$ with randomly sampled augmentations.
**end for**
$\mathbf{u}^{(0)} = \mathbb{1}$, $\mathbf{v}^{(0)} = \mathbb{1}$, $f = 0$, $g = 0$, $\mathbf{C}_{ij} = c(\mathbf{z}_i', \mathbf{z}_j'')$
$d_1 \leftarrow d_1/(d_1 + \varepsilon)$, $d_2 \leftarrow d_2/(d_2 + \varepsilon)$, $\mathbf{K} = \exp(C_{ij}/\varepsilon^{-1})$
**for** $i = 1$ **to** $T$ **do**
    $\delta f \leftarrow \exp -f/(\varepsilon + d_1)$, $\delta g \leftarrow \exp -g/(\varepsilon + d_2)$
    $\mathbf{u} \leftarrow \delta f \cdot \text{Prox}_{\mathcal{F}}(K\mathbf{v} + \delta_{\text{eps}})^{fi}$
    $\mathbf{v} \leftarrow \delta g \cdot \text{Prox}_{\mathcal{G}}(K^T\mathbf{u} + \delta_{\text{eps}})^{fi}$
    **if** $\mathbf{u} > \tau$ or$\mathbf{v} > \tau$ **then**
        $f \leftarrow f + \varepsilon \cdot \log(\max(\mathbf{u}))$, $g \leftarrow g + \varepsilon \cdot \log(\max(\mathbf{v}))$
        $K \leftarrow \exp{(f + g - \mathbf{C})}/\varepsilon$, $\mathbf{v} = \mathbb{1}$
    **end if**
**end for**
$\log \mathbf{u} \leftarrow f/(\varepsilon + \mathbf{u})$, $\log \mathbf{v} \leftarrow g/(\varepsilon + \mathbf{v})$
Compute transport plan as:
$\mathbf{P} \leftarrow \exp(\log \mathbf{u} + \log \mathbf{v} - C/\varepsilon)$
Normalize $\mathbf{P}_u^{(T)}$ by its column sums.
Loss: $\mathcal{L}_{GCA} = d_M(\mu \otimes \nu, \mathbf{P}_u^{(T)})$
Update networks $f_\theta$ and $g_\theta$ to minimize $\mathcal{L}_{GCA}$
```

Let $\mathbf{K} \in \mathbb{R}_{+,*}^{n\times m}$, then for $(\mathbf{v}, \mathbf{v}') \in (\mathbb{R}_{+,*}^m)^2$ we have

$$d_H(\mathbf{u}^{(t+1)}, \mathbf{u}^*) = d_H\left(\frac{\mathbb{1}_n}{\mathbf{K}\mathbf{v}^{(t)}}, \frac{\mathbb{1}_n}{\mathbf{K}\mathbf{v}^*}\right) = d_H(\mathbf{K}\mathbf{v}^{(t)}, \mathbf{K}\mathbf{v}^*) \leq \lambda(\mathbf{K}) d_H(\mathbf{v}^{(t)}, \mathbf{v}^*),$$

where

$$\lambda(\mathbf{K}) := \frac{\sqrt{\eta(\mathbf{K})} - 1}{\sqrt{\eta(\mathbf{K})} + 1} < 1, \quad \eta(\mathbf{K}) := \max_{i,j,k,\ell} \frac{\mathbf{K}_{i,k}\mathbf{K}_{j,\ell}}{\mathbf{K}_{j,k}\mathbf{K}_{i,\ell}}. \tag{42}$$

Based on the contraction mapping theory, one has $(\mathbf{u}^{(\ell)}, \mathbf{v}^{(\ell)}) \to (\mathbf{u}^*, \mathbf{v}^*)$ and

$$\begin{aligned} d_H(\mathbf{u}^{(t)}, \mathbf{u}^*) &\leq d_H(\mathbf{u}^{(t+1)}, \mathbf{u}^{(t)}) + d_H(\mathbf{u}^{(t+1)}, \mathbf{u}^*) && (43)\\ &\leq d_H\left(\frac{\mu}{\mathbf{K}\mathbf{v}^{(t)}}, \mathbf{u}^{(t)}\right) + \lambda(\mathbf{K})^2 d_H(\mathbf{u}^{(t)}, \mathbf{u}^*) && (44)\\ &= d_H\left(\mu, \mathbf{u}^{(t)} \odot (\mathbf{K}\mathbf{v}^{(t)})\right) + \lambda(\mathbf{K})^2 d_H(\mathbf{u}^{(t)}, \mathbf{u}^*), && (45) \end{aligned}$$

$$d_H(\mathbf{u}^{(t)}, \mathbf{u}^*) \leq \frac{d_H(\mathbf{P}^{(t)}\mathbb{1}_m, \mu)}{1 - \lambda(\mathbf{K})^2}, \quad d_H(\mathbf{v}^{(t)}, \mathbf{v}^*) \leq \frac{d_H(\mathbf{P}^{(t)\top}\mathbb{1}_n, \nu)}{1 - \lambda(\mathbf{K})^2}, \tag{46}$$

where we denoted $\mathbf{P}^{(t)} := \text{diag}(\mathbf{u}^{(t)})\mathbf{K}\,\text{diag}(\mathbf{v}^{(t)})$. Last, one has

$$\|\log(\mathbf{P}^{(t)}) - \log(\mathbf{P}^*)\|_\infty \leq d_H(\mathbf{u}^{(t)}, \mathbf{u}^*) + d_H(\mathbf{v}^{(t)}, \mathbf{v}^*), \tag{47}$$

where $\mathbf{P}^*$ is the unique solution of Equation (3). The above formula also shows that the t-step solution gives a better lower bound than the 1-step solution. □

B.1.2 Convergence of the Djkstra's projection algorithms

The previous subsection proved the convergence of GCA-INCE. Here, we extend this to show the convergence of all generalized proximal operators in Algorithm 1. Additionally, we demonstrate that these operators can iteratively solve alignment problems in the forward pass, following Dykstra's projection algorithm.

We present a general convergence proof for Dykstra's projection algorithm, sharing the form in Definition (1). First, we define d_Γ as a generic Bregman divergence on some convex set $\mathcal{B}$, and the proximal map of a convex function d_ϕ according to this divergence is:

$$\mathrm{Prox}^{d_\Gamma}_{d_\phi,\mathcal{B}}(\mathbf{K}) := \arg\min_{\tilde{\mathbf{P}}\in\mathcal{B}} d_\Gamma(\tilde{\mathbf{P}}\|\mathbf{K}) + d_\phi(\tilde{\mathbf{P}}). \tag{48}$$

Γ is a strictly convex function smooth on int$(\mathcal{B})$, and $\mathrm{Prox}^{d_\Gamma}_{d_\phi}(\mathbf{K}) \in \mathrm{int}(\mathcal{B})$ is always uniquely defined by strict convexity. As $\mathcal{B} = \mathrm{dom}(\Gamma)$,

$$\forall(\mathbf{P},\mathbf{K}) \in \mathcal{B}\times\mathrm{int}(\mathcal{B}), d_\Gamma(\mathbf{P}\|\mathbf{K}) = \Gamma(\mathbf{P}) - \Gamma(\mathbf{K}) - \langle\nabla\Gamma(\mathbf{K}), \mathbf{P}-\mathbf{K}\rangle,$$

which has its Legendre transform is also smooth and strictly convex:

$$\Gamma^*(\rho) = \max_{\mathbf{P}\in\mathcal{B}}\langle\mathbf{P},\rho\rangle - \Gamma(\mathbf{P})$$

In particular, one has that $\nabla\Gamma$ and $\nabla\Gamma^*$ are bijective maps between int$(\mathcal{B})$ and int(dom(Γ^*)) such that $\nabla\Gamma^* = (\nabla\Gamma)^{-1}$. For $\Gamma = ||\cdot||^2$, one recovers the squared Euclidean norm $d_\Gamma = ||\cdot||^2$. One has KL = d_Γ for $\Gamma(\mathbf{P}) = h(\mathbf{P}) = -\sum_{i,j=1}^{B}(\mathbf{P}_{ij}(\log\mathbf{P}_{ij}-1))$. Dykstra's algorithm starts by initializing $\mathbf{P}^{(0)} := \mathbf{K}$ and $\mathbf{U}^{(0)} = \mathbf{U}^{(-1)} := 0$. One then iterative defines, for $k > 0$,

$$\mathbf{P}^{(k)} := \mathrm{Prox}^{d_\Gamma}_{d_{\phi_{[k]_2}}}(\nabla\Gamma^*(\nabla\Gamma(\mathbf{P}^{(k-1)}) + \mathbf{U}^{(k-2)})), \tag{49}$$

$$\mathbf{U}^{(k)} := \mathbf{U}^{(k-2)} + \nabla\Gamma(\mathbf{P}^{(k-1)}) - \nabla\Gamma(\mathbf{P}^{(k)}), \tag{50}$$

Proposition 2. *Giving d_{ϕ_1}, d_{ϕ_2} are two proper, lower-semicontinuous convex functions defined on $\mathcal{B}$. We also assume that the following qualification constraint holds:*

$$ri(dom(d_{\phi_1})) \cap ri(dom(d_{\phi_2})) \cap ri(dom(d_\Gamma)) = \emptyset, \tag{51}$$

where ri is the relative interior and dom$(\phi) = \{\pi;\phi(\pi) = +\infty\}$. Then the $\mathbf{P}^t$ converges to the solution of the following equation:

$$prox^{d_\Gamma}_{h,\mathcal{B}}(\mathbf{K}) = \arg\min_{\mathbf{P}\in\mathcal{B}}\{d_\Gamma(\mathbf{P}\|\mathbf{K}) + \lambda_1 d_{\phi_1}(\mathbf{P}) + \lambda_2 d_{\phi_2}(\mathbf{P})\} \tag{52}$$

Proof of the Proposition 2: Proof in [42] section 3.2.

B.1.3 Convergence of Bregman projection

This section aims to show for the continuous measure, the convergence of Bregman projection holds [4]. Finding the optimal transport map T could be derived in minimizes some functionals derived from the potential function f defined on $\mathcal{X}$, with Legendre transform in Equation (36) defined on $\mathcal{Y}$:

$$J(f) := \int_{\mathcal{X}} f\mu(x) + \int_{\mathcal{Y}} f^{\mathrm{c}}\nu(y) = I_\mu - L \tag{53}$$

Lemma 3 (*Uniformly convergence*)**.** *[4] When $t_1 \to \infty$, $f^{(t_1)}$ converges uniformly to a fixed point $f^{(\infty)}$, with $f^{(t_1)} \leq f^{(\infty)}$.*

Proof for the Lemma 3 (*Uniformly convergence*): We follow the procedures of methods in [4].

Giving push-forward measure T_μ and T_ν and a composed operator $S = T_\nu \circ T_\mu$, which yields an iteration on $C(\mathcal{X})$ as $S : C(\mathcal{X}) \to C(\mathcal{X}), f \to f \circ g \circ f$, $f^{(m+1)} = S(f^{(m)})$, and $e^{S(f)-f}\mu$ is the probability measure on $\mathcal{X}$.

Lemma 4 (*Existence and uniqueness*)**.** *The following conditions are equivalent for a function f in the space $C(X)$, where $C(X)$ denotes the space of continuous functions on a set X:*

- *f is a critical point for the functional F on $C(X)$.*
- *The function $\exp(S(f) - f) = 0$ hold almost everywhere (a.e.) with respect to (w.r.t.) μ.*

Moreover, if f is a critical point, then $f^* := S(f)$ is a fixed point for the operator S on $C(X)$.

Proof of the Lemma 1 Consider the functional L defined in Equation (53), the differential of L at an element $f \in C(X)$ is represented by the probability measure $\exp(S(f) - f)\mu$. For some iterations $f^{(m+1)} - f^{(m)} = S(f^{(m)}) - f^{(m)}$, when f is a critical point (derivative is zero or undefined) for the functional J on $C(X)$, and $f^* := S(f)$ is a fixed point for the operator S on $C(X)$, proved by realizing for any $\dot{f} \in C(X)$:

$$\frac{d}{dt} L(f + t\dot{f})\Big|_{t=0} = \int_X \dot{f} e^{(S(f)-f)} d\mu. \tag{54}$$

This follows readily from the definitions by differentiating $t \mapsto g[(f + t\dot{f})]$ to get an integral over (X, μ) and then switching the order of integration. As a consequence, f is a critical point of the functional F on $C^0(X)$ if and only if $e^{(S(f)-f)}\mu = \mu$, i.e., if and only if $e^{(S(f)-f)} = 1$ almost everywhere with respect to μ. Finally, if this is the case, then $S(f) = f$ almost everywhere with respect to μ and hence $S(S(f)) = S(f)$ (since $S(f)$ only depends on f viewed as an element in $L^1(X, \mu)$).

Lemma 5. *Given a point $x_0 \in X$, the subset K_{x_0} of $C(\mathcal{X})$ defined as all elements f in the image of S satisfying $f(x_0) = 0$ is compact in $C(\mathcal{X})$.*

Proof of the Lemma 5: Based on the compactness of the product space $\mathcal{X} \times \mathcal{Y}$, the continuous function c is uniformly continuous on $\mathcal{X}$. So $S(C(\mathcal{X}))$ is an equicontinuous family of continuous functions on X. By Arzelà-Ascoli theorem, it follows that the set K_{x_0} is compact in $C(\mathcal{X})$.

Proposition 3. *The operator S has a fixed point f^* in $C(\mathcal{X})$. Moreover, f^* is uniquely determined a.e. wrt μ up to an additive constant, and f^* minimizes the functional F. More precisely, there exists a unique fixed point in $S(C(\mathcal{X}))/\mathbb{R}$.*

Proof of the Proposition (3): Then based on the Jensen's inequality, we have

$$I_\mu(f^{(m+1)}) - I_\mu(f^{(m)}) = \int \log \exp\left(S(f^{(m)}) - f^{(m)}\right) d\mu \leq \log \int \exp\left(S(f^{(m)}) - f^{(m)}\right) d\mu = 0, \tag{55}$$

$$L(f^{(m)}) - L(f^{(m+1)}) = \int \log \exp\left(S(g^{(m)}) - g^{(m)}\right) d\nu \leq \log \int \exp\left(S(g^{(m)}) - g^{(m)}\right) d\nu = 0. \tag{56}$$

So we know the functionals are strictly decreasing at $f^{(m)}$ unless $S(f^*) = f^*$ for $f^* := S(f^{(m)})$. Then based on the Lemma 5, we know for each initial data f_0, the closure of its images denoted as K_{f_0} in $C(\mathcal{X})/\mathbb{R}$ is compact, under the operator S. Hence, $f^{(m)} \to f^{(\infty)}$ in $C(\mathcal{X})/\mathbb{R}$. And J is decreasing along the orbit but has lower bound:

$$J(f^{(\infty)}) = \inf_{K_{f^{(0)}}} J.$$

By the condition for strict monotonicity, it must be that $S(f^{(\infty)}) = f^{(\infty)}$ a.e. wrt μ. It then follows from the Proposition (3) that $f^{(\infty)}$ is uniquely determined in $C(\mathcal{X})/\mathbb{R}$ (by the initial data $f^{(0)}$), i.e. the whole sequence converges in $C(\mathcal{X})/\mathbb{R}$. We first show that there exists a number $\lambda \in \mathbb{R}$ such that $\lim_{m\to\infty} I_\mu(f^{(m)}) = \lambda$. I_μ is decreasing and hence it is enough to show that $I_\mu(f^{(m)})$ is bounded from below. By $I_\mu = J + L$, and J is bounded from below (by $F(f^{(\infty)})$). Moreover, by the first step $L(f^{(m)}) \geq L(f^{(0)})$. Next, decompose

$$f^{(m)} = \tilde{f}^{(m)} + f^{(m)}(x_0),$$

By the Lemma 5 the sequence $(\tilde{f}^{(m)}$ is relatively compact in $C(\mathcal{X})$ and we claim that $|f^{(m)}(x_0)| \leq C$ for some constant C. Indeed, if this is not the case then there is a subsequence $f^{(m_j)}$ such that $|f^{(m_j)}| \to \infty$ uniformly on X. But this contradicts that $I_\mu(f^{(m)})$ is uniformly bounded. It follows that the sequence $(f^{(m)})$ is also relatively compact. Hence, by the previous step the whole sequence $f^{(m)}$ converges to the unique minimizer f^* of F in $S(C(\mathcal{X}))$ satisfying $I_\mu(f^*) = \lambda$.

B.2 GCA version of unbalanced optimal transport (GCA-UOT)

In this section, we are going to introduce the relaxation of the EOT plan as Unbalanced optimal transport plan (UOT). And its relationship with the dual formula of EOT. Here we need to emphasize that the GCA-UOT not just add constraint to the proximal operators which computes the coupling matrix $\mathbf{P}_\theta$, but also add the penalty (i.e. KL-divergence) to the loss function d_M. For the specific function we used in the method of GCA-UOT in Table 2, we employed a version with the loss in Equation (11) plus the loss in Equation (10) with a weight control parameter.

B.2.1 Explanation of the unbalanced OT

Unbalanced optimal transport (UOT) in Equation (9) seeks to generalize the OT problem in Equation (24) by allowing for the relaxation of these constraints [11], as penalization by certain divergence measures d_{ϕ_1} and d_{ϕ_2} (e.g., Kullback-Leibler divergence). Here we provide the unbalanced OT for the entropic regularization optimal transport in Equation (3), which ensure that the transported mass respects the given source μ and target distributions ν:

$$\mathbf{U}_{OT}(\mu,\nu) = \min_{\mathbf{P}} \langle \mathbf{P},\mathbf{C}\rangle + \lambda_1 d_{\phi_1}(\mathbf{P}\mathbb{1}||\mu) + \lambda_2 d_{\phi_2}(\mathbf{P}^\top\mathbb{1}||\nu) + \varepsilon H(\mathbf{P}) \tag{57}$$

Here $\langle P, C\rangle$ represents the total transport cost. λ_1 and λ_2 are regularization parameters that control the trade-off between the transport cost and the divergence penalties.

B.2.2 Connection to dual formula of EOT

Lemma 6. *The entropy regularized OT problem is a special case of a structured convex optimization problem of Equation (57) the by giving functions $h_{\mathcal{F}}$ and $h_{\mathcal{G}}$, $h_{\mathcal{F}} = \iota_{\{C_1^\mu\}}$ and $h_{\mathcal{G}} = \iota_{\{C_2^\nu\}}$, as the indicator function of a closed convex set $C_1^\mu := \{\mathbf{P} : \mathbf{P}\mathbb{1}_m = \mu\}, C_2^\nu := \{\mathbf{P} : \mathbf{P}^\top\mathbb{1}_n = \nu\}$.*

$$\min_{\mathbf{P}} \langle \mathbf{P},\mathbf{C}\rangle + \varepsilon H(\mathbf{P}) + h_{\mathcal{F}}(\mathbf{P}\mathbb{1}_m) + h_{\mathcal{G}}(\mathbf{P}^\top\mathbb{1}_n). \quad \iota_C(x) = \begin{cases} 0 & \textit{if } x \in C, \\ +\infty & \textit{otherwise}, \end{cases} \tag{58}$$

Proof of the Lemma 6: Let's start with the dual formula of the Equation (3) with $\mathcal{B} = C_1^\mu \cap C_2^\nu$, we can introduce the Lagrangian $\mathcal{E}(\mathbf{P}, f, g)$ of Equation (3) reads:

$$\text{Prox}_{\mathcal{B}}^{\text{KL}}(\mathbf{K}) := \min_{\mathbf{P}\in\mathcal{B}} \langle \mathbf{P},\mathbf{C}\rangle - \varepsilon H(\mathbf{P}) = \mathcal{E}(\mathbf{P}, f, g) \tag{59}$$

$$= \min_{P} \max_{f\in\mathbb{R}^n, g\in\mathbb{R}^m} \langle \mathbf{P},\mathbf{C}\rangle - \varepsilon H(\mathbf{P}) - \langle f, \mathbf{P}\mathbb{1}_m - \mu\rangle - \langle g, \mathbf{P}^T\mathbb{1}_n - \nu\rangle. \quad \square \tag{60}$$

To solve this problem, we can use the first order condition:

$$\frac{\partial \mathcal{E}(\mathbf{P}, f, g)}{\partial \mathbf{P}_{ij}} = \mathbf{C}_{ij} + \varepsilon\log(\mathbf{P}_{ij}) - f_i - g_j = 0 \quad \Rightarrow \quad log\mathbf{P} = \frac{1}{\varepsilon}(f\mathbb{1}_m^T + \mathbb{1}_n g^T - \mathbf{C}) \tag{61}$$

The solution to the Equation (3) is unique with scaling variabl $(\mathbf{u},\mathbf{v}) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$ in Equation (23). And each items in the optimal transport matrix $\mathbf{P}$ is, and optimal (f, g) are linked to non-negative vectors $(\mathbf{u},\mathbf{v})$ through $(\mathbf{u},\mathbf{v}) = (e^{f/\varepsilon}, e^{g/\varepsilon})$.

$$\mathbf{P}_{ij} = e^{f_i/\varepsilon}e^{-\mathbf{C}_{ij}/\varepsilon}e^{g_j/\varepsilon} = \mathbf{u}_i\mathbf{K}_{ij}\mathbf{v}_j, \quad (f^{(t)}, g^{(t)}) = \varepsilon(\log(\mathbf{u}^{(t)}), \log(\mathbf{v}^{(t)})), \tag{62}$$

B.3 Equivalence of INCE objective with single step Bregman projection

In this section, we are going to discuss how to build the equivalence between minimizing the KL-divergence d_M between the $\mathbf{P}^{(1)}$ and the $\mathbf{P}_{tgt}$ with respect to θ in GCA objective:

$$\min_\theta \text{KL}\big(\mathbf{I}||\text{Prox}_{C_1^\mu}^{KL}(\mathbf{K}_\theta)),$$

with the INCE loss minimization in Equation (1). Here $\mathbf{P}^{(1)}$ is the nearest point of $\mathbf{K}_\theta$ on constraint set C_1^μ measured by the KL-divergence d_Γ defined in Equation (18), through one step of proximal operator (Bregman projection). And $\mathbf{K}_\theta$ denote the augmentation kernel as in Definition (3) with cosine similarity.

B.3.1 Proof of the Theorem 1

Suppose we had a encoder f_θ with parameter θ in INCE, with $\widetilde{f_\theta}$ to represent its normalized form,then we can use the following proposition to assist our proof:

Proposition 4. *Given the cost matrix as* $\mathbf{C}_{i,j} = 1 - \widetilde{f_\theta}(\mathbf{x}'_i)^\top \widetilde{f_\theta}(\mathbf{x}''_j)$*, and Gibbs kernel* $\mathbf{K}_\theta = \exp(-\mathbf{C}_{i,j}/\varepsilon)$*, based on the cosine dissimilarity scores of the inner products* $\langle \mathbf{z}_{\theta i}, \mathbf{z}_{\theta j} \rangle$*, with* $\mathbf{z}_i = \frac{f_\theta(\mathbf{x}'_i)}{\|f_\theta(\mathbf{x}'_i)\|}$ *and* $\mathbf{z}_j = \frac{f_\theta(\mathbf{x}''_j)}{\|f_\theta(\mathbf{x}''_j)\|}$*. Set* d_M *and* d_Γ *to KL-divergence, and the target transport plan* $\mathbf{P}_{tgt} = \mathbf{I}$*. The probability matrix* $\mathbf{P}$ *after one-step Bregman iteration of entropy optimal transport problem could be represented as:*

$$\mathbf{P}_{ij} = \frac{\mathbf{K}_{\theta ij}}{\sum_{j=1}^{B} \mathbf{K}_{\theta ij}} = \frac{\exp\left(\varepsilon^{-1}\langle \mathbf{z}_i, \mathbf{z}_j \rangle\right)}{\sum_{j=1}^{B} \exp\left(\varepsilon^{-1}\langle \mathbf{z}_i, \mathbf{z}_k \rangle\right)} \tag{63}$$

Proof of the Proposition (4): We assume that gibbs kernel $\mathbf{K}_\theta$ is a matrix which can be expressed as:

$$\mathbf{K}_{\theta ij} = \exp\left(-\varepsilon^{-1}\mathbf{C}_{i,j}\right) = \exp\left(-\varepsilon^{-1}|1 - \langle \mathbf{z}_i, \mathbf{z}_j \rangle|\right),$$

with a temperature parameter ε. μ, ν, $\mathbf{u}^{(0)}$ and $\mathbf{v}^{(0)}$ can be initialized as a vector of ones with the same size as B, the batch size,

$$\mu = \mathbb{1}, \nu = \mathbb{1} \quad \mathbf{u}^{(0)} = \mathbb{1}, \mathbf{v}^{(0)} = \mathbb{1}.$$

For t iterations of the Sinkhorn algorithm, $\mathbf{u}^{(t)}$ is updated as:

$$\mathbf{u}^{(t+1)} \overset{\text{def}}{=} \frac{\mu}{\mathbf{K}_\theta \mathbf{v}^{(t)}}, \quad \mathbf{v}^{(t+1)} \overset{\text{def}}{=} \frac{\nu}{\mathbf{K}_\theta^T \mathbf{u}^{(t)}}.$$

So we know that:

$$\mathbf{u}^{(1)} = \frac{1}{\sum_{j=1}^{b} \mathbf{K}_{\theta ij}}.$$

Thus, half-step sinkhorn iteration or one-step Bregman interation for $\mathbf{P}$ can be expressed as:

$$\mathbf{P}_{ij} = \mathbf{u}_i^{(1)} \mathbf{K}_{\theta ij} \mathbf{v}_j^{(0)} = \frac{\mathbf{K}_{\theta ij}}{\sum_{j=1}^{b} \mathbf{K}_{\theta ij}} = \frac{\exp\left(\varepsilon^{-1}\langle \mathbf{z}_i, \mathbf{z}_j \rangle\right)}{\sum_{j=1}^{b} \exp\left(\varepsilon^{-1}\langle \mathbf{z}_i, \mathbf{z}_k \rangle\right)} \quad \square$$

This concludes the expressions of $\mathbf{P}$ at half-step iteration. Reminds us the formula of the KL divergence $\mathrm{KL}(\mathbf{I}\|\mathbf{P})$ and the entropy $H(\mathbf{P})$:

$$\mathrm{KL}(\mathbf{I}\|\mathbf{P}) \overset{\text{def}}{=} \sum_{i,j} \mathbf{I}_{i,j} \log \frac{\mathbf{I}_{i,j}}{\mathbf{P}_{i,j}} - \mathbf{I}_{i,j} + \mathbf{P}_{i,j}, \quad \text{where} \quad \mathbf{I}_{i,j} \log \frac{\mathbf{I}_{i,j}}{\mathbf{P}_{i,j}} = 0, \quad \text{if} \quad \mathbf{I}_{i,j} = 0. \tag{64}$$

And after the batch normalization of $\mathbf{P}$, the value of $\sum_{i,j} \mathbf{P}_{i,j}$ is equal to the batch size B and exactly the same as the $\sum_{i,j} \mathbf{I}_{i,j}$, we can obtain:

$$\mathrm{KL}(\mathbf{I}\|\mathbf{P}) = \sum_i \log\left(\frac{1}{\mathbf{P}_{ii}}\right) = -\sum_i \log \frac{\exp\left(\varepsilon^{-1}\langle \mathbf{z}_i, \mathbf{z}_i \rangle\right)}{\sum_{j=1}^{b} \exp\left(\varepsilon^{-1}\langle \mathbf{z}_i, \mathbf{z}_j \rangle\right)}$$

j represents the elements on the diagonal of the similarity matrix, which is the same structure as the INCE loss as:

$$\mathcal{L}_{\text{INCE}} = -\sum_i \log\left(\frac{\exp(f_\theta(\mathbf{x}'_i)^\top f_\theta(\mathbf{x}''_i))}{\sum_{j=1}^{b} \exp(f_\theta(\mathbf{x}'_i)^\top f_\theta(\mathbf{x}''_j))}\right) \quad \square$$

B.4 Proximal operator version of RINCE

In this section, we are going to discuss how to build the equivalence between minimizing the some convex function of d_M with adjustable parameters q and λ between the $\mathbf{P}^{(1)}$ and the $\mathbf{P}_{tgt}$ as:

$$d_M(\mathbf{I},\mathbf{P}) = -\frac{1}{q}\left(\left(\frac{\operatorname{diag}(\mathbf{P}^{(1)}_\theta)}{\mathbf{u}^{(1)}}\right)^q - \left(\frac{\lambda\mathbf{I}}{\mathbf{u}^{(1)}}\right)^q\right) \tag{65}$$

with respect to θ in GCA objective:

$$L^{\lambda,q}_{\text{RINCE}} = \min_\theta -\frac{1}{q}\left(\frac{\operatorname{diag}(\mathbf{P}^{(1)}_\theta)}{\mathbf{u}^{(1)}}\right)^q + \frac{1}{q}\left(\frac{\lambda\mathbf{I}}{\mathbf{u}^{(1)}}\right)^q, \text{ with } \mathbf{P}^{(1)}_\theta = \operatorname{Prox}^{\text{KL}}_{C^\mu_1}(\mathbf{K}_\theta), \ \mathbf{u}^{(1)} = \operatorname{diag}\left(\frac{\mu}{\mathbf{P}^{(0)}\mathbb{1}}\right)$$

with the RINCE loss minimization in Equation (2). Here $\mathbf{P}^{(1)}$ is the nearest point of $\mathbf{K}_\theta$ on constraint set C^μ_1 measured by the KL-divergence d_Γ defined in Equation (18), through one step of proximal operator (Bregman projection). And $\mathbf{K}_\theta$ denote the augmentation kernel as in Definition (3) with cosine similarity.

Also, we are going to discuss when the q=1, RINCE loss is the symmetry loss, which provides the robustness in the noisy view.

B.4.1 Proof of the Theorem 2

The loss function of RINCE looks like:

$$\mathcal{L}^{\lambda,q}_{\text{RINCE}} = \frac{1}{q}\big(-e^{qs_{ii}} + \lambda^q(e^{s_{ii}} + \textstyle\sum_{i\neq j} e^{s_{ij}})^q\big) \tag{66}$$

For the specific parameters θ, we record the normalized latent of the $\mathbf{z}^i_{\theta+} = s_{ii}$, and $\mathbf{z}^i_{\theta-} = s_{ij}, j \neq i$. The positive pairs are stored in the diagonal of the gibbs kernel $\mathbf{K}_\theta$, and the negative pairs are stored in the off-diagonal elements, which means:

$$\mathbf{K}_{ii} = \exp\left(-\varepsilon^{-1}\mathbf{C}_{i,i}\right) = \exp\left(-\varepsilon^{-1}|1-\langle\mathbf{z}'_{\theta i},\mathbf{z}''_{\theta i}\rangle|\right) = \exp\left(\varepsilon^{-1}\langle\mathbf{z}'_{\theta i},\mathbf{z}''_{\theta i}\rangle - \varepsilon^{-1}\right) \propto e^{\mathbf{z}^i_{\theta+}}. \tag{67}$$

$$\mathbf{K}_{ij} = \exp\left(-\varepsilon^{-1}\mathbf{C}_{i,j}\right) = \exp\left(\varepsilon^{-1}\langle\mathbf{z}'_{\theta i},\mathbf{z}''_{\theta j}\rangle - \varepsilon^{-1}\right) \propto e^{\mathbf{z}^i_{\theta-}}, j \neq i. \tag{68}$$

By solving the $\langle\mathbf{u}^{(1)}\mathbf{K},\mathbb{1}\rangle = \mu$ in the Equation (21), we have the ith column elements $\sum_{j=1}\mathbf{K}_{\theta ij} = \frac{\mu}{\mathbf{u}^{(1)}_i}$, in which $\mathbf{u}^{(1)}$ is given in 21:

$$\frac{\mu}{\mathbf{u}^{(1)}_i} = \sum_{j=1}^{B}\mathbf{K}_{\theta ij} = \frac{1}{e^{\varepsilon^{-1}}}(e^{\varepsilon^{-1}\langle\mathbf{z}'_{\theta i},\mathbf{z}''_{\theta i}\rangle} + \sum_{j=1,j\neq i}^{B} e^{\varepsilon^{-1}\langle\mathbf{z}'_{\theta i},\mathbf{z}''_{\theta j}\rangle}), i \neq j, \tag{69}$$

$$\operatorname{diag}(\mathbf{K}_\theta) = \frac{e^{\mathbf{z}^i_{\theta+}}}{e^{\varepsilon^{-1}}} = \frac{\operatorname{diag}(\mathbf{P}^{(1)})}{\mathbf{u}^{(1)}}. \tag{70}$$

The diagonal of K matrix contains the positive views and the marginal distribution of the u contains the negative view, we have:

$$L^{\lambda,q}_{\text{RINCE}}(s^i_\theta) = -\frac{e^{qs^i_{\theta+}}}{q} + \frac{(\lambda\cdot(e^{s^i_{\theta+}} + \sum_{j=1,j\neq i}^{B} e^{s^{ij}_{\theta-}}))^q}{q} \propto -\frac{\operatorname{diag}(\mathbf{K}_\theta)^q_{ii}}{q} + \frac{(\lambda\cdot(\sum_{j=1}^{B}\mathbf{K}_{\theta ij}))^q}{q} \tag{71}$$

Furthermore, we have:

$$-E(L^{\lambda,q}_{\text{RINCE}}(\mathbf{K}_\theta)) = \frac{1}{q}\bigg(\operatorname{diag}(\mathbf{K}_\theta)\bigg)^q - \frac{1}{q}\left(\frac{\lambda\mathbf{I}}{\mathbf{u}^{(1)}}\right)^q. \tag{72}$$

where $\mathbf{P}^{(0)} = \operatorname{diag}(\mathbb{1})\mathbf{K}_\theta\operatorname{diag}(\mathbb{1})$, $\mathbf{P}^{(1)} = \operatorname{diag}(\mathbf{u}^{(1)})\mathbf{K}_\theta\operatorname{diag}(\mathbb{1})$, we have:

$$L^{\lambda,q}_{\text{RINCE}}(\mathbf{P}^{(1)}_\theta) = -\frac{1}{q}(\frac{\operatorname{diag}(\mathbf{P}^{(1)})}{\mathbf{u}^{(1)}})^q + \frac{1}{q}(\frac{\lambda\mathbf{I}}{\mathbf{u}^{(1)}})^q. \tag{73}$$

B.4.2 Proof of the Symmetry and robustness of RINCE

Symmetry loss is said to be noise tolerant as the classifier will keep performance with the label noise in **Empirical Risk Minimization (ERM)**. In many practical machine learning scenarios, we aim to select a model or function f_θ that minimizes the expected loss across all possible inputs and outputs from a distribution $\mathcal{D}$, which is typically unknown. Instead of minimizing the true risk, which is often not feasible due to the unknown distribution $\mathcal{D}$, we minimize what is called the **empirical risk** $\hat{R}_L(\widetilde{f}_\theta)$, which is defined as the average loss over the training dataset of size B, which consists of independently and identically distributed (iid) data points. Mathematically, it is given by the following formula:

$$\hat{R}_L(f_\theta) = \frac{1}{B}\sum_{i=1}^{B} L(\widetilde{f}_\theta(\mathbf{x}_i), \mathbf{y}_i) \tag{74}$$

Here, $L(\widetilde{f}_\theta(\mathbf{x}_i), \mathbf{y}_i)$ represents the loss function, which measures the discrepancy between the predicted value $\widetilde{f}_\theta(\mathbf{x}_i)$ and the true value $\mathbf{y}_i$. The function $\widetilde{f}_\theta$ that minimizes this empirical risk is chosen as the model for making predictions. This approach is based on the assumption that minimizing the empirical risk will also approximate the minimization of the true risk, especially as the size of the training set increases.

First we show the symmetry loss is robust to the noisy view with the following Lemma [20], which means they will achieve the same performance in ERM with the noisy labels. Then we show RINCE satisfy the symmetry condition when $q = 1$, so the lemma is:

Lemma 7. *Give a loss function $L(\widetilde{f}_\theta(\mathbf{x}), \mathbf{y})$ exhibits a certain symmetry for some positive constant K, with respect to the labels $\mathbf{y} = 1$ and $\mathbf{y} = -1$:*

$$L(\widetilde{f}_\theta(\mathbf{x}), 1) + L(\widetilde{f}_\theta(\mathbf{x}), -1) = K, \quad \forall x, \forall f, \quad \textit{(Symmetry)} \tag{75}$$

Symmetry loss is noise tolerant given the label noise $\eta < 0.5$, which corresponds to the flipped labels:

$$P_D[sign(\widetilde{f}^*_\theta(x)) = \mathbf{y}_\mathbf{x}] = P_D[sign(\widetilde{f}^*_{\theta\eta}(\mathbf{x})) = \mathbf{y}_\mathbf{x}], \quad \textit{(Noisy tolerant)} \tag{76}$$

Proof of the Lemma 7 is in [20].

Second we show the RINCE loss is a symmetry loss with $q \to 1$, so we have the Equation (2):

$$L^{\lambda,q=1}_{\text{RINCE}} = -e^{\mathbf{z}^{ii}_{\theta+}} + \lambda \cdot (e^{\mathbf{z}^{ii}_{\theta+}} + \sum_{j=1,j\neq i}^{B} e^{\mathbf{z}^{ij}_{\theta-}}) \tag{77}$$

As we know that this formula has the same structure as the exponential loss function: $L(\mathbf{z}_\theta, \mathbf{y}) = -\mathbf{y}e^{\mathbf{z}_\theta}$. To check for symmetry, we define a new binary classification loss function as:

$$\tilde{L}_x(\mathbf{z}_\theta(\mathbf{x}), \mathbf{y}) = B + L_x(\widetilde{f}_\theta(\mathbf{x}), \mathbf{y}) = B - \mathbf{y} \cdot e^{\widetilde{f}_\theta(\mathbf{x})} \geq 0$$

where the prediction score $\widetilde{f}_\theta(\mathbf{x})$ is bounded by $s_{\text{max}} = \log(B)$. Then we can establish that the loss satisfies the symmetry property:

$$\tilde{L}(\widetilde{f}_\theta(\mathbf{x}), 1) + \tilde{L}(\widetilde{f}_\theta(\mathbf{x}), -1) = 2B \tag{78}$$

So we prove that this loss function is symmetry.

B.5 Proof for RINCE is the upper bound of the 1-Wasserstein distance

In this section, we are trying to build the connection when change the d_M from the KL-divergence in Equation (10) to the 1-Wasserstein distance in Equation (12), when q=1 in the RINCE loss.

B.5.1 Proof of the Theorem 3

WDM [12] is proposed as a replacement for the KL divergence by Wasserstein distance in Mutual Information estimation. The Wasserstein distance between the joint distribution π on $\mathcal{X} \times \mathcal{Y}$ and the product of the marginal distributions μ and ν on $\mathcal{X}$ and $\mathcal{Y}$, respectively, is given by:

$$W(\pi, \mu \otimes \nu) = \sup_{f \in \mathcal{C}(\mathcal{X}\times\mathcal{Y})} \left(\mathbb{E}_{\pi(x,y)}[f(x,y)] - \mathbb{E}_{\mu\otimes\nu(x,y)}[f(x,y)]\right)$$

where $\mathcal{C}(\mathcal{X} \times \mathcal{Y})$ denotes the set of all 1-Lipschitz functions from $\mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$. A function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is defined to be 1-Lipschitz if, for any two points $(x_1, y_1), (x_2, y_2) \in \mathcal{X} \times \mathcal{Y}$, the following condition is satisfied:

$$|f(x_1, y_1) - f(x_2, y_2)| \leq d((x_1, y_1), (x_2, y_2))$$

where $d((x_1, y_1), (x_2, y_2))$ denotes the metric on $\mathcal{X} \times \mathcal{Y}$ typically defined, for example, by the Euclidean distance:

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Based on the Lipschitz continuity and inner product, it is easy to know for two given point (x_1, y_1), (x_2, y_2), the following properties hold with $-\frac{1}{\varepsilon} \leq s \leq \frac{1}{\varepsilon}$, which implies $|\nabla_s e^s| \leq e^{1/\varepsilon}$. Therefore, by the mean value theorem, we have:

$$\begin{aligned} &|e^{x_1^T y_1/\varepsilon} - e^{x_2^T y_2/\varepsilon}| \leq e^{1/\varepsilon}\frac{1}{\varepsilon}|\langle x_1, y_1\rangle - \langle x_2, y_2\rangle| = e^{1/\varepsilon}\frac{1}{\varepsilon}|\langle x_1 - x_2, y_1\rangle + \langle x_2, y_1 - y_2\rangle| \\ &\leq e^{1/\varepsilon}\frac{1}{\varepsilon}\left(\|x_1 - x_2\|\|y_1\| + \|y_1 - y_2\|\|x_2\|\right) = e^{1/\varepsilon}\frac{1}{\varepsilon}\left(\|x_1 - x_2\| + \|y_1 - y_2\|\right) \end{aligned} \tag{79}$$

Consider two pairs of views, $(\mathbf{z}'_{\theta 1}, \mathbf{z}''_{\theta 1})$ and $(\mathbf{z}'_{\theta 2}, \mathbf{z}''_{\theta 2})$, sampled from the joint distribution π of μ and ν. Thus, each pair $(\mathbf{z}'_{\theta i}, \mathbf{z}''_{\theta i})$ for $i = 1, 2$ represents a sample from the joint distribution π, where $\mathbf{z}'_{\theta i} \sim \mu$ and $\mathbf{z}''_{\theta i} \sim \nu$. The RINCE loss is a symmetry loss with $q = 1$, so we have the Equation (2):

$$L_{\text{RINCE}}^{\lambda, q=1} = -e^{\mathbf{z}_{\theta +}^{ii}} + \lambda \cdot (e^{\mathbf{z}_{\theta +}^{ii}} + \sum_{j=1, j\neq i}^{B} e^{\mathbf{z}_{\theta -}^{ij}}), \begin{cases} \mathbf{z}_{\theta +}^{ii} &= \varepsilon^{-1}\widetilde{f}_\theta(\mathbf{x}'_i)^\top \widetilde{f}_\theta(\mathbf{x}''_i), \quad \text{for } i = i, \\ \mathbf{z}_{\theta -}^{ij} &= \varepsilon^{-1}\widetilde{f}_\theta(\mathbf{x}'_i)^\top \widetilde{f}_\theta(\mathbf{x}''_j), \quad \text{for } i \neq j. \end{cases} \tag{80}$$

So we know that:

$$\begin{aligned} &- \mathbb{E}(L_{\text{RINCE}}^{\lambda, q=1}(z_\theta)) = \mathbb{E}_{\substack{\mathbf{z}'_{\theta i} \sim \mu \\ \mathbf{z}''_{\theta i} \sim \nu | \mu = \mathbf{z}'_{\theta i} \\ \mathbf{z}''_{\theta j} \sim \nu}} \left[(1-\lambda)e^{\varepsilon^{-1}\mathbf{z}'^T_{\theta i}\mathbf{z}''_{\theta i}} - \lambda \sum_{j=1}^{B-1} e^{\varepsilon^{-1}\mathbf{z}'^T_{\theta i}\mathbf{z}''_{\theta j}}\right] \\ &= \mathbb{E}_{(\mathbf{z}'_{\theta i}, \mathbf{z}''_{\theta i}) \sim \pi}\left[(1-\lambda)e^{\frac{\mathbf{z}'^T_{\theta i}\mathbf{z}''_{\theta i}}{\varepsilon}}\right] - \lambda(B-1)\mathbb{E}_{\mathbf{z}'^T_{\theta i} \sim \mu, \mathbf{z}''^T_{\theta j} \sim \nu}\left[e^{\frac{\mathbf{z}'^T_{\theta i}\mathbf{z}''_{\theta j}}{\varepsilon}}\right] \\ &\leq (1-\lambda)\left(\mathbb{E}_{(\mathbf{z}'_\theta, \mathbf{z}''_\theta) \sim \pi}\left[e^{\frac{\mathbf{z}'^T_\theta \mathbf{z}''_\theta}{\varepsilon}}\right] - \mathbb{E}_{\mathbf{z}'^T_\theta \sim \mu, \mathbf{z}''_\theta \sim \nu}\left[e^{\frac{\mathbf{z}'^T_\theta \mathbf{z}''_\theta}{\varepsilon}}\right]\right) \text{(Giving setting } \lambda(B-1) > 1-\lambda) \end{aligned}$$

If we give two couples of two views $(\mathbf{z}'_{\theta 1}, \mathbf{z}''_{\theta 1})$ and $(\mathbf{z}'_{\theta 2}, \mathbf{z}''_{\theta 2})$ from joint distribution π of μ and ν, $\mathbf{z}'_\theta \sim \mu$ and $\mathbf{z}''_\theta \sim \nu$, which means to maximize:

$$\begin{aligned} &|e^{\varepsilon^{-1}\mathbf{z}'^T_{\theta 1}\mathbf{z}''_{\theta 1}} - e^{\varepsilon^{-1}\mathbf{z}'^T_{\theta 2}\mathbf{z}''_{\theta 2}}| \\ &\leq (1-\lambda)e^{\frac{1}{\varepsilon}}\frac{1}{\varepsilon}\left(\|\mathbf{z}'_{\theta 1} - \mathbf{z}'_{\theta 2}\|\|\mathbf{z}''_{\theta 1}\| + \|\mathbf{z}''_{\theta 1} - \mathbf{z}''_{\theta 2}\|\|\mathbf{z}'_{\theta 2}\|\right) \text{(Mean value theorem from Equation (79))} \\ &= (1-\lambda)e^{\frac{1}{\varepsilon}}\frac{1}{\varepsilon}\left(\|\mathbf{z}'_{\theta 1} - \mathbf{z}'_{\theta 2}\|_2 + \|\mathbf{z}''_{\theta 1} - \mathbf{z}''_{\theta 2}\|_2\right) \\ &= (1-\lambda)e^{\frac{1}{\varepsilon}}\frac{1}{\varepsilon}d\left((\mathbf{z}'_{\theta 1}, \mathbf{z}''_{\theta 1}), (\mathbf{z}'_{\theta 2}, \mathbf{z}''_{\theta 2})\right) \\ &\leq (1-\lambda)e^{1/\varepsilon}\frac{1}{\varepsilon}W_1(\pi, \mu \otimes \nu). \end{aligned}$$

B.6 Proof of connection with BYOL

In this section, we are going to who how the change of the augmetation kernel from the $\mathbf{K}_\theta$ in Definition (3) into the BYOL kernel $\mathbf{S}_\theta$ would lead to the BYOL loss.

Proof for the Theorem 4

BYOL has the online network parameterized by θ and target network parameterized by ξ, where $\mathbf{z}'_\theta = \widetilde{f}_\theta(\mathbf{x}')$ and $\mathbf{z}''_\xi = \widetilde{f}_\xi(\mathbf{x}'')$ are the normalized outputs of the online and target networks, respectively. The kernel of BYOL looks like:

$$\mathbf{S}_\theta(\mathbf{x}'_i, \mathbf{x}''_j) = \exp(-\langle \widetilde{q}_\theta(\widetilde{f}_\theta(\mathbf{x}'_i)), \widetilde{f}_\xi(\mathbf{x}''_j)\rangle),$$

The kernel here involves both the parameters θ and ξ, however, the target network has the stop gradient. Therefore, the only θ needs to be updated, so we can rewrite the kernel as $\mathbf{S}_\theta(\mathbf{x}'_i, \mathbf{x}''_j)$ as we show in the main text. As we give in the equation, the corresponding proximal operators evolving with d_Γ is equal to L2-distance has the formula, and $h(x) = 0$ for all $\mathbf{P} \in \mathcal{R}^{B\times B}$:

$$\text{Prox}^{\|\cdot\|^2}_{\mathcal{R}^{B\times B}}(\mathbf{S}_\theta) = \arg \min_{\mathbf{P}\in\mathcal{R}^{B\times B}} \left\{ h(\mathbf{P}) + \frac{1}{2}\|\mathbf{P} - \mathbf{S}_\theta\|_2^2 \right\} \Rightarrow \mathbf{P} = \mathbf{S}_\theta$$

The BYOL loss can be written as normalized L2-distance between the normalized output after online network $\widetilde{q}_\theta(\mathbf{z}'_\theta)$ in which $\widetilde{q}_\theta$ is predictor and the stop gradient results for the target network $\widetilde{q}_\theta(\mathbf{z}')$, and the formula of BYOL object reads as $L_{\text{BYOL}} = \|\widetilde{q}_\theta(\mathbf{z}'_\theta) - \mathbf{z}''_\xi\|_2^2$.

In this case, there exists equivalence between

$$\text{KL}(\mathbf{I}\|\mathbf{S}_\theta) = -\sum_i^B \log \mathbf{S}_{\theta ii} = \sum_i^B \|\widetilde{q}_\theta(\mathbf{z}'_\theta) - \mathbf{z}''_\xi\|_2^2 \tag{81}$$

which is the BYOL loss.

B.7 Complexity Analysis for GCA

In the forward pass, iteratively running the GCA does not involve inner optimization for gradient back-propagation. In the Sinkhorn algorithm, the transport plan $\mathbf{P}_\theta$ is computed as:

$$\mathbf{P}_\theta = \exp(f + g - \mathbf{C}_\theta)/\epsilon,$$

where f and g are dual variables iteratively updated in the Sinkhorn algorithm but do not involve gradients with respect to θ. The Sinkhorn optimization primarily entails scaling the rows and columns of $\mathbf{P}$ to satisfy the marginal constraints, which can be viewed as element-wise operations (scaling and exponentiation) on the cost matrix $\mathbf{C}_\theta$.

Since $\mathbf{P}_\theta$ is computed through the fixed-point iteration of f and g that depend only on the current values of $\mathbf{C}_\theta$, the gradient back-propagation process is simplified. Specifically, the gradient of the loss with respect to the cost matrix $\mathbf{C}_\theta$ is the key part that needs to be differentiated, rather than through each iterative update of f and g. A typical workflow of these algorithms was shown in Figure 2 of [17], the gradient flow primarily involves differentiating through $\mathbf{C}_\theta$, which is done only once, and not through each step of the Sinkhorn iterations. This approach reduces computational complexity and avoids the need for back-propagation through every iterative update within the Sinkhorn algorithm, which might otherwise be computationally expensive.

C Proofs that GCA methods improve the alignment and uniformity

C.1 Improving Alignment

In this section, we are going to show the GCA methods minimize the difference between the target alignment plan with the coupling matrix on latent. The uniformity and alignment loss have been used to exam the quality of the representation in self-supervised learning, which is defined as the following [57]:

Definition 9 (Alignment loss)**.** *Given π as joint distribution of positive samples on the latent, $(\mathbf{z}'_{\theta i}, \mathbf{z}''_{\theta i})$ are the normalized positive pairs sampled from the joint distribution π with encoder parameterized by θ, the alignment loss is:*

$$\mathcal{L}_{align} = \min_\theta \mathbb{E}_{(\mathbf{z}'_{\theta i}, \mathbf{z}''_{\theta i})\sim\pi}\left[\|\mathbf{z}'_{\theta i} - \mathbf{z}''_{\theta i}\|_2^2\right] = \min_\theta \sum_i \text{diag}(\mathbf{C}_{ii}), \tag{82}$$

where $\mathbf{C}$ is the cost matrix defined in Equation (24).

We can alter the constraint sets of proximal operators to provide the better alignment plans, i.e. GCA-INCE changes the constraint sets by considering both row and column normalization in coupling matrix Rather than just the row normalization. Such change will not affect the alignment loss in forward pass, it will benefit the alignment loss in the backward pass through a tighter bound of empirical risk minimization with the identity matrix.

C.1.1 Proof of the tighter bound of GCA in ERM

In this section, we provide the evidence for using the converged coupling plan $\mathbf{P}_\theta^{(\infty)}$ is better than the $\mathbf{P}_\theta^{(1)}$ or $\mathbf{P}_\theta^{(t)}$ in Equation (18) for the GCA-methods loss in table 1. This loss function will correspond to different alignment loss on the latent. And here the ERM is the definition as we provided in Appendix B.4.2.

Lemma 8. *Denote $f^{(t_1)}$ and $g^{(t_2)}$ the two dual variables in their t_1 and t_2 iterations, respectively. Then the objective loss in Equation (1) could be written as* $\mathrm{KL}(\mathbf{I}\|\mathbf{P}_\theta) = \varepsilon^{-1}(\mathrm{diag}(\mathbf{C}) - (f^{(t_1)} + g^{(t_2)}))$.

Proof of the Lemma 8:

The above Lemma 8 be derived form Equation (23). Recall that $\mathbf{u} = \exp{(f/\varepsilon)}, \quad \mathbf{v} = \exp{(g/\varepsilon)}, \quad \mathbf{K} = \exp{(-\mathbf{C}/\varepsilon)}$

$$\mathrm{KL}(\mathbf{I}\|\mathbf{P}_\theta) = -\sum_i \log{(\mathbf{P}_{ii})} = -\sum_i (\log \mathrm{diag}(\mathbf{u})_{ii} + \log \mathbf{K}_{ii} + \log \mathrm{diag}(\mathbf{v})_{ii})$$
$$= -\varepsilon^{-1}\sum_i (f_i - C_{ii} + g_i) = \varepsilon^{-1}(\mathrm{diag}(C) - (f+g)) \quad \square$$

Here we provide the proof of the *Best Alignment* in **Theorem 5:**

Proof of the Theorem 5

Based on the Lemma 8, to show $\mathrm{KL}(\mathbf{I}\|\mathbf{P}^{(\infty)}) \leq \mathrm{KL}(\mathbf{I}\|\mathbf{P}^{(1)})$. We have to show:

$$\varepsilon^{-1}(\mathrm{diag}(C) - (f^{(\infty)} + g^{(\infty)})) \leq \varepsilon^{-1}(\mathrm{diag}(C) - (f^{(1)} + g^{(1)})).$$

Then give the Lemma 3, we know the $f^{(t_1)}$ and $g^{(t_2)}$ increase and converge weakly to their upper bound. As the $\mathrm{diag}(\mathbf{C})$ will be unchanged in each proximal operations, we know the objective function $\mathrm{KL}(\mathbf{I}\|\mathbf{P}_\theta)$ have lower upper bound with $f^{(t_1)}$ and $g^{(t_2)}$ increase and finally converged. □

Based on the Lemma 8, We have to show:

$$\varepsilon^{-1}(\mathrm{diag}(C) - (f^{(t)} + g^{(t)})) \geq \varepsilon^{-1}(\mathrm{diag}(C) - (f^{(\infty)} + g^{(\infty)})).$$

Then give the Lemma 3, when $t_1 \to \infty$, $f^{(t_1)}$ converge uniformly to a fixed point $f^{(\infty)}$ with $f^{(t_1)} \leq f^{(\infty)}$. So we know the $f^{(t_1)}$ and $g^{(t_2)}$ increase and converge weakly to their upper bound. As the $\mathrm{diag}(\mathbf{C})$ will be unchanged in each proximal operations, we know the objective function finally converged to $f^{(\infty)}$ and $g^{(\infty)}$ (Similarly, we prove $g^{(t_1)} \leq g^{(\infty)}$ in Appendix B.1.3). □

C.1.2 Proof of the Theorem 6:

To show: $L^{\lambda,q=1,\varepsilon}_{\text{GCA-RINCE}}(\mathbf{P}_\theta^{(t)}) \leq L^{\lambda,q=1,\varepsilon}_{\text{RINCE}}(\mathbf{P}_\theta^{(1)})$.

we know that:

$$L^{\lambda,q=1,\varepsilon}_{\text{GCA-RINCE}}(\mathbf{P}_\theta^{(t)}) = -\frac{\mathrm{diag}(\mathbf{P}^{(t)})}{\mathbf{u}^{(t)}} + \frac{\lambda \mathbf{I}}{\mathbf{u}^{(t)}}, \tag{83}$$

$$L^{\lambda,q=1,\varepsilon}_{\text{RINCE}}(\mathbf{P}_\theta^{(1)}) = -\frac{\mathrm{diag}(\mathbf{P}^{(1)})}{\mathbf{u}^{(1)}} + \frac{\lambda \mathbf{I}}{\mathbf{u}^{(1)}}, \tag{84}$$

Given that the Lemma 3, we know $\{\mathbf{u}^{(t)}\}$ and $\{\mathbf{v}^{(t)}\}$ are a monotonically increasing sequence where

$$\mathbf{u}^{(1)} \leq \mathbf{u}^{(t)} \Rightarrow \frac{\lambda \mathbf{I}}{\mathbf{u}^{(1)}} \geq \frac{\lambda \mathbf{I}}{\mathbf{u}^{(t)}} \tag{85}$$

$$-\mathrm{diag}(\mathbf{K}_\theta)\mathbf{v}^{(0)} \geq -\mathrm{diag}(\mathbf{K}_\theta)\mathbf{v}^{(t)} \Rightarrow -\frac{\mathrm{diag}(\mathbf{P}^{(1)})}{\mathbf{u}^{(1)}} \geq -\frac{\mathrm{diag}(\mathbf{P}^{(t)})}{\mathbf{u}^{(t)}} \tag{86}$$

Combine the above two items, we have the equation like $L^{\lambda,q=1,\varepsilon}_{\text{GCA-RINCE}}(\mathbf{P}^{(t)}_{\theta}) \leq L^{\lambda,q=1,\varepsilon}_{\text{RINCE}}(\mathbf{P}^{(1)}_{\theta})$.

C.2 GCA methods improve the uniformity and benefit downstream classification tasks

In this section, we provide theoretical evidence that the GCA approaches could improve the performance of downstream task, i.e. classification tasks, by providing the maximum uniformity through solving the EOT, as **Theorem** (7) stated. Here, the uniformity loss is defined as [50]:

Definition 10 (Uniformity loss)**.** *Let $\mathbf{z}'_{\theta i} \sim \mu$ and $\mathbf{z}''_{\theta j} \sim \nu$ in which μ and ν are two distributions on the representation space, we define the uniformity loss as the following:*

$$L_{uniform} = \log \mathbb{E}_{\mathbf{z}'_{\theta i}, \mathbf{z}''_{\theta j}\ i.i.d. \sim p_{data}}[e^{-\varepsilon\|\mathbf{z}'_{\theta i} - \mathbf{z}''_{\theta j}\|_2^2}] \tag{87}$$

, in which $p_{data}(\cdot)$ is the sample distribution over latent space $\mathbb{R}^n$.

Here, $p_{\text{data}}(\cdot)$ should be the marginal distribution of the samples. As the $\mathbf{z}'_{\theta i}$ and $\mathbf{z}''_{\theta} j$ are normalized latent variables, we have the right items of the uniformity loss $e^{-\varepsilon\|\mathbf{z}'_{\theta i} - \mathbf{z}''_{\theta j}\|_2^2}$ is the same as the entropy-regularized kernel $\mathbf{K}_{ij} = e^{-\varepsilon \mathbf{C}_{ij}}$ with cost matrix items $\mathbf{C}_{ij} = \|\mathbf{z}'_{\theta i} - \mathbf{z}''_{\theta j}\|_2^2$.

C.2.1 Proof of the Theorem 7

Here we are going to compare two different coupling plans, $\mathbf{P}^{(1)}_{\theta}$ and $\mathbf{P}^{(\infty)}_{\theta}$, and show the converged plan $\mathbf{P}^{(\infty)}_{\theta}$ will achieve higher the uniformity after the forward pass. The general logic is that we show the equivalence for the solving EOT with the minimizing the uniformity loss objective. Then we use the convergence of iterative Bregman projections to show it could achieve higher uniformity.

Based on the Entropy regularized OT defined in Definition 8, we have:

$$W_{c,\varepsilon}(\mu, \nu) := \min_{\pi \in \Pi(\mu,\nu)} \int_{X \times Y} c(x, y) d\pi(x, y) + \varepsilon H(\pi|\mu \otimes \nu) \tag{88}$$

in which the entropy could be defined as:

$$H(\pi|\mu \otimes \nu) := \int_{X \times Y} \left(\log \left(\frac{d\pi(x, y)}{d\mu(x) d\nu(y)} \right) - 1 \right) d\pi(x, y) + 1, \tag{89}$$

is the relative entropy of the transport plan π with respect to the product measure $\mu \otimes \nu$. So the corresponding dual problem of this EOT one is shown in the following formula:

$$W_{c,\varepsilon}(\mu, \nu) = \max_{f \in C(\mathcal{X}), g \in C(\mathcal{Y})} \int_{\mathcal{X}} f(x)\, d\mu(x) + \int_{\mathcal{Y}} g(y)\, d\nu(y) \tag{90}$$

$$- \varepsilon \int_{\mathcal{X} \times \mathcal{Y}} e^{\frac{f(x)+g(y)-c(x,y)}{\varepsilon}} d\mu(x) d\nu(y) + \varepsilon \tag{91}$$

$$= \max_{f \in C(\mathcal{X}), g \in C(\mathcal{Y})} \mathbb{E}_{\mu \otimes \nu} \left[f(x) + g(y) - e^{\frac{f(x)+g(y)-c(x,y)}{\varepsilon}} \right] + \varepsilon \tag{92}$$

The $\mu(x)$ and $\nu(x)$ are defined as the uniformly distribution with Dirac delta function we have on the two latent supports $\{\mathbf{z}'_{\theta i}\}_{i=1}^{B}$ and $\{\mathbf{z}''_{\theta i}\}_{i=1}^{B}$, so the function $f(x)$ and $g(y)$ could be pull out of the expectation operators. Since the $\|\mathbf{z}'_{\theta i} - \mathbf{z}''_{\theta j}\|_2^2$ is the element in the cost matrix $\mathbf{C}_{ij}$, which is computed through the cost function $c(x, y)$. As the $\mathbf{z}'_{\theta i}$ and $\mathbf{z}''_{\theta j}$ are drawn independently from the latent distribution, so the remaining item $\mathbb{E}_{\mu \otimes \nu}[e^{\frac{-c(x,y)}{\epsilon}}]$ is equivalent to the uniformity loss. The the above integral could be turned into the sum of the elements in matrix of dual variables of $f^{(t_1)}$ and$g^{(t_1)}$ in each iteration. Meanwhile, based on the convergence provided in the Lemma 3, When $t_1 \to \infty$, $f^{(t_1)}$ converge uniformly to a fixed point $f^{(\infty)}$ with $f^{(t_1)} \leq f^{(\infty)}$, which would provided the maximum value of the dual formula in the $f^{(\infty)}$, which corresponding to the coupling plan the $\mathbf{P}^{(\infty)}$. □

C.2.2 GCA benefits the downstream supervised classification task

Here, we further show how the minimizing the uniformity loss is equivalent to minimize the downstream supervised loss in classification tasks under several assumptions [16]. Giving a labeled dataset $\mathcal{D} = \{(\bar{\mathbf{x}}_i, \mathbf{y}_i)\} \in \bar{\mathcal{X}} \times \mathcal{Y}$ where $\mathcal{Y} = [1..M]$ with M classes, we consider a fixed, pre-trained encoder $f_\theta \in \mathcal{F} : \mathcal{X} \to \mathcal{S}$ with its representation $f_\theta(\mathcal{X})$ and the input space $\mathcal{X}$ contains both positive and negative views of n original samples $(\bar{\mathbf{x}}_i)_{i\in[1..n]} \in \bar{\mathcal{X}}$, sampled from the data distribution $p(\bar{\mathbf{x}})$. For each positive views $\bar{\mathbf{x}}'_i$ in $\mathcal{X}$, we sample from $\bar{\mathbf{x}}_i$ using $\mathbf{x}'_i \sim \mathcal{A}(\cdot|\bar{\mathbf{x}}_i)$, $\mathcal{A}(\cdot|\bar{\mathbf{x}}_i)$ is augmentation distribution (e.g., by applying color jittering, flip, or crop with a given probability). For consistency, we assume $\mathcal{A}(\bar{\mathbf{x}}) = p(\bar{\mathbf{x}})$ so that the distributions $\mathcal{A}(\cdot|\bar{\mathbf{x}})$ and $p(\bar{\mathbf{x}})$ induce a marginal distribution $p(\mathbf{x})$ over $\mathcal{X}$. Given an anchor $\bar{\mathbf{x}}_i$, all views $\mathbf{x}'' \sim \mathcal{A}(\cdot|\bar{\mathbf{x}}_j), j \neq i$ from different samples $\bar{\mathbf{x}}_j$ are considered as negatives.

Proof of claim 1: From assumption 1 we know that the representation ability of encoders is good enough via the augmented samples in the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_{\bar{\mathcal{X}}}$ of the original sample spaces $\bar{\mathcal{X}}$. And the kernel $K_{\bar{\mathcal{X}}}$ with any function g RKHS defined by $(\mathcal{H}_{f_\theta}, \mathbf{K}_\theta)$ also belongs to $H_{\bar{\mathcal{X}}}$ when conditioned on the distribution $\mathcal{A}(\mathbf{x}|\cdot)$. So based on the assumption we have, we can obtain a centroid estimator by [16]:

Definition 11 (Kernel-based centroid estimator). *Let* $(\mathbf{x}_i, \bar{\mathbf{x}}_i)_{i\in[1..n]} \sim \mathcal{A}(\mathbf{x}, \bar{\mathbf{x}})$, *asssuming a consistent estimator of* $\mu_{\bar{\mathbf{x}}}$ *is.*

$$\forall \bar{\mathbf{x}} \in \bar{\mathcal{X}}, \hat{\mu}_{\bar{\mathbf{x}}} = \sum_{i=1}^{n} \alpha_i(\bar{\mathbf{x}}) f(\mathbf{x}_i),$$

where $\alpha_i(\bar{\mathbf{x}}) = \sum_{j=1}^{n} [(\mathbf{K}_n + n\lambda \mathbf{I}_n)^{-1}]_{ij} \mathbf{K}_{\bar{\mathcal{X}}}(\bar{\mathbf{x}}_j, \bar{\mathbf{x}})$ *and* $\mathbf{K}_n = [\mathbf{K}_{\bar{\mathcal{X}}}(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j)]_{i,j\in[1..n]}$. *It converges to* $\mu_{\bar{\mathbf{x}}}$ *with the* ℓ_2 *norm at a rate* $\mathcal{O}(n^{-1/4})$ *for* $\lambda = \mathcal{O}(n^{-1/2})$.

The above estimator allows us to use representations of images close to an anchor $\bar{\mathbf{x}}$ to estimate $\mu_{\bar{\mathbf{x}}}$. From the assumption 2, we assume that all the samples in the same class is achievable when give the ideal augmentation or at least close to the augmented points in an ϵ region.

Consequently, if the prior is "good enough" to connect intra-class images disconnected in the augmentation graph suggested by Assumption 1, then this estimator allows us to tightly control the classification risk of the representation of f_θ on a classification task with a linear classifier $g(\bar{\mathbf{x}}) = \mathcal{W} f_\theta(\bar{\mathbf{x}})$ (with f_θ fixed) that minimizes the multi-class classification loss.

First we show the cross-entropy could be transformed into centroid based distance (optimal supervised loss): The cross-entropy (CE) to measure the difference between the true distribution (actual labels) and the estimated probability distribution (predicted probabilities from the model), which usually computes logits $\mathbf{z}_k$ from the model, then apply the softmax function to obtain probabilities p_k. The logits $\mathbf{z}_k$ could be defined as negative distances between $f(\bar{\mathbf{x}})$ and class centroids μ_k after the representation:

$$\mathbf{z}_k = -\|f(\bar{\mathbf{x}}) - \mu_k\|^2, \quad \mu_k = \mathbb{E}_{p(\bar{\mathbf{x}}|\mathbf{y}=k)} \mu_{\bar{\mathbf{x}}}$$

which encourages the model to reduce the distance to the correct class centroid while increasing distances to others. The probability of class k in M classes given input $\bar{\mathbf{x}}$ is:

$$p(\mathbf{y} = k|\bar{\mathbf{x}}) = \frac{e^{\mathbf{z}_k}}{\sum_{j=1}^{M} e^{\mathbf{z}_j}}, \quad p(\mathbf{y}|\bar{\mathbf{x}}) \propto e^{-\|f(\bar{\mathbf{x}})-\mu_{\mathbf{y}}\|^2}.$$

If the model predictions $p(\mathbf{y}|\bar{\mathbf{x}})$ are influenced by the distances between $\bar{\mathbf{x}}$ and the class centroids $\mu_{\mathbf{y}}$, then minimizing cross-entropy indirectly affects these distances. The standard CE loss in supervised learning for classification tasks is:

$$\mathcal{L}_{\text{CE}}(f_\theta) = -\mathbb{E}_{(\bar{\mathbf{x}},\mathbf{y})\sim\mathcal{D}} \left[\log p(\mathbf{y}|\bar{\mathbf{x}})\right] \tag{93}$$

$$= -\mathbb{E}_{(\bar{\mathbf{x}},\mathbf{y})\sim\mathcal{D}} \left[-\|f(\bar{\mathbf{x}}) - \mu_{\mathbf{y}}\|^2 - \log Z\right] = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{M} \mathbf{y}_{i,k} \log(p_{i,k}) \tag{94}$$

which focuses on maximizing the likelihood $\hat{\mathbf{y}} = \arg\max_k p(\mathbf{y} = k|\bar{\mathbf{x}})$ of the correct class for each individual sample $\bar{\mathbf{x}}_i$, where $y_{i,k}$ is the true label indicator for example i and class k, $p_{i,k}$ is the predicted probability for example i and class k. Therefore, we can rewrite the CE loss as optimal supervised loss in [16], which is defined as:

Lemma 9 (Optimal supervised loss)**.** *Let a downstream task D with M classes. We assume that $M \leq d+1$ (i.e., a big enough representation space), that all classes are balanced and the realizability of an encoder $f^* = \arg\min_{f \in F} \mathcal{L}_{sup}(f_\theta)$ with*

$$\mathcal{L}_{sup}(f_\theta) = \log \mathbb{E}_{y,y' \sim p(y)p(y')} \left[e^{-\|\mu_y - \mu_{y'}\|^2} \right],$$

and $\mu_y = \mathbb{E}_{p(\bar{x}|y)} \mu_{\bar{x}}$. Then the optimal centroids $(\mu_y^*)_{y \in Y}$ associated to f^* make a regular simplex on the hypersphere S^{d-1} and they are perfectly linearly separable, i.e.,

$$\min_{(w_y)_{y \in \mathcal{Y}} \in \mathbb{R}^d} \mathbb{E}_{(\bar{x},y) \sim D} \mathbb{1}(w_y \cdot \mu_y^* < 0) = 0.$$

Proof of the Lemma 9 All "labeled" centroids $\mu_y = \mathbb{E}_{p(\bar{x}|y)} \mu_{\bar{x}}$ are bounded by 1 ($\|\mu_y\| \leq \mathbb{E}_{p(\bar{x}|y)} \mathbb{E}_{A(x|x')} \|f(x)\| = 1$ by Jensen's inequality). Then, since all classes are balanced, we can re-write the supervised loss as:

$$\mathcal{L}_{\text{sup}}(f_\theta) = \log \frac{1}{C^2} \sum_{y,y'=1}^{C} e^{-\|\mu_y - \mu_{y'}\|^2}.$$

We have:

$$\Gamma_{\mathcal{Y}}(\mu) := \sum_{y,y'} \|\mu_y - \mu_{y'}\|^2 = \sum_{y,y'} \|\mu_y\|^2 + \|\mu_{y'}\|^2 - 2\mu_y \cdot \mu_{y'} \leq \sum_{y,y'} (2 - 2\mu_y \cdot \mu_{y'}) = 2C^2 - 2\|\sum_y \mu_y\|^2 \leq 2C^2,$$

with equality if and only if $\sum_{y=1}^{C} \mu_y = 0$ and $\forall y \in [1..C], \|\mu_y\| = 1$. By the strict convexity of $u \to e^{-u}$, we have:

$$\sum_{y \neq y'} \exp(-\|\mu_y - \mu_{y'}\|^2) \geq C(C-1) \exp\left(-\frac{\Gamma_Y(\mu)}{C(C-1)}\right) \geq C(C-1) \exp\left(-\frac{2C}{C-1}\right),$$

with equality if and only if all pairwise distances $\|\mu_y - \mu_{y'}\|$ are equal (equality case in Jensen's inequality for a strict convex function), $\sum_{y=1}^{C} \mu_y = 0$, and $\|\mu_y\| = 1$. Thus, all centroids must form a regular $(C-1)$-simplex inscribed on the hypersphere S^{d-1} centered at 0. Furthermore, since $\|\mu_y\| = 1$, we have equality in Jensen's inequality:

$$\|\mu_y\| = \|\mathbb{E}_{\mathcal{A}(\mathbf{x}|\bar{\mathbf{x}}')} f_\theta(x)\| \leq \mathbb{E}_{\mathcal{A}(\mathbf{x}|\bar{\mathbf{x}}')} \|f_\theta(x)\| = 1,$$

so f must be perfectly aligned for all samples belonging to the same class: $\forall x, \bar{x}' \sim p(\cdot|y)$, $f_\theta(\bar{x}) = f_\theta(\bar{x}')$. □

Seond we show optimizing the uniformity loss is equivalent to the supervised loss:

As we have uniformity Loss defined in Equation (87)

$$L_{\text{uniform}}(f_\theta) = \log \mathbb{E}_{\mathbf{z}_i', \mathbf{z}_j'' \sim p_{\text{data}}} \left[e^{-\varepsilon \left\|\mathbf{z}_i' - \mathbf{z}_j''\right\|^2} \right], \tag{95}$$

where $\mathbf{z}_i' = f(\mathbf{x}_i)$ and $\mathbf{z}_j'' = f(\mathbf{x}_j)$. Supervised Loss:

$$\mathcal{L}_{\text{sup}}(f_\theta) = \log \mathbb{E}_{y,y' \sim p(y)p(y')} \left[e^{-\left\|\mu_y - \mu_{y'}\right\|^2} \right],$$

where $\mu_y = \mathbb{E}_{p(\bar{\mathbf{x}}|y)} \hat{\mu}_{\bar{\mathbf{x}}}$. Express the expectation over all pairs in terms of class labels:

$$\mathbb{E}_{\mathbf{z}_i', \mathbf{z}_j''} = \mathbb{E}_{y,y'} \mathbb{E}_{\mathbf{z}_i' \sim p(\mathbf{z}|y), \mathbf{z}_j'' \sim p(\mathbf{z}|y')}.$$

So the uniformity loss could be decomposed into intra-class and inter-class components:

$$L_{\text{uniform}}(f_\theta) = \log \left(\underbrace{\mathbb{E}_y \left[\mathbb{E}_{\mathbf{z}_i', \mathbf{z}_j'' \sim p(\mathbf{z}|y)} \left[e^{-\varepsilon \left\|\mathbf{z}_i' - \mathbf{z}_j''\right\|^2} \right] \right]}_{\text{Intra-Class Term}} + \underbrace{\mathbb{E}_{y \neq y'} \left[\mathbb{E}_{\mathbf{z}_i' \sim p(\mathbf{z}|y), \mathbf{z}_j'' \sim p(\mathbf{z}|y')} \left[e^{-\varepsilon \left\|\mathbf{z}_i' - \mathbf{z}_j''\right\|^2} \right] \right]}_{\text{Inter-Class Term}} \right).$$

Based on the assumption 2, we can approximate the Intra-Class term by:

$$\left\|\mathbf{z}'_i - \mathbf{z}''_j\right\|^2 = \|(\mu_y + \delta_i) - (\mu_{y'} + \delta_j)\|^2 = \|\mu_y - \mu_{y'} + \delta_i - \delta_j\|^2 \approx \|\mu_y - \mu_{y'}\|^2$$
$$\Longrightarrow \mathbb{E}_{\mathbf{z}'_i\sim p(\mathbf{z}|y),\mathbf{z}''_j\sim p(\mathbf{z}|y')}\left[e^{-\varepsilon\left\|\mathbf{z}'_i-\mathbf{z}''_j\right\|^2}\right] \approx e^{-\varepsilon\left\|\mu_y-\mu_{y'}\right\|^2}$$

for $y = y'$, $\mathbf{z}'_i$ and $\mathbf{z}''_j$ are close to μ_y

$$\left\|\mathbf{z}'_i - \mathbf{z}''_j\right\|^2 \approx \|(\mu_y + \delta_i) - (\mu_y + \delta_j)\|^2 = \|\delta_i - \delta_j\|^2 .$$

Since δ_i and δ_j are small deviations:

$$\mathbb{E}_{\mathbf{z}'_i,\mathbf{z}''_j\sim p(\mathbf{z}|y)}\left[e^{-\varepsilon\left\|\mathbf{z}'_i-\mathbf{z}''_j\right\|^2}\right] \approx 1, \quad e^{-\varepsilon\|\delta_i-\delta_j\|^2} \approx 1.$$

Then with M terms for $y = y'$ and $M(M-1)$ terms of $y \neq y'$, we have:

$$L_{\text{uniform}} = \log(\frac{1}{M}e^{-\varepsilon\|\delta_i-\delta_j\|^2} + \frac{1}{M^2}\sum_{y\neq y'} e^{-\varepsilon\left\|\mu_y-\mu_{y'}\right\|^2}) \tag{96}$$

The supervised loss is:

$$\mathcal{L}_{\text{sup}}(f_\theta) = \log(\frac{1}{M^2}\sum_{y,y'} e^{-\left\|\mu_y-\mu_{y'}\right\|^2}) = \log(\frac{1}{M}e^{-\|\mu_y-\mu_y\|^2} + \frac{1}{M^2}\sum_{y\neq y'} e^{-\left\|\mu_y-\mu_{y'}\right\|^2})$$

Since $e^{-\|\mu_y-\mu_y\|^2} = 1$ (for $y = y'$), the difference will be mainly dependent on the inter-class term. Therefore, a tighter (smaller) uniformity loss leads to smaller values of the supervised loss. This supports the idea that improving uniformity in representations can benefit downstream supervised classification tasks. □

C.3 Unbalanced OT assists to alleviate the feature suppression

Although minimizing the uniformity loss can enhance downstream classification tasks, it may also lead the model to learn shortcut features that could impair the encoder's generalization ability. To show this, we incorporate two propositions from previous work by Robinson et al. [48].

C.3.1 The uniformity loss causes feature suppression

For an encoder $f_\theta : \mathcal{X} \to \mathbb{S}^{d-1}$ to map input data $\mathbf{x}$ to the surface of the unit sphere $\mathbb{S}^{d-1} = \{u \in \mathbb{R}^d : ||u||_2 = 1\}$. Suppose we have the latent feature spaces $\mathbf{Z}^1, \dots, \mathbf{Z}^n$ with a distribution p_j on each latent space $\mathbf{Z}^j$ with $j \in [n]$ to model a distinct feature. We write $\mathbf{Z}$ instead of $\mathbf{Z}^{[n]}$ for the product as $\mathbf{Z}^S = \prod_{j\in S}\mathbf{Z}^j$, where $[n] = \{1, \dots, n\}$. So the latent sample $\mathbf{z}$ could be represented as a set of feature vectors $\mathbf{z} = (\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^n) = (\mathbf{z}^j)_{j\in S} \in \mathbf{Z}$, where each $\mathbf{z}^j$ comes from $\mathbf{Z}^j$. Further, let λ denote the measure on $\mathbf{Z}$ induced by $\mathbf{z}$ and $\lambda(\cdot|\mathbf{z}^S)$ denote the conditional measure on $\mathbf{Z}$ for fixed $\mathbf{z}^S$. For $S \subseteq [n]$ we use $\mathbf{z}^S$ to denote the projection of $\mathbf{z}$ onto Z^S. Finally, an injective map $g : \mathbf{Z} \to \mathcal{X}$ produces observations $\mathbf{x} = g(\mathbf{z})$. The feature suppression is defined as:

Definition 12. *Consider an encoder $f_\theta : \mathcal{X} \to \mathbb{S}^{d-1}$ and features $S \subseteq [n]$. For each $\mathbf{z}^S \in Z^S$, let $\mu(\cdot|\mathbf{z}^S)$ be the pushforward measure on S^{d-1} by $f_\theta \circ g$ of the conditional $\lambda(\cdot|\mathbf{z}^S)$.*

1. *f_θ suppresses S if for any pair $\mathbf{z}^S, \tilde{\mathbf{z}}^S \in Z^S$, we have $\mu(\cdot|\mathbf{z}^S) = \mu(\cdot|\tilde{\mathbf{z}}^S)$.*
2. *f_θ distinguishes S if for any pair of distinct $\mathbf{z}^S, \tilde{\mathbf{z}}^S \in \mathbf{Z}^S$, measures $\mu(\cdot|\mathbf{z}^S), \mu(\cdot|\tilde{\mathbf{z}}^S)$ have disjoint support.*

If one feature is uniformly distributed on the latent space, it might cause feature suppression due to different features could both achieve the minimization of the uniformity loss as the following propositions [48]:

Proposition 5 (Feature suppression). *For a set $S \subseteq [n]$ of features let*

$$L_S(f_\theta) = L_{align}(f_\theta) + \mathbb{E}_{\mathbf{x}^+}\left[-\log \mathbb{E}_{\mathbf{x}}\left[e^{f(\mathbf{x}^+)^\top f(\mathbf{x}^-)} \Big| \mathbf{z}^S = \mathbf{z}^{S^-}\right]\right]$$

denote the (limiting) InfoNCE conditioned on $\mathbf{x}^+, \mathbf{x}^-$ having the same features S. Suppose that p_j is uniform on $Z^j = S^{d-1}$ for all $j \in [n]$. Then the infimum $\inf L_S$ is attained, and every $f_\theta \in \arg\min_f L_S(f'_\theta)$ suppresses features S almost surely.

Proof of proposition 5 is in [48].

C.3.2 How the GCA methods and unbalanced OT and alleviates the feature suppression

Here we extended the unbalanced OT in the Equation (9) as the following:

$$\min_\theta \; d_M(\mathbf{P}_{tgt} \| \mathbf{P}_\theta) + \lambda_1 d_{\phi_1}(\mathbf{P}_\theta) + \lambda_2 d_{\phi_2}(\mathbf{P}_\theta) + \cdots + \lambda_n d_{\phi_n}(\mathbf{P}_\theta) \tag{97}$$

The UOT equation can be converted with finding the transport plan $\mathbf{P}_\theta$ that minimizes the transportation cost between two probability measures μ and ν. Here we only need to show that the relaxation or adding penalties will change the optimal transport plan $\mathbf{P}_\theta$, which is empirically exhibited in the Figure A4.

Suppose we have empirical samples $\{\mathbf{z}'_i\}_{i=1}^n$ from μ and $\{\mathbf{z}''_j\}_{j=1}^m$ from ν. We can approximate the measures using empirical distributions:

$$\mu \approx \frac{1}{n}\sum_{i=1}^n \delta_{\mathbf{z}'_i}, \quad \nu \approx \frac{1}{m}\sum_{j=1}^m \delta_{\mathbf{z}''_j},$$

where $\delta_{\mathbf{z}}$ is the Dirac delta function at point $\mathbf{z}$. The standard UOT objective can be written as:

$$\min_{\mathbf{P}\geq 0} \sum_{i=1}^n \sum_{j=1}^m \mathbf{C}(\mathbf{z}'_i, \mathbf{z}''_j)\mathbf{P}_{ij} + \lambda_1 d_{\phi_1}\left(\sum_{j=1}^m \mathbf{P}_{ij} \Big\| \frac{1}{n}\right) + \lambda_2 d_{\phi_2}\left(\sum_{i=1}^n \mathbf{P}_{ij} \Big\| \frac{1}{m}\right) \tag{98}$$

$$= \min_{\mathbf{P}\geq 0} \sum_{i=1}^n \sum_{j=1}^m \left[\mathbf{C}_{ij}\mathbf{P}_{ij} + \lambda_1 \mathbf{P}_{ij}\left(\log \frac{\mathbf{P}_{ij}}{r_i} - 1\right) + \lambda_2 \mathbf{P}_{ij}\left(\log \frac{\mathbf{P}_{ij}}{c_j} - 1\right)\right] \tag{99}$$

where $\mathbf{C}$ is the cost matrix d_ϕ could be any divergence (e.g., Kullback-Leibler divergence) with respect to a convex function ϕ. $\mathbf{P}\mathbf{1}_\mu$ and $\mathbf{P}^\top \mathbf{1}_\nu$ are the marginal distributions. λ_1, λ_2 are regularization parameters controlling the unbalancedness and $r_i = \frac{1}{n}$ (source marginal mass for $\mathbf{z}'_i$), $c_j = \frac{1}{m}$ (target marginal mass for $\mathbf{z}''_j$). Based on the UOT, here we can choose the divergence as $\mathcal{L}$:

$$\mathcal{L}(\mathbf{P}) = \sum_{i,j}\left[C_{ij}\mathbf{P}_{ij} + \lambda_1 \mathbf{P}_{ij}\left(\log \frac{\mathbf{P}_{ij}}{r_i} - 1\right) + \lambda_2 \mathbf{P}_{ij}\left(\log \frac{\mathbf{P}_{ij}}{c_j} - 1\right)\right]$$

To find the minimizer, we take the partial derivative of $L(\mathbf{P})$ with respect to $\mathbf{P}_{ij}$ and set it to zero:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{P}_{ij}} = C_{ij} + \lambda_1\left(\log \frac{\mathbf{P}_{ij}}{r_i}\right) + \lambda_2\left(\log \frac{\mathbf{P}_{ij}}{c_j}\right) = 0 \tag{100}$$

$$\implies \lambda_1 (\log \mathbf{P}_{ij} - \log r_i) + \lambda_2 (\log \mathbf{P}_{ij} - \log c_j) = -C_{ij} \tag{101}$$

$$\implies (\lambda_1 + \lambda_2)\log \mathbf{P}_{ij} - \lambda_1 \log r_i - \lambda_2 \log c_j = -C_{ij} \tag{102}$$

$$\implies \log \mathbf{P}_{ij} = \frac{-C_{ij} + \lambda_1 \log r_i + \lambda_2 \log c_j}{\lambda_1 + \lambda_2} \tag{103}$$

$$\implies \mathbf{P}_{ij} = \exp\left(\frac{-C_{ij} + \lambda_1 \log r_i + \lambda_2 \log c_j}{\lambda_1 + \lambda_2}\right) \tag{104}$$

The minimizer $\mathbf{P}_{ij}$ depends on λ_1 and λ_2 and the weights of r_i and c_j, which determine the influence of the marginals r_i and c_j, and through the scaling of the cost C_{ij} by $\lambda_1 + \lambda_2$. This explicit relationship shows how λ_1 and λ_2 determine the minimizer.

D Details of Experiments

The following experiments involving with the GPU was set up on NVIDIA GeForce RTX 3090.

D.1 Experimental details on image classification task

In Table 2 standard settings, we used two different experimental setups. The first setup, referred to as the C0 or standard settings, was applied specifically to the CIFAR10 and CIFAR100 tasks. The second setup was used for the SVHN and ImageNet100 tasks, respectively. Below, we present the settings for CIFAR10 and CIFAR100, followed by the setups for SVHN and ImageNet100. Here is the setups for CIFAR10 and CIFAR100:

- The SSL model has 512 feature dimensions with the base model (ResNet-18), which first convolutional changed as a layer with 3 input channels, 64 output channels, kernel size 3, stride 1, padding 1, and no bias. We replace the max-pooling layer as the identity.
- A sequential projector comprising a linear layer mapping from feature dimension to 2048, ReLU activation, and another linear layer mapping from 2048 to 128.
- For SSL training, an SGD optimizer is used with a learning rate of 0.6, momentum 0.9, and a weight decay of 1.0e-6. A LambdaLR scheduler is employed with linearly decay the learning rate to 1.0e-3 over total steps, which equals the length of the SSL training loader times the maximum epochs. The SSL model is trained for a maximum of 500 epochs, without loading a pre-trained model. The parameters of encoders are frozen after training. Temperature or epsilon: 0.5.
- For supervised training, an Adam optimizer is also used with a learning rate of 0.2, momentum 0.9 and a weight decay of 0. A same LambdaLR scheduler is applied, where the learning rate is reduced by a factor of 1.0e-3. For supervised training, the model is trained for a maximum of 200 epochs using the specified train and test loaders.

The setups for SVHN and ImageNet100 are:

- The SSL model has number of feature dimensions equal to the fc layer incoming features of base model (ResNet-50). We replace the max-pooling layer as the identity.
- A sequential projector comprising a linear layer mapping from feature dimension to 2048, ReLU activation, and another linear layer mapping from 2048 to 128.
- For SSL training, an Adam optimizer is used with a learning rate of 3e-4. The SSL model is trained for a maximum of 200 epochs for ImageNet100 and 500 epochs for the SVHN, without loading a pre-trained model. The parameters of encoders are frozen after training. Temperature or epsilon: 0.5.
- For supervised training, an Adam optimizer is also used with a learning rate of 3e-4. The model is trained for a maximum of 100 epochs using the specified train and test loaders.

D.2 Settings for extreme data augmentations

There is the "extreme DA" (Ex DA) column in Table 2, which is the average of the following three settings:

- C1: Large Erase Settings: Here, we first employed the same standard augmentation as C0 in Appendix D.1 does, than we apply the random erase with 'p=1' (random erasing is applied every time), the 'scale=(0.10, 0.33)'. The large erase is applied before the normalization.
- C2: Strong Crop Setting: This involves a strong cropping operation followed by resizing, which applied by 'transforms.RandomCrop' and 'transforms.Resize'. The crop size varies based on the severity level, with values ranging from 96 to 224 pixels. We selected level 3 during our experiments, than Resizes the cropped image back to 32x32 pixels.
- C3: Brightness settings: This augmentation alters the brightness of the images. We have 'severity' determines the degree of brightness change, with predefined levels ranging from '.05' to '.3', corresponding to level 1 and level 5. And we chosse the level 5 as our C3 augmentation. The brightness is adjusted in the HSV color space, specifically altering the value channel to change the brightness.

To evaluate performance on CIFAR10-C, we use a pretrained SSL model with frozen parameters. Fine-tuning is performed by training only the linear layer with 10% of CIFAR10-C data for 50 epochs. We compute the final score by averaging results across all corruption types and severity levels in CIFAR10-C. And the details of each column are provided in Table A2, Table A3 and Table A4.

Table A2: *Test accuracy for contrastive methods on CIFAR-10.* Test accuracy for different contrastive methods and their GCA equivalents on CIFAR-10 for ResNet-18 under extreme augmentation conditions, averaged over 5 seeds.

Conditions	**INCE**	**GCA-INCE**	**RINCE**	**GCA-RINCE**	**SimCLR**	**BYOL**	**IOT**	**IOT-uni**	**GCA-UOT**
Standard	92.01 ± 0.40	93.02 ± 0.19	93.27±0.20	93.47±0.32	92.16 ±0.16	90.56 ± 0.59	92.10 ± 0.22	91.49 ± 0.11	**93.50±0.31**
Erase	88.40 ± 0.17	88.16 ± 0.89	88.80±1.01	89.21±0.59	88.44 ± 0.24	88.77 ± 0.58	87.02 ± 0.43	87.83 ± 0.30	**89.84 ± 0.58**
Crop	72.45 ± 0.40	72.79 ± 0.62	73.02±0.39	73.10±0.31	71.84 ± 1.02	70.78 ± 0.62	70.44 ± 0.64	70.78 ± 0.21	**73.35 ± 0.41**
Brightness	85.24 ± 0.41	85.60 ± 0.57	85.97±0.50	85.98 ± 0.58	85.32 ± 0.32	85.10 ± 0.29	84.31 ± 0.84	83.77 ± 0.21	**86.36±0.34**

Table A3: *Test accuracy for contrastive methods on CIFAR-100.* Test accuracy for different contrastive methods and their GCA equivalents on CIFAR-100 using ResNet-18 under extreme augmentation conditions, averaged over 5 seeds.

Conditions	**INCE**	**GCA-INCE**	**RINCE**	**GCA-RINCE**	**SimCLR**	**BYOL**	**IOT**	**IOT-uni**	**GCA-UOT**
Standard	71.09 ± 0.31	71.55 ± 0.12	71.63 ± 0.36	71.95 ± 0.48	70.85 ± 0.50	69.75 ± 0.37	68.37 ± 0.42	68.62 ± 0.35	**72.16 ± 0.38**
Large Erase	62.54 ± 0.20	62.65 ± 0.17	63.55 ± 0.14	63.14 ± 0.41	62.94 ± 0.13	62.70 ± 0.31	62.69 ± 0.34	62.56 ± 0.22	**63.62 ± 0.27**
Strong Crop	45.67 ± 0.31	46.31 ± 0.22	46.47 ± 0.20	46.50 ± 0.35	46.05 ± 0.34	43.11 ± 0.41	45.36 ± 0.19	45.29 ± 0.12	**46.60 ± 0.34**
Brightness	59.87 ± 0.36	59.68 ± 0.33	60.38 ± 0.25	60.56 ± 0.25	60.46 ± 0.19	55.74 ± 0.63	57.52 ± 0.32	57.42 ± 0.18	**61.02 ± 0.55**

Table A4: *Test accuracy for contrastive methods on CIFAR-10C.* Test accuracy for different contrastive methods and their GCA equivalents on CIFAR-10C for ResNet-18 under extreme augmentation conditions, averaged over 5 seeds.

Conditions	**INCE**	**GCA-INCE**	**RINCE**	**GCA-RINCE**	**SimCLR**	**BYOL**	**IOT**	**IOT-uni**	**GCA-UOT**
Standard	81.52 ± 1.04	82.63 ± 0.28	82.86 ± 0.21	82.87 ± 0.11	81.74 ± 1.54	82.43 ± 0.06	82.01 ± 0.80	81.18 ± 1.12	**82.90 ± 0.49**
Large Erase	82.67 ± 0.31	82.19 ± 0.35	83.11 ± 0.36	**83.44 ± 0.55**	82.49 ± 0.24	82.53 ± 0.34	82.30 ± 0.69	82.30 ± 0.48	83.09 ± 0.62
Strong Crop	43.55 ± 0.90	41.12 ± 3.28	42.51 ± 1.48	44.96 ± 1.46	42.74 ± 1.27	43.86 ± 0.91	42.36 ± 0.87	42.43 ± 1.03	**45.20 ± 1.20**
Brightness	84.00 ± 0.47	84.08 ± 0.26	84.24 ± 0.36	84.54 ± 0.26	83.86 ± 0.46	83.87 ± 0.19	82.33 ± 0.31	80.86 ± 0.72	**84.88 ± 0.44**

D.3 Experimental setting for domain generalization

This section is going to show the settings of experiments in Figure 1, which involves the domain generalization task. Training was executed under the DomainBed framework. Each model underwent training across multiple domains, with 5 distinct seeds (seed 71, 68, 42, 36, 15) used to ensure reproducibility:

- For SSL model configuration, we employed a ResNet-18 architecture as the encoder, following with a 2048-dimensional, 3-layer projector equipped with BatchNorm1D and ReLU activations. We improved the framework of the SelfReg algorithm in Domainbed [23] by a self-supervised contrastive learning phase which involves the GCA-INCE, with regularized parameters $\varepsilon = 0.2$.
- For SSL training hyperparameters, an Adam optimizer is used with a learning rate of 3e-4, and a weight decay of 1.5e-6. A Cosine Annealing learning rate scheduler is employed with a maximum number of 200 iterations equal to the length of the SSL training. The learning rate is scheduled to decrease to a minimum value of 0. The SSL model is trained for a maximum of 1500 epochs.
- In the self-supervised learning phase, we utilized 20% of the data from each of the four datasets in the PACS dataset. The unsupervised holdout part employed contrastive learning augmentations to enhance generalization capabilities. Specifically, we implemented dual augmentation, including operations such as random resized crops, flips, color jitter, and grayscale conversion, standardized to an input shape of $3 \times 224 \times 224$.
- The supervised learning rate was set at 5×10^{-5} using MSE loss, and the Adam optimizer with no weight decay. Training involved both domain and class labels over 3000 epochs, with checkpoints every 300 epochs to capture the model's best performance. This approach was supplemented by fine-tuning the model post-unsupervised training phase. Domain labels were categorized into four types corresponding to the PACS dataset, and class labels were divided into five categories. In domain classification, all four domains are used for

training, with 70% of the data held out for training and the remaining 30% used for testing. Four domains are utilized for class classification tasks. We train supervised models on three domains and test on the fourth.

- The domain accuracy is computed as the average of the highest domain accuracies across five seeds, with each of the four test domains set sequentially as the test domain. The standard deviation for domain accuracy is calculated from the results across these five seeds.
- Class label accuracy is determined by averaging the accuracies of the four test environments for each domain. The average of highest performance across the domain is taken as the mean accuracy. The standard deviation for each domain is computed from the five seeds, and these values are then averaged to obtain the final class standard deviation.

Both the label classification tasks and the domain classification tasks use the Mean Squared Error (MSE) loss.

E Additional Experiments

E.1 Complexity Analysis of GCA Algorithms

Time complexity analysis: The computational complexity of GCA including the forward pass and backward propagation phases. The complexity varies in different variants. For GCA-INCE, the computational complexity of forward pass is related to the speed of Sinkhorn when solving the EOT problem as $O(n^2/\varepsilon^3)$, in which ε is the regularization parameter . For GCA-UOT, the forward complexity is the Sinkhorn algorithm solving unbalanced OT, which is characterized by

$$O(\tau(\alpha+\beta)^2/\varepsilon \log(n)[\log(\|C\|_\infty) + \log(\log(n)) + \log(1/\varepsilon)]),$$

where C is the cost matrix, α and β denote the total masses of the measures, and τ is a regularization parameter related to KL divergences in the UOT framework [44]. Notably, the gradient backpropagation speed is not seriously affected by scaling operations in the EOT as we explained in Section B.7. Moreover, the relaxations of penalties in UOT provide a even faster speed compared with the INCE and GCA-INCE (see Figure A2).

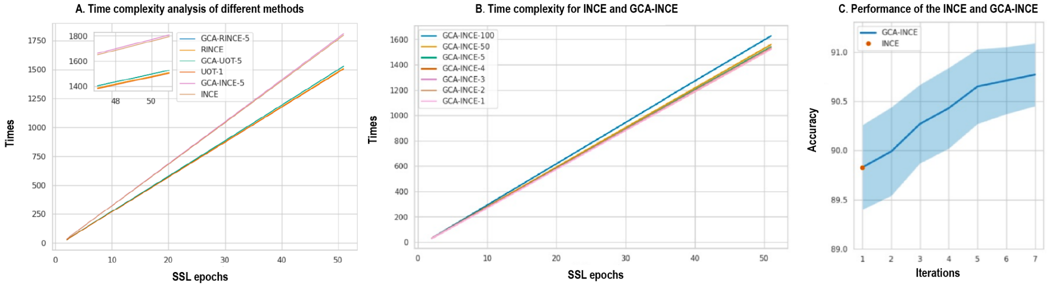

Figure A2: *Time complexity analysis* (A) Time complexity analysis of different methods. Here, we provide the time complexity for different contrastive methods (INCE, RINCE) and GCA-based methods (GCA-INCE, GCA-RINCE, and GCA-UOT) on CIFAR-10. (B) Time complexity for INCE (GCA-INCE-1), and GCA-INCE with different number of iterations GCA-INCE-100 denotes GCA-INCE with 100 iterations. We ran the methods on the CIFAR-10 as self-supervised learning task for 50 epochs, and compared their run time. (C) Performance of the INCE (iteration=1) and GCA-INCE (iterations>1) on the CIFAR10 with different number of iterations. The shaded blue region is the standard deviation across 5 seeds.

The complexity of the forward pass is affected by the choice of proximal operator, whereas the complexity of the gradient backward pass is influenced by the form of d_M [37]. Notably, utilizing Sinkhorn algorithms in GCA-UOT, GCA-RINCE, and GCA-INCE, only requires updating the coupling matrix $\mathbf{P}$ $(B \times B)$ without impacting the complexity of the backward pass, where B is the batch size. OT is known to have B^2 complexity and in many cases can converge very quickly in fewer than 10 iterations. In practice, we use a simple stopping criterion for the multiple iterations using a convergence criterion.

Upon analyzing the run time for the different methods (see Figure A2) we observe that the GCA-based variants of the different base approaches (INCE, or RINCE) achieve very similar run time as their equivalent loss, but different losses (RINCE vs INCE) exhibit more significant variability. Specifically, we find that RINCE and GCA-RINCE have lower time complexity than INCE and GCA-INCE. So the runnning speed is even quicker if we utillized different d_M in Equation (8).

E.2 Measuring the representation quality using alignment and uniformity

We study the uniformity and alignment of the representations learned by our GCA-INCE vs. INCE variants of GCA in Algorithms 1.We train the model through the corresponding settings (C0: standard provided in the , C1: erase, C2: crop, C3: brightness) provided in the Appendix D.1 and Appendix D.2. We find that in general, the GCA variants improve the representation quality evaluated by alignment and uniformity on both CIFAR-10 and CIFAR-10C datasets.

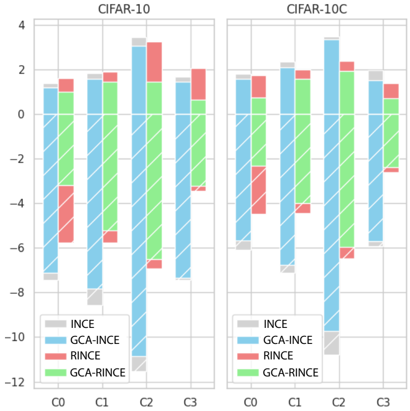

Figure A3: *Alignment and uniformity metrics on CIFAR-10.* To visualize the ability of uniformity and alignment with different methods under different augmentation settings (C0: standard, C1: erase, C2: crop, C3: brightness). The bar above the x axis (zero line) represents the alignment loss, while the bar under the x axis represents the uniformity loss. The shorter the color bars i.e with lower alignment loss and higher uniformity loss, correspond to the better performance of SSL models.

E.3 Visualizing transport plans of different methods after training

Here we compared the optimal transport (OT) plans of different methods after training for 500 epochs under standard augmentation C0 settings in Appendix D.1. Specifically, we analyzed the $-\log(\mathbf{P})$ matrices of INCE, GCA-INCE, GCA-RINCE, and GCA-UOT, as shown in Figure A4. In these matrices, darker blue regions represent higher similarity, while lighter blue areas indicate less similarity. The matrices are rearranged based on class labels, so an effective model should display empty diagonals and block structures aligned along the main diagonal and sub-diagonals—reflecting high intra-class similarity and low inter-class similarity.

Figure A4(A) shows that INCE results in a matrix with only row normalization. In contrast, Figures A4(B) and (C) demonstrate that GCA-INCE and GCA-RINCE achieve both row and column normalization, leading to more uniform distributions. Figure A4(D) reveals that GCA-UOT produces a matrix highlighting greater differences between positive and negative pairs, underscoring its effectiveness in distinguishing them.

E.4 Hyperparameter Tuning and Sensitivity Analysis

In our hyperparameter modifying experiments, we investigate the influence of key parameters in transport plan regularization, iteration counts, and augmentation strengths on CIFAR-10 classification performance.

Figure A5 visualizes transport plans under varying entropic regularization (ϵ values from 0.01 to 1) across INCE and GCA-UOT models, illustrating adjustments after five iterations using the same ResNet-18 weights. Figure A6 examines the impact of iteration number and entropic regularization on compactness—measured by the average L2 distance to class centers—and accuracy, with 20 pre-training epochs followed by fine-tuning. Figure A7 highlights the sensitivity of GCA-RINCE to the hyperparameters q and λ, testing classification accuracy for different settings under strong augmentation conditions; this includes a comparison against INCE with large erase augmentation after substantial pre-training and evaluation epochs.

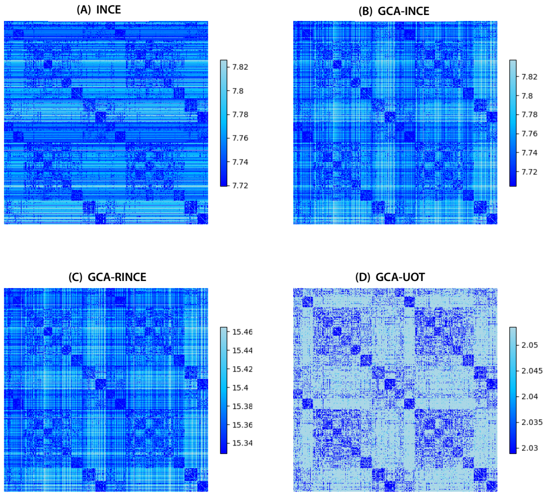

Figure A4: *Comparison of the* $-\log(\mathbf{P})$ *matrix across different methods.* (A) The INCE matrix with row normalization. (B) The $-\log(\mathbf{P})$ matrix of GCA-INCE with five iterations in forward pass, both row and column normalization. (C) The $-\log(\mathbf{P})$ matrix of GCA-RINCE with five iterations in forward pass. (D) The $-\log(\mathbf{P})$ matrix of GCA-UOT with five iterations in forward pass

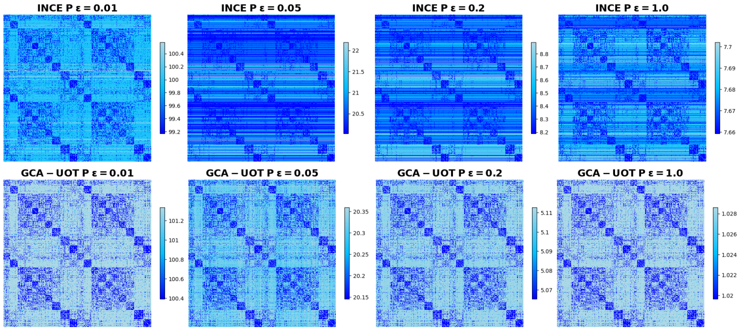

Figure A5: *Visualization transport plan P for different amounts of entropic regularization.* (Top) The transport plans for ϵ from 0.01 to 1 for INCE and (Bottom) GCA-UOT after 5 iterations. To compute each plan, we took a mini-batch on CIFAR-10 with 1024 samples, and loaded the same weights of Resnet-18 for each subfigure.

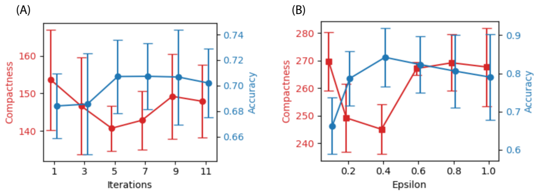

Figure A6: *Hyperparameter sensitivity study.* The compactness and accuracy as a function of the (A) number of iterations and the (B) entropic regularization parameter. In our experiments, we use the same weights and perform 20 pre-training epochs for each point, then evaluate their performance by fine-tuning linear classifiers for 20 epochs. Here the compactness is the average L2 distance of each point to their corresponding class center on the representation space after the encoder.

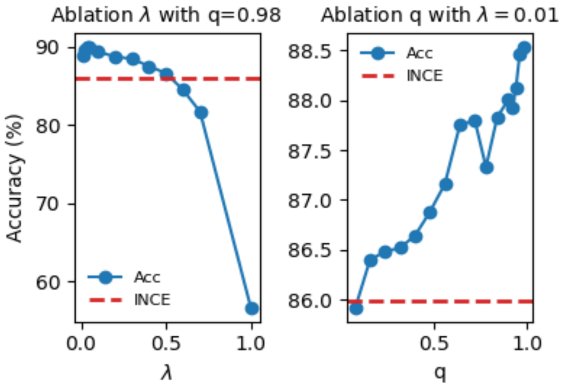

Figure A7: *Hyperparameter sensitivity for q and λ in GCA-RINCE.* Both experiments are tested on the CIFAR-10 dataset with a ResNet-18 encoder and involve strong augmentation with large erase. (Left) Given $q = 0.98$, we change λ from 0 to 1. (Right) Given $\lambda = 0.01$, we change q from 0 to 1. The red threshold line is the INCE performance with the large erase augmentation. Each point represents the CIFAR-10 classification accuracy of the ResNet-18 model pre-trained for 400 epochs and evaluated after 300 epochs.