# Contrasting Multiple Representations with the Multi-Marginal Matching Gap

**Zoe Piran** [*1,2] **Michal Klein** [*1] **James Thornton** [1] **Marco Cuturi** [1]

## Abstract

Learning meaningful representations of complex objects that can be seen through multiple ($k \geq 3$) views or modalities is a core task in machine learning. Existing methods use losses originally intended for paired views, and extend them to $k$ views, either by instantiating $\frac{1}{2}k(k-1)$ loss-pairs, or by using reduced embeddings, following a *one vs. average-of-rest* strategy. We propose the multi-marginal matching gap (M3G), a loss that borrows tools from multi-marginal optimal transport (MM-OT) theory to simultaneously incorporate all $k$ views. Given a batch of $n$ points, each seen as a $k$-tuple of views subsequently transformed into $k$ embeddings, our loss contrasts the cost of matching these $n$ ground-truth $k$-tuples with the MM-OT polymatching cost, which seeks $n$ optimally arranged $k$-tuples chosen within these $n \times k$ vectors. While the exponential complexity $O(n^k)$ of the MM-OT problem may seem daunting, we show in experiments that a suitable generalization of the Sinkhorn algorithm for that problem can scale to, e.g., $k = 3 \sim 6$ views using mini-batches of size $64 \sim 128$. Our experiments demonstrate improved performance over multiview extensions of pairwise losses, for both self-supervised and multimodal tasks.

## 1. Introduction

Learning meaningful representations of complex objects that can be seen through multiple views or modalities is a core task in machine learning. These representations may be trained separately for each modality, as a preliminary step towards zero-shot learning (Palatucci et al., 2009; Socher et al., 2013; Frome et al., 2013). In that scenario, modalities can be heterogeneous, as with images and text (Radford et al., 2021; Schuhmann et al., 2022), or beyond (Deldari et al., 2022); or homogeneous, e.g. various channels of the same timeseries (Khaertdinov et al., 2021; Cheng et al., 2020; Wen et al., 2020; Brüsch et al., 2023; Banville et al., 2021; Tonekaboni et al., 2021; Kiyasseh et al., 2021). In the closely related task of self-supervised learning (SSL), a single embedding backbone may be considered instead, and applied to multiple views/augmentations of the same object (Chen et al., 2020; Caron et al., 2020; Bardes et al., 2022b; Assran et al., 2023; Tsai et al., 2021).

**Learning with Pairs.** Whether applied to multiview or multimodal learning, these approaches were originally proposed for $k = 2$ different representations (e.g. arising from two modalities or two augmentations). Most of them rely on contrastive learning (Gutmann & Hyvärinen, 2010; Oord et al., 2018) as a blueprint, using, for instance, the InfoNCE loss. The InfoNCE loss promotes encoders that produce nearby representations for *two* inputs that arise from the same object (either with different views or modalities), and far-away representations for any other pair. Alternatively, BYOL (Grill et al., 2020) uses only positive pairs, and relies instead on a pair of encoders with tied parameters.

**Learning with $k \geq 3$ Representations.** As representation learning eyes more ambitious tasks, practitioners are tempted to incorporate more than two views/modalities (Alayrac et al., 2020; Akbari et al., 2021; Girdhar et al., 2023). Various strategies have been proposed to handle $k \geq 3$ representations that can cope with the limitation of pairwise losses (Bachman et al., 2019; Tian et al., 2020; Tsai et al., 2020). For instance, one may handle $k$ representations by averaging all possible $\frac{1}{2}k(k-1)$ pairwise losses (Bachman et al., 2019; Tian et al., 2020); Alternatively, one may average embeddings (Pototzky et al., 2022), effectively comparing each of the $k$ representations to the average of the remaining $k-1$ embeddings. None of these approaches do leverage, however, the knowledge that these $k$ representations should be *simultaneously* coherent, by looking at $k$-tuple of points rather than $\frac{1}{2}k(k-1)$ pairs.

**Our Contributions.** We propose a novel approach that fully leverages the ground-truth knowledge that a single input data point should be viewed, holistically, as $k$-tuples of embeddings. Our contrastive loss is tailored for multiple ($k \geq 3$) views, without a reduction to pairwise compar-

[*]Equal contribution [1]Apple [2]Hebrew University Jerusalem. Correspondence to: Marco Cuturi <cuturi@apple.com>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. 

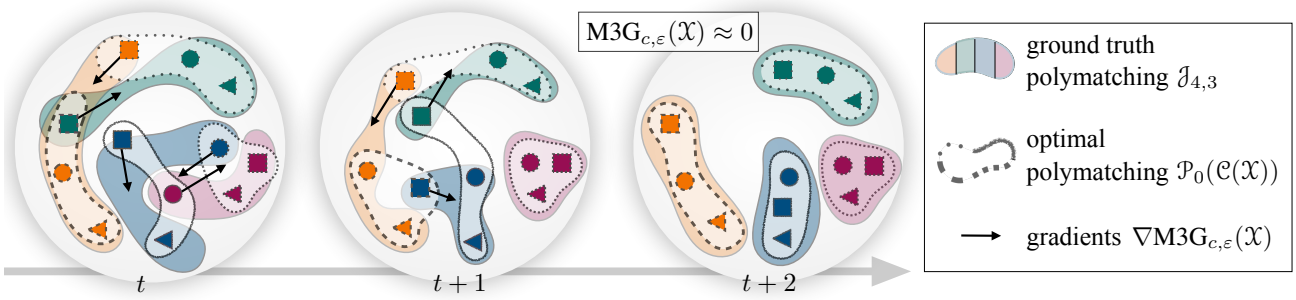


Figure 1: *(left)* Embeddings for $n = 4$ points (identified using 4 colors), each given in $k = 3$ views (differentiated using 3 shapes) in $d = 2$ dimensions. The ground-truth polymatching of these points is known: to each color its 3 shapes, as illustrated with colored cliques, and described mathematically as a tensor $\mathcal{J}_{4,3}$. Their initial configuration in space indicates, assuming one solves a multi-marginal optimal transport problem parameterized with the cost tensor $\mathcal{C}(\mathcal{X})$, a *different* polymatching $\mathcal{P}_0(\mathcal{C}(\mathcal{X}))$. That difference (quantified as a difference in their *matching objectives*) defines the M3G loss (see Def. 3.1 for a precise definition of what $c, \varepsilon$ refer to). A high M3G indicates, as shown on the left, a large discrepancy between the ground-truth matching's cost and that of the optimal polymatching. This loss will gradually displace points so that, ideally, upon convergence and after consecutive updates (visualized in *(middle)* and *(right)* plots), both ground-truth and optimal polymatchings coincide in their objective. For additional intuition see Animation 1, presenting the gradient flow of M3G over a toy problem.

isons. This global view is provided by solving polymatching problems using entropy-regularized multi-marginal optimal transport (MM-OT). More precisely:

- After providing background on SSL and MM-OT in § 2, we present in § 3 the M3G loss to measure the contrast of a configuration of an $n$-batch of $k$-tuples of points. M3G subtracts the lowest matching cost achieved by an MM-OT solver (MM-Sinkhorn) to the matching cost of the ground-truth identity matching tensor available to the user. We study computational and theoretical properties of M3G, highlight the freedom to choose any multiway cost function defined on $k$-tuple of points, and show how to use M3G for representation learning.
- We provide experimental evidence in § 4 that the M3G loss improves on extensions of pairwise losses in a variety of self-supervised and multimodal tasks, using the ImageNet-1k dataset (Deng et al., 2009), DomainNet (He et al., 2020)) and time-series electroencephalography (EEG) data from PhysioNet (Goldberger et al., 2000; Ghassemi et al., 2018; Kemp et al., 2000).

**Notation.** We use bold fonts for vectors $\mathbf{x}, \mathbf{y}, \dots$ in $\mathbb{R}^d$ and matrices $\mathbf{X}, \mathbf{Y}, \mathbf{P}, \mathbf{C} \dots$ in $\mathbb{R}^{n \times d}$ or $\mathbb{R}^{n \times n}$; curved fonts $\mathcal{X}, \mathcal{Y}, \mathcal{P}, \mathcal{C} \dots$ for tensors of dimension 3 and more. For an integer $k$, we set $[\![k]\!] := (1, \dots, k)$, and for two integers $\ell < m$, $[\![\ell, m]\!] := (\ell, \ell + 1, \dots, m)$.

## 2. Background: SSL and MM-OT

### 2.1. Joint embeddings with student-teacher architecture

We rely in this work on joint embedding student-teacher architectures (Balestriero et al., 2023; Grill et al., 2020). This setting consists of a tied pair of *online* (student) and *target* (teacher) networks. The student network contains three components–an encoder, a projector and a predictor. The teacher is based on the online network, omitting the predictor head. Importantly, the parameters of the latter are updated using an exponential moving average (EMA) of the student's parameters. Setting, for instance, the index $s$ to be the student, and $t$ the teacher, the parameters $\theta_t$ are simply updated as $\theta_t \leftarrow (1 - \rho)\theta_s + \rho\theta_s$ after each $\theta_s$ update, with EMA parameter $0 < \rho < 1$.

### 2.2. Learning embeddings with pairwise losses

**Learning with $k = 2$ views.** In a standard SSL setup, one selects a batch of $n$ items $(z_i)_i := (z_1, \cdots, z_n)$ alongside two augmentation pipelines $\mathcal{A}_1$ and $\mathcal{A}_2$; applies both augmentations to each item in the batch, yielding a list of $n$ pairs of objects, $(\mathcal{A}_1(z_i), \mathcal{A}_2(z_i))_i$. These are then passed through parameterized neural networks, $f_{\theta_1}, g_{\theta_2}$, that produce vector representations $\mathbf{x}_i^1 := f_{\theta_1}(\mathcal{A}_1(z_i))$ and $\mathbf{x}_i^2 := g_{\theta_2}(\mathcal{A}_2(z_i))$. This results in $\mathbf{X}^1 := (\mathbf{x}_i^1)_i, \mathbf{X}^2 = (\mathbf{x}_i^2)_i$, two $n \times d$ matrices of embeddings, which we assume throughout the paper to lie on the $d$-sphere, i.e. their norms are equal to 1. These views are then fed to a pairwise loss, $\mathcal{L}_{\text{pair}}\left(\mathbf{X}^1, \mathbf{X}^2\right)$ used to fit either or both parameters $\theta_1, \theta_2$. The seminal approach of SimCLR (Chen et al., 2020) con-

sidered a variant of the InfoNCE loss (Oord et al., 2018; Gutmann & Hyvärinen, 2010) defined as:

$$\mathcal{L}_{\text{InfoNCE}}(\mathbf{X}^1, \mathbf{X}^2) = -\frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{e^{\frac{\langle \mathbf{x}_i^1, \mathbf{x}_i^2\rangle}{\tau}}}{\sum_j e^{\frac{\langle \mathbf{x}_i^1, \mathbf{x}_j^2\rangle}{\tau}}}\right). \quad (1)$$

Grill et al. (2020) propose an alternative that leverages an asymetric student-teacher setting (§ 2.1), to focus exclusively on paired positive examples:

$$\mathcal{L}_{\text{BYOL}}(\mathbf{X}^1, \mathbf{X}^2) = 2 - \frac{2}{n}\sum_{i=1}^{n} \langle \mathbf{x}_i^1, \mathbf{x}_i^2\rangle. \quad (2)$$

**Extending pairwise losses to $k \geq 3$ views.** Recent contrastive approaches rely on $k \geq 3$ views to improve model performance (Caron et al., 2020; Tian et al., 2020; Zhou et al., 2022; Bardes et al., 2022a). For example, a multi-crop strategy for images adds various views at different resolutions, rather than two full-resolution views (Caron et al., 2020), to extract more information from a single input object (Balestriero et al., 2023; Hoffer et al., 2020). The number of samples the model sees is effectively increased at each batch to $n \times k$ instances, $k$ views for a batch of $n$ images. Each of these instances is then represented as a $d$-dimensional vector, all of which can be stored as a 3D tensor: $k$-views for $n$ points in $d$-dimensions, $\mathcal{X} \in \mathbb{R}^{k\times n\times d}$, $\mathcal{X} = [\mathbf{X}^1, \dots, \mathbf{X}^k]$ where each $\mathbf{X}^l$ gathers the $n$ objects as seen from the $\ell$-th view, namely an $n \times d$ matrix $\mathbf{X}^\ell = [\mathbf{x}_1^\ell, \dots, \mathbf{x}_n^\ell]$. To handle multiple views, a *pairwise* contrastive loss $\mathcal{L}_{\text{pwe}}$ can be defined by aggregating all possible pairs, $\frac{1}{2}k(k-1)$ in total, of $\mathcal{L}_{\text{pair}}$ losses,

$$\mathcal{L}_{\text{pwe}}(\mathcal{X}) = \frac{2}{k(k-1)}\sum_{\ell<m}^{k} \mathcal{L}_{\text{pair}}\left(\mathbf{X}^\ell, \mathbf{X}^m\right), \quad (3)$$

The summation can be performed on all, a subset of the pairs, e.g. restricting $\ell$ to sweep $1, 2$ and taking $m \in [\![k]\!]$ (Caron et al., 2021; Grill et al., 2020), or with a different aggregation method (Shidani et al., 2024). An alternative way to fall back on using a parwiese loss, is averaging representations beforehand, and applying the loss in a *one vs. average-of-rest* fashion. That is, for each view $\ell$, the embeddings $\mathbf{X}^\ell$, are compared to the *average* of all remaining views, $\bar{\mathbf{X}}^{-\ell} := \frac{1}{k-1}\sum_{m\neq\ell} \mathbf{X}^m$ as presented in (Pototzky et al., 2022; Liang et al., 2024), defining the loss,

$$\mathcal{L}_{\text{ave}}(\mathcal{X}) = \frac{1}{k}\sum_{\ell=1}^{k} \mathcal{L}_{\text{pair}}\left(\mathbf{X}^\ell, \bar{\mathbf{X}}^{-\ell}\right). \quad (4)$$

These two approaches only look at the entire representation tensor $\mathcal{X}$ two slices at a time, either by comparing $\mathbf{X}^\ell$ to another $\mathbf{X}^m$, $\bar{\mathbf{X}}^{-\ell}$, or $\mathbf{X}^\ell$ to a combination of the other $(\mathbf{X}^m)_{m\neq\ell}$. MM-OT, introduced next, will serve as the workhorse to provide the first holistic loss for multiple views, leveraging the entire distribution described in $\mathcal{X}$.

### 2.3. Multi-Marginal Optimal Transport (MM-OT)

We borrow notations from (Peyré & Cuturi, 2019, Chap. 4), restricting our attention to *matching* problems (uniform marginals of the same size). As a warm-up to the multi-marginal case, we start with two marginals.

**Regularized Bistochastic Matching.** Consider a cost matrix $\mathbf{C} \in \mathbb{R}^{n\times n}$.The entropy regularized matching cost of $\mathbf{C}$, parameterized by regularization $\varepsilon \geq 0$, is the output of the following minimization:

$$\text{OT}_{2,\varepsilon}(\mathbf{C}) = \min_{\mathbf{P}\in\mathcal{B}_{n,2}} \langle \mathbf{P}, \mathbf{C}\rangle + \varepsilon\langle \mathbf{P}, \log \mathbf{P} - 1\rangle, \quad (5)$$

where $\mathcal{B}_{n,2}$ is the Birkhoff polytope of bistochastic *matrices*,

$$\mathcal{B}_{n,2} := \{\mathbf{P} \in \mathbb{R}_+^{n\times n} \,|\, \mathbf{P}\mathbb{1}_n = \mathbf{P}^T\mathbb{1}_n = \mathbb{1}_n/n\}. \quad (6)$$

For $\varepsilon = 0$, one recovers the optimal assignment problem, used for instance to compute a loss between lists of annotations in an image (Carion et al., 2020). When $\varepsilon > 0$, the problem can be solved with the Sinkhorn fixed point iterations, with faster execution on accelerators (Cuturi, 2013).

**Regularized Polystochastic Matchings.** We consider the generalization to multidimensional cost tensors of the matching problem, moving away from the bipartite setting described above. Such problems arise when comparing $k \geq 3$ families of points simultaneously, to solve *polypartite* matching problems. The MM-OT (Gangbo & Święch, 1998; Pass, 2015) problem and its entropic regularization (Benamou et al., 2015) generalize Eq. (5) by searching for $k$-polymatchings, represented with their relaxation as polystochastic tensors (Benson-Putnins, 2014). To introduce these approaches, we need a few more notations.

**Polystochastic Tensors.** We consider the set of $k$-dimensional tensors, of size $n$ for each slice:

$$\mathcal{T}_{n,k} := \mathbb{R}^{\overbrace{n\times\cdots\times n}^{k \text{ times}}}.$$

Let $\mathbb{1}_{n,k}$ be the tensor in $\mathcal{T}_{n,k}$ containing ones, including $\mathbb{1}_n := \mathbb{1}_{n,1}$ the $n$-vector of ones. For a tensor $\mathcal{P} \in \mathcal{T}_{n,k}$, and $\ell \leq k$, we write $\text{m}_\ell$ for the contraction of the tensor along all of its slices expect for the $\ell$-th one. Using the tensordot operator (with 1-indexing, not 0 as used by default in python), this is equivalent to, writing $I_\ell = ([\![k]\!] \setminus \ell, [\![k-1]\!])$ for the pair of contraction indices,

$$\text{m}_\ell(\mathcal{P}) = \texttt{tensordot}(\mathcal{P}, \mathbb{1}_{n,k-1}, I_\ell) \in \mathbb{R}^n.$$

We define, by analogy to Eq. (6), the polytope of $k$-polystochastic tensors,

$$\mathcal{B}_{n,k} := \{\mathcal{P} \in \mathcal{T}_{n,k} \,|\, \forall \ell \leq k,\, \mathrm{m}_\ell(\mathcal{P}) = \mathbb{1}_n/n\}\,. \quad (7)$$

We can now generalize Eq. (5) by replacing the suffix 2 by $k$ in these expressions, to define, for any cost tensor $\mathcal{C} \in \mathcal{T}_{n,k}$,

$$\mathrm{OT}_{k,\varepsilon}(\mathcal{C}) := \min_{\mathcal{P}\in\mathcal{B}_{n,k}} h_{k,\varepsilon}(\mathcal{P},\mathcal{C}),\ \text{where}$$
$$h_{k,\varepsilon}(\mathcal{P},\mathcal{C}) := \langle \mathcal{P}, \mathcal{C}\rangle + \varepsilon\langle \mathcal{P}, \log \mathcal{P} - 1\rangle\,. \quad (8)$$

**Dual Formulation.** Eq. (8) admits an unconstrained dual formulation, using a few more notations: For an $n\times k$ matrix $\mathbf{F}$, stored as $k$ column vectors of size $n$, $\mathbf{F} = (\mathbf{f}^1, \dots, \mathbf{f}^k) \in \mathbb{R}^{n\times k}$, we define the tensor sum operator, which to a matrix in $\mathbb{R}^{n\times k}$ associates a tensor in $\mathcal{T}_{n,k}$ as follows,

$$\bigoplus \mathbf{F} := \mathbf{f}^1 \oplus \cdots \oplus \mathbf{f}^k,\ \text{i.e.}\ \left[\bigoplus \mathbf{F}\right]_{i_1\dots i_k} = \mathbf{f}^1_{i_1} + \cdots + \mathbf{f}^k_{i_k}.$$

where all indices $1 \leq i_1, \dots, i_k \leq n$. In that case, one has the following equivalence with Eq. (8),

$$\mathrm{OT}_{k,\varepsilon}(\mathcal{C}) = \max_{\mathbf{F}} \tfrac{1}{n}\mathbb{1}_n^T\mathbf{F}\mathbb{1}_k - \varepsilon\langle e^{\frac{\bigoplus\mathbf{F}-\mathcal{C}}{\varepsilon}}, \mathbb{1}_{n,k}\rangle \quad (9)$$

and the following primal-dual relationship among the optimal solutions $\mathcal{P}^\star$ of Eq. (8) and $\mathbf{F}^\star$ of Eq. (9):

$$\mathcal{P}^\star = \exp\left(\left(\bigoplus \mathbf{F}^\star - \mathcal{C}\right)/\varepsilon\right)\,. \quad (10)$$

**Multi-Marginal Sinkhorn.** Eq. (9) can be solved using the multi-marginal Sinkhorn (MM-S) algorithm described in Alg. 1. This algorithm outputs, using Eq. (10), the optimal polystochastic tensor associated to $\mathcal{C}$:

$$\mathcal{P}_\varepsilon(\mathcal{C}) := \operatorname*{arg\,min}_{\mathcal{P}\in\mathcal{B}_{n,k}} h_{k,\varepsilon}(\mathcal{P},\mathcal{C}), \quad (11)$$

Note that Alg. 1 requires introducing the log-sum-exp operator: For a tensor $\mathcal{A} \in \mathcal{T}_{n,k}$, and a subset $I \subset [\![k]\!]$ of slices, $\mathrm{LSE}(\mathcal{A}, I)$ denotes the log-sum-exp operator on such slices (this corresponds to $\texttt{log sum}(\exp(\mathcal{A}), \texttt{axis} = I)$, with a 1 indexing convention).

The theoretical complexity of MM-S (Lin et al., 2022) is $\mathcal{O}(k^3 n^k \varepsilon^{-2})$, and involves in practice $k$ tensor reductions at each step, as highlighted in the for loop of Alg. 1. As with the standard Sinkhorn algorithm, smaller $\varepsilon$ requires a larger number of iterations, and early stopping can be controlled with the tolerance parameter $\alpha$. In our experiments, we set $\alpha = 10^{-3}$ and study the impact of $\varepsilon$ on the number of iterations needed to converge in §4.

**Algorithm 1** Multi-marginal Sinkhorn (MM-S)

**input:** cost tensor $\mathcal{C} \in \mathcal{T}_{n,k}$, regularization $\varepsilon$, tol. $\alpha$.
$\mathbf{F} = [\mathbf{f}^1, \dots, \mathbf{f}^k] = \mathbf{0}_{n\times k}$
**repeat**
  **for** $\ell, 1 \leq \ell \leq k$ **do**
    $\mathbf{f}^\ell \leftarrow -\varepsilon\left(\mathrm{LSE}\left(\frac{\bigoplus\mathbf{F}-\mathcal{C}}{\varepsilon}, [\![k]\!] \setminus \ell\right) + \log n\right)$
  **end for**
  $\mathcal{P} \leftarrow \exp\left(\left(\bigoplus \mathbf{F} - \mathcal{C}\right)/\varepsilon\right)$,
  $\delta \leftarrow \sum_{\ell=1}^k \|\mathrm{m}_\ell(\mathcal{P}) - \frac{\mathbb{1}_n}{n}\|_1$
**until** $\delta < \alpha$
**output:**
  Polystochastic tensor $\mathcal{P} \in \mathcal{T}_{n,k}$,
  MM-OT cost $\mathrm{OT}_{k,\varepsilon}(\mathcal{C}) = \frac{1}{n}\langle \mathbf{F}, \mathbb{1}_{n\times k}\rangle - \varepsilon\langle \mathcal{P}, \mathbb{1}_{n,k}\rangle$.

## 3. Multi-Marginal Matching Gap (M3G)

We present our main contribution, the multi-marginal matching gap (M3G) loss. The loss takes a $k \times n \times d$ tensor $\mathcal{X}$ of $k$ views for $n$ points of a batch, all represented as $d$-dimensional vectors. As sketched in Figure 1, the loss quantifies, informally speaking, whether the $k$ views for each of the $n$ points cluster sufficiently when taken as a whole, relative to all other points.

### 3.1. Ground-Truth and Multiway Costs

We introduce two crucial elements needed to define the M3G: the ground-truth polymatching provided by batches of aligned points, and a cost function that quantifies the concentration of a $k$-tuple of vectors.

**Ground-Truth Polymatching.** Let $\mathcal{J}_{n,k}$ be the identity tensor in $\mathcal{T}_{n,k}$ divided by $n$. This is the tensor of zeros, except for the $n$ diagonal indices, which are all equal to $\frac{1}{n}$:

$$[\mathcal{J}_{n,k}]_{i_1,\dots,i_n} = \tfrac{1}{n}\mathbf{1}_{i_1=\cdots=i_n}\,.$$

Naturally, $\mathcal{J}_{n,k} \in \mathcal{B}_{n,k}$. The polymatching described in that tensor could not be more simple: the $k$ views $\left(\mathbf{x}_i^1, \dots, \mathbf{x}_i^k\right)$ of each point $i \leq n$ are matched together.

**From Embeddings to Cost Tensors.** We use a multiway cost function $c : \mathbb{R}^{\overbrace{d\times\cdots\times d}^{k\text{ times}}} \to \mathbb{R}$ to construct, using the information contained in $\mathcal{X}$, a cost *tensor* that evaluates that multiway cost on *all* $n^k$ possible combinations of points (for each of the $k$ views, choose one among $n$ available points). We call $\mathcal{M}_c$ the operator from $\mathbb{R}^{k\times n\times d}$ to $\mathcal{T}_{n,k}$, defined as:

$$\mathcal{M}_c(\mathcal{X}) = [c\left(\mathbf{x}_{i_1}^1, \cdots, \mathbf{x}_{i_k}^k\right)]_{i_1,\dots,i_k}\,,$$

where all indices $1 \leq i_1, \dots, i_k \leq n$. The multiway cost function $c$ can be seen equivalently as a function from $\mathbb{R}^{n\times d}$ to $\mathbb{R}$. While several costs have been considered in the MM-OT literature, e.g. repulsive Coulomb costs in density functional theory (Pass, 2015)), we use the simplest cost in our

**Algorithm 2** Cost Tensor from Embeddings $\mathcal{M}_{c_{\text{cv}}}(\mathcal{X})$

**input:** embeddings tensor $\mathcal{X} \in \mathbb{R}^{k\times n\times d}$
$\mathcal{A} = \mathbf{0} \in \mathcal{T}_{n,k}$
$\mathcal{W} = \mathcal{X}(\texttt{None}, :, \texttt{None}) \cdot \mathcal{X}(:, \texttt{None}, :, \texttt{None})$
$\mathcal{D} = 2 - 2\mathcal{W}.\text{sum}(-1))$
**for** $\ell, 1 \leq \ell \leq k$ **do**
  **for** $m, \ell + 1 \leq m \leq k$ **do**
    $I = [\![\ell - 1]\!] + [\![\ell + 1, m - 1]\!] + [\![m + 1, k]\!]$
    $\mathcal{A} = \mathcal{A} + \text{expand\_dims}(\mathcal{D}[\ell, m, ...], I - 1)$
  **end for**
**end for**
**output:** Cost tensor $\mathcal{M}_{c_{\text{cv}}}(\mathcal{X}) = \left(\frac{2}{k-1}\right)^2 \mathcal{A} \in \mathcal{T}_{n,k}$

setting, quantified as the norm of the average of $k$ points on the sphere (whose norm is necessarily smaller than 1), following insights from directional statistics (Ley & Verdebout, 2017). We define first the *resultant* length of a set of a points on the sphere,

$$R^2(\mathbf{z}_1, \ldots, \mathbf{z}_k) := \|\tfrac{1}{k}\sum_{\ell} \mathbf{z}_\ell\|^2 = 1 - 2\tfrac{k}{k-1}\sum_{\ell<m} \|\mathbf{z}_\ell - \mathbf{z}_m\|^2$$

Note that the rightmost reformulation above, using pairwise distances, allows for a more efficient computation, provided in Algorithm 2. The *circular variance* can quantify dispersion for these $k$ points as:

$$c_{\text{cv}}(\mathbf{z}_1, \ldots, \mathbf{z}_k) = 1 - R^2(\mathbf{z}_1, \ldots, \mathbf{z}_k)\,. \tag{12}$$

We have also tested an alternative, the MLE variance parameter of the wrapped Gaussian given these $k$ points, a.k.a. the circular standard deviation,

$$c_{\text{csd}}(\mathbf{z}_1, \ldots, \mathbf{z}_k) = -\log(R^2(\mathbf{z}_1, \ldots, \mathbf{z}_k))\,. \tag{13}$$

We use $c_{cv}$ by default in all experiments, and only consider $c_{csd}$ in the ablation studies in § 5.

### 3.2. Multi-Marginal Matching Gap

With these definitions, we can define the M3G loss:

**Definition 3.1.** The multimodal multi-marginal matching gap (M3G) of data tensor $\mathcal{X}$, parameterized by a multiway cost $c$ and $\varepsilon > 0$, is the gap to optimality of the ground-truth matching tensor $\mathcal{J}_{n,k}$:

$$\begin{aligned}\text{M3G}_{c,\varepsilon}(\mathcal{X}) &:= h_{k,\varepsilon}(\mathcal{J}_{n,k}, \mathcal{M}_c(\mathcal{X})) - \inf_{\mathcal{P}\in\mathcal{B}_{n,k}} h_{k,\varepsilon}(\mathcal{P}, \mathcal{M}_c(\mathcal{X})) \\ &= \langle \mathcal{J}_{n,k}, \mathcal{M}_c(\mathcal{X})\rangle + \varepsilon \log n - \text{OT}_{k,\varepsilon}(\mathcal{M}_c(\mathcal{X}))\,. \end{aligned} \tag{14}$$

The idea of contrasting the loss of a ground-truth solution to that achieved by a solver parameterized by actionable inputs (here, ultimately, the encoder parameters) can be traced back to, e.g. structured SVMs (Tsochantaridis et al., 2005), and was investigated in more depth in the elegant framework of Fenchel-Young losses (Blondel et al., 2020). In the context of OT, a similar idea was used to define a regularizer for vector-to-vector mappings (Uscidda & Cuturi, 2023).

**Proposition 3.2.** *The* M3G *loss is non-negative. The gradient of the* M3G *losses only requires applying the vector-Jacobian operator (Blondel & Roulet, 2024, §2.3.5) of* $\mathcal{M}$*,* $\partial\mathcal{M}(\cdot)^*[\cdot]$*, evaluated at* $\mathcal{X}$*, to the difference of two polystochastic tensors, the ground-truth* $\mathcal{J}_{n,k}$ *and the optimal* $\mathcal{P}_\varepsilon(\mathcal{M}(\mathcal{X}))$ *given in Eq. (11):*

$$\nabla\text{M3G}(\mathcal{X}) = \partial\mathcal{M}(\mathcal{X})^*\left[\mathcal{J}_{n,k} - \mathcal{P}_\varepsilon(\mathcal{M}(\mathcal{X}))\right] \in \mathbb{R}^{k\times n\times d}\,.$$

These results come from an application of Fenchel-Young losses (Blondel et al., 2020). Briefly, the first result comes from the fact that M3G is an optimality gap; the second follows from the fact that $\text{OT}_{k,\varepsilon}$ is an unconstrained convex optimization problem, and therefore an application of Danskin's theorem (assuming Alg. 1 is run to low tolerance $\alpha$, which we do by setting it to $10^{-3}$) states that $\nabla\text{OT}_{k,\varepsilon}(\mathcal{C}) = \mathcal{P}_\varepsilon(\mathcal{C})$. This, combined with the chain-rule, gives the result.

**Deeper Dive into $k = 2$.** Although the case $k = 2$ is not the main focus of our work, we highlight that M3G does not reduce to an InfoNCE-like loss, even for two views. Indeed, for $k = 2$ only, and using notations from § 2.2 one recovers, up to the constant $\varepsilon \log n$, that:

$$\text{M3G}_{c,\varepsilon}([\mathbf{X}^1, \mathbf{X}^2]) = \tfrac{1}{n}\|\mathbf{X}^1 - \mathbf{X}^2\|^2 - \text{OT}_{2,\varepsilon}\left(\left[c(\mathbf{x}_i^1, \mathbf{x}_j^2)\right]_{ij}\right).$$

For $k = 2$, the M3G loss provides an alternative to the classic InfoNCE loss, and is related, but not equivalent to, the recent "inverse optimal transport" approach advocated in (Shi et al., 2023). We briefly discuss this link in §6.1.

### 3.3. Learning Representations with M3G

Suppose we are given a batch of $n$ objects $z_1, \ldots, z_n$, and that each of these objects is available in $k$ multiple views, either through data collection or augmentations $((z_i^1, \ldots, z_i^k))_i$. Broadly speaking, we consider parameterized networks, $f_{\theta_\ell}$, $\ell \leq k$, in which case $\theta$ would stand for the list of all parameters $(\theta_\ell)_\ell$. We assume that all networks take values in the $d$-sphere, $\{\mathbf{x} \in \mathbb{R} : \|\mathbf{x}\| = 1\}$. We propose to minimize the M3G loss on the $n \times k$ encodings of all these objects, for each minibatch.

$$\mathcal{L}(\theta) := \text{M3G}_{c,\varepsilon}\left([f_{\theta_\ell}(z_i^\ell)]_{\ell,i}\right)\,.$$

## 4. Experiments

We test the M3G loss in an SSL setting (ImageNet-1k) and two multimodal tasks (DomainNet and PhysioNet). In § 4.1 and § 4.2, we use a joint embedding student-teacher

Table 1: **Multiview models performance as a function of number of views, $k$, for models pre-trained on ImageNet-1k.** Evaluation of classification performance of M3G ($\varepsilon = 0.2$) in comparison to paiwise losses, BYOL and InfoNCE extended to multiview using either the *pairwise* sum across views ($\mathcal{L}_{\text{pwe}}$), or a *one vs. average-of-rest* ($\mathcal{L}_{\text{ave}}$). We evaluate the performance for varying $k$ views, with $n = 64$ batch size, trained for 300 epochs. Reported are mean and standard variation over five independent repetitions per setting. In bold is the top performing method per setting.

| # Views | Method | | | | |
|---|---|---|---|---|---|
| | $\text{BYOL}_{\text{ave}}$ | $\text{BYOL}_{\text{pwe}}$ | $\text{InfoNCE}_{\text{ave}}$ | $\text{InfoNCE}_{\text{pwe}}$ | M3G |
| $k = 2$ | $74.62 \pm 0.14$ | | $74.61 \pm 0.16$ | | $\mathbf{74.75 \pm 0.48}$ |
| $k = 3$ | $74.60 \pm 0.16$ | $75.16 \pm 0.09$ | $74.24 \pm 0.13$ | $75.36 \pm 0.10$ | $\mathbf{75.61 \pm 0.12}$ |
| $k = 4$ | $75.04 \pm 0.18$ | $75.06 \pm 0.10$ | $74.80 \pm 0.09$ | $75.26 \pm 0.16$ | $\mathbf{75.75 \pm 0.11}$ |

architecture, see § 2.1. The student network is evaluated on all modalities $[1, k] - i$, apart from one index $1 \leq i \leq k$, for which the teacher is used. Gradients are aggregated on the $k - 1$ evaluations. Index $i$ loops then across all $k$ modalities to form an aggregated loss. In§ 4.3 we use a single common network, as proposed by (Brüsch et al., 2023).

### 4.1. Multiview SSL Performance on ImageNet-1k

We use $k$ random augmentations for each image, and study the impact of $k$ on the linear performance of encoder models pre-trained on ImageNet-1k (Deng et al., 2009). We compare our loss M3G, with the previously suggested extensions of contrastive losses to $k \geq 3$, using either aggregation of *pairwise* contributions ($\mathcal{L}_{\text{pwe}}$, Eq. (3)), or the *one vs. average-of-rest* approach ($\mathcal{L}_{\text{ave}}$, Eq. (4)). We train and evaluate each setting in five independent repetitions, and report mean and standard deviation.

**Augmentations.** We use the augmentations introduced in BYOL (Grill et al., 2020) and SimCLR (Chen et al., 2020). These vary in the parameters used for the aggregated transformations–cropping, random flipping, random color jittering, Gaussian blur, and grayscale or solarization. See §A for details on the $k$ augmentation stacks.

**Architecture.** The backbone encoder is a ViT-B/16 architecture (Dosovitskiy et al., 2020), followed by an MLP projection head (see §A for details). We use a batch size of $n = 64$ per GPU. We train all models for 300 epochs.

**Results.** Table 1 indicates that the M3G loss performs slightly better than baselines alternatives for multiview learning, improving the linear classification accuracy by .25% and .49% respectively for $k = 3$ and $k = 4$, validating the soundness of M3G with a fairly small batch size.

**On Increasing Batch size.** As mentioned earlier in this section, results reported here use $k - 1$ student branches vs. 1 teacher. This allows dropping the forward activations of the teacher branch. Informally, for a batch size of $n$, given $s$ student branches, $k - s$ teacher branches, and writing $M$ for the memory cost needed to store all activations for a parameter $\theta$ (of a single student branch, for a single point) this yields a total memory cost of $\approx O(snM + n^k)$, taking into account the cost of the MM-S cost tensor. This results in a trade-off, since smaller $s$ can allow for larger $n$ of $k$, depending on the magnitude of $M$. We leave this direction for future research.

### 4.2. Multimodal Domain Adaptation

We consider a domain adaptation (DA) task, where the goal is to learn a common encoder, followed by one or multiple classifiers, using labeled data from multiple domains. We quantify the generalization power of this pre-trained encoder with a classification task, tested on data coming from a new, completely unseen domain (Peng et al., 2019; Gulrajani & Lopez-Paz, 2020). We assume that the new data, despite coming from an unseen domain, still falls in the same classes. For that purpose, we use the DomainNet dataset (Peng et al., 2019). The dataset consists of 569,010 images divided into 345 different categories, and subdivided in 6 different domains–real photos (*rel*), cliparts (*clp*), sketches (*skt*), infographic images (*inf*), artistic paintings (*pnt*), and quickdraw (*qdr*).

**Training Procedure.** We consider the same losses: M3G and the baselines, InfoNCE and BYOL (each evaluated using both aggregation choices, pwe and ave). We pick one domain that acts as the unseen modality, and train representations on the $k = 5$ remaining domains. Following the conventions set in §3, the $n \times k$ points are sampled by picking randomly $n$ classes, and for each class, $k$ images in the dataset, coming from each of the $k = 5$ domains. For each of the settings we pre-train four independent representation encoders. The encoders' backbone is identical to that used on the ImageNet-1 dataset (ViT-B-16). We repeat this for each of the six modalities–implying a total of $5 \times 4 \times 6 = 120$ models. All models are trained using the same architecture and parameters choice (for details see §A), and evaluated on two tasks:

**5 Domains *vs.* 1.** For the first assessment, we train a linear classifier jointly on all 5 seen domains, and report test

Table 2: **Predictions accuracy over unseen unlabeled domains over the DomainNet dataset.** Evaluation of classification performance of M3G ($\varepsilon = 0.05$) compared to two pairwise losses, considering $\mathcal{L}_{\text{pair}} \in \{\text{InfoNCE}, \text{BYOL}\}$. For each we evaluate the *pairwise* sum across views ($\mathcal{L}_{\text{pwe}}$), and a *one vs. average-of-rest* ($\mathcal{L}_{\text{ave}}$). All models are trained with a fixed batch size of $n = 16$ for 300 epochs. Mean and standard deviation of performance reported for four independent repetitions. In bold is the top performing method per setting.

| Domains | Method | | | | |
|---|---|---|---|---|---|
| seen $\rightarrow$ unseen | $\text{BYOL}_{\text{ave}}$ | $\text{BYOL}_{\text{pwe}}$ | $\text{InfoNCE}_{\text{ave}}$ | $\text{InfoNCE}_{\text{pwe}}$ | M3G |
| $\neg$*clp* $\rightarrow$ *clp* | $24.1 \pm 0.2$ | $9.6 \pm 4.8$ | $23.2 \pm 0.4$ | $6.9 \pm 10.3$ | $\mathbf{32.4 \pm 0.3}$ |
| $\neg$*inf* $\rightarrow$ *inf* | $10.2 \pm 0.1$ | $5.8 \pm 0.8$ | $11.0 \pm 0.2$ | $10.1 \pm 0.4$ | $\mathbf{12.2 \pm 0.1}$ |
| $\neg$*pnt* $\rightarrow$ *pnt* | $28.3 \pm 0.1$ | $21.3 \pm 0.9$ | $30.6 \pm 0.6$ | $24.7 \pm 1.8$ | $\mathbf{31.3 \pm 0.1}$ |
| $\neg$*qdr* $\rightarrow$ *qdr* | $7.8 \pm 0.2$ | $3.4 \pm 2.2$ | $8.8 \pm 0.2$ | $8.5 \pm 0.5$ | $\mathbf{10.4 \pm 0.3}$ |
| $\neg$*rel* $\rightarrow$ *rel* | $43.1 \pm 0.3$ | $30.9 \pm 9.3$ | $44.6 \pm 0.1$ | $42.0 \pm 0.6$ | $\mathbf{46.3 \pm 0.2}$ |
| $\neg$*skt* $\rightarrow$ *skt* | $21.6 \pm 0.6$ | $10.1 \pm 2.1$ | $24.9 \pm 0.1$ | $20.5 \pm 1.6$ | $\mathbf{26.5 \pm 0.4}$ |

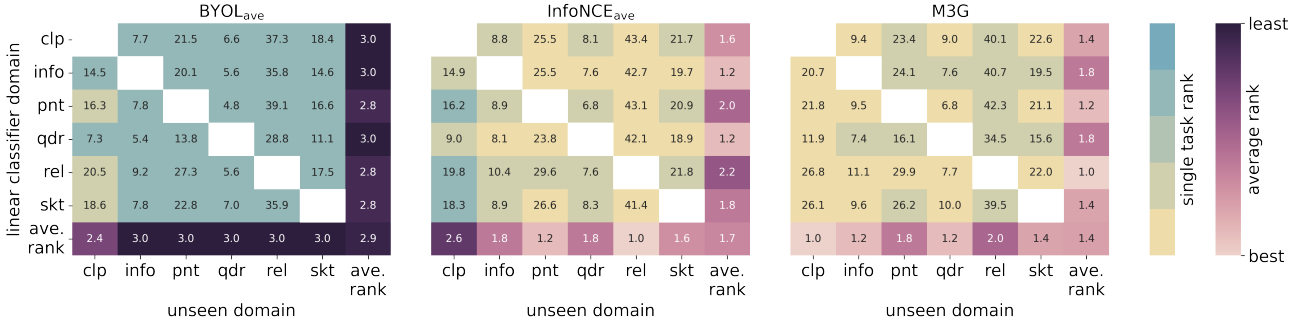


Figure 2: **Pairwise domain prediction accuracy on the DomainNet dataset.** The prediction accuracy over the unseen domain using a linear classifier trained on a single domain in the pre-training train set. Each table presents the performance of a different model choice. From left to right, baseline approaches, using the pairwise losses to evaluate *one vs. average-of-rest*, $\text{BYOL}_{\text{ave}}$ (left) and $\text{InfoNCE}_{\text{ave}}$ (center), compared to M3G (right). Columns correspond to the unseen domains and rows to the domains used for the linear classifier training. Mean performance reported for four independent repetitions.

results on the unseen domain, see Table 2. While prediction accuracy in this task varies according to domain, ranging from 11% to 30% for M3G, M3G is consistently ranked 1st, with a mean improvement of 3.1% over the 2nd best model ($\text{InfoNCE}_{\text{ave}}$). In contrast to the ImageNet-1k task (§4.1), the *ave* baseline outperforms the *one vs. rest*.

**1 Domain *vs.* 1.** Next, we consider a harder task, training five independent classifiers, one per domain. Figure 2 reports the prediction accuracy of these classifiers when tested using images from the unseen domain. The performance of the pwe approach is overall much worse in this task as well, and presented in § C, Figure A1.

We find that globally M3G outperforms baseline approaches, with an average rank of 1.4 across all tasks (30 linear evaluations $\times$ 4 independent repetitions $=$ 120).

### 4.3. EEG Data

Health records often contain multi-channel time series data, available in vast amounts, but that require manual annotations by domain experts. Because channels provide aligned data points, they provide a testbed for multimodal embedding approaches. We apply directly the M3G loss on an EEG dataset using $k = 6$ channels, taken from the PhysioNet Challenge 2018 (Goldberger et al., 2000; Ghassemi et al., 2018). EEG is a neurophysiological technique that records and measures the brain's electrical activity. The train data contains segmented samples of 994 individuals, and the evaluation dataset, SleepEDFx (Goldberger et al., 2000; Kemp et al., 2000), contains 153 nights of sleep recordings from 78 individuals, each annotated as belonging to one among five classes of sleep stage.

**Classification.** The task is to accurately predict the sleep stage using a sample of 30s. Freezing the pre-trained representation model, we train a linear encoder over samples of 10, 50, 100, and 1000 data points for each of the five classes, and evaluate the prediction accuracy over the same number of samples respectively. We reuse the codebase provided by (Brüsch et al., 2023).

**Results.** In Table 3 we compare M3G to the *pairwise* In-

foNCE loss. In accordance with previous results we find the M3G performs better than $\text{InfoNCE}_{\text{pwe}}$.

Table 3: **Prediction accuracy over multichannel EEG dataset.** Linear prediction accuracy of M3G and baseline method on a $k = 6$ EEG channels dataset using different numbers of samples per class $s \in \{10, 50, 100, 1000\}$. We report mean results, averaged over 5 seeds. $\text{InfoNCE}_{\text{pwe}}$ models are trained using a batch size of $n = 64$ and M3G uses a batch size of $n = 16$. All models are pre-trained for 10 epochs and fine-tuned on the classification task for a maximum of 40 epochs.

| Method | samples per class | | | |
|---|---|---|---|---|
| | $s = 10$ | $s = 50$ | $s = 100$ | $s = 1000$ |
| $\text{InfoNCE}_{\text{pwe}}$ | 35.5 | **49.3** | 52.2 | 56.3 |
| M3G, $\varepsilon = 0.2$ | **36.6** | 49.2 | **56.1** | **64.6** |

## 5. Ablation studies

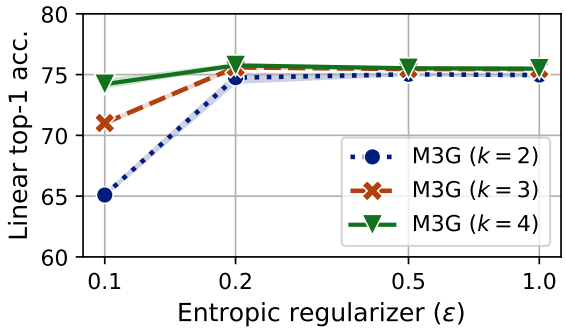


Figure 3: **Linear performance on ImageNet-1k as a function of the entropic regularizer $\varepsilon$.** We report the linear top-1 accuracy for different values of the MM-OT entropic regularizer $\varepsilon$, as we vary the view multiplicity, $k$. All results are given for the same batch size ($n = 64$) and training duration (300 epochs). Solid line and band depict the mean and 95% confidence interval over five independent repetitions.

The M3G loss is parameterized by a multiway cost function $c$, and by entropic regularization $\varepsilon$. We study how these two choices affect performance. We limit our study to the ImageNet-1K multiview task.

**Entropic Regularization.** Because embeddings are always normalized, and costs depend directly on dot-products, the range of cost values is constrained. Thanks to this, setting $\varepsilon$ was fairly easy. As shown in Fig. 3, and observed in most of our other experiments, overall performance is fairly robust to $\varepsilon$. Setting $\varepsilon = 0.2$ returned consistently good results.

**Cost Function.** The multiway cost function $c$ is the other important degree of freedom available to the user to shape the M3G loss. Apart from the circular variance used throughout our experiments ($\text{M3G}_{\text{cv}}$) we evaluate the performance using a circular standard deviation cost ($\text{M3G}_{\text{csd}}$). We observe that M3G is robust to this choice, attaining similar performance under both cost choices, see Table 4.

| Views | Method | |
|---|---|---|
| | $\text{M3G}_{\text{cv}}$ | $\text{M3G}_{\text{csd}}$ |
| $k = 2$ | 74.75 | 73.81 |
| $k = 3$ | 75.61 | 75.63 |
| $k = 4$ | 75.75 | 75.73 |

Table 4: **Robustness to cost function.** Classification performance of M3G models pre-trained on ImageNet-1k, using either $c_{\text{cv}}$, Eq. (12) or $c_{\text{csd}}$, Eq. (13), with $\varepsilon = 0.2$.

**Compute Overhead.** Despite the daunting cost of running the MM-Sinkhorn (1964) algorithm in $n^k$, we show that in the most computationally demanding of our tasks (ImageNet-1k), using M3G only incurs a relatively minor compute overhead. This is summarized in Figure 4.

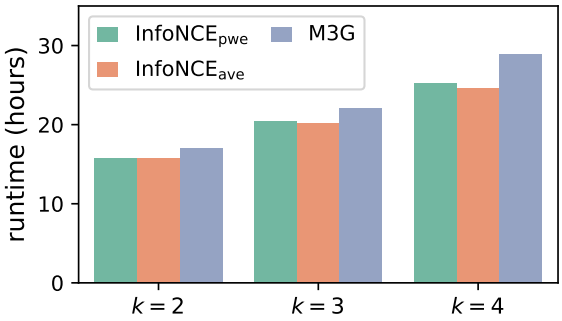


Figure 4: **Compute overhead incurred by M3G on ImageNet-1k as a function of $k$.** All results are given for the same per GPU batch size ($n = 64$), 300 epochs, $\varepsilon = 0.2$ for M3G, run on 4 nodes of 8 A100 GPUs.

## 6. Discussion

### 6.1. Related Works

**MM-OT.** When the number of views $k \geq 3$, we are not aware of any other work that uses MM-OT to study multiple representations of objects. Compared to regular OT, MM-OT has been used in far fewer applications, notably to handle density functional theory in chemistry with Coulomb costs (Pass, 2015; Benamou et al., 2017). MM-OT has very recently started playing a role in core ML tasks, e.g. with recent links to adversarial multiclass classification (Trillos et al., 2023). Much like regular OT, MM-OT has also been extended to accommodate unbalanced constraints (Beier et al., 2023) or quadratic (Gromov-Wasserstein-like) objectives (Beier et al., 2022). Solving the MM-OT problem raises many challenges that are increasingly better understood in theory (Le et al., 2022; Lin et al., 2022; Altschuler & Boix-Adsera, 2021). Finding alternative schemes to compute or approximate MM-OT is a very recent and active

research subject, using e.g. ODEs (Nenna & Pass, 2023) or by exploiting a more specific structure in costs (Haasler et al., 2021b;a). These ideas might be employed to speed up our scheme, as using Danskin's theorem leaves ample room for solving MM-OT with any forward pass, without having to go through a differentiable solver.

**The case $k = 2$.** When restricting our contribution to the simplest case $k = 2$, our method reduces to a "classic" OT formulation, solved with the usual Sinkhorn algorithm. Closest to our method lies the proposal of Shi et al. (2023) to use Inverse OT as a loss in SSL. There is, however, a significant discrepancy between their work and our loss for $k = 2$: Shi et al. (2023) define their loss as the KL divergence between the ground truth identity matching matrix $\mathbf{J}_n$, and the optimal coupling returned by Sinkhorn $\mathbf{P}_\varepsilon(\mathbf{C}(\mathbf{X}))$. To compute the gradient of their loss, they need, therefore, to differentiate through Sinkhorn iterations. While this can be done by either unrolling iterations, or using the implicit function theorem (Luise et al., 2018; Cuturi et al., 2022), this adds memory and compute requirements. Because $\mathrm{M3G}$ is an optimality gap, namely a Fenchel-Young loss (Blondel et al., 2020), we can avoid that backward pass thanks to Danskin's theorem. The recent proposal of Jiang et al. (2023) is also closely related to $\mathrm{M3G}$ loss when $k = 2$, since they propose to apply OT weights within the negative sample reweighting approach of Robinson et al. (2020).

### 6.2. Limitations

An important limitation of the $\mathrm{M3G}$ loss lies in solving an MM-OT problem, using the MM-Sinkhorn, see Alg. 1. Computing the $\mathrm{M3G}$ loss incurs a cost that scales as $O(n^k)$, preventing, in practice, using large batch sizes. We believe this exponential scaling is likely the price to pay to account simultaneously for all $k$-tuples of views. Our experiments show that for small batch sizes (e.g. $n = 16, 32, 64$) and small $k$ (we considered $k \leq 6$), this compute overhead was reasonable, notably when compared to the cost of running large encoders, such as ViT-B/16 models (See 4). However, this increase will remain intractable for larger $k$ values if one uses, as we did, a generic multiway cost. While we studied an alternative cost ($\mathrm{csd}$) with similar compute, this did not yield significantly different results. As future work, we believe one might explore better cost functions, either for computational or modeling reasons, e.g. using domain-specific knowledge that focuses on specific subsets of the $k$ views. Aside from this limitation, our loss remains, however, fairly simple, since it only has two hyperparameters: the cost function $c$ itself and the $\varepsilon$ regularization. We have observed good performance for most $\varepsilon$ choices but suspect that $\varepsilon$ should depend on the batch-size for best performance.

### 6.3. Conclusion

To our knowledge, the $\mathrm{M3G}$ loss is the first contrastive loss proposed to learn multi-representations that takes a holistic view of all $k$ views (when $k \geq 3$) of a given object. Specifically, $\mathrm{M3G}$ avoids contrasting views in a pairwise approach, and relies instead on a cost function that scores instead the coherence of a family of $k$ point embeddings. That score is computed for each of the $n$ objects, seen through their $k$-views, and subsequently averaged. It is then compared with the cost of the best polymatching tensor that can be obtained using $n \times k$ views pooled together. The latter is approximated using the multi-marginal Sinkhorn algorithm. The $\mathrm{M3G}$ loss and its gradient can be computed with a single forward execution of the multi-marginal Sinkhorn algorithm. While the application of the $\mathrm{M3G}$ loss to practical tasks may seem daunting, because of the exponential complexity in $k$ incurred when running the MM-Sinkhorn algorithm, we show that the overhead paid to compute this loss, in terms of running time, is manageable as long as batch size $n$ and $k$ are not too large. We have presented promising performance on a variety of self-supervised and multimodal tasks, paving the way for future extensions that can leverage more informed cost structures.

## Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## References

Akbari, H., Yuan, L., Qian, R., Chuang, W.-H., Chang, S.-F., Cui, Y., and Gong, B. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34: 24206–24221, 2021.

Alayrac, J.-B., Recasens, A., Schneider, R., Arandjelović, R., Ramapuram, J., De Fauw, J., Smaira, L., Dieleman, S., and Zisserman, A. Self-supervised multimodal versatile networks. *Advances in Neural Information Processing Systems*, 33:25–37, 2020.

Altschuler, J. M. and Boix-Adsera, E. Hardness results for multimarginal optimal transport problems. *Discrete Optimization*, 42:100669, 2021.

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., and Ballas, N. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/ddf354219aac374f1d40b7e760ee5bb7-Paper.pdf.

Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian, Y., et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

Banville, H., Chehab, O., Hyvärinen, A., Engemann, D.-A., and Gramfort, A. Uncovering the structure of clinical eeg signals with self-supervised learning. *Journal of Neural Engineering*, 18(4):046020, 2021.

Bardes, A., Ponce, J., and LeCun, Y. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Bardes, A., Ponce, J., and LeCun, Y. Vicregl: Self-supervised learning of local visual features. *Advances in Neural Information Processing Systems*, 35:8799–8810, 2022b.

Beier, F., Beinert, R., and Steidl, G. Multi-marginal gromov-wasserstein transport and barycenters. *arXiv preprint arXiv:2205.06725*, 2022.

Beier, F., von Lindheim, J., Neumayer, S., and Steidl, G. Unbalanced multi-marginal optimal transport. *Journal of Mathematical Imaging and Vision*, 65(3):394–413, 2023.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

Benamou, J.-D., Carlier, G., and Nenna, L. A numerical method to solve multi-marginal optimal transport problems with coulomb cost. In *Splitting Methods in Communication, Imaging, Science, and Engineering*, pp. 577–601. Springer, 2017.

Benson-Putnins, D. Counting integer points in multi-index transportation polytopes. *arXiv preprint arXiv:1402.4715*, 2014.

Birkhoff, G. Tres observaciones sobre el algebra lineal. *Universidad Nacional de Tucumán Revista Series A*, 5: 147–151, 1946.

Blondel, M. and Roulet, V. The elements of differentiable programming. *arXiv preprint arXiv:2403.14606*, 2024.

Blondel, M., Martins, A. F., and Niculae, V. Learning with fenchel-young losses. *The Journal of Machine Learning Research*, 21(1):1314–1382, 2020.

Brüsch, T., Schmidt, M. N., and Alstrøm, T. S. Multi-view self-supervised learning for multivariate variable-channel time series. In *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2023.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chen, X., Xie, S., and He, K. An Empirical Study of Training Self-Supervised Vision Transformers. *arXiv e-prints*, art. arXiv:2104.02057, April 2021. doi: 10.48550/arXiv.2104.02057.

Cheng, J. Y., Goh, H., Dogrusoz, K., Tuzel, O., and Azemi, E. Subject-aware contrastive learning for biosignals. *arXiv preprint arXiv:2007.04871*, 2020.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Cuturi, M., Meng-Papaxanthos, L., Tian, Y., Bunne, C., Davis, G., and Teboul, O. Optimal transport tools (ott): A jax toolbox for all things wasserstein. *arXiv preprint arXiv:2201.12324*, 2022.

Danskin, J. M. *The Theory of Max-Min and its Applications to Weapons Allocation Problems*, volume 5. Springer, 1967.

Deldari, S., Xue, H., Saeed, A., He, J., Smith, D. V., and Salim, F. D. Beyond just vision: A review on self-supervised representation learning on multimodal and temporal data. *arXiv preprint arXiv:2206.02353*, 2022.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *preprint arXiv:2010.11929*, 2020.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

Gangbo, W. and Święch, A. Optimal maps for the multidimensional monge-kantorovich problem. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 51(1): 23–45, 1998.

Ghassemi, M. M., Moody, B. E., Lehman, L.-W. H., Song, C., Li, Q., Sun, H., Mark, R. G., Westover, M. B., and Clifford, G. D. You snooze, you win: the physionet/computing in cardiology challenge 2018. In *2018 Computing in Cardiology Conference (CinC)*, volume 45, pp. 1–4. IEEE, 2018.

Girdhar, R., El-Nouby, A., Liu, Z., Singh, M., Alwala, K. V., Joulin, A., and Misra, I. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15180–15190, 2023.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220, 2000.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Bilal, P., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Haasler, I., Ringh, A., Chen, Y., and Karlsson, J. Multimarginal optimal transport with a tree-structured cost and the schrodinger bridge problem. *SIAM Journal on Control and Optimization*, 59(4):2428–2453, 2021a.

Haasler, I., Singh, R., Zhang, Q., Karlsson, J., and Chen, Y. Multi-marginal optimal transport and probabilistic graphical models. *IEEE Transactions on Information Theory*, 67(7):4647–4668, 2021b.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Hoffer, E., Ben-Nun, T., Hubara, I., Giladi, N., Hoefler, T., and Soudry, D. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8129–8138, 2020.

Jiang, R., Ishwar, P., and Aeron, S. Hard negative sampling via regularized optimal transport for contrastive representation learning. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2023.

Kemp, B., Zwinderman, A. H., Tuk, B., Kamphuisen, H. A., and Oberye, J. J. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering*, 47(9): 1185–1194, 2000.

Khaertdinov, B., Ghaleb, E., and Asteriadis, S. Contrastive self-supervised learning for sensor-based human activity recognition. In *2021 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–8. IEEE, 2021.

Kiyasseh, D., Zhu, T., and Clifton, D. A. Clocs: Contrastive learning of cardiac signals across space, time, and patients. In *International Conference on Machine Learning*, pp. 5606–5615. PMLR, 2021.

Le, K., Nguyen, H., Nguyen, K., Pham, T., and Ho, N. On multimarginal partial optimal transport: Equivalent forms and computational complexity. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I. (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4397–4413. PMLR, 28–30 Mar

2022. URL https://proceedings.mlr.press/v151/le22a.html.

Ley, C. and Verdebout, T. *Modern directional statistics*. CRC Press, 2017.

Liang, Z., Luo, Y., Beese, M., and Drexlin, D. J. Multiple positive views in self-supervised learning, 2024. URL https://openreview.net/forum?id=WGP2pHtLtn.

Lin, T., Ho, N., Cuturi, M., and Jordan, M. I. On the complexity of approximating multimarginal optimal transport. *The Journal of Machine Learning Research*, 23(1):2835–2877, 2022.

Luise, G., Rudi, A., Pontil, M., and Ciliberto, C. Differential properties of sinkhorn approximation for learning with wasserstein distance. *Advances in Neural Information Processing Systems*, 31, 2018.

Nenna, L. and Pass, B. An ode characterisation of multimarginal optimal transport with pairwise cost functions, 2023.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Palatucci, M., Pomerleau, D., Hinton, G. E., and Mitchell, T. M. Zero-shot learning with semantic output codes. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper_files/paper/2009/file/1543843a4723ed2ab08e18053ae6dc5b-Paper.pdf.

Pass, B. Multi-marginal optimal transport: theory and applications. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 49(6):1771–1790, 2015.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Peyré, G. and Cuturi, M. Computational Optimal Transport. *Foundations and Trends in Machine Learning*, 11(5-6), 2019. ISSN 1935-8245.

Pototzky, D., Sultan, A., and Schmidt-Thieme, L. Fastsiam: Resource-efficient self-supervised learning on a single gpu. In *DAGM German Conference on Pattern Recognition*, pp. 53–67. Springer, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Robinson, J. D., Chuang, C.-Y., Sra, S., and Jegelka, S. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2020.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294, 2022.

Shi, L., Zhang, G., Zhen, H., Fan, J., and Yan, J. Understanding and generalizing contrastive learning from the inverse optimal transport perspective. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31408–31421. PMLR, 23–29 Jul 2023.

Shidani, A., Hjelm, R. D., Ramapuram, J., Webb, R., Dhekane, E. G., and Busbridge, D. Poly-view contrastive learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=iHcTLIor0m.

Sinkhorn, R. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. Zero-shot learning through cross-modal transfer. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/2d6cc4b2d139a53512fb8cbb3086ae2e-Paper.pdf.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 776–794. Springer, 2020.

Tonekaboni, S., Eytan, D., and Goldenberg, A. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.

Trillos, N. G., Jacobs, M., and Kim, J. The multimarginal optimal transport formulation of adversarial multiclass classification. *Journal of Machine Learning Research*, 24 (45):1–56, 2023.

Tsai, Y.-H. H., Wu, Y., Salakhutdinov, R., and Morency, L.-P. Self-supervised learning from a multi-view perspective. In *International Conference on Learning Representations*, 2020.

Tsai, Y.-H. H., Wu, Y., Salakhutdinov, R., and Morency, L.-P. Self-supervised learning from a multi-view perspective. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-bdp_8Itjwp.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(50):1453–1484, 2005. URL http://jmlr.org/papers/v6/tsochantaridis05a.html.

Uscidda, T. and Cuturi, M. The monge gap: A regularizer to learn all transport maps. In *International Conference on Machine Learning*, pp. 34709–34733. PMLR, 2023.

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., and Xu, H. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

Zhou, P., Zhou, Y., Si, C., Yu, W., Ng, T. K., and Yan, S. Mugs: A multi-granular self-supervised learning framework. *arXiv preprint arXiv:2203.14415*, 2022.

# A. Implementation Details

**Multi-Marginal Sinkhorn Optimization.** To perform experiments, we implemented the multi-marginal Sinkhorn algorithm (Alg. 1) in PyTorch Paszke et al. (2019).

**Hyperparameters for Models Training.** In Table A1 we provide the hyperparameters used to train ImageNet-1k and DomainNet models. In all cases the encoder is based on ViT-B/16 architecture and the following projection and predictor heads consist of a linear layer with output size 4096 followed by Gaussian error linear units (GeLU) (Hendrycks & Gimpel, 2016)), and an additional linear layer with output dimension 256.

For the EEG dataset, we follow the setting reported in (Brüsch et al., 2023), using the implementation provided in the GitHub repository[1]. The network is composed of six convolutional blocks consisting of a 1D convolution, a dropout layer, a group normalization layer, and a GELU activation function. The kernel width and stride is three in the first layer and two in the remaining five layers. 256 kernels are used for all intermediate layers and the final output dimension is 64. A readout layer with kernel width and stride set to 1 is added at the end. We train models for 5 different seeds and report avergage results. All models are trained for 10 epochs and a batch size of $n = 16, 64$ for $\mathrm{M3G}, \mathrm{InfoNCE}_{\mathrm{pwe}}$ respectively.

**Augmentations.** For image datasets (ImageNet-1k and DomainNet), we use augmentation settings introduced in BYOL (Grill et al., 2020) and SimCLR (Chen et al., 2020). We provide the pseudocode for the augmentations used in Pseudocode 1. For ImageNet-1k the maximal stack ($k = 4$) is defined as `A = [byol-global1, byol-global2, simclr, byol-global1]`, for lower $k$ we take `A[:k]`. For DomainNet training we use the `simclr` augmentation for all views. In all cases, for test augmentations we follow the standard practice–resize, center crop and normalization.

Pseudocode 1: Definition of the train augmentations.

```
byol-global1 = [
    RandomResizedCrop(
        size=224,
        scale=(0.08, 1.0),
        interpolation=Image.BICUBIC
    ),
    RandomHorizontalFlip(p=0.5),
    RandomApply([
            ColorJitter(
                brightness=0.4,
                contrast=0.4,
                saturation=0.2,
                hue=0.1
            )
        ], p=0.8,
    ),
    RandomGrayscale(p=0.2),
    GaussianBlur(),
    Normalize(
        mean=(0.485, 0.456, 0.406),
        std=(0.229, 0.224, 0.225)
    )
]

byol-global2 = [
    RandomResizedCrop(
        size=224,
        scale=(0.08, 1.0),
        interpolation=Image.BICUBIC
    ),
    RandomHorizontalFlip(p=0.5),
    RandomApply([
            ColorJitter(
                brightness=0.4,
                contrast=0.4,
                saturation=0.2,
                hue=0.1
            )
        ], p=0.8,
    ),
    RandomGrayscale(p=0.2),
    RandomApply([GaussianBlur()], p=0.1),
    RandomApply([Solarization()], p=0.2),
    Normalize(
        mean=(0.485, 0.456, 0.406),
        std=(0.229, 0.224, 0.225)
    )
]

simclr = [
    RandomResizedCrop(
            size=224,
            scale=(0.08, 1.0),
            interpolation=Image.BICUBIC
        ),
    RandomHorizontalFlip(p=0.5),
    RandomApply([
            ColorJitter(
                brightness=0.8,
                contrast=0.8,
                saturation=0.8,
                hue=0.2
            )
        ], p=0.8,
    ),
    RandomGrayscale(p=0.2),
    RandomApply([GaussianBlur(kernel_size
        =23, sigma=[0.1, 2.0])], p=0.5),
    Normalize(
        mean=(0.485, 0.456, 0.406),
        std=(0.229, 0.224, 0.225)
    )
]
```

[1] https://github.com/theabrusch/Multiview_TS_SSL

# B. Training and Evaluation

**Linear Evaluation of Image Models.** For the image datasets tasks (ImageNet-1k and DomainNet) we follow the standard linear evaluation pipeline (Chen et al., 2021; He et al., 2016). We freeze the backbone encoder of the pre-trained model and train a linear classifier for 100 epochs on the data used for pre-training (ImageNet-1k or DomainNet respectively). We use the SGD optimizer with zero `weight_decay`. For the `learning_rate` we sweep over two possible values $(0.01, 0.001)$. Random horizontal flipping, random resized cropping and normalization are applied during training.

Table A1: **Vision models hyperparameters.**

| | |
|---|---|
| Encoder architecture | ViT-B/16 |
| Weight initialization | `trunc_normal(.02)` |
| Backbone normalization | LayerNorm |
| Batch size | 2048 (ImageNet-1k)<br>512 (DomainNet) |
| Head normalization | LayerNorm |
| Synchronized BatchNorm over replicas | True |
| Learning rate schedule | Single Cycle Cosine |
| Learning rate warmup (epochs) | 10 |
| Learning rate minimum value | $5 \times 10^{-5}$ |
| Training duration (epochs) | 300 |
| Optimizer | AdamW |
| Optimizer scaling rule | Adam |
| Base $(\beta_1, \beta_2)$ | (0.9, 0.95) |
| Base learning rate | $6.5 \times 10^{-4}$ |
| Per GPU Batch size | 64 (ImageNet-1k)<br>16 (DomainNet) |
| Base teacher momentum | 0.99 |
| Weight decay | 0.04 |
| Weight decay end | 0.4 |
| Weight decay warmup | 0.0 |

**DomainNet Downstream Evaluation** Using a pre-trained model with an unseen domain we freeze the encoder and test the model performance in two regimes: (i) train a single linear classifier, as described above, using the five seen domains. (ii) train five different classifiers, each over a single seen domain. All linear classifiers are tested over the unseen domain.

**Linear Evaluation of the EEG dataset.** We follow precisely the evaluation suggested by (Brüsch et al., 2023). Given the pre-trained model, consisting of a single encoder, a linear layer is used to combine all channel representations. A linear classifier is trained over the frozen joint representation for the classification task. Both layers are retrained from scratch.

## C. Additional Results

**Pairwise domain evaluation over DomainNet dataset.**

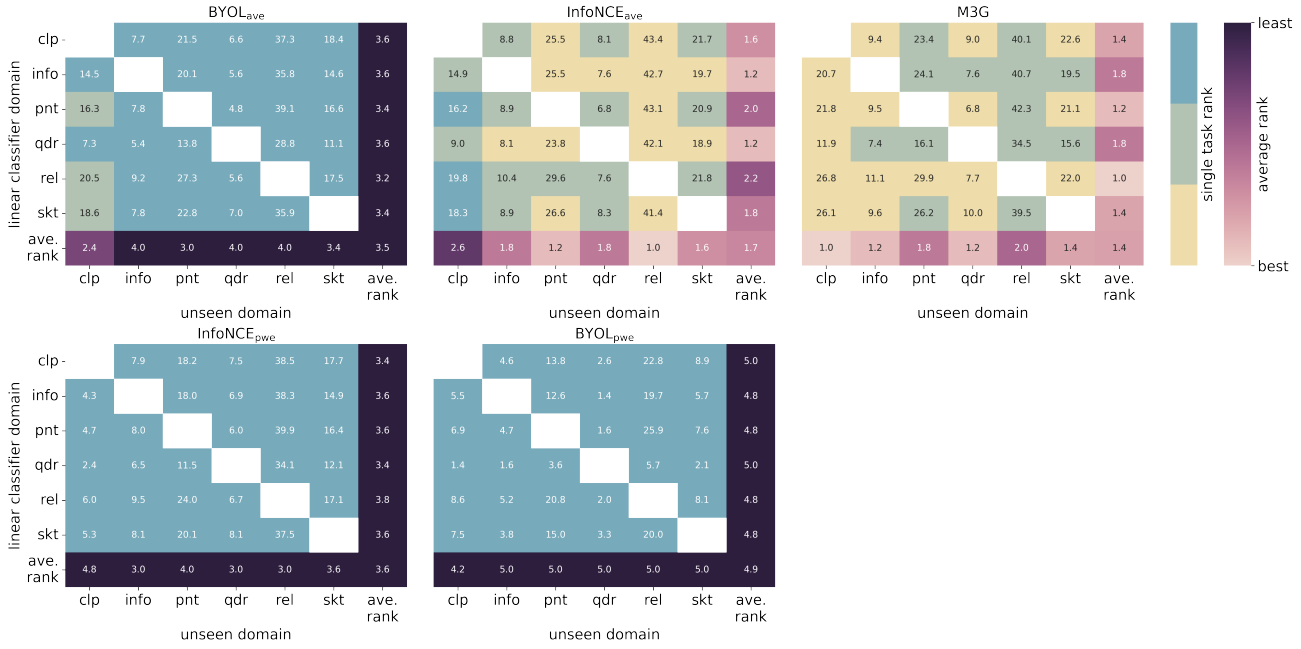


Figure A1: **Pairwise domain prediction accuracy on the DomainNet dataset.** The prediction accuracy over the unseen domain using a linear classifier trained on a single domain in the pre-training train set. Each table presents the performance of a different model choice. Two left columns present baseline approaches, using the pairwise losses to evaluate *one vs. average-of-rest* (top row) and *pairwise* (bottom row). Columns from left to right consider BYOL (left), InfoNCE (center), and M3G loss. In each subplot, columns correspond to the unseen domains and rows to the domains used for the linear classifier training. Mean performance reported for four independent repetitions.