Minimum Spanning Tree Regularization for Self-Supervised Learning

Julie Mordacq
Joint Work with David Loiseaux, Vicky Kalogeiton, Steve Oudot

Journées de Géométrie Algorithmique

École Polytechnique
Institut Polytechnique de Paris
Inria

Table of content

1. Introduction

Deep Learning

Given a training set $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{N}$, with $x_i \in \mathrm{R}^d$ an observation and $y_i \in \mathscr{Y} = \{0, \ldots, J\}$, we want to learn:

Deep Learning

Given a training set $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{N}$, with $x_i \in \mathrm{R}^d$ an observation and $y_i \in \mathscr{Y} = \{0, \ldots, J\}$, we want to learn:

Neural network:

A parametrized function (neural network) $f_\theta : \mathrm{R}^d \to \mathscr{Y}$, indexed by some parameters $\theta \in \Theta \subset \mathrm{R}^b$

Deep Learning

Given a training set $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{N}$, with $x_i \in \mathrm{R}^d$ an observation and $y_i \in \mathscr{Y} = \{0, \ldots, J\}$, we want to learn:

Neural network:

A parametrized function (neural network) $f_\theta : \mathrm{R}^d \to \mathscr{Y}$, indexed by some parameters $\theta \in \Theta \subset \mathrm{R}^b$

Optimization:

We want to find the **optimal parameters** $\boldsymbol{\theta}$**,** which minimise the empirical risk:

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} M_N(\theta) \quad \text{with} \quad M_N := \frac{1}{N} \mathscr{L}(y_i, f_\theta(x_i))$$

Where $\mathscr{L}$ is a cost function such as a mean squared error or cross-entropy.

Deep Learning

Given a training set $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{N}$, with $x_i \in \mathrm{R}^d$ an observation and $y_i \in \mathscr{Y} = \{0, \dots, J\}$, we want to learn:

Neural network:

A parametrized function (neural network) $f_\theta : \mathrm{R}^d \to \mathscr{Y}$, indexed by some parameters $\theta \in \Theta \subset \mathrm{R}^b$

Optimization:

We want to find the **optimal parameters θ,** which minimise the empirical risk:

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} M_N(\theta) \quad \text{with} \quad M_N := \frac{1}{N} \mathscr{L}(y_i, f_\theta(x_i))$$

Where $\mathscr{L}$ is a cost function such as a mean squared error or cross-entropy.

Gradient-based methods: back-propagation

Deep Learning

Given a training set $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{N}$, with $x_i \in \mathrm{R}^d$ an observation and $y_i \in \mathscr{Y} = \{0,\ldots,J\}$, we want to learn:

Neural network:

A parametrized function (neural network) $f_\theta : \mathrm{R}^d \to \mathscr{Y}$, indexed by some parameters $\theta \in \Theta \subset \mathrm{R}^b$

Optimization:

We want to find the **optimal parameters θ,** which minimise the empirical risk:

$$\hat{\theta} \in \arg\min_{\theta\in\Theta} M_N(\theta) \quad \text{with} \quad M_N := \frac{1}{N}\mathscr{L}(y_i, f_\theta(x_i))$$

Where $\mathscr{L}$ is a cost function such as a mean squared error or cross-entropy.

Universal approximation theorem

Neural networks with a given structure can, in principle, approximate any continuous function to any desired degree of accuracy

Deep Unsupervised Learning

Given a training set $\mathscr{D} = \{(x_i, \boxed{y_i})\}_{i=1}^N$, with $x_i \in \mathrm{R}^d$ an observation and $\boxed{y_i \in \mathscr{Y} = \{0, \ldots, J\}}$, we want to learn:

Neural network: a parametrized function $f_\theta : \mathrm{R}^d \to \boxed{\mathscr{Y}}$, indexed by some parameters $\theta \in \Theta \subset \mathrm{R}^b$.

Optimization:
We want to find the **optimal parameters θ,** which minimise the empirical risk:

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} M_N(\theta) \quad \text{with} \quad M_N := \frac{1}{N} \boxed{\mathscr{L}(y_i, f_\theta(x_i))}$$

Deep Unsupervised Learning

Given a training set $\mathscr{D} = \{x_i\}_{i=1}^{N}$, with $x_i \in \mathrm{R}^d$ an observation

Neural network: a parametrized function $f_\theta : \mathrm{R}^d \to \mathrm{R}^p$ typically with $d > p$, indexed by some parameters $\theta \in \Theta \subset \mathrm{R}^b$.

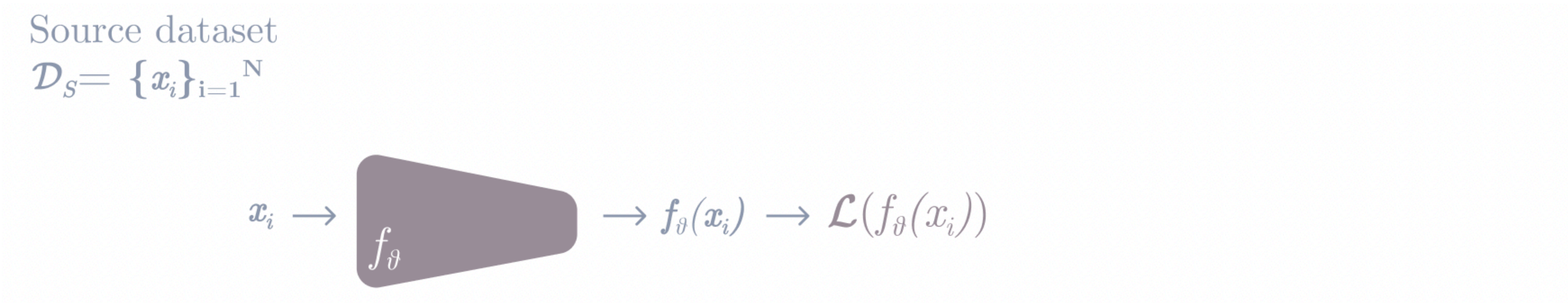
Optimization:

We want to find the **optimal parameters θ,** which minimise the empirical risk:

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} M_N(\theta) \quad \text{with} \quad M_N := \frac{1}{N} \mathscr{L}(f_\theta(x_i))$$

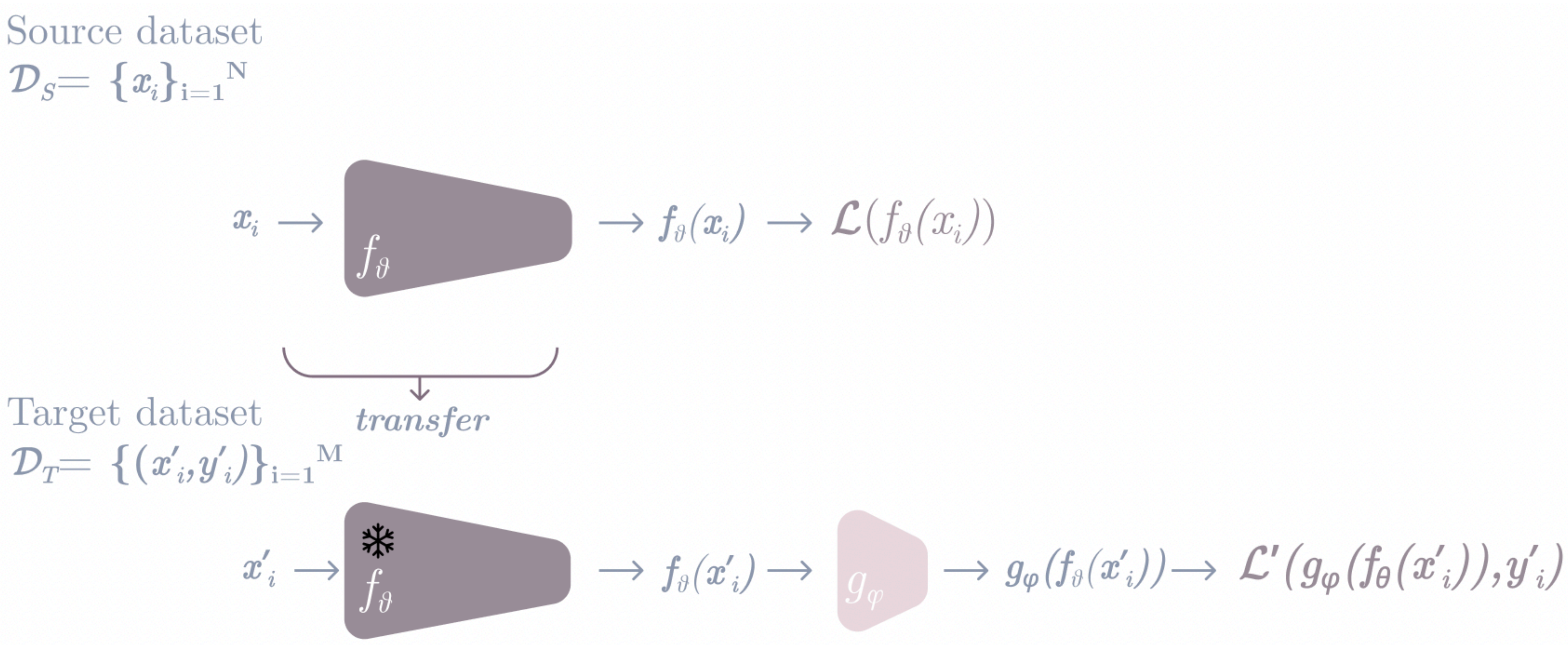
Self-supervised learning

Transfer Learning: transferring knowledge (e.g., learned weights: f_θ) from a source domain to improve performance in a target domain and task



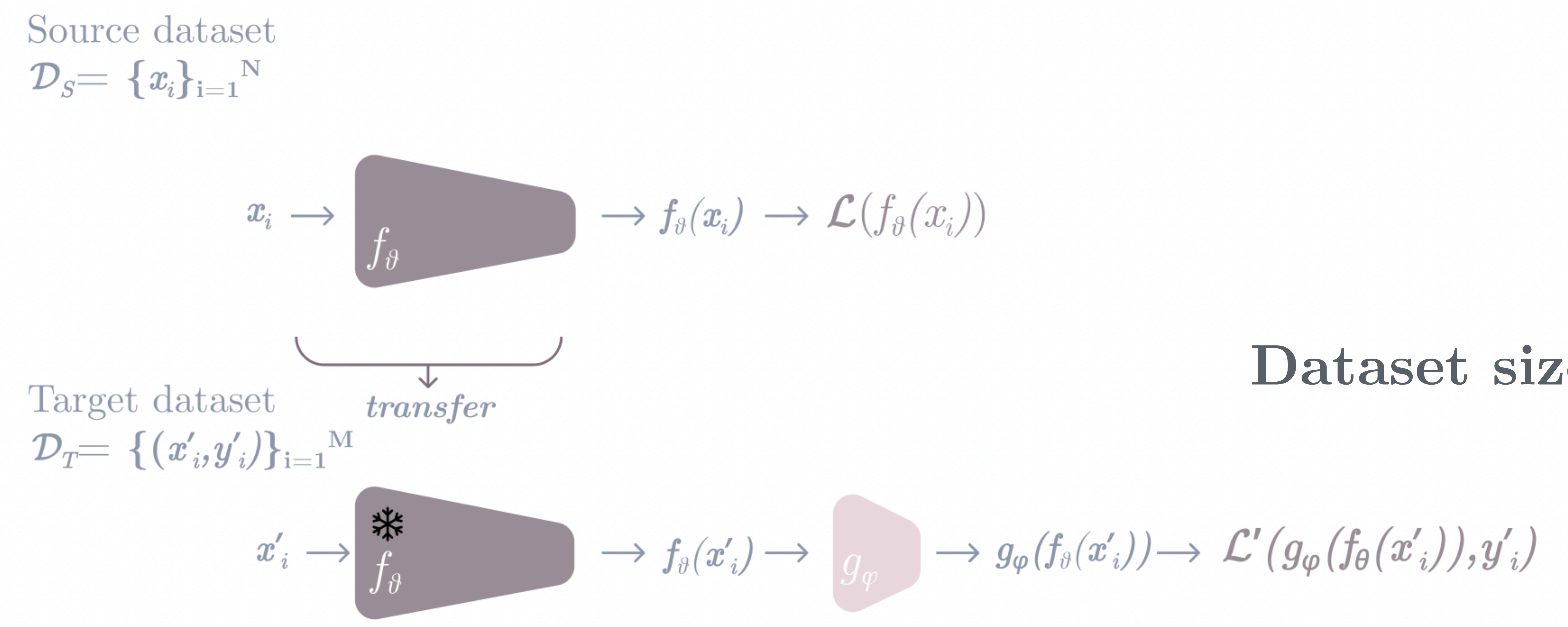
Self-supervised learning

Transfer Learning: transferring knowledge (e.g., learned weights: f_θ) from a source domain to improve performance in a target domain and task



Self-supervised learning

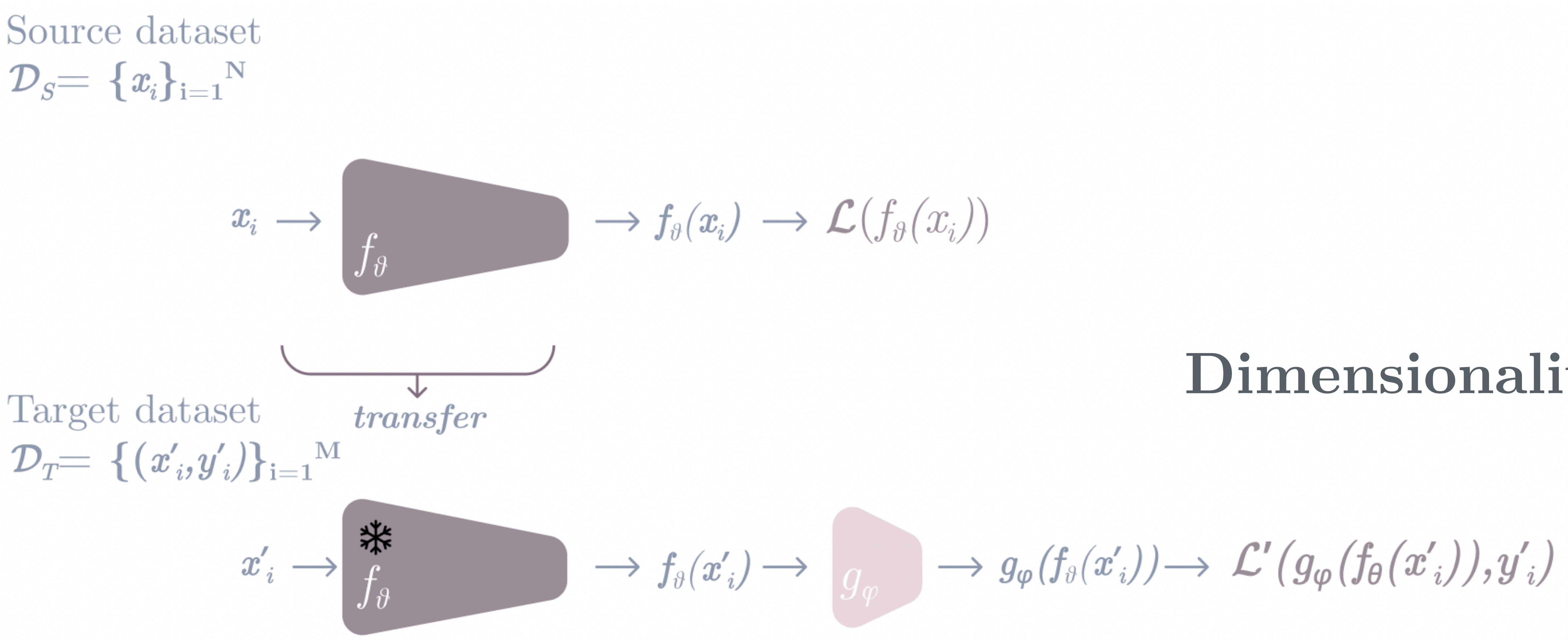
Transfer Learning: transferring knowledge (e.g., learned weights: f_θ) from a source domain to improve performance in a target domain and task



Dataset sizes: $N > M$

Self-supervised learning

Transfer Learning: transferring knowledge (e.g., learned weights: f_θ) from a source domain to improve performance in a target domain and task



Dimensionality reduction

Self-supervised learning

Transfer Learning: transferring knowledge (e.g., learned weights: f_θ) from a source domain to improve performance in a target domain and task

How to get **‘good’** and **‘general**’ representations without knowing the downstream task(s) ?

Self-supervised learning

Self-supervised learning: paradigm to learn meaningful data representations without relying on annotations, by leveraging supervision from the raw data itself

Self-supervised learning

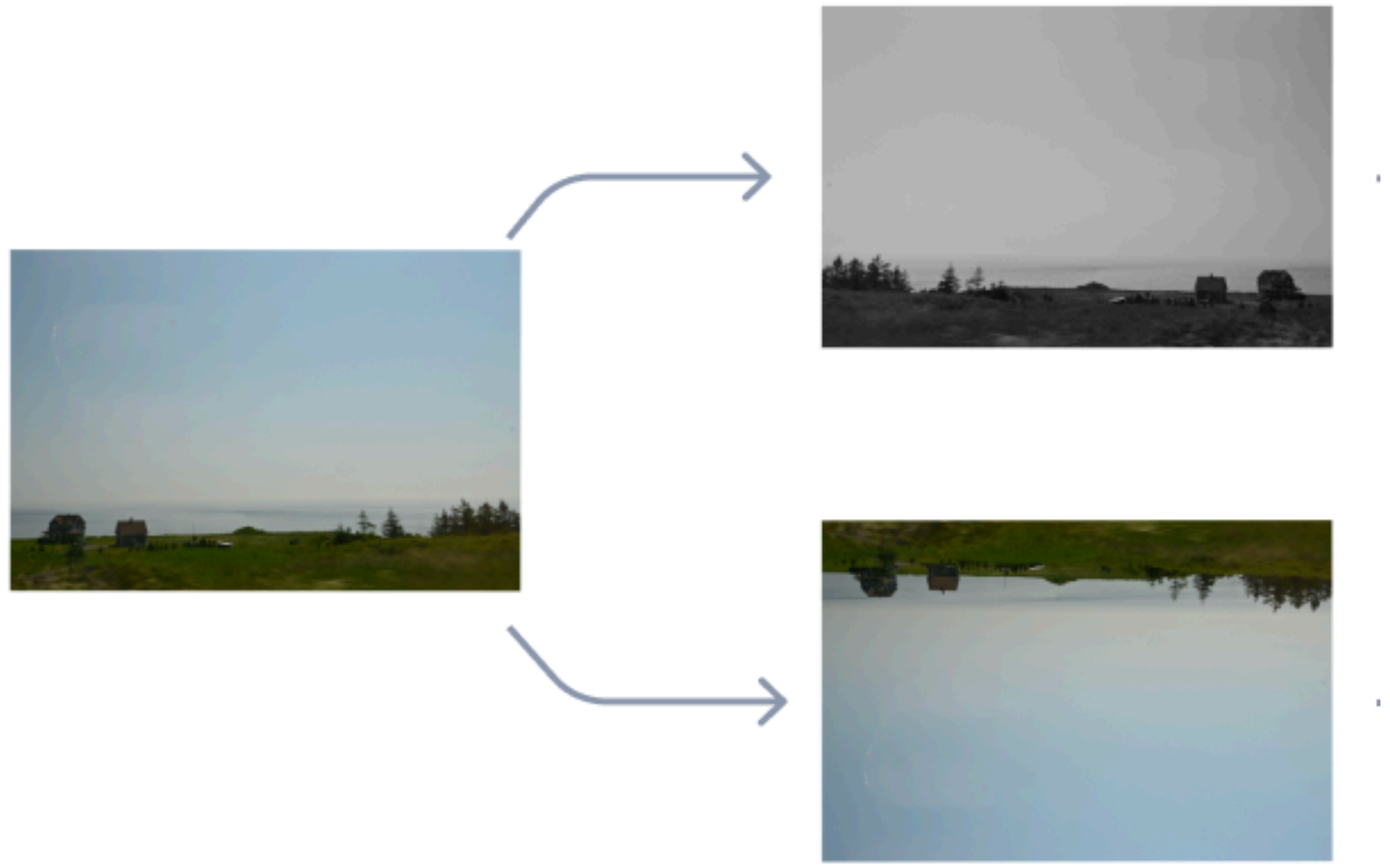
Self-supervised learning: paradigm to learn meaningful data representations without relying on annotations, by leveraging supervision from the raw data itself.

Invariance to a given class of transforms

Joint-Embedding Self-Supervised Learning

$i \in \mathrm{R}^{w \times h}$

Joint-Embedding Self-Supervised Learning

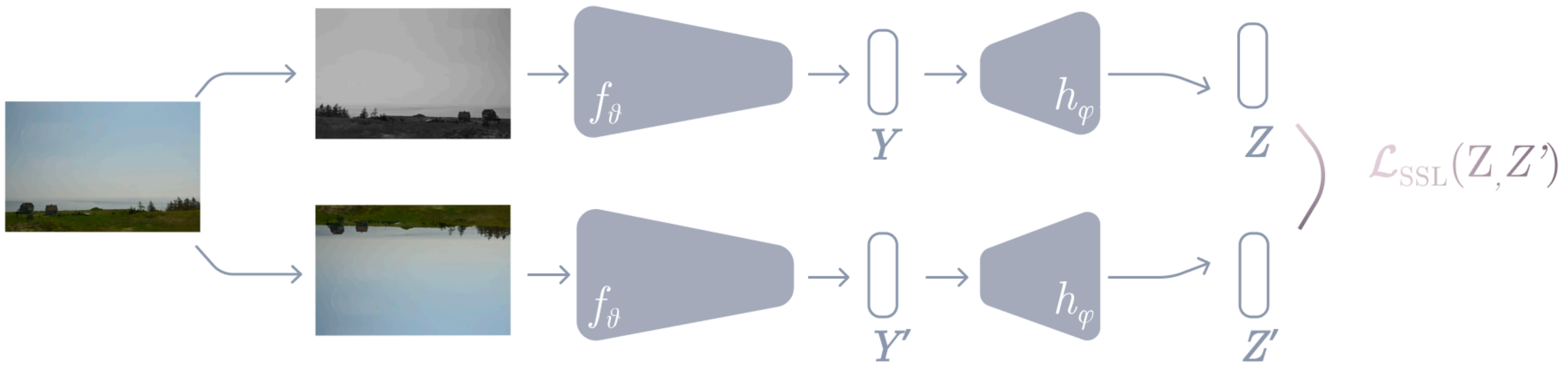


$i \in \mathrm{R}^{w \times h}$

$t, t' \sim T$

$x = t(i)$ and $x' = t'(i)$

Joint-Embedding Self-Supervised Learning



: $\mathcal{L}_{\mathrm{SSL}}(Z, Z')$ enforces view-invariance

$i \in \mathbb{R}^{w \times h}$

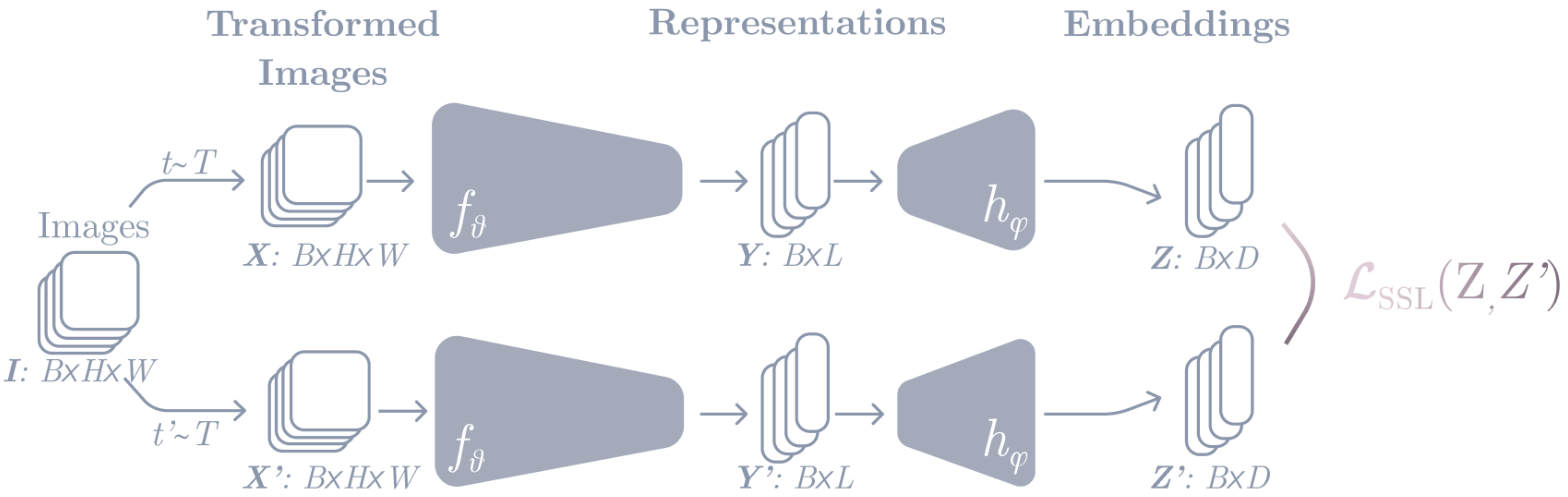
$t, t' \sim T$

$x = t(i)$ and $x' = t'(i)$

$y = f_\theta(x)$ and $y' = f_\theta(x')$

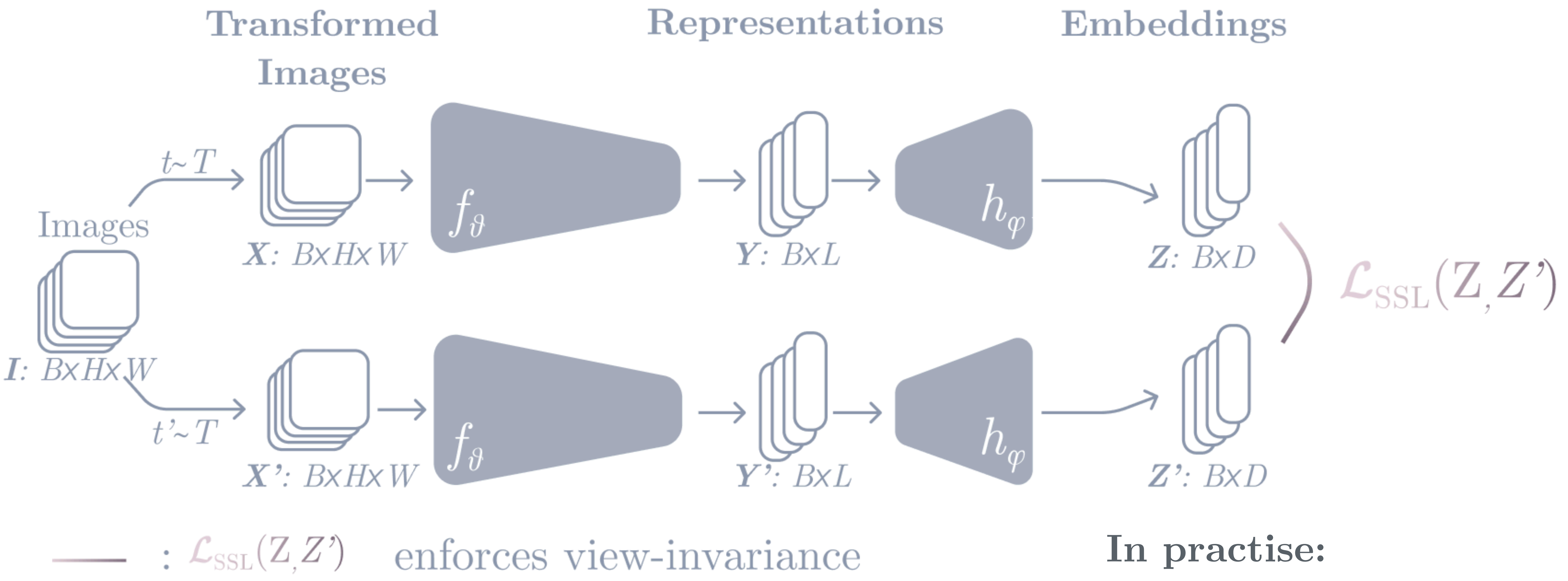
$z = h_\phi(y)$ and $z' = h_\phi(y')$

Joint-Embedding Self-Supervised Learning



: $\mathcal{L}_{\mathrm{SSL}}(\mathrm{Z}, Z')$ enforces view-invariance

Joint-Embedding Self-Supervised Learning

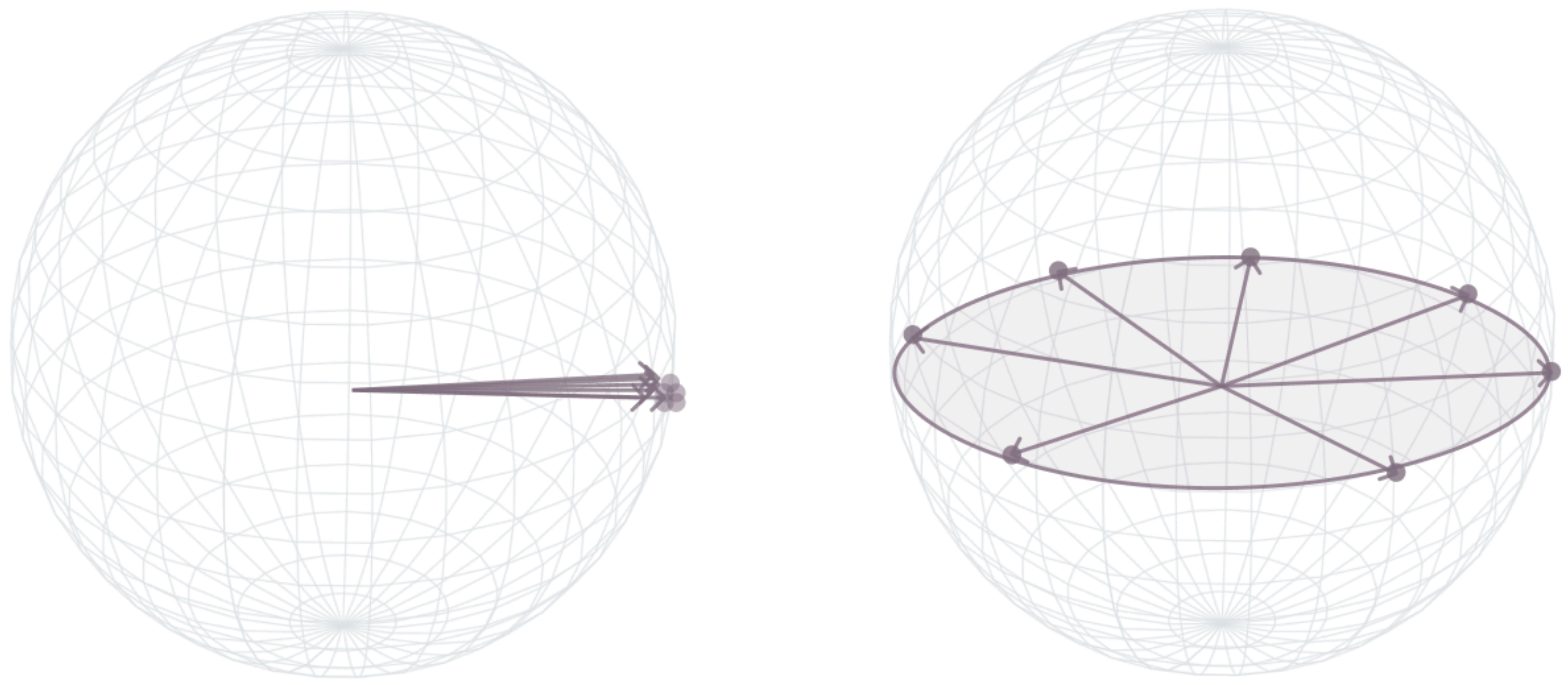


In practise:

- we may consider f_θ and h_ϕ as one network
- can be viewed as a dimensionality reduction problem as $D < H \times W$

Joint-Embedding Self-Supervised Learning

The fundamental challenge in JE-SSL:
preventing **collapse** either representational or dimensional



(a) Representational Collapse *(b)* Dimensional Collapse

Figure: **Illustration of collapse.**

Joint-Embedding Self-Supervised Learning

i. Contrastive approaches, e.g, SimCLR [1]
encourage embeddings of different views of the same image to be similar while pushing away embeddings of different images.

ii. Asymmetric architecture, e.g., BYOL [2]

iii. Covariance-based approaches, e.g., VICReg [3]
enforce embedding decorrelation by promoting an identity covariance matrix

[1] A Simple Framework for Contrastive Learning of Visual Representations, ICML 2020
[2] Bootstrap your own latent: A new approach to self-supervised Learning, NeurIPS 2020
[3] VICReg: Variance-Invariance-Covariance Regularization For Self-Supervised Learning, ICLR 2022

Joint-Embedding Self-Supervised Learning

Fang et al. [1] formalized desirable properties of JE-SSL:

1. **Mitigating dimensional collapse**: encourage the embeddings to span the entire space
2. **Promoting sample uniformity**: ensure the embeddings are evenly distributed across the representation space

[1] Rethinking uniformity in self-supervised learning, Fang et al, ICLR 2024

Joint-Embedding Self-Supervised Learning

Fang et al. [1] formalized desirable properties of JE-SSL:

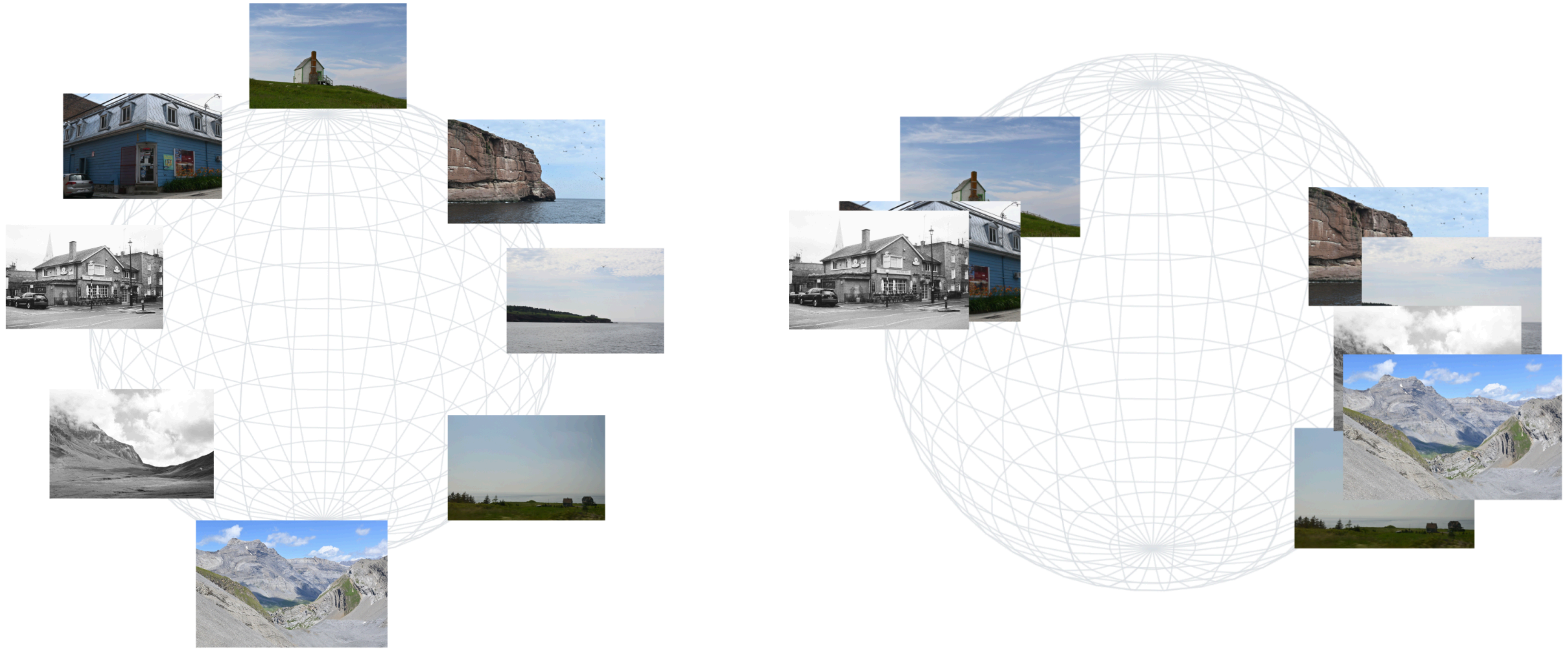
1. **Mitigating dimensional collapse**: encourage the embeddings to span the entire space
2. **Promoting sample uniformity**: ensure the embeddings are evenly distributed across the representation space

[1] Rethinking uniformity in self-supervised learning, Fang et al, ICLR 2024

Joint-Embedding Self-Supervised Learning

Fang et al. [1] formalized desirable properties of JE-SSL:

1. **Mitigating dimensional collapse**: encourage the embeddings to span the entire space
2. **Promoting sample uniformity**: ensure the embeddings are evenly distributed across the representation space



[1] Rethinking uniformity in self-supervised learning, Fang et al, ICLR 2024

Joint-Embedding Self-Supervised Learning

Fang et al. [1] formalized desirable properties of JE-SSL:

1. **Mitigating dimensional collapse**: encourage the embeddings to span the entire space
2. **Promoting sample uniformity**: ensure the embeddings are evenly distributed across the representation space

How? Maximizing the length of the Minimum Spanning Tree

[1] Rethinking uniformity in self-supervised learning, Fang et al, ICLR 2024

Minimum spanning tree

Definition:
Given a point cloud Z in a Euclidean space, a **spanning tree** (ST) of Z is an undirected graph $G = (V, E)$ with the vertex set $V = Z$ and edge set $E \subset V \times V$ such that G is connected without cycles.

Minimum spanning tree

Definition:

Given a point cloud Z in a Euclidean space, a **spanning tree** (ST) of Z is an undirected graph $G = (V, E)$ with the vertex set $V = Z$ and edge set $E \subset V \times V$ such that G is connected without cycles.

The length of G is:

$$E(G) := \sum_{(z,z') \in E} ||z - z'||_2$$

A **MST(Z)**, is an ST of Z that minimises length E.

Minimum spanning tree

Definition:
Given a point cloud Z in a Euclidean space, a **spanning tree** (ST) of Z is an undirected graph $G = (V, E)$ with the vertex set $V = Z$ and edge set $E \subset V \times V$ such that G is connected without cycles.

The length of G is:

$$E(G) := \sum_{(z,z') \in E} ||z - z'||_2$$

A **MST(Z)**, is an ST of Z that minimises length E.

The MST also relates to the persistence in degree 0 of the Rips filtration.

Minimum spanning tree

<u>**Definition**</u>:
Given a point cloud Z in a Euclidean space, a **spanning tree** (ST) of Z is an undirected graph $G=(V,E)$ with the vertex set $V=Z$ and edge set $E \subset V \times V$ such that G is connected without cycles.

The length of G is:

$$E(G) := \sum_{(z,z') \in E} ||z - z'||_2$$

A **MST(Z)**, is an ST of Z that minimises length E.

The MST also relates to the persistence in degree 0 of the Rips filtration.
Specifically, there is a bijection between the edges of the minimal spanning tree of a finite metric space $x = \{x_1, \ldots, x_n\}$ and the points in the persistence diagram $\mathbf{PH}_0(x)$ obtained from the Rips filtration.

Minimum spanning tree

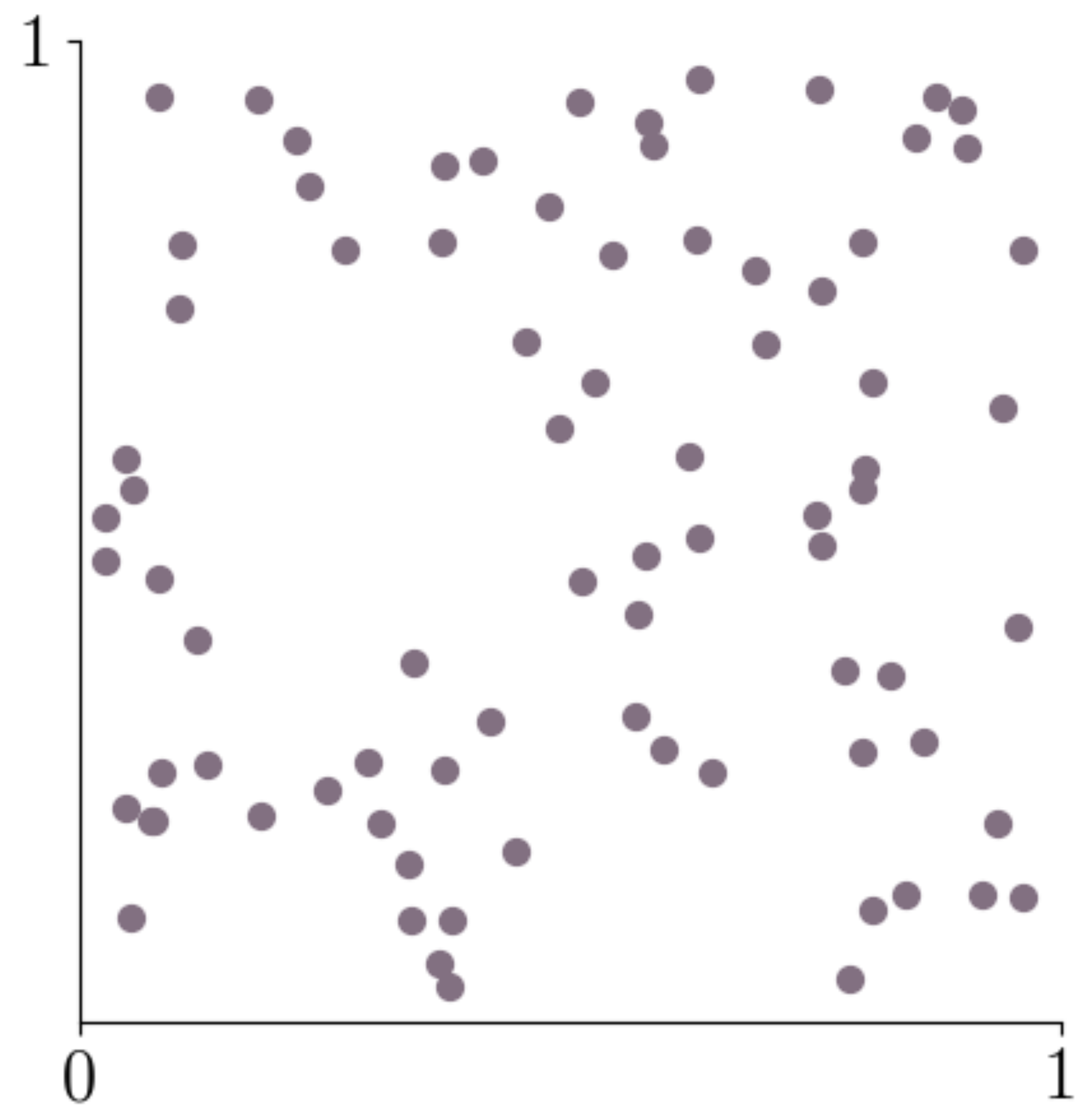


Figure: **Example of MST in 2-d**

Minimum spanning tree

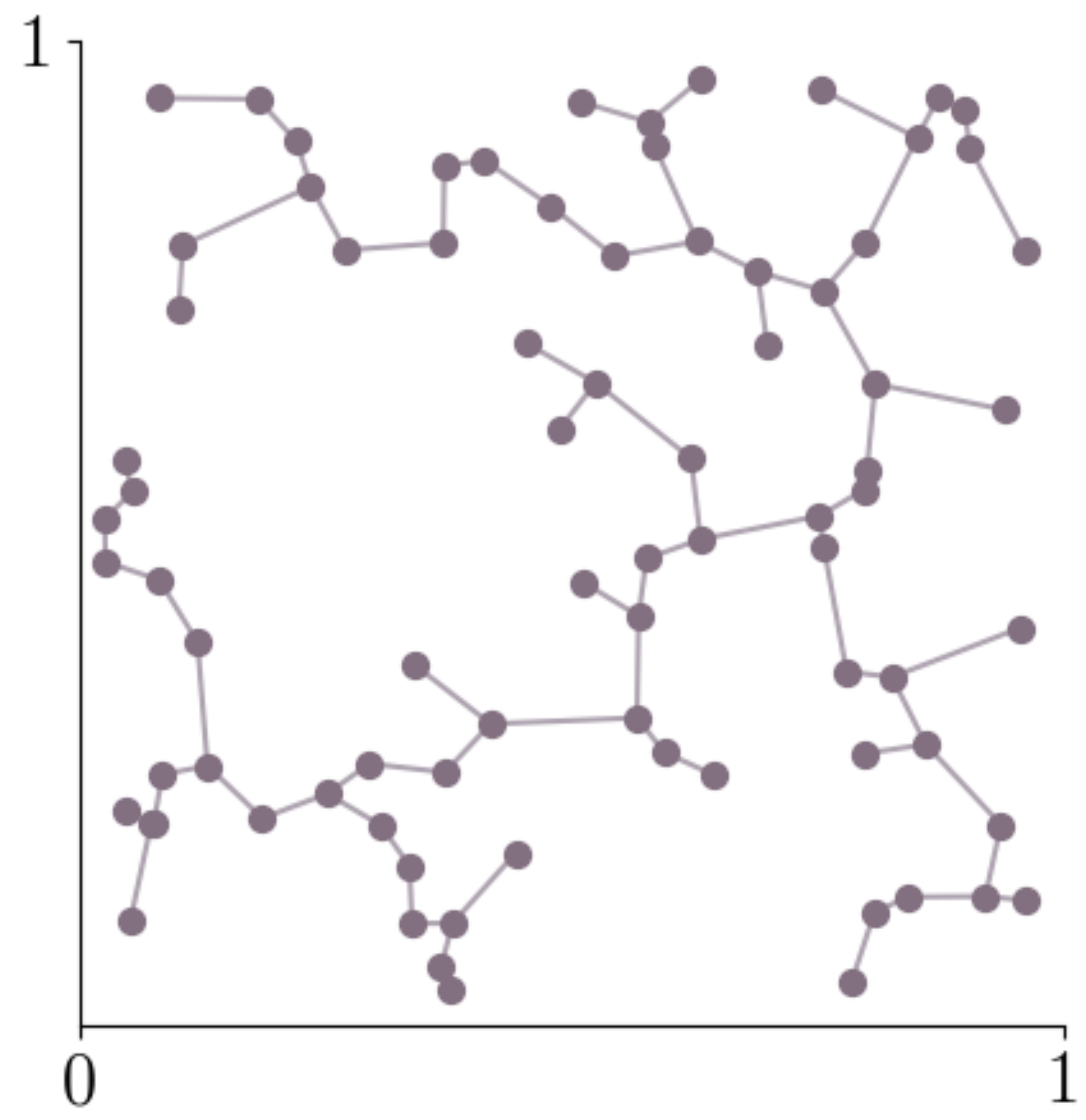


Figure: **Example of MST in 2-d**

2. Background

Minimum spanning tree and dimension estimation

Steele [1] studied the length of the minimum spanning tree of random Euclidean spaces. Let E_n be an i.i.d $n-$sample drawn from a probability measure P_X with compact support on R^d. For $d \geq 2$, Theorem 1 [1] controls the growth rate of the length of the MST(X_n) as follows:

$$E(MST(X_n) \sim Cn^{(d-1)/d} \text{ almost surely, as } n \to \infty$$

The asymptotic rate allows to derive several estimators of intrinsic dimension

[1] Growth rates of euclidean minimal spanning trees with power weighted edges, Steele, 1988

Minimum spanning tree and persistence

Persistence provides a mathematical framework to optimize the length of the MST [1]:

$$\forall x \in X, \quad \nabla_x E(MST(X)) = \sum_{\substack{(x,z) \text{ edge} \\ \text{of } MST(X)}} \nabla_x \| x - z \|_2 = \sum_{\substack{(x,z) \text{ edge} \\ \text{of } MST(X)}} \| x - z \|_2^{-1} (x - z)\,.$$

Each pair of points forming an edge in the MST exerts a repulsive force on the other during optimization.

[1] Optimising persistence homology-based functions, Carrière et al, ICML 2021

3. T-REG: Minimum Spanning Tree based regularization

T-REG

Given, $Z = \{z_0, \ldots, z_n\} \in \mathbb{R}^d$, the MST length is defined as:

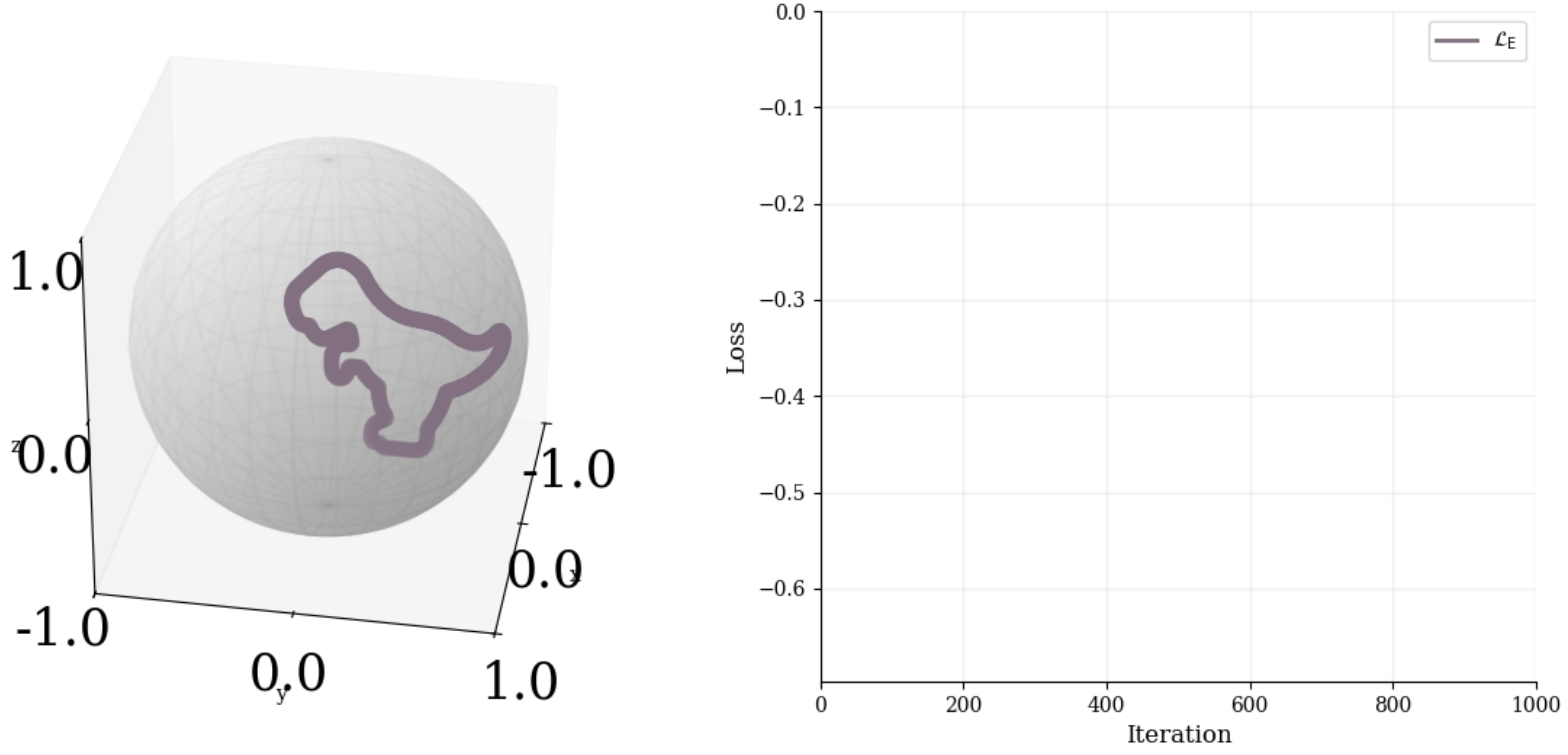
$$\mathscr{L}_{\mathbf{E}} = -\frac{1}{n}E(\mathbf{MST}(\mathrm{Z}))$$

T-REG

Given, $Z = \{z_0, \ldots, z_n\} \in \mathbb{R}^d$, the MST length is defined as:

$$\mathscr{L}_{\mathbf{E}} = -\frac{1}{n} E(\mathbf{MST}(\mathrm{Z}))$$

Point Cloud optimization with $\mathcal{L}_{\mathrm{E}}$



T-REG

Given, $Z = \{z_0, \ldots, z_n\} \in \mathbb{R}^d$, the MST length is defined as:

$$\mathscr{L}_{\mathbf{E}} = -\frac{1}{n} E(\mathbf{MST}(\mathrm{Z}))$$

The soft-sphere constraint is given by:

$$\mathscr{L}_{\mathbf{S}} = \frac{1}{n} \sum_i (||z_i||_2 - 1)^2$$

T-REG

Given, $Z = \{z_0, \ldots, z_n\} \in \mathbb{R}^d$, the MST length is defined as:

$$\mathscr{L}_{\mathbf{E}} = -\frac{1}{n}E(\mathbf{MST}(\mathrm{Z}))$$

The soft-sphere constraint is given by:

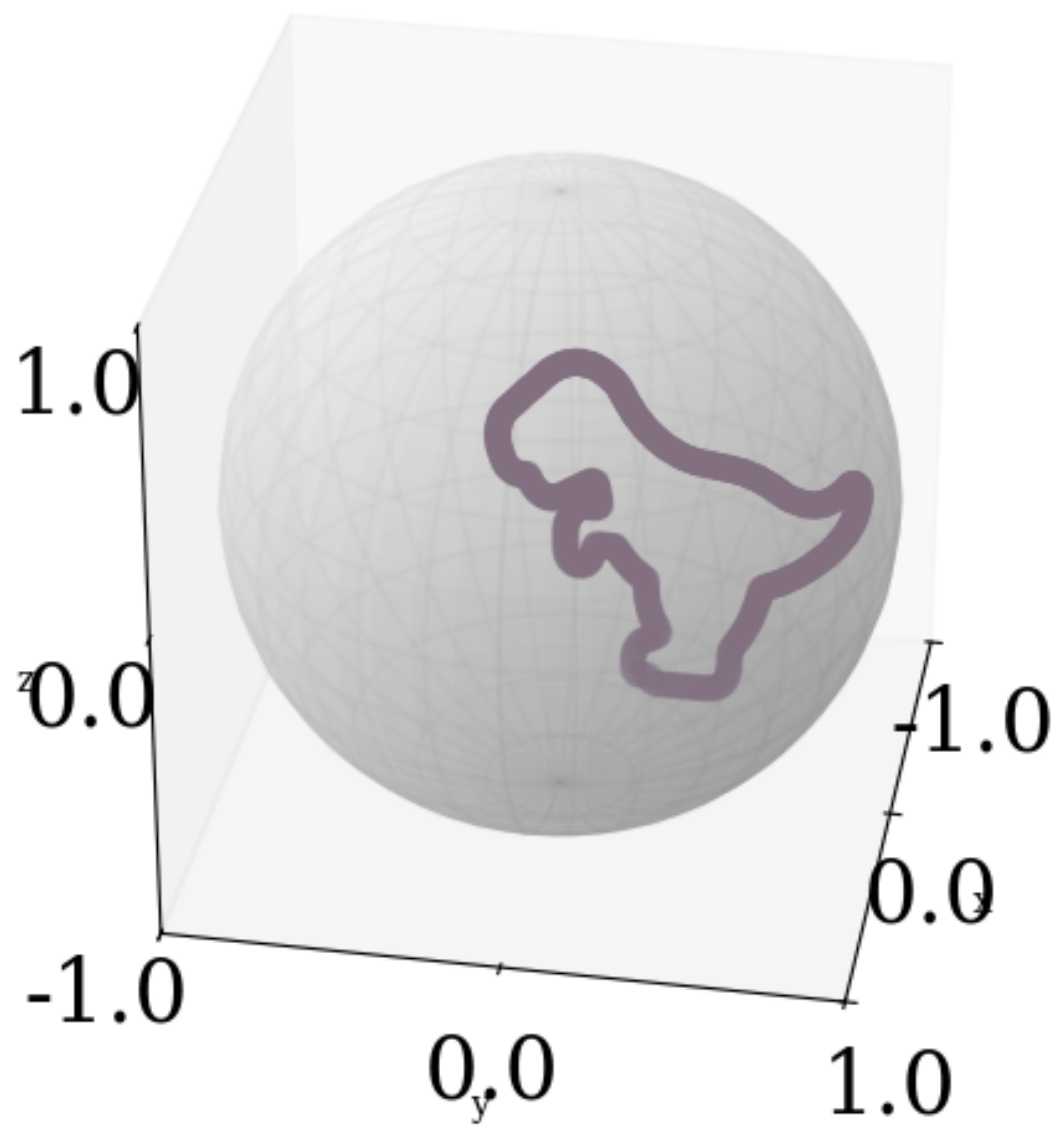
$$\mathscr{L}_{\mathbf{S}} = \frac{1}{n}\sum_i (||z_i||_2 - 1)^2$$

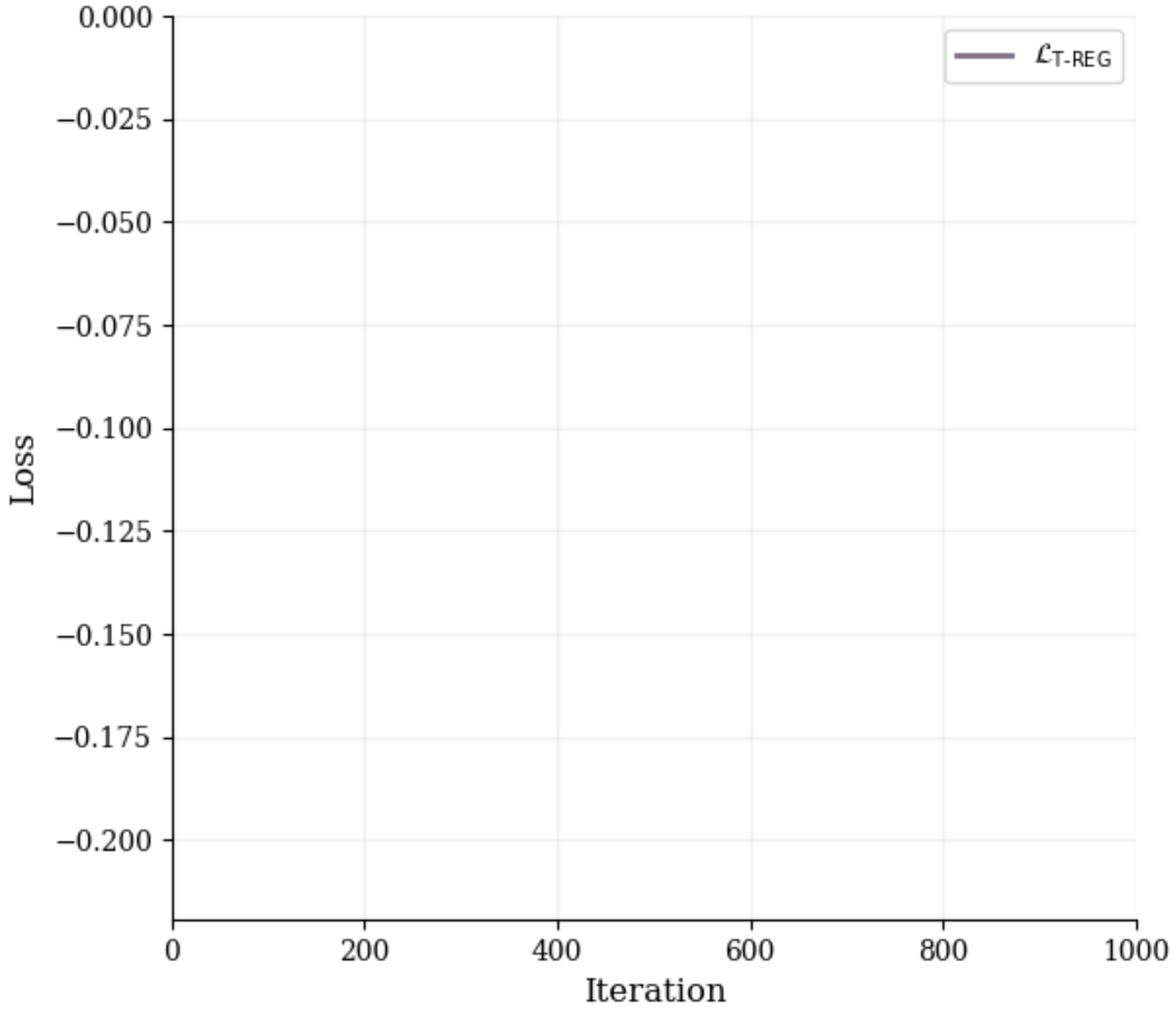
The overall loss is defined as: $\mathscr{L}_{\text{T-REG}} = \gamma\mathscr{L}_{\text{E}} + \lambda\mathscr{L}_{\text{S}}$
(with γ, λ are hyper-parameters)

T-REG

The overall loss is defined as: $\mathscr{L}_{\text{T-REG}} = \gamma\,\mathscr{L}_{\text{E}} + \lambda\,\mathscr{L}_{\text{S}}$

Point Cloud optimization with $\mathcal{L}_{\text{T-REG}}$





3. T-REG: Minimum Spanning Tree based regularization

Theoretical analysis

Theoretical analysis

Asymptotic behavior on large samples $(n > d + 1)$:

Derived from Steele [1], let E_n be an i.i.d $n-$sample drawn from a probability measure P_X with compact support on R^d.
For $d \geq 2$, Theorem 1 [1] controls the growth rate of the length of the MST(X_n) as follows:

$$E(MST(X_n) \sim Cn^{(d-1)/d} \text{ almost surely, as } n \to \infty$$

[1] Growth rates of euclidean minimal spanning trees with power weighted edges, Steele, 1988

Theoretical analysis

Asymptotic behavior on large samples $(n > d + 1)$:

We fix a compact Riemannian d-manifold, $\mathscr{M}$, equipped with the d-dimensional Hausdorff measure μ.

Theorem 4.4 [1]: Let X_n be an iid n-sample of a probability measure on $\mathscr{M}$ with density f_X w.r.t. μ. Then, there exists a constant independent of f_X and of $\mathscr{M}$ such that:

$$n^{(d-1)/d} \cdot E(MST(X_n)) \underset{n\to\infty}{\longrightarrow} C' \int f^{\frac{d-1}{d}} \mathrm{d}\mu$$

[1] Determining Intrinsic Dimension and Entropy of High-Dimensional Shape Spaces. Costa et al, 2006
[2] Shannon entropy, Renyi entropy, and information. Bromiley et al. 2004
[3] Maximum entropy autoregressive conditional heteroskedasticity model. Park et al, 2009

Theoretical analysis

Asymptotic behavior on large samples $(n > d + 1)$:

We fix a compact Riemannian d-manifold, $\mathscr{M}$, equipped with the d-dimensional Hausdorff measure μ.

Theorem 4.4 [1]: Let X_n be an iid n-sample of a probability measure on $\mathscr{M}$ with density f_X w.r.t. μ. Then, there exists a constant independent of f_X and of $\mathscr{M}$ such that:

$$n^{(d-1)/d} \cdot E(MST(X_n)) \xrightarrow[n\to\infty]{} C' \int f^{\frac{d-1}{d}} \mathrm{d}\mu$$

The limit is related to the intrinsic Rényi $\frac{d}{d-1}$-entropy which is known to converge to the Shannon entropy as $\frac{d-1}{d} \to 1$[2]. The Shannon entropy, in turn, achieves its **maximum at the uniform distribution** on compact sets [3].

[1] Determining Intrinsic Dimension and Entropy of High-Dimensional Shape Spaces. Costa et Hero, 2006
[2] Shannon entropy, Renyi entropy, and information. Bromiley et al. 2004
[3] Maximum entropy autoregressive conditional heteroskedasticity model. Park et al, 2009

Theoretical analysis

Behaviour on small samples ($n \leq d+1$, e.g. batch sizes are often smaller than or comparable to the ambient dimension):

Theoretical analysis

Behaviour on small samples ($n \leq d + 1$, e.g. batch sizes are often smaller than or comparable to the ambient dimension):
To account for the effect of the soft sphere constraint, we assume the points of X lie inside some fixed closed Euclidean d-ball B of radius r centered at the origin

Theorem 4.1:

Under the above conditions, the maximum of the $E(MST(Z))$, over the point sets $X \subset B$ of fixed cardinality n is attained when the points of X lie on the sphere $S = \partial B$, at the vertices of a regular $(n-1)$-simplex that has S as its smallest circumscribing sphere.

Theoretical analysis

Behaviour on small samples ($n \leq d + 1$, e.g. batch sizes are often smaller than or comparable to the ambient dimension):
In order to account for the effect of the soft sphere constraint, we assume the points of X lie inside some fixed closed Euclidean d-ball B of radius r centered at the origin

Theorem 4.1:

Under the above conditions, the maximum of the $E(MST(Z))$, over the point sets $X \subset B$ of fixed cardinality n is attained when the points of X lie on the sphere $S = \partial B$, at the vertices of a regular $(n-1)$-simplex that has S as its smallest circumscribing sphere.

Behavior of T-REG:

- First, $\mathscr{L}_{\mathrm{E}}$ in $\mathscr{L}_{\text{T-REG}}$ expands the point cloud until the sphere constraint term $\mathscr{L}_{\mathrm{S}}$ becomes the dominating term

- Then, the points stop expanding and start spreading themselves out uniformly along the sphere of directions.

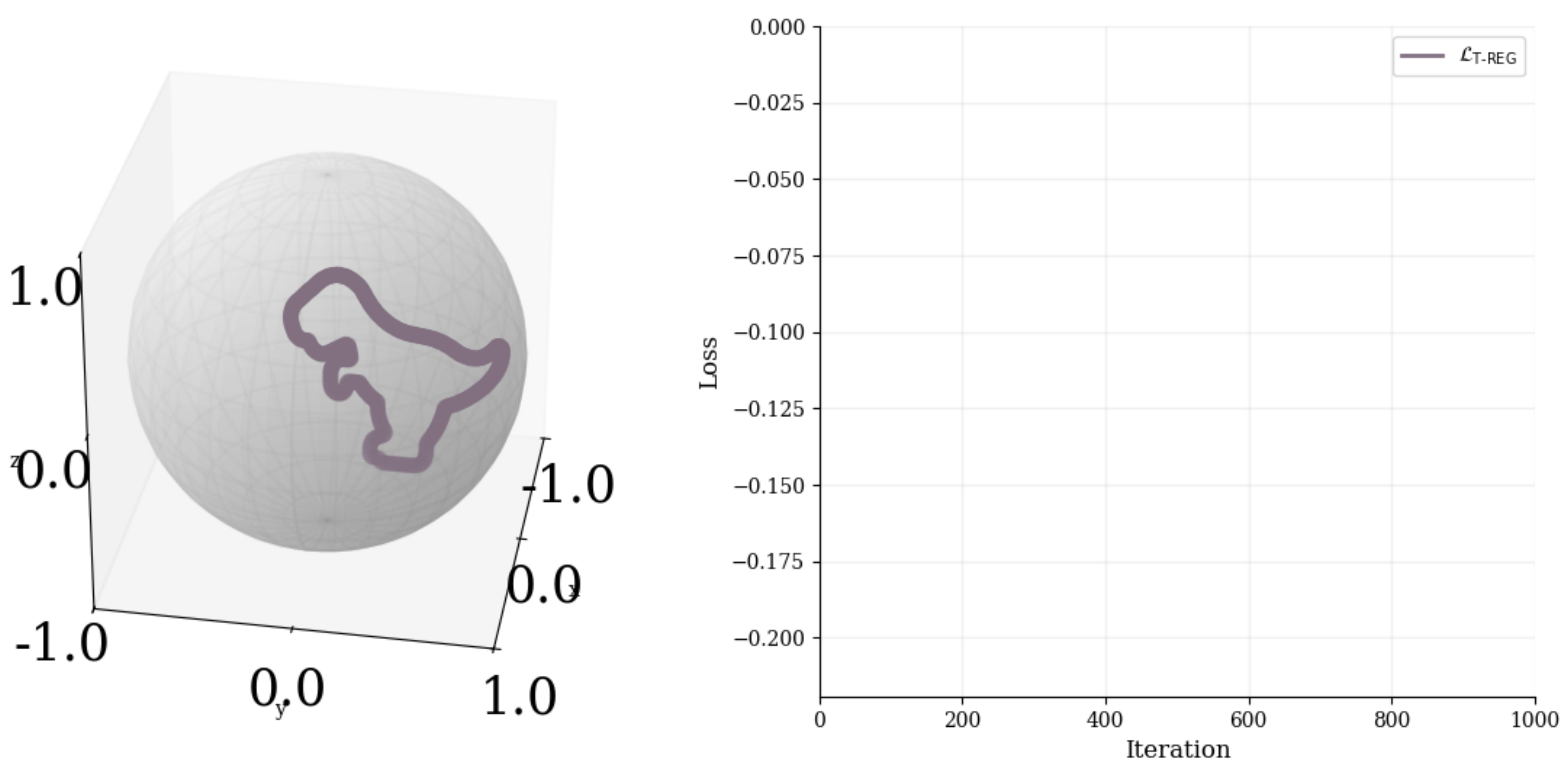
3. T-REG: Minimum Spanning Tree based regularization

Empirical analysis

Empirical analysis, promoting sample uniformity

Set-up: We apply T-REG alone to optimize the positions of a given point cloud, and we analyze its behavior when $n > d + 1$.

Point Cloud optimization with $\mathcal{L}_{\text{T-REG}}$

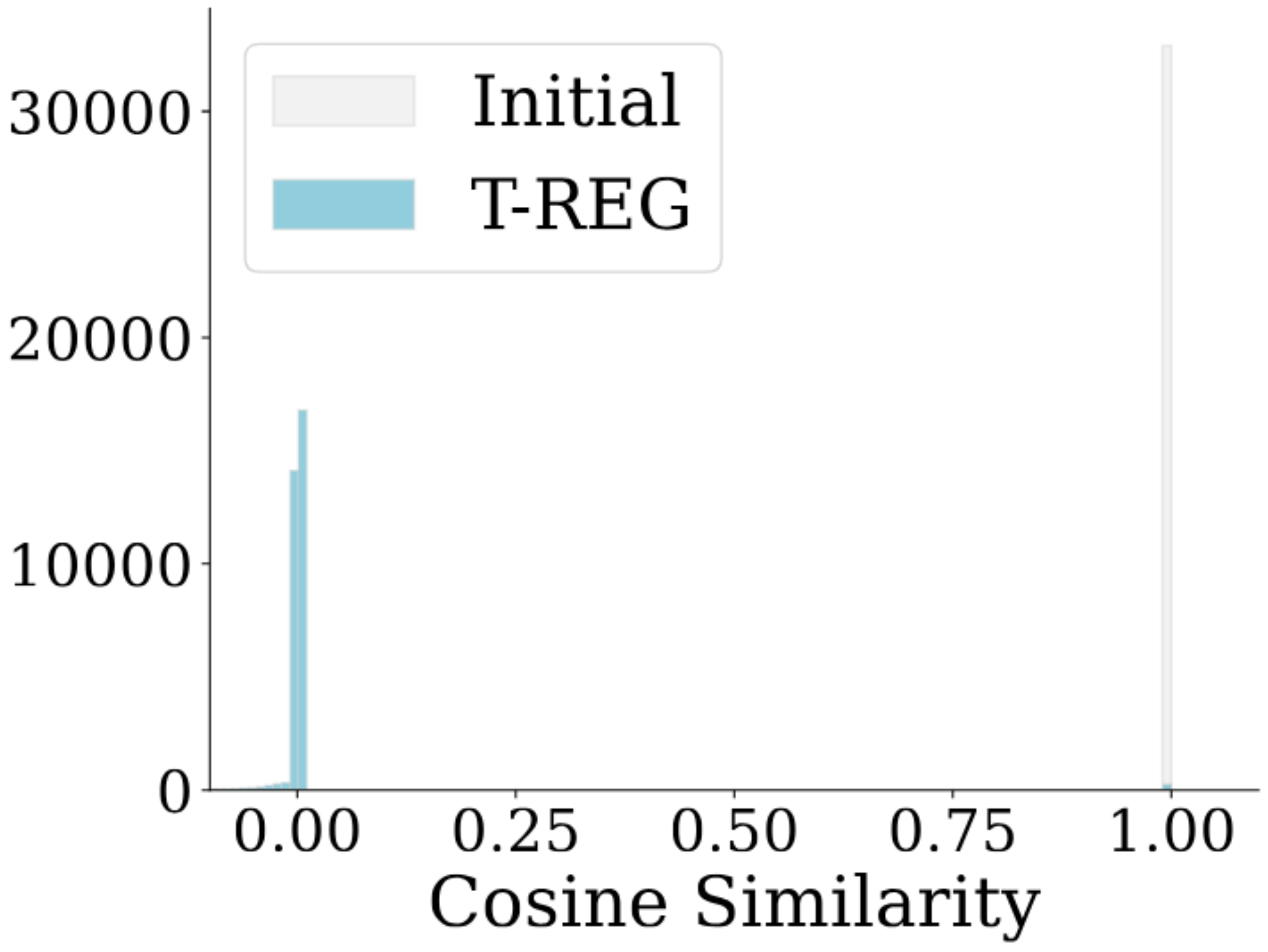


Empirical analysis, promoting sample uniformity

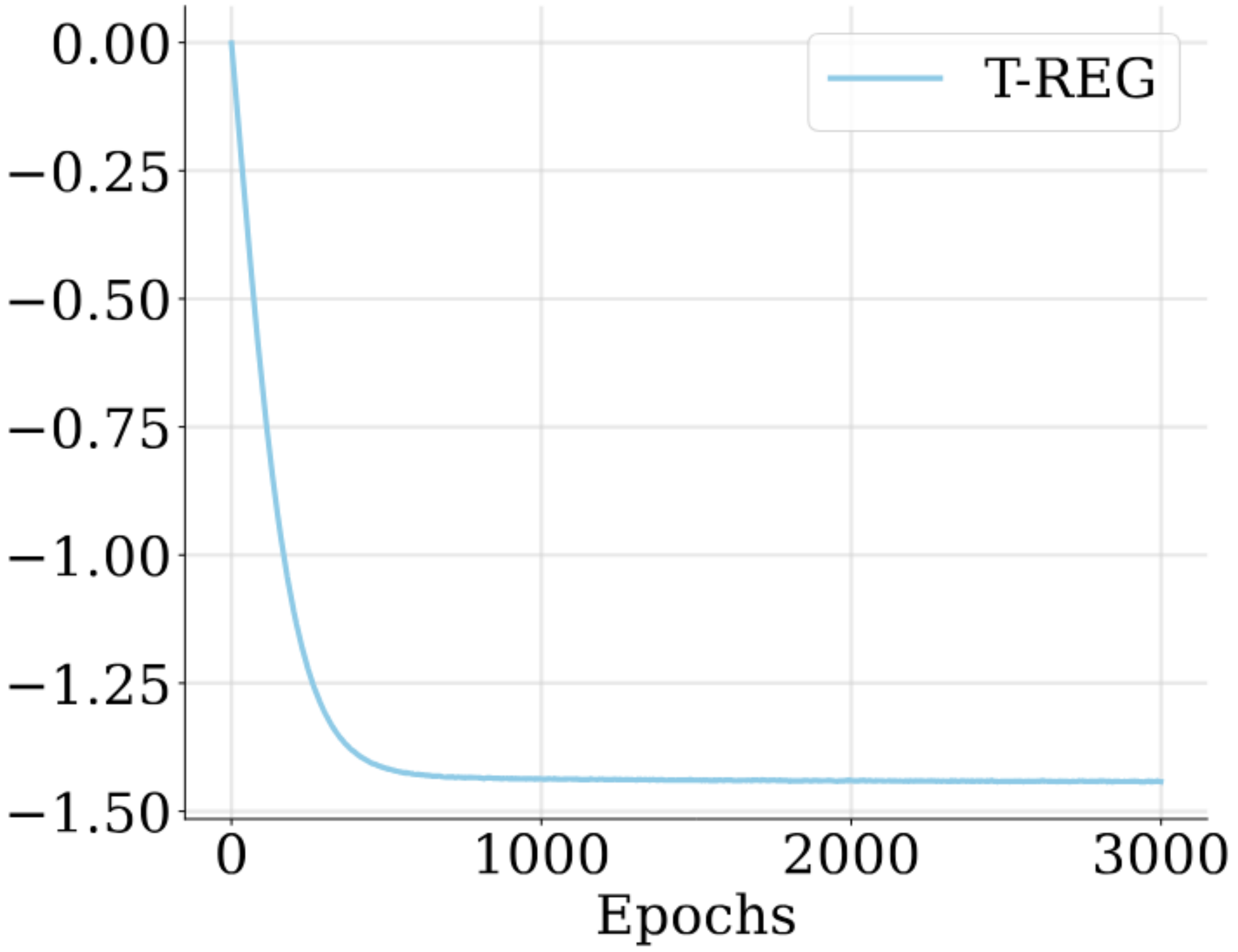
Set-up: We apply T-REG alone to optimize the positions of a given point cloud, and we analyze its behavior when $n \leq d + 1$ (specifically $n = 1024, d = 1024$).

Empirical analysis, promoting sample uniformity

Set-up: We apply T-REG alone to optimize the positions of a given point cloud, and we analyze its behavior when $n \leq d+1$ (specifically $n = 1024, d = 1024$).



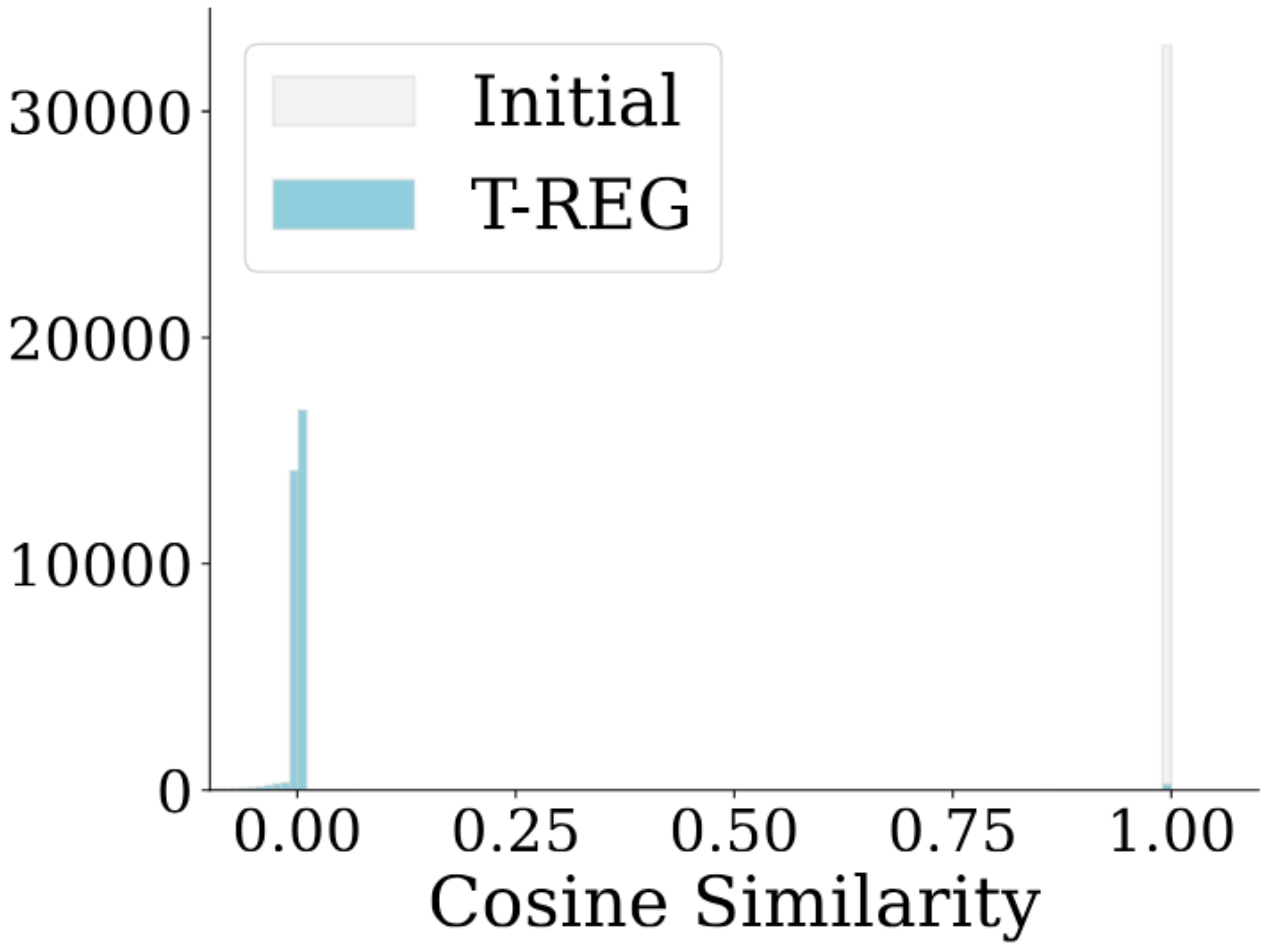
(d) Optimization w/ T-REG



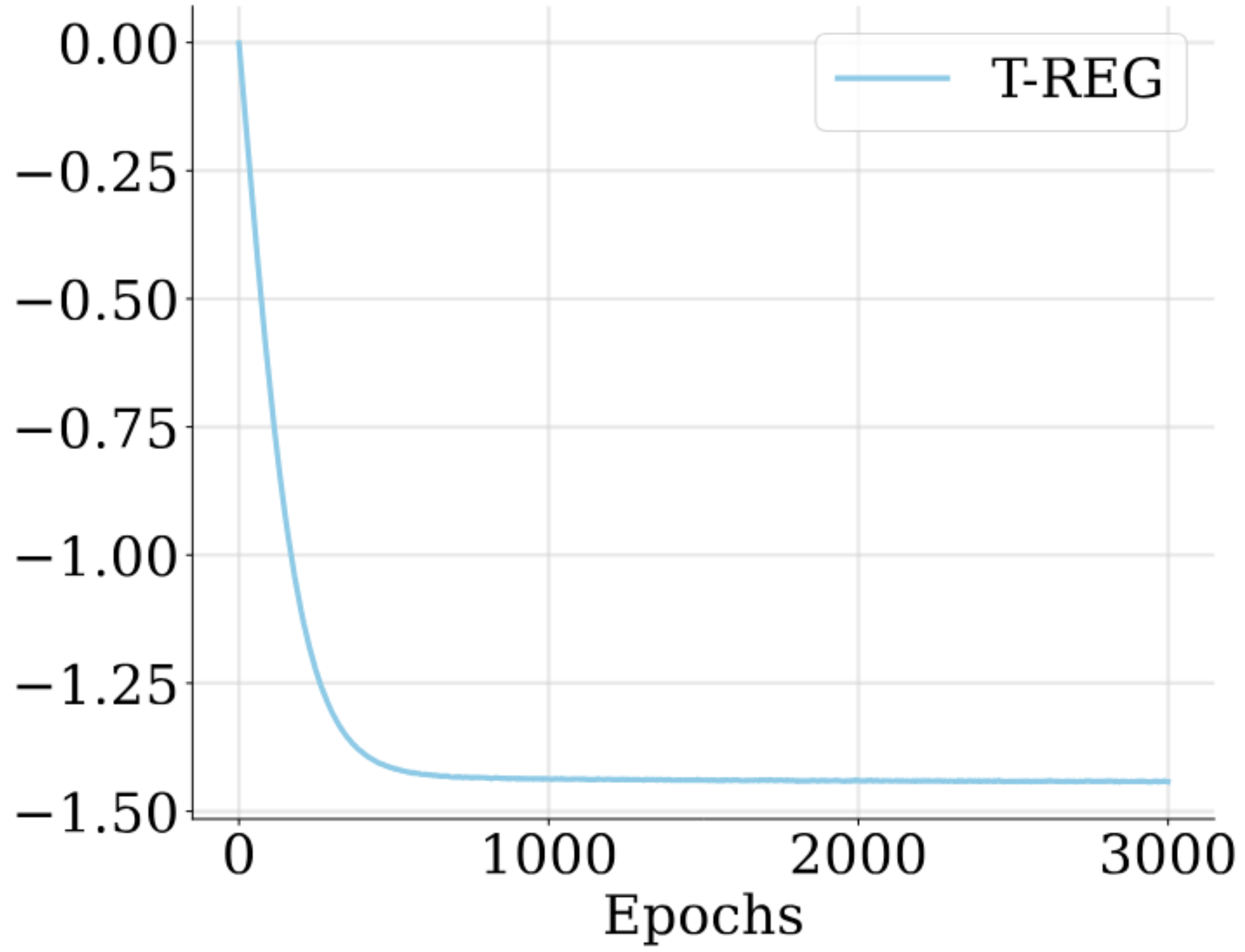
(e) Loss

Empirical analysis, promoting sample uniformity

Set-up: We apply T-REG alone to optimize the positions of a given point cloud, and we analyze its behavior when $n \leq d+1$ (specifically $n = 1024, d = 1024$).



(d) Optimization w/ T-REG

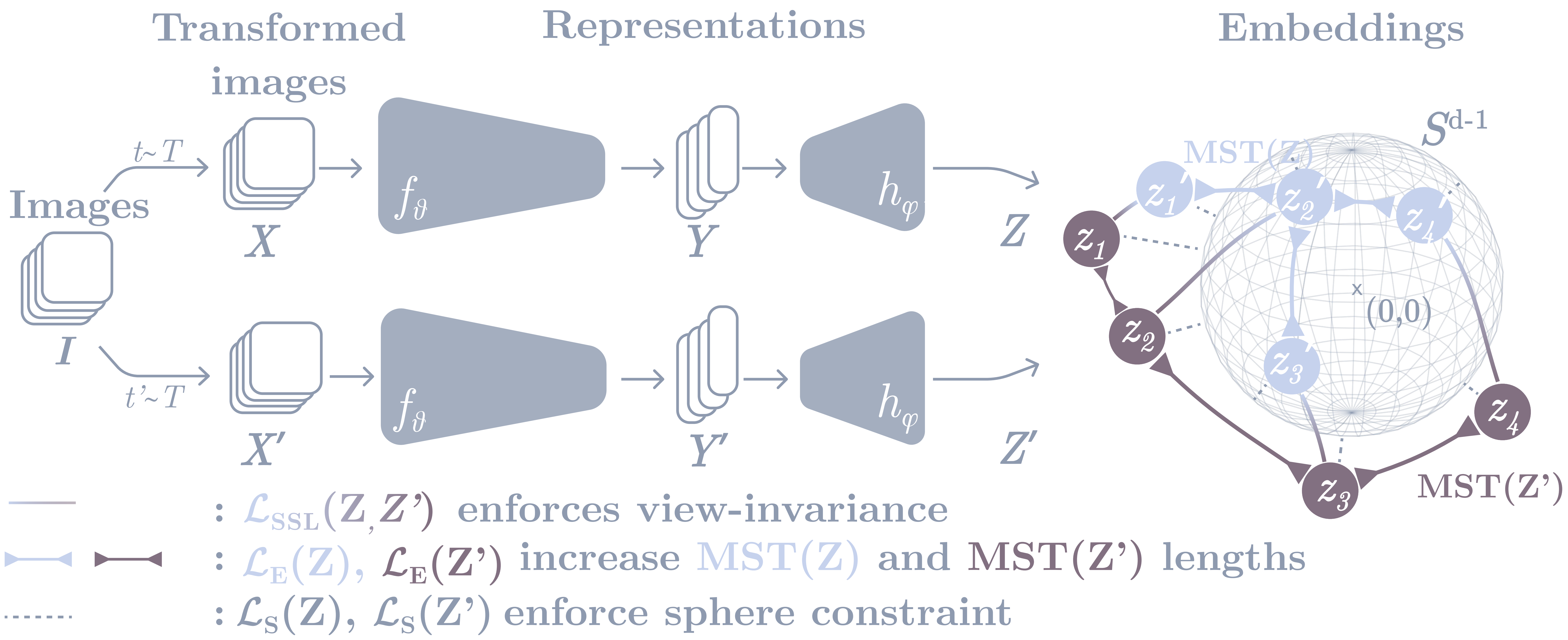


(e) Loss

After optimization with T-REG, the distribution becomes almost a Dirac slightly below 0, which indicates that the configuration of the points is close to that of the vertices of the regular simplex.

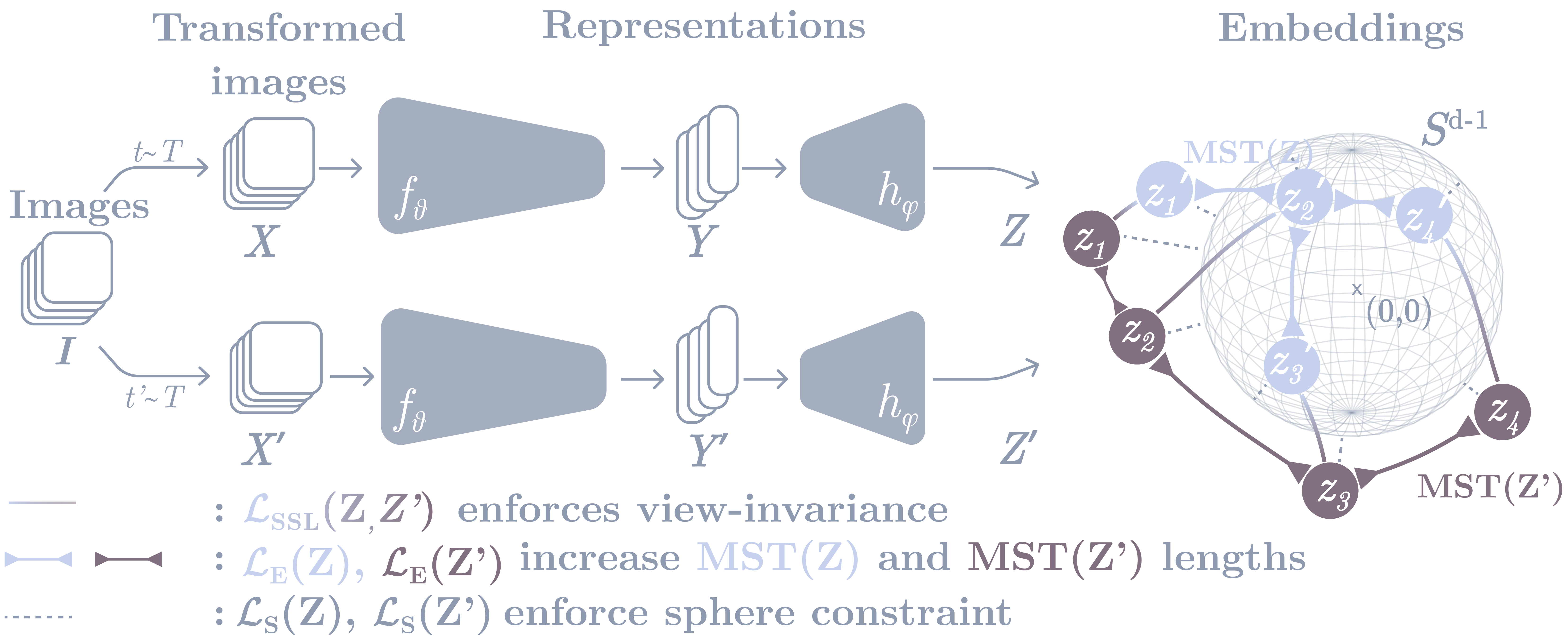
4. T-REGS: T-REG for Self-Supervised Learning

T-REGS



Transformed images
Representations
Embeddings
Images
I
$t \sim T$
$t' \sim T$
X
X'
f_ϑ
Y
Y'
h_φ
Z
Z'
S^{d-1}
MST(Z)
MST(Z')
(0,0)
z_1
z_2
z_3
z_4
z_1'
z_2'
z_3'
z_4'
: $\mathcal{L}_{\mathrm{SSL}}(Z, Z')$ enforces view-invariance
: $\mathcal{L}_{\mathrm{E}}(\mathrm{Z})$, $\mathcal{L}_{\mathrm{E}}(\mathrm{Z'})$ increase MST(Z) and MST(Z') lengths
: $\mathcal{L}_{\mathrm{S}}(\mathrm{Z})$, $\mathcal{L}_{\mathrm{S}}(\mathrm{Z'})$ enforce sphere constraint

T-REGS



$$\mathscr{L}(Z, Z') = \beta\,\mathscr{L}_{\mathrm{SSL}}(Z, Z') + \boxed{\gamma\,\mathscr{L}_{\mathrm{E}}(Z) + \lambda\,\mathscr{L}_{\mathrm{S}}(Z)} + \boxed{\gamma\,\mathscr{L}_{\mathrm{E}}(Z') + \lambda\,\mathscr{L}_{\mathrm{S}}(Z')}$$

T-REGS: T-REG for Self-supervised learning

Evaluation on Standard SSL benchmark

Evaluation on standard SSL Benchmark

Method		CIFAR-10 [37]	CIFAR-100 [37]
Zero-CL [67]		91.3	**68.5**
	+ $\mathcal{L}_u$	91.3	68.4
	+ $\mathcal{W}_2$	91.4	**68.5**
MoCo v2 [14]		90.7	60.3
	+ $\mathcal{L}_u$	91.0	61.2
	+ $\mathcal{W}_2$	91.4	63.7
BYOL [28]		89.5	63.7
	+ $\mathcal{L}_u$	90.1	62.7
	+ $\mathcal{W}_2$	90.1	65.2
	+ T-REGS	90.4	65.7
Barlow Twins [64]		91.2	68.2
	+ $\mathcal{L}_u$	91.4	68.4
	+ $\mathcal{W}_2$	91.4	**68.5**
	+ T-REGS	**91.8**	**68.5**
$\mathcal{L}_{\text{MSE}}$	+ T-REGS	91.3	67.4

Table 1: **Comparison with $\mathcal{W}_2$ regularization [23] on CIFAR-10/100.** The table is inherited from Fang et al. [23], and we follow the same protocol: ResNet-18 models are pre-trained for 500 epochs on CIFAR-10/100, with a batch size of 256, followed by linear probing. We report Top-1 accuracy (%), **bold** indicates best performance.

On CIFAR-10/100.

T-REGS demonstrate strong standalone performances.

T-REGS

Evaluation on standard SSL Benchmark

Method		CIFAR-10 [37]	CIFAR-100 [37]
Zero-CL [67]		91.3	**68.5**
	+ $\mathcal{L}_u$	91.3	68.4
	+ $\mathcal{W}_2$	91.4	**68.5**
MoCo v2 [14]		90.7	60.3
	+ $\mathcal{L}_u$	91.0	61.2
	+ $\mathcal{W}_2$	91.4	63.7
BYOL [28]		89.5	63.7
	+ $\mathcal{L}_u$	90.1	62.7
	+ $\mathcal{W}_2$	90.1	65.2
	+ T-REGS	90.4	65.7
Barlow Twins [64]		91.2	68.2
	+ $\mathcal{L}_u$	91.4	68.4
	+ $\mathcal{W}_2$	91.4	**68.5**
	+ T-REGS	**91.8**	**68.5**
$\mathcal{L}_{\text{MSE}}$	+ T-REGS	91.3	67.4

Table 1: **Comparison with $\mathcal{W}_2$ regularization [23] on CIFAR-10/100.** The table is inherited from Fang et al. [23], and we follow the same protocol: ResNet-18 models are pre-trained for 500 epochs on CIFAR-10/100, with a batch size of 256, followed by linear probing. We report Top-1 accuracy (%), **bold** indicates best performance.

On CIFAR-10/100.

Using T-REGS as an auxiliary loss consistently improves performance over the respective baselines, and over variants that use $\mathscr{L}_{\mathrm{u}}$ or $\mathscr{W}_2$ as additional regularization terms.

Evaluation on standard SSL Benchmark

# views	Method		Imagenet-100 [58] Top-1	ImageNet-1k [18] Batch Size	ImageNet-1k [18] Top-1
8	SwAV [9]		74.3	4096	66.5
	FroSSL [55]		79.8	-	-
	SSOLE [34]		**82.5**	256	**73.9**
2	SimCLR [12]		77.0	4096	66.5
	MoCo v2 [14]		79.3	256	67.4
	SimSiam [13]		78.7	256	68.1
	W-MSE [20]		69.1	512	65.1
	Zero-CL [67]		79.3	1024	**68.9**
	VICReg [4]		79.4	1024	68.3
	CW-RGP [61]		77.0	512	67.1
	INTL [62]		**81.7**	512	**69.5**
	BYOL [28]		80.3	1024	66.5
		+ T-REGS	**80.8**	1024	67.2
	Barlow Twins [64]		80.2	2048	67.7
		+ T-REGS	**80.9**	2048	67.8
	$\mathcal{L}_{\mathrm{MSE}}$	+ T-REGS	80.3	512	**68.8**

Table 2: **Linear Evaluation on ImageNet-100/1k.** We report Top-1 accuracy (%). Top-4 best methods are **bolded**. For ImageNet-100, ResNet-18 are pre-trained for 400 epochs using a batch size of 256; for ImageNet-1k, ResNet-50 are pre-trained for 100 epochs. The table is mostly inherited from Weng et al. [62].

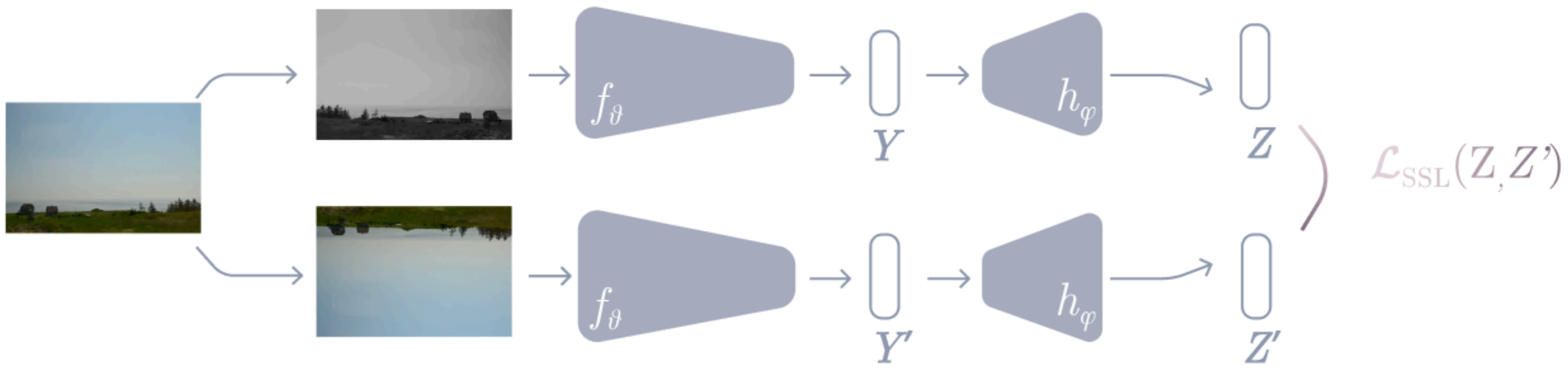
On ImageNet-100/1k.

T-REGS is competitive with method that use the same number of views.

T-REGS: T-REG for Self-supervised learning

Further applications

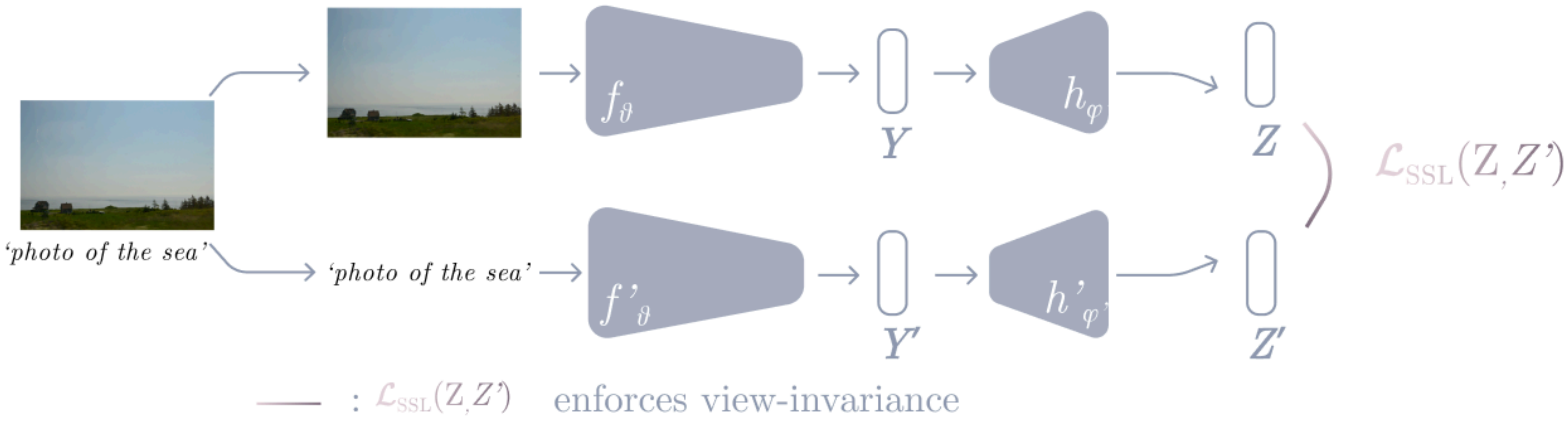
Further applications



: $\mathcal{L}_{\mathrm{SSL}}(Z, Z')$ enforces view-invariance

Further applications

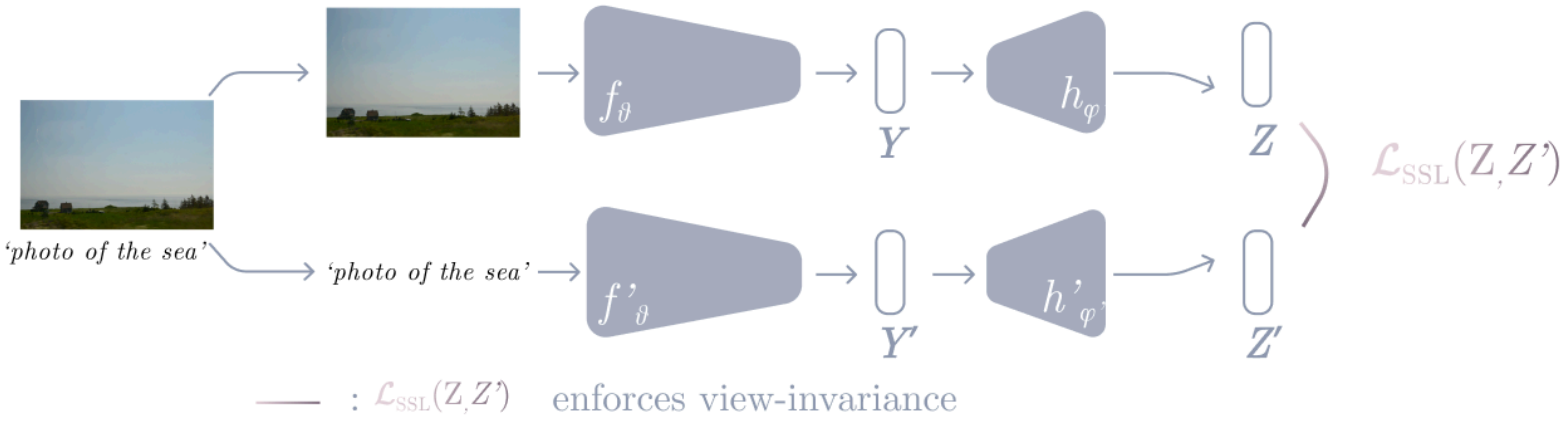
Joint-Embedding multimodal models [1]:



[1] Learning Transferable Visual Models From Natural Language Supervision, Radford et al, ICML 2021

Further applications

Joint-Embedding multimodal models [1]:



Such models preserves distinct subspaces for text and image: **the *modality gap*** [2]**.**

[1] Learning Transferable Visual Models From Natural Language Supervision, Radford et al, ICML 2021
[2] Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning, Liang et al, NeurIPS 2022

Further applications

We fine-tune CLIP using T-REGS as regularization to the standard CLIP loss to encourage more robust and uniformly distributed representations.

	Flickr30k [48]				MS-COCO [42]			
	i → t		t → i		t → i		t → i	
Method	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
Zero-Shot	71.1	90.4	68.5	88.9	31.9	56.9	28.5	53.1
Finetune	81.2	95.5	80.7	95.8	36.7	63.6	36.9	63.9
ES [40]	71.8	90.0	68.5	88.9	31.9	56.9	28.7	53.0
i-Mix [39]	72.3	91.7	69.0	91.1	34.0	63.0	34.6	62.2
Un-Mix [53]	78.5	95.4	74.1	91.8	38.8	66.2	33.4	61.0
m^3-Mix [44]	82.3	95.9	**82.7**	96.0	41.0	68.3	39.9	67.9
$\mathcal{L}_{\text{CLIP}}$ + T-REGS	**83.2**	**96.0**	80.8	**96.4**	**41.6**	**68.7**	**41.5**	**68.7**

Table 3: **Cross-Modal Retrieval after finetuning CLIP.** Image-to-text (i → t) and text-to-image (t → i) retrieval results (top 1/5 Recall: R@1, R@5). The table is mostly inherited from Oh et al. [44]. **Bold** indicates the best performance.

T-REGS: T-REG for Self-supervised learning

Analysis

Complexity

The MSTs are computed with Kruskal's algorithm, whose worst-case time is $\mathcal{O}(B^2D)$, with B the batch size and D the embedding dimension.
Empirically T-REGS matches the per-step wall-clock of VICReg and simCLR.

Complexity

The MSTs are computed with Kruskal's algorithm, whose worst-case time is $\mathcal{O}(B^2D)$, with B the batch size and D the embedding dimension.
Empirically T-REGS matches the per-step wall-clock of VICReg and simCLR.

Method	Complexity	B range	D range	Wall-clock time
SimCLR [12]	$\mathcal{O}(B^2 \cdot D)$	[2048-4096]	[256-1024]	0.22 ± 0.03
VICReg [4]	$\mathcal{O}(B \cdot D^2)$	[1024-4096]	[4096-8192]	0.23 ± 0.02
$\mathcal{L}_{\text{MSE}}$ + T-REGS	$\mathcal{O}(B^2(D \cdot \log B))$	[512-1024]	[512-2048]	0.20 ± 0.001

Table 5: **Complexity and computational cost.** Comparison between different methods is performed, with training on ImageNet-1k distributed across 4 Tesla H100 GPUs. The wall-clock time (sec/step) is averaged over 500 steps. B, D ranges are reported from Bardes et al. [4], Garrido et al. [25].

Analysis

$$\mathscr{L}(Z, Z') = \beta\, \mathscr{L}_{\mathrm{SSL}}(Z, Z') + \gamma\, \mathscr{L}_{\mathrm{E}}(Z) + \lambda\, \mathscr{L}_{\mathrm{S}}(Z) + \gamma\, \mathscr{L}_{\mathrm{E}}(Z') + \lambda\, \mathscr{L}_{\mathrm{S}}(Z')$$

T-REGS coefficients			**Scaling**		
β	γ	λ	$\frac{\beta}{\gamma}$	$\frac{\gamma}{\lambda}$	**Top-1**
1	-	-	-	-	collapse
1	1	-	1	-	collapse
10	1	1	10	1	collapse
10	0.5	5e-2	20	10	25.7
10	0.2	2e-2	50	10	45.4
10	0.5	2.5e-3	20	200	65.0
10	0.2	1e-3	50	200	65.3
10	0.5	2e-3	20	250	64.9
10	0.2	8e-4	50	250	**66.1**
10	0.02	8e-5	100	300	63.3

Table 4: **Impact of coefficients.** $\mathcal{L}_{\mathrm{MSE}}$ + T-REGS top-1 accuracy (%) on ImageNet-1k with online evaluation protocol over 50 epochs. **Bold** indicates best performance.

Thank you!