Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere

Tongzhou Wang [1] **Phillip Isola** [1]

Abstract

Contrastive representation learning has been outstandingly successful in practice. In this work, we identify two key properties related to the contrastive loss: (1) *alignment* (closeness) of features from positive pairs, and (2) *uniformity* of the induced distribution of the (normalized) features on the hypersphere. We prove that, asymptotically, the contrastive loss optimizes these properties, and analyze their positive effects on downstream tasks. Empirically, we introduce an optimizable metric to quantify each property. Extensive experiments on standard vision and language datasets confirm the strong agreement between *both* metrics and downstream task performance. Directly optimizing for these two metrics leads to representations with comparable or better performance at downstream tasks than contrastive learning.

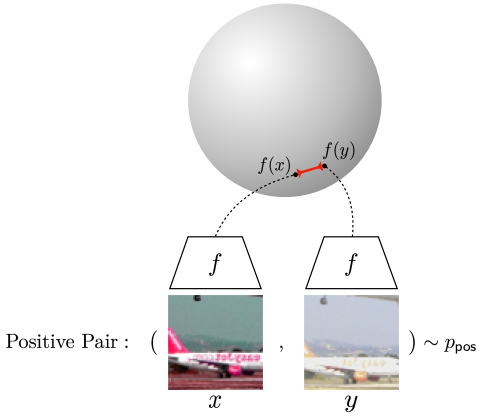
Project Page: ssnl.github.io/hypersphere.
Code: github.com/SsnL/align_uniform.
github.com/SsnL/moco_align_uniform.

1. Introduction

A vast number of recent empirical works learn representations with a unit ℓ_2 norm constraint, effectively restricting the output space to the unit hypersphere (Parkhi et al., 2015; Schroff et al., 2015; Liu et al., 2017; Hasnat et al., 2017; Wang et al., 2017; Bojanowski & Joulin, 2017; Mettes et al., 2019; Hou et al., 2019; Davidson et al., 2018; Xu & Durrett, 2018), including many unsupervised contrastive representation learning methods (Wu et al., 2018; Bachman et al., 2019; Tian et al., 2019; He et al., 2019; Chen et al., 2020a).

Intuitively, having the features live on the unit hypersphere leads to several desirable traits. Fixed-norm vectors are known to improve training stability in modern machine learning where dot products are ubiquitous (Xu & Durrett, 2018; Wang et al., 2017). Moreover, if features of a class are sufficiently well clustered, they are linearly separable with the rest of feature space (see Figure 2), a common criterion used to evaluate representation quality.

While the unit hypersphere is a popular choice of feature space, not all encoders that map onto it are created equal. Recent works argue that representations should additionally be invariant to unnecessary details, and preserve as much information as possible (Oord et al., 2018; Tian et al., 2019; Hjelm et al., 2018; Bachman et al., 2019). Let us call these two properties *alignment* and *uniformity* (see Figure 1). *Alignment* favors encoders that assign similar



Alignment: Similar samples have similar features.
(Figure inspired by Tian et al. (2019).)

Uniformity: Preserve maximal information.

Figure 1: Illustration of alignment and uniformity of feature distributions on the output unit hypersphere. STL-10 (Coates et al., 2011) images are used for demonstration.

[1]MIT Computer Science & Artificial Intelligence Lab (CSAIL). Correspondence to: Tongzhou Wang <tongzhou@mit.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

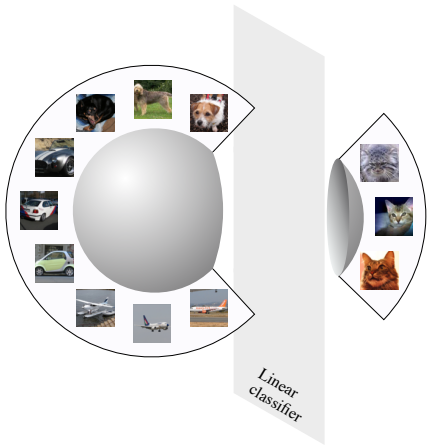


Figure 2: **Hypersphere:** When classes are well-clustered (forming spherical caps), they are linearly separable. The same does not hold for Euclidean spaces.

features to similar samples. *Uniformity* prefers a feature distribution that preserves maximal information, i.e., the uniform distribution on the unit hypersphere.

In this work, we analyze the *alignment* and *uniformity* properties. We show that a currently popular form of contrastive representation learning in fact directly optimizes for these two properties in the limit of infinite negative samples. We propose theoretically-motivated metrics for alignment and uniformity, and observe strong agreement between them and downstream task performance. Remarkably, directly optimizing for these two metrics leads to comparable or better performance than contrastive learning.

Our main contributions are:

- We propose quantifiable metrics for *alignment* and *uniformity* as two measures of representation quality, with theoretical motivations.
- We prove that the contrastive loss optimizes for alignment and uniformity asymptotically.
- Empirically, we find strong agreement between *both* metrics and downstream task performance.
- Despite being simple in form, our proposed metrics, when directly optimized with no other loss, empirically lead to comparable or better performance at downstream tasks than contrastive learning.

2. Related Work

Unsupervised Contrastive Representation Learning has seen remarkable success in learning representations for image and sequential data (Logeswaran & Lee, 2018; Wu et al., 2018; Oord et al., 2018; Hénaff et al., 2019; Tian et al., 2019; Hjelm et al., 2018; Bachman et al., 2019; Tian et al., 2019; He et al., 2019; Chen et al., 2020a). The common motivation behind these work is the InfoMax principle (Linsker, 1988), which we here instantiate as maximizing the mutual information (MI) between two views (Tian et al., 2019; Bachman et al., 2019; Wu et al., 2020). However, this interpretation is known to be inconsistent with the actual behavior in practice, e.g., optimizing a tighter bound on MI can lead to worse representations (Tschannen et al., 2019). What the contrastive loss exactly does remains largely a mystery. Analysis based on the assumption of latent classes provides nice theoretical insights (Saunshi et al., 2019), but unfortunately has a rather large gap with empirical practices: the result that representation quality suffers with a large number of negatives is inconsistent with empirical observations (Wu et al., 2018; Tian et al., 2019; He et al., 2019; Chen et al., 2020a). In this paper, we analyze and characterize the behavior of contrastive learning from the perspective of alignment and uniformity properties, and empirically verify our claims with standard representation learning tasks.

Representation learning on the unit hypersphere. Outside contrastive learning, many other representation learning approaches also normalize their features to be on the unit hypersphere. In variational autoencoders, the hyperspherical latent space has been shown to perform better than the Euclidean space (Xu & Durrett, 2018; Davidson et al., 2018). Directly matching uniformly sampled points on the unit hypersphere is known to provide good representations (Bojanowski & Joulin, 2017), agreeing with our intuition that uniformity is a desirable property. Mettes et al. (2019) optimizes prototype representations on the unit hypersphere for classification. Hyperspherical face embeddings greatly outperform the unnormalized counterparts (Parkhi et al., 2015; Liu et al., 2017; Wang et al., 2017; Schroff et al., 2015). Its empirical success suggests that the unit hypersphere is indeed a nice feature space. In this work, we formally investigate the interplay between the hypersphere geometry and the popular contrastive representation learning.

Distributing points on the unit hypersphere. The problem of uniformly distributing points on the unit hypersphere is a well-studied one. It is often defined as minimizing the total pairwise potential w.r.t. a certain kernel function (Borodachov et al., 2019; Landkof, 1972), e.g., the Thomson problem of finding the minimal electrostatic potential energy configuration of electrons (Thomson, 1904), and minimization of the Riesz s-potential (Götz & Saff, 2001; Hardin & Saff, 2005; Liu et al., 2018). The uniformity metric we propose is based on the Gaussian potential, which can be used to represent a very general class of kernels and is closely related to the universally optimal point configurations (Borodachov et al., 2019; Cohn & Kumar, 2007). Additionally, the best-packing problem on hyperspheres (often called the Tammes problem) is also well studied (Tammes, 1930).

3. Preliminaries on Unsupervised Contrastive Representation Learning

The popular unsupervised contrastive representation learning method (often referred to as *contrastive learning* in this paper) learns representations from unlabeled data. It assumes a way to sample *positive pairs*, representing similar samples that should have similar representations. Empirically, the positive pairs are often obtained by taking two independently randomly augmented versions of the same sample, e.g. two crops of the same image (Wu et al., 2018; Hjelm et al., 2018; Bachman et al., 2019; He et al., 2019; Chen et al., 2020a).

Let $p_{\mathsf{data}}(\cdot)$ be the data distribution over $\mathbb{R}^n$ and $p_{\mathsf{pos}}(\cdot,\cdot)$ the distribution of positive pairs over $\mathbb{R}^n \times \mathbb{R}^n$. Based on empirical practices, we assume the following property.

Assumption. Distributions p_{data} and p_{pos} should satisfy

- Symmetry: $\forall x, y,\ p_{\mathsf{pos}}(x, y) = p_{\mathsf{pos}}(y, x)$.
- Matching marginal: $\forall x,\ \int p_{\mathsf{pos}}(x, y)\,\mathrm{d}y = p_{\mathsf{data}}(x)$.

We consider the following specific and widely popular form of contrastive loss for training an encoder $f\colon \mathbb{R}^n \to \mathcal{S}^{m-1}$, mapping data to ℓ_2 normalized feature vectors of dimension m. This loss has been shown effective by many recent representation learning methods (Logeswaran & Lee, 2018; Wu et al., 2018; Tian et al., 2019; He et al., 2019; Hjelm et al., 2018; Bachman et al., 2019; Chen et al., 2020a).

$$\mathcal{L}_{\mathsf{contrastive}}(f;\tau,M) \triangleq \mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\mathsf{pos}} \\ \{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{data}}}} \left[-\log \frac{e^{f(x)^\mathsf{T} f(y)/\tau}}{e^{f(x)^\mathsf{T} f(y)/\tau} + \sum_i e^{f(x_i^-)^\mathsf{T} f(y)/\tau}} \right], \tag{1}$$

where $\tau > 0$ is a scalar temperature hyperparameter, and $M \in \mathbb{Z}_+$ is a fixed number of negative samples.

The term *contrastive loss* has also been generally used to refer to various objectives based on positive and negative samples, e.g., in Siamese networks (Chopra et al., 2005; Hadsell et al., 2006). In this work, we focus on the specific form in Equation (1) that is widely used in modern unsupervised contrastive representation learning literature.

Necessity of normalization. Without the norm constraint, the `softmax` distribution can be made arbitrarily sharp by simply scaling all the features. Wang et al. (2017) provided an analysis on this effect and argued for the necessity of normalization when using feature vector dot products in a cross entropy loss, as is in Eqn. (1). Experimentally, Chen et al. (2020a) also showed that normalizing outputs leads to superior representations.

The InfoMax principle. Many empirical works are motivated by the InfoMax principle of maximizing $I(f(x); f(y))$ for $(x, y) \sim p_{\mathsf{pos}}$ (Tian et al., 2019; Bachman et al., 2019; Wu et al., 2020). Usually they interpret $\mathcal{L}_{\mathsf{contrastive}}$ in Eqn. (1) as a lower bound of $I(f(x); f(y))$ (Oord et al., 2018; Hjelm et al., 2018; Bachman et al., 2019; Tian et al., 2019). However, this interpretation is known to have issues in practice, e.g., maximizing a tighter bound often leads to worse downstream task performance (Tschannen et al., 2019). Therefore, instead of viewing it as a bound, we investigate the exact behavior of directly optimizing $\mathcal{L}_{\mathsf{contrastive}}$ in the following sections.

4. Feature Distribution on the Hypersphere

The contrastive loss encourages learned feature representation for positive pairs to be similar, while pushing features from the randomly sampled negative pairs apart. Conventional wisdom says that representations should extract the most shared information between positive pairs and remain invariant to other noise factors (Linsker, 1988; Tian et al., 2019; Wu et al., 2020; Bachman et al., 2019). Therefore, the loss should prefer two following properties:

- *Alignment*: two samples forming a positive pair should be mapped to nearby features, and thus be (mostly) invariant to unneeded noise factors.
- *Uniformity*: feature vectors should be roughly uniformly distributed on the unit hypersphere $\mathcal{S}^{m-1}$, preserving as much information of the data as possible.

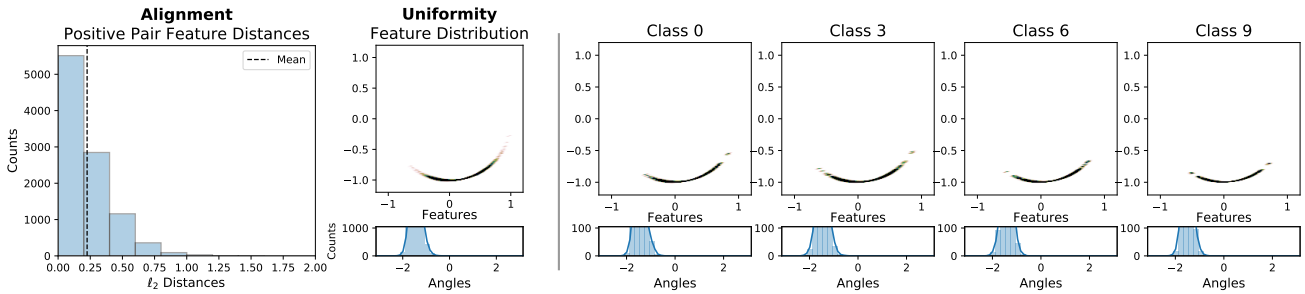
To empirically verify this, we visualize CIFAR-10 (Torralba et al., 2008; Krizhevsky et al., 2009) representations on $\mathcal{S}^1$ ($m = 2$) obtained via three different methods:

- Random initialization.
- Supervised predictive learning: An encoder and a linear classifier are jointly trained from scratch with cross entropy loss on supervised labels.
- Unsupervised contrastive learning: An encoder is trained w.r.t. $\mathcal{L}_{\mathsf{contrastive}}$ with $\tau = 0.5$ and $M = 256$.

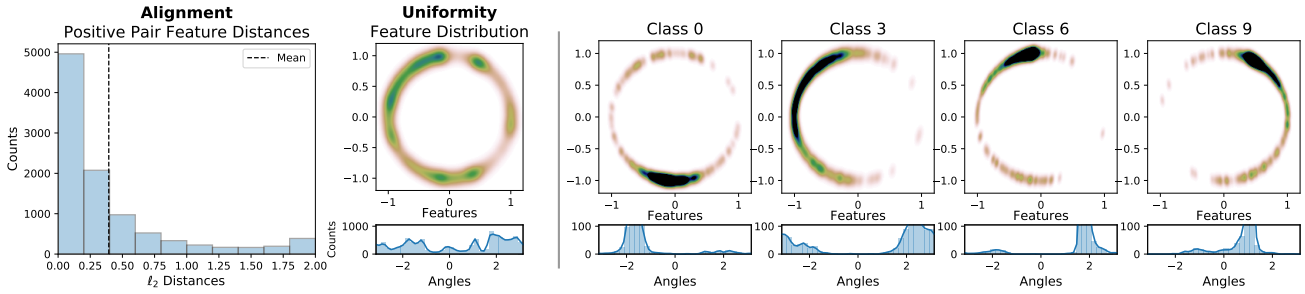
All three encoders share the same AlexNet based architecture (Krizhevsky et al., 2012), modified to map input images to 2-dimensional vectors in $\mathcal{S}^1$. Both predictive and contrastive learning use standard data augmentations to augment the dataset and sample positive pairs.

Figure 3 summarizes the resulting distributions of validation set features. Indeed, features from unsupervised contrastive learning (bottom in Figure 3) exhibit the most uniform distribution, and are closely clustered for positive pairs.

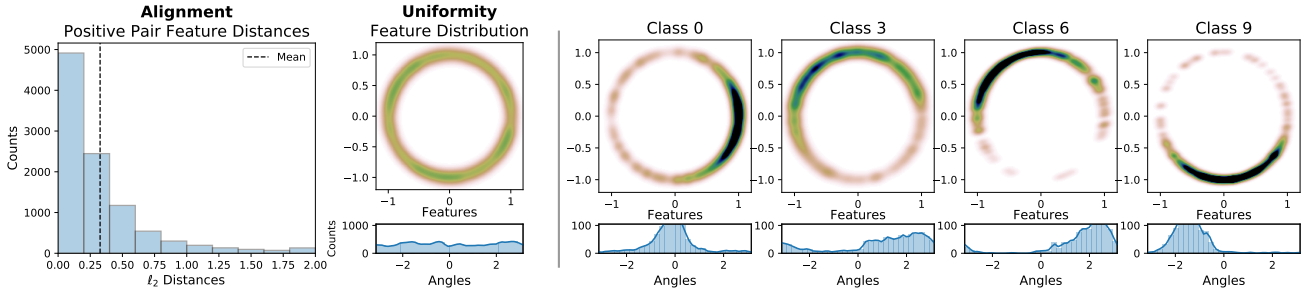
The form of the contrastive loss in Eqn. (1) also suggests this. We present informal arguments below, followed by more formal treatment in Section 4.2. From the symmetry



(a) **Random Initialization.** Linear classification validation accuracy: 12.71%.



(b) **Supervised Predictive Learning.** Linear classification validation accuracy: 57.19%.



(c) **Unsupervised Contrastive Learning.** Linear classification validation accuracy: 28.60%.

Figure 3: Representations of CIFAR-10 validation set on $\mathcal{S}^1$. **Alignment analysis:** We show distribution of distance between features of positive pairs (two random augmentations). **Uniformity analysis:** We plot feature distributions with Gaussian kernel density estimation (KDE) in $\mathbb{R}^2$ and von Mises-Fisher (vMF) KDE on angles (i.e., $\arctan2(y, x)$ for each point $(x, y) \in \mathcal{S}^1$). **Four rightmost plots** visualize feature distributions of selected specific classes. Representation from contrastive learning is both *aligned* (having low positive pair feature distances) and *uniform* (evenly distributed on $\mathcal{S}^1$).

of p, we can derive

$$\mathcal{L}_{\mathsf{contrastive}}(f;\tau,M) = \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{pos}}}\left[-f(x)^\mathsf{T} f(y)/\tau\right] + \mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\mathsf{pos}} \\ \{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{data}}}} \left[\log\left(e^{f(x)^\mathsf{T} f(y)/\tau} + \sum_i e^{f(x_i^-)^\mathsf{T} f(x)/\tau}\right)\right].$$

Because the $\sum_i e^{f(x_i^-)^\mathsf{T} f(x)/\tau}$ term is always positive and bounded below, the loss favors smaller $\mathbb{E}\left[-f(x)^\mathsf{T} f(y)/\tau\right]$, i.e., having more aligned positive pair features. Suppose the encoder is perfectly aligned, i.e., $\mathbb{P}\left[f(x) = f(y)\right] = 1$, then minimizing the loss is equivalent to optimizing

$$\mathop{\mathbb{E}}_{\substack{x\sim p_{\mathsf{data}} \\ \{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{data}}}} \left[\log\left(e^{1/\tau} + \sum_i e^{f(x_i^-)^\mathsf{T} f(x)/\tau}\right)\right],$$

which is akin to maximizing pairwise distances with a `LogSumExp` transformation. Intuitively, pushing all features away from each other should indeed cause them to be roughly uniformly distributed.

4.1. Quantifying Alignment and Uniformity

For further analysis, we need a way to measure alignment and uniformity. We propose the following two metrics (losses).

4.1.1. Alignment

The alignment loss is straightforwardly defined with the expected distance between positive pairs:

$$\mathcal{L}_{\mathsf{align}}(f;\alpha) \triangleq \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{pos}}}\left[\|f(x) - f(y)\|_2^\alpha\right], \quad \alpha > 0.$$

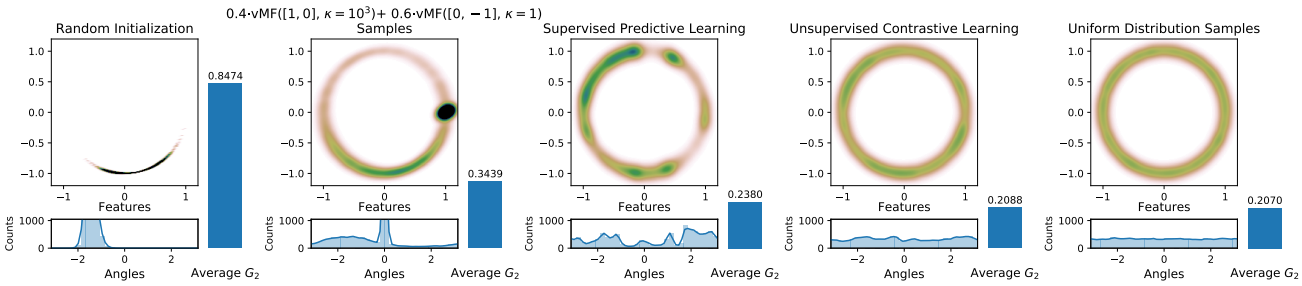


Figure 4: Average pairwise G_2 potential as a measure of uniformity. Each plot shows 10000 points distributed on $\mathcal{S}^1$, obtained via either applying an encoder on CIFAR-10 validation set (same as those in Figure 3) or sampling from a distribution on $\mathcal{S}^1$, as described in plot titles. We show the points with Gaussian KDE and the angles with vMF KDE.

4.1.2. Uniformity

We want the uniformity metric to be both asymptotically correct (i.e., the distribution optimizing this metric should converge to uniform distribution) and empirically reasonable with finite number of points. To this end, we consider the Gaussian potential kernel (also known as the Radial Basis Function (RBF) kernel) $G_t \colon \mathcal{S}^d \times \mathcal{S}^d \to \mathbb{R}_+$ (Cohn & Kumar, 2007; Borodachov et al., 2019):

$$G_t(u, v) \triangleq e^{-t\|u-v\|_2^2} = e^{2t \cdot u^\top v - 2t}, \quad t > 0,$$

and define the uniformity loss as the logarithm of the average pairwise Gaussian potential:

$$\begin{aligned}\mathcal{L}_{\mathsf{uniform}}(f; t) &\triangleq \log \mathop{\mathbb{E}}_{\substack{x,y \overset{\text{i.i.d.}}{\sim} p_{\mathsf{data}}}} [G_t(u, v)] \\ &= \log \mathop{\mathbb{E}}_{\substack{x,y \overset{\text{i.i.d.}}{\sim} p_{\mathsf{data}}}} \left[e^{-t\|f(x)-f(y)\|_2^2}\right], \quad t > 0.\end{aligned}$$

The average pairwise Gaussian potential is nicely tied with the uniform distribution on the unit hypersphere.

Definition (Uniform distribution on $\mathcal{S}^d$)**.** σ_d denotes the normalized surface area measure on $\mathcal{S}^d$.

First, we show that the uniform distribution is the unique distribution that minimize the expected pairwise potential.

Proposition 1. *For $\mathcal{M}(\mathcal{S}^d)$ the set of Borel probability measures on $\mathcal{S}^d$, σ_d is the unique solution of*

$$\min_{\mu \in \mathcal{M}(\mathcal{S}^d)} \int_u \int_v G_t(u, v) \, \mathrm{d}\mu \, \mathrm{d}\mu.$$

Proof. See appendix. □

In addition, as number of points goes to infinity, distributions of points minimizing the average pairwise potential converge weak* to the uniform distribution. Recall the definition of the weak* convergence of measures.

Definition (Weak* convergence of measures)**.** A sequence of Borel measures $\{\mu_n\}_{n=1}^\infty$ in $\mathbb{R}^p$ converges weak* to a Borel measure μ if for all continuous function $f \colon \mathbb{R}^p \to \mathbb{R}$, we have

$$\lim_{n \to \infty} \int f(x) \, \mathrm{d}\mu_n(x) = \int f(x) \, \mathrm{d}\mu(x).$$

Proposition 2. *For each $N > 0$, the N point minimizer of the average pairwise potential is*

$$\mathbf{u}_N^* = \mathop{\arg\min}_{u_1, u_2, \ldots, u_N \in \mathcal{S}^d} \sum_{1 \le i < j \le N} G_t(u_i, u_j).$$

The normalized counting measures associated with the $\{\mathbf{u}_N^*\}_{N=1}^\infty$ sequence converge weak* to σ_d.

Proof. See appendix. □

Designing an objective minimized by the uniform distribution is in fact nontrivial. For instance, average pairwise dot products or Euclidean distances is simply optimized by any distribution that has zero mean. Among kernels that achieve uniformity at optima, the Gaussian kernel is special in that it is closely related to the universally optimal point configurations and can also be used to represent a general class of other kernels, including the Riesz s-potentials. We refer readers to Borodachov et al. (2019) and Cohn & Kumar (2007) for in-depth discussions on these topics. Moreover, as we show below, $\mathcal{L}_{\mathsf{uniform}}$, defined with the Gaussian kernel, has close connections with $\mathcal{L}_{\mathsf{contrastive}}$.

Empirically, we evaluate the average pairwise potential of various finite point collections on $\mathcal{S}^1$ in Figure 4. The values nicely align with our intuitive understanding of uniformity.

We further discuss properties of $\mathcal{L}_{\mathsf{uniform}}$ and characterize its optimal value and range in the appendix.

4.2. Limiting Behavior of Contrastive Learning

In this section, we formalize the intuition that contrastive learning optimizes alignment and uniformity, and characterize its asymptotic behavior. We consider optimization problems over all measurable encoder functions from the p_{data} measure in $\mathbb{R}^n$ to the Borel space $\mathcal{S}^{m-1}$.

We first define the notion of optimal encoders for each of these two metrics.

Definition (Perfect Alignment)**.** We say an encoder f is *perfectly aligned* if $f(x) = f(y)$ a.s. over $(x, y) \sim p_{\mathsf{pos}}$.

Definition (Perfect Uniformity)**.** We say an encoder f is *perfectly uniform* if the distribution of $f(x)$ for $x \sim p_{\mathsf{data}}$ is the uniform distribution σ_{m-1} on $\mathcal{S}^{m-1}$.

Realizability of perfect uniformity. We note that it is not always possible to achieve perfect uniformity, e.g., when the data manifold in $\mathbb{R}^n$ is lower dimensional than the feature space $\mathcal{S}^{m-1}$. Moreover, in the case that p_{data} and p_{pos} are formed from sampling augmented samples from a finite dataset, there cannot be an encoder that is *both* perfectly aligned and perfectly uniform, because perfect alignment implies that all augmentations from a single element have the same feature vector. Nonetheless, perfectly uniform encoder functions do exist under the conditions that $n \geq m - 1$ and p_{data} has bounded density.

We analyze the asymptotics with infinite negative samples. Existing empirical work has established that larger number of negative samples consistently leads to better downstream task performances (Wu et al., 2018; Tian et al., 2019; He et al., 2019; Chen et al., 2020a), and often uses very large values (e.g., $M = 65536$ in He et al. (2019)). The following theorem nicely confirms that optimizing w.r.t. the limiting loss indeed requires both alignment and uniformity.

Theorem 1 (Asymptotics of $\mathcal{L}_{\mathsf{contrastive}}$)**.** *For fixed $\tau > 0$, as the number of negative samples $M \to \infty$, the (normalized) contrastive loss converges to*

$$\lim_{M\to\infty} \mathcal{L}_{\mathsf{contrastive}}(f;\tau,M) - \log M = -\frac{1}{\tau} \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{pos}}} \left[f(x)^\top f(y)\right] + \mathop{\mathbb{E}}_{x\sim p_{\mathsf{data}}} \left[\log \mathop{\mathbb{E}}_{x^-\sim p_{\mathsf{data}}} \left[e^{f(x^-)^\top f(x)/\tau}\right]\right]. \tag{2}$$

We have the following results:

1. *The first term is minimized iff f is perfectly aligned.*
2. *If perfectly uniform encoders exist, they form the exact minimizers of the second term.*
3. *For the convergence in Equation (2), the absolute deviation from the limit decays in $\mathcal{O}(M^{-1/2})$.*

Proof. See appendix. □

Relation with $\mathcal{L}_{\mathsf{uniform}}$. The proof of Theorem 1 in the appendix connects the asymptotic $\mathcal{L}_{\mathsf{contrastive}}$ form with minimizing average pairwise Gaussian potential, i.e., minimizing $\mathcal{L}_{\mathsf{uniform}}$. Compared with the second term of Equation (2), $\mathcal{L}_{\mathsf{uniform}}$ essentially pushes the log outside the outer expectation, without changing the minimizer (perfectly uniform encoders). However, due to its pairwise nature, $\mathcal{L}_{\mathsf{uniform}}$ is much simpler in form and avoids the computationally expensive `softmax` operation in $\mathcal{L}_{\mathsf{contrastive}}$ (Goodman, 2001; Bengio et al.; Gutmann & Hyvärinen, 2010; Grave et al., 2017; Chen et al., 2018).

Relation with feature distribution entropy estimation. When p_{data} is uniform over finite samples $\{x_1, x_2, \ldots, x_N\}$ (e.g., a collected dataset), the second term in Equation (2) can be alternatively viewed as a resubstitution entropy estimator of $f(x)$ (Ahmad & Lin, 1976), where x follows the underlying distribution p_{nature} that generates $\{x_i\}_{i=1}^N$, via a von Mises-Fisher (vMF) kernel density estimation (KDE):

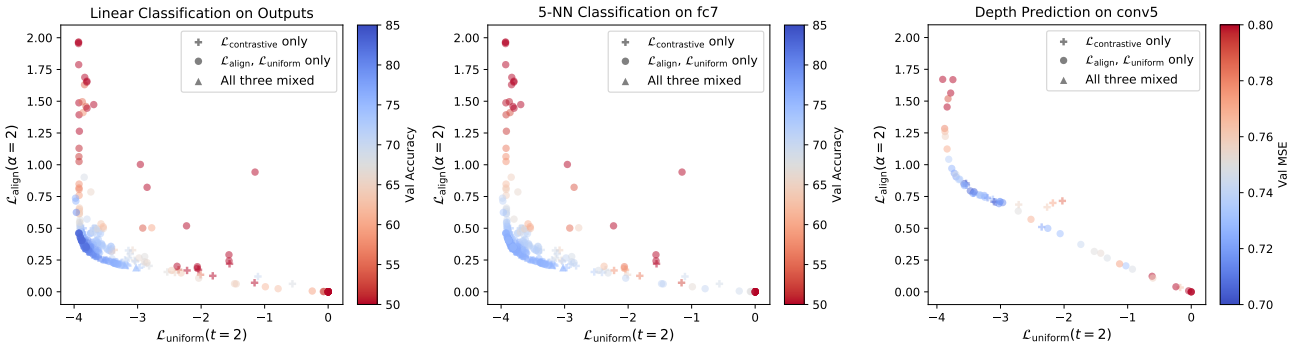
$$\begin{aligned}
&\mathop{\mathbb{E}}_{x\sim p_{\mathsf{data}}} \left[\log \mathop{\mathbb{E}}_{x^-\sim p_{\mathsf{data}}} \left[e^{f(x^-)^\top f(x)/\tau}\right]\right] \\
&= \frac{1}{N}\sum_{i=1}^N \log\left(\frac{1}{N}\sum_{j=1}^N e^{f(x_i)^\top f(x_j)/\tau}\right) \\
&= \frac{1}{N}\sum_{i=1}^N \log \hat{p}_{\mathsf{vMF\text{-}KDE}}(f(x_i)) + \log Z_{\mathsf{vMF}} \\
&\triangleq -\hat{H}(f(x)) + \log Z_{\mathsf{vMF}}, && x \sim p_{\mathsf{nature}} \\
&\triangleq -\hat{I}(x; f(x)) + \log Z_{\mathsf{vMF}}, && x \sim p_{\mathsf{nature}},
\end{aligned}$$

where

- $\hat{p}_{\mathsf{vMF\text{-}KDE}}$ is the KDE based on samples $\{f(x_j)\}_{j=1}^N$ using a vMF kernel with $\kappa = \tau^{-1}$,
- Z_{vMF} is the normalization constant for vMF distribution with $\kappa = \tau^{-1}$,
- $\hat{H}$ denotes the resubstitution entropy estimator,
- $\hat{I}$ denotes the mutual information estimator based on $\hat{H}$, since f is a deterministic function.

Relation with the InfoMax principle. Many empirical works are motivated by the InfoMax principle, i.e., maximizing $I(f(x); f(y))$ for $(x, y) \sim p_{\mathsf{pos}}$. However, the interpretation of $\mathcal{L}_{\mathsf{contrastive}}$ as a lower bound of $I(f(x); f(y))$ is known to be inconsistent with its actual behavior in practice (Tschannen et al., 2019). Our results instead analyze the properties of $\mathcal{L}_{\mathsf{contrastive}}$ itself. Considering the identity $I(f(x); f(y)) = H(f(x)) - H(f(x) \mid f(y))$, we can see that while uniformity indeed favors large $H(f(x))$, alignment is stronger than merely desiring small $H(f(x) \mid f(y))$. In particular, both Theorem 1 and the above connection with maximizing an entropy estimator provide alternative interpretations and motivations that $\mathcal{L}_{\mathsf{contrastive}}$ optimizes for *aligned* and *information-preserving* encoders.

Finally, even for the case where only a single negative sample is used (i.e., $M = 1$), we can still prove a weaker result, which we describe in details in the appendix.



(a) 306 STL-10 encoders are evaluated with linear classification on output features and 5-nearest neighbor (5-NN) on fc7 activations. Higher accuracy (blue color) is better.

(b) 64 NYU-DEPTH-V2 encoders are evaluated with CNN depth regressors on conv5 activations. Lower MSE (blue color) is better.

Figure 5: Metrics and performance of STL-10 and NYU-DEPTH-V2 experiments. Each point represents a trained encoder, with its x- and y-coordinates showing $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ metrics and color showing the performance on validation set. **Blue** is better for both tasks. Encoders with low $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ are consistently the better performing ones (lower left corners).

```
# bsz : batch size (number of positive pairs)
# d   : latent dim
# x   : Tensor, shape=[bsz, d]
#       latents for one side of positive pairs
# y   : Tensor, shape=[bsz, d]
#       latents for the other side of positive pairs
# lam : hyperparameter balancing the two losses

def lalign(x, y, alpha=2):
    return (x - y).norm(dim=1).pow(alpha).mean()

def lunif(x, t=2):
    sq_pdist = torch.pdist(x, p=2).pow(2)
    return sq_pdist.mul(-t).exp().mean().log()

loss = lalign(x, y) + lam * (lunif(x) + lunif(y)) / 2
```

Figure 6: PyTorch implementation of $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$.

5. Experiments

In this section, we empirically verify the hypothesis that alignment and uniformity are desired properties for representations. Recall that our two metrics are

$$\mathcal{L}_{\text{align}}(f;\alpha) \triangleq \mathbb{E}_{(x,y)\sim p_{\text{pos}}} \left[\|f(x) - f(y)\|_2^\alpha\right]$$
$$\mathcal{L}_{\text{uniform}}(f;t) \triangleq \log \mathbb{E}_{x,y \overset{\text{i.i.d.}}{\sim} p_{\text{data}}} \left[e^{-t\|f(x)-f(y)\|_2^2}\right].$$

We conduct extensive experiments with convolutional neural network (CNN) and recurrent neural network (RNN) based encoders on four popular representation learning benchmarks with distinct types of downstream tasks:

- STL-10 (Coates et al., 2011) classification on AlexNet-based encoder outputs or intermediate activations with a linear or k-nearest neighbor (k-NN) classifier.
- NYU-DEPTH-V2 (Nathan Silberman & Fergus, 2012) depth prediction on CNN encoder intermediate activations after convolution layers.
- IMAGENET and IMAGENET-100 (random 100-class subset of IMAGENET) classification on CNN encoder penultimate layer activations with a linear classifier.
- BOOKCORPUS (Zhu et al., 2015) RNN sentence encoder outputs used for Moview Review Sentence Polarity (MR) (Pang & Lee, 2005) and Customer Product Review Sentiment (CR) (Wang & Manning, 2012) binary classification tasks with logisitc classifiers.

For image datasets, we follow the standard practice and choose positive pairs as two independent augmentations of the same image. For BOOKCORPUS, positive pairs are chosen as neighboring sentences, following Quick-Thought Vectors (Logeswaran & Lee, 2018).

We perform majority of our analysis on STL-10 and NYU-DEPTH-V2 encoders, where we calculate $\mathcal{L}_{\text{contrastive}}$ with negatives being other samples within the minibatch following the standard practice (Hjelm et al., 2018; Bachman et al., 2019; Tian et al., 2019; Chen et al., 2020a), and $\mathcal{L}_{\text{uniform}}$ as the logarithm of average pairwise feature potentials also within the minibatch. Due to their simple forms, these two losses can be implemented in PyTorch (Paszke et al., 2019) with less than 10 lines of code, as shown in Figure 6.

To investigate *alignment* and *uniformity* properties on recent contrastive learning methods and larger datasets, we also analyze IMAGENET and IMAGENET-100 encoders trained with Momentum Contrast (MoCo) (He et al., 2019; Chen et al., 2020b), and BOOKCORPUS encoders trained with Quick-Thought Vectors (Logeswaran & Lee, 2018), with these methods modified to also allow $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$.

We optimize a total of 306 STL-10 encoders, 64 NYU-DEPTH-V2 encoders, 45 IMAGENET-100 encoders, and 108 BOOKCORPUS encoders without supervision. The encoders are optimized w.r.t. weighted combinations of $\mathcal{L}_{\text{contrastive}}$, $\mathcal{L}_{\text{align}}$, and/or $\mathcal{L}_{\text{uniform}}$, with varying

	Loss Formula	Validation Set Accuracy ↑			
		Output + Linear	Output + 5-NN	fc7 + Linear	fc7 + 5-NN
Best $\mathcal{L}_{\text{contrastive}}$ only	$\mathcal{L}_{\text{contrastive}}(\tau=0.19)$	80.46%	78.75%	83.89%	76.33%
Best $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ only	$0.98 \cdot \mathcal{L}_{\text{align}}(\alpha=2) + 0.96 \cdot \mathcal{L}_{\text{uniform}}(t=2)$	**81.15%**	78.89%	**84.43%**	**76.78%**
Best among all encoders	$\mathcal{L}_{\text{contrastive}}(\tau=0.5) + \mathcal{L}_{\text{uniform}}(t=2)$	81.06%	**79.05%**	84.14%	76.48%

Table 1: STL-10 encoder evaluations. Numbers show linear and 5-nearest neighbor (5-NN) classification accuracies on the validation set. The best result is picked by encoder outputs linear classifier accuracy from a 5-fold training set cross validation, among all 150 encoders trained from scratch with 128-dimensional output and 768 batch size.

	Loss Formula	Validation Set MSE ↓	
		conv5	conv4
Best $\mathcal{L}_{\text{contrastive}}$ only	$0.5 \cdot \mathcal{L}_{\text{contrastive}}(\tau=0.1)$	0.7024	**0.7575**
Best $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ only	$0.75 \cdot \mathcal{L}_{\text{align}}(\alpha=2) + 0.5 \cdot \mathcal{L}_{\text{uniform}}(t=2)$	**0.7014**	0.7592
Best among all encoders	$0.75 \cdot \mathcal{L}_{\text{align}}(\alpha=2) + 0.5 \cdot \mathcal{L}_{\text{uniform}}(t=2)$	**0.7014**	0.7592

Table 2: NYU-DEPTH-V2 encoder evaluations. Numbers show depth prediction mean squared error (MSE) on the validation set. The best result is picked based on conv5 layer MSE from a 5-fold training set cross validation, among all 64 encoders trained from scratch with 128-dimensional output and 128 batch size.

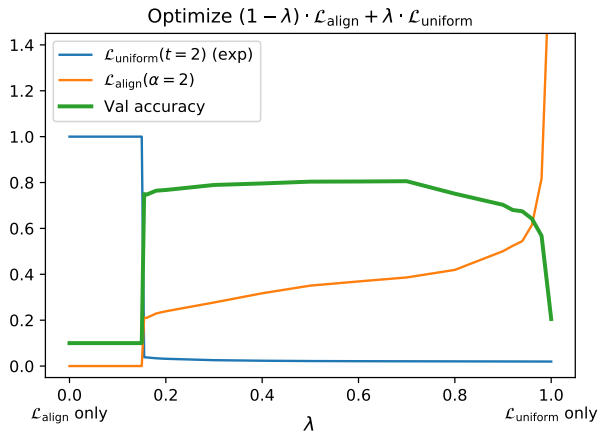


Figure 7: Effect of optimizing different weighted combinations of $\mathcal{L}_{\text{align}}(\alpha=2)$ and $\mathcal{L}_{\text{uniform}}(t=2)$ for STL-10. For each encoder, we show the $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ metrics, and validation accuracy of a linear classifier trained on encoder outputs. $\mathcal{L}_{\text{uniform}}$ is exponentiated for plotting purposes.

- (possibly zero) weights on the three losses,
- temperature τ for $\mathcal{L}_{\text{contrastive}}$,
- $\alpha \in \{1, 2\}$ for $\mathcal{L}_{\text{align}}$,
- $t \in \{1, 2, \ldots, 8\}$ for $\mathcal{L}_{\text{uniform}}$,
- batch size (affecting the number of (negative) pairs for $\mathcal{L}_{\text{contrastive}}$ and $\mathcal{L}_{\text{uniform}}$),
- embedding dimension,
- number of training epochs and learning rate,
- initialization (from scratch vs. a pretrained encoder).

See the appendix for more experiment details and the exact configurations used.

$\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ strongly agree with downstream task performance. For each encoder, we measure the downstream task performance, and the $\mathcal{L}_{\text{align}}$, $\mathcal{L}_{\text{uniform}}$ metrics on the validation set. Figure 5 visualizes the trends between both metrics and representation quality. We observe that the two metrics strongly agrees the representation quality overall. In particular, the best performing encoders are exactly the ones with low $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$, i.e., the lower left corners in Figure 5.

Directly optimizing only $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ can lead to better representations. As shown in Tables 1 and 2, encoders trained with only $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ consistently outperform their $\mathcal{L}_{\text{contrastive}}$-trained counterparts, for both tasks. Theoretically, Theorem 1 showed that $\mathcal{L}_{\text{contrastive}}$ optimizes alignment and uniformity asymptotically with infinite negative samples. This empirical performance gap suggests that directly optimizing these properties can be superior in practice, when we can only have finite negatives.

Both alignment and uniformity are necessary for a good representation. Figure 7 shows how the final encoder changes in response to optimizing differently weighted combinations of $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ on STL-10. The trade-off between the $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ indicates that perfect alignment and perfect uniformity are likely hard to simultaneously achieve in practice. However, the inverted-U-shaped accuracy curve confirms that both properties are indeed necessary for a good encoder. When $\mathcal{L}_{\text{align}}$ is weighted much higher than $\mathcal{L}_{\text{uniform}}$, degenerate solution occurs and all inputs are mapped to the same feature vector ($\exp \mathcal{L}_{\text{uniform}} = 1$). However, as long as the ratio between two weights is not too large (e.g., < 4), we observe that the representation quality remains relatively good and insensitive to the exact weight choices.

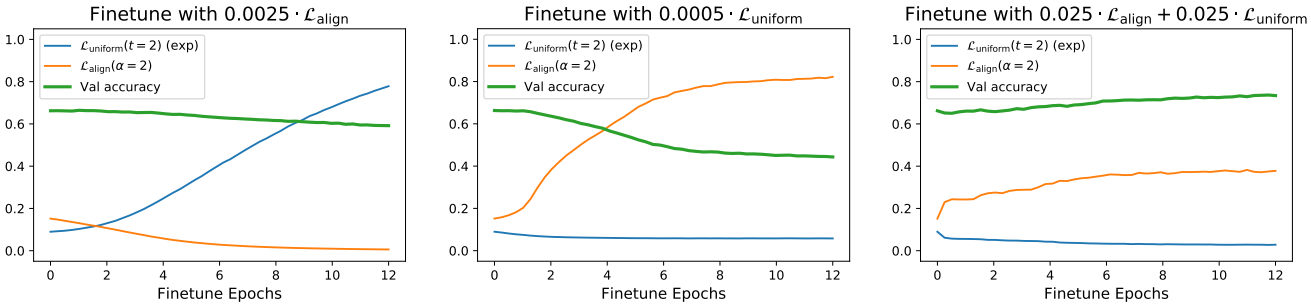
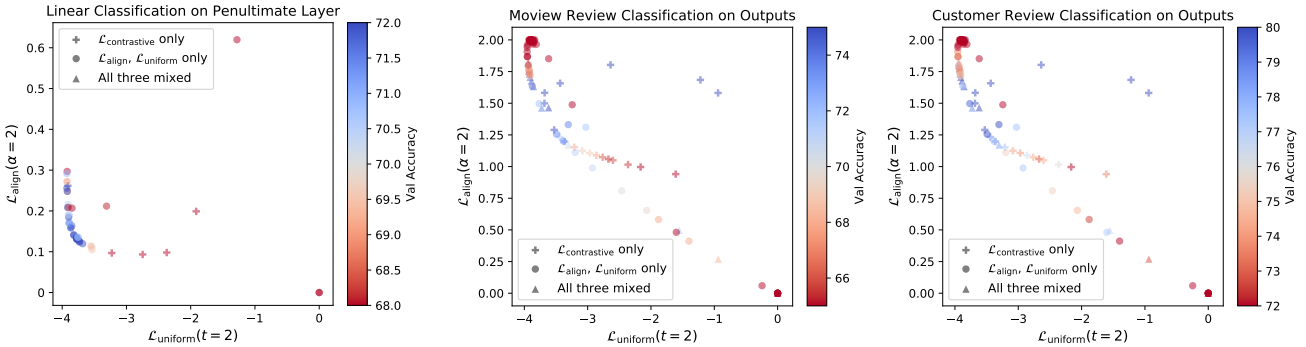


Figure 8: Finetuning trajectories from a STL-10 encoder trained with $\mathcal{L}_{\text{contrastive}}$ using a suboptimal temperature $\tau = 2.5$. Finetuning objectives are weighted combinations of $\mathcal{L}_{\text{align}}(\alpha=2)$ and $\mathcal{L}_{\text{uniform}}(t=2)$. For each intermediate checkpoint, we measure $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ metrics, as well as validation accuracy of a linear classifier trained from scratch on the encoder outputs. $\mathcal{L}_{\text{uniform}}$ is exponentiated for plotting purpose. **Left and middle:** Performance degrades if only one of alignment and uniformity is optimized. **Right:** Performance improves when both are optimized.



(a) 45 IMAGENET-100 encoders are trained with MoCo-based methods, and evaluated with linear classification.

(b) 108 BOOKCORPUS encoders are trained with Quick-Thought-Vectors-based methods, and evaluated with logistic binary classification on Movie Review Sentence Polarity and Customer Product Review Sentiment tasks.

Figure 9: Metrics and performance of IMAGENET-100 and BOOKCORPUS experiments. Each point represents a trained encoder, with its x- and y-coordinates showing $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ metrics and color showing the validation accuracy. **Blue** is better. Encoders with low $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ consistently perform well (lower left corners), even though the training methods (based on MoCo and Quick-Thought Vectors) are different from directly optimizing the contrastive loss in Equation (1).

$\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ causally affect downstream task performance. We take an encoder trained with $\mathcal{L}_{\text{contrastive}}$ using a suboptimal temperature $\tau = 2.5$, and finetune it according to $\mathcal{L}_{\text{align}}$ and/or $\mathcal{L}_{\text{uniform}}$. Figure 8 visualizes the finetuning trajectories. When only one of alignment and uniformity is optimized, the corresponding metric improves, but both the other metric and performance degrade. However, when both properties are optimized, the representation quality steadily increases. These trends confirm the causal effect of alignment and uniformity on the representation quality, and suggest that directly optimizing them can be a reasonable choice.

Alignment and uniformity also matter in other contrastive representation learning variants. MoCo (He et al., 2019) and Quick-Thought Vectors (Logeswaran & Lee, 2018) are contrastive representation learning variants that have nontrivial differences with directly optimizing $\mathcal{L}_{\text{contrastive}}$ in Equation (1). MoCo introduces a memory queue and a momentum encoder. Quick-Thought Vectors uses two different encoders to encode each sentence in a positive pair, only normalizes encoder outputs during evaluation, and does not use random sampling to obtain minibatches. After modifying them to also allow $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$, we train these methods on IMAGENET-100 and BOOKCORPUS, respectively. Figure 9 shows that $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ metrics are still correlated with the downstream task performances. Tables 3 and 4 show that directly optimizing them also leads to comparable or better representation quality. Table 5 also shows improvements on full IMAGENET when we use $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ to train MoCo v2 (Chen et al., 2020b) (an improved version of MoCo). These results suggest that alignment and uniformity are indeed desirable properties for representations, for *both* image and text modalities, and are likely connected with general contrastive representation learning methods.

	Loss Formula	Validation Set Accuracy ↑	
		top1	top5
Best $\mathcal{L}_{\text{contrastive}}$ only	$\mathcal{L}_{\text{contrastive}}(\tau=0.07)$	72.80%	91.64%
Best $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ only	$3 \cdot \mathcal{L}_{\text{align}}(\alpha=2) + \mathcal{L}_{\text{uniform}}(t=3)$	**74.60%**	**92.74%**
Best among all encoders	$3 \cdot \mathcal{L}_{\text{align}}(\alpha=2) + \mathcal{L}_{\text{uniform}}(t=3)$	**74.60%**	**92.74%**

Table 3: IMAGENET-100 encoder evaluations. Numbers show validation set accuracies of linear classifiers trained on encoder penultimate layer activations. The encoders are trained using MoCo-based methods. The best result is picked based on top1 accuracy from a 3-fold training set cross validation, among all 45 encoders trained from scratch with 128-dimensional output and 128 batch size.

	MR Classification		CR Classification	
	Loss Formula	Val. Set Accuracy ↑	Loss Formula	Val. Set Accuracy ↑
Best $\mathcal{L}_{\text{contrastive}}$ only	$\mathcal{L}_{\text{contrastive}}(\tau=0.075)$	**77.51%**	$\mathcal{L}_{\text{contrastive}}(\tau=0.05)$	**83.86%**
Best $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ only	$0.9 \cdot \mathcal{L}_{\text{align}}(\alpha=2) + 0.1 \cdot \mathcal{L}_{\text{uniform}}(t=5)$	73.76%	$0.9 \cdot \mathcal{L}_{\text{align}}(\alpha=2) + 0.1 \cdot \mathcal{L}_{\text{uniform}}(t=5)$	80.95%
Best among all encoders	$\mathcal{L}_{\text{contrastive}}(\tau=0.075)$	**77.51%**	$\mathcal{L}_{\text{contrastive}}(\tau=0.05)$	**83.86%**

Table 4: BOOKCORPUS encoder evaluations. Numbers show Movie Review Sentence Polarity (MR) and Customer Product Sentiment (CR) validation set classification accuracies of logistic classifiers fit on encoder outputs. The encoders are trained using Quick-Thought-Vectors-based methods. The best result is picked based on accuracy from a 5-fold training set cross validation, individually for MR and CR, among all 108 encoders trained from scratch with 1200-dimensional output and 400 batch size.

Loss Formula	Validation Set top1 Accuracy ↑
$\mathcal{L}_{\text{contrastive}}(\tau=0.2)$ (MoCo v2 Chen et al. (2020b))	$67.5\% \pm 0.1\%$
$3 \cdot \mathcal{L}_{\text{align}}(\alpha=2) + \mathcal{L}_{\text{uniform}}(t=3)$	**67.69%**

Table 5: IMAGENET encoder evaluations with MoCo v2, and its variant with $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$. MoCo v2 results are from the MoCo v2 official implementation (Chen et al., 2020c), with mean and standard deviation across 5 runs. Both settings use 200 epochs of unsupervised training.

6. Discussion

Alignment and *uniformity* are often alluded to as motivations for representation learning methods (see Figure 1). However, a thorough understanding of these properties is lacking in the literature.

Are they in fact related to the representation learning methods? Do they actually agree with the representation quality (measured by downstream task performance)?

In this work, we have presented a detailed investigation on the relation between these properties and the popular paradigm of contrastive representation learning. Through theoretical analysis and extensive experiments, we are able to relate the contrastive loss with the alignment and uniformity properties, and confirm their strong connection with downstream task performances. Remarkably, we have revealed that directly optimizing our proposed metrics often leads to representations of better quality.

Below we summarize several suggestions for future work.

Niceness of the unit hypersphere. Our analysis was based on the empirical observation that representations are often ℓ_2 normalized. Existing works have motivated this choice from a manifold mapping perspective (Liu et al., 2017; Davidson et al., 2018) and computation stability (Xu & Durrett, 2018; Wang et al., 2017). However, to our best knowledge, the question of why the unit hypersphere is a nice feature space is not yet rigorously answered. One possible direction is to formalize the intuition that connected sets with smooth boundaries are nearly linearly separable in the hyperspherical geometry (see Figure 2), since linear separability is one of the most widely used criteria for representation quality and is related to the notion of disentanglement (Higgins et al., 2018).

Beyond contrastive learning. Our analysis focused on the relationship between contrastive learning and the alignment and uniformity properties on the unit hypersphere. However, the ubiquitous presence of ℓ_2 normalization in the representation learning literature suggests that the connection may be more general. In fact, several existing empirical methods are directly related to uniformity on the hypersphere (Bojanowski & Joulin, 2017; Davidson et al., 2018; Xu & Durrett, 2018). We believe that relating a broader class of representations to uniformity and/or alignment on the hypersphere will provide novel insights and lead to better empirical algorithms.

Acknowledgements

We thank Philip Bachman, Ching-Yao Chuang, Justin Solomon, Yonglong Tian, and Zhenyang Zhang for many helpful comments and suggestions. Tongzhou Wang was supported by the MIT EECS Merrill Lynch Graduate Fellowship.

Major Changelog

8/24/2020:

- Added results on full ImageNet and MoCo v2.

11/6/2020:

- Added discussions on the range of $\mathcal{L}_{\mathsf{uniform}}$.
- Corrected Theorem 1's convergence rate to $\mathcal{O}(M^{-1/2})$.

References

Ahmad, I. and Lin, P.-E. A nonparametric estimation of the entropy for absolutely continuous distributions (corresp.). *IEEE Transactions on Information Theory*, 22(3):372–375, 1976.

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pp. 15509–15519, 2019.

Bengio, Y. et al. Quick training of probabilistic neural nets by importance sampling.

Bochner, S. Monotone funktionen, stieltjessche integrale und harmonische analyse. *Collected Papers of Salomon Bochner*, 2: 87, 1992.

Bojanowski, P. and Joulin, A. Unsupervised learning by predicting noise. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 517–526. JMLR. org, 2017.

Borodachov, S. V., Hardin, D. P., and Saff, E. B. *Discrete energy on rectifiable sets*. Springer, 2019.

Chen, P. H., Si, S., Kumar, S., Li, Y., and Hsieh, C.-J. Learning to screen for fast softmax inference on large vocabulary neural networks. 2018.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020a.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. GitHub repository https://github.com/facebookresearch/moco/tree/78b69cafae80bc74cd1a89ac3fb365dc20d157d3, 2020c.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 539–546. IEEE, 2005.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223, 2011.

Cohn, H. and Kumar, A. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*, 20 (1):99–148, 2007.

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*, 2018.

Goodman, J. Classes for fast maximum entropy training. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 1, pp. 561–564. IEEE, 2001.

Götz, M. and Saff, E. B. Note on d—extremal configurations for the sphere in r d+1. In *Recent Progress in Multivariate Approximation*, pp. 159–162. Springer, 2001.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Grave, E., Joulin, A., Cissé, M., Jégou, H., et al. Efficient softmax approximation for gpus. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1302–1310. JMLR. org, 2017.

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, 2010.

Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

Hardin, D. and Saff, E. Minimal riesz energy point configurations for rectifiable d-dimensional manifolds. *Advances in Mathematics*, 193(1):174–204, 2005.

Hasnat, M., Bohné, J., Milgram, J., Gentric, S., Chen, L., et al. von mises-fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264*, 2017.

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. *arXiv preprint arXiv:1911.05722*, 2019.

Hénaff, O. J., Razavi, A., Doersch, C., Eslami, S., and Oord, A. v. d. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*, 2015.

Kobayashi, S. Homemade bookcorpus. GitHub repository https://github.com/soskek/bookcorpus/tree/5fe0cec8d7fd83940e48c799739496dc68ab2798, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Landkof, N. S. *Foundations of modern potential theory*, volume 180. Springer, 1972.

Linsker, R. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220, 2017.

Liu, W., Lin, R., Liu, Z., Liu, L., Yu, Z., Dai, B., and Song, L. Learning towards minimum hyperspherical energy. In *Advances in Neural Information Processing Systems*, pp. 6222–6233. 2018.

Logeswaran, L. and Lee, H. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*, 2018.

Mettes, P., van der Pol, E., and Snoek, C. Hyperspherical prototype networks. In *Advances in Neural Information Processing Systems*, pp. 1485–1495, 2019.

Nathan Silberman, Derek Hoiem, P. K. and Fergus, R. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

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

Pang, B. and Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pp. 115–124. Association for Computational Linguistics, 2005.

Parkhi, O. M., Vedaldi, A., and Zisserman, A. Deep face recognition. 2015.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8026–8037. 2019.

Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., and Khandeparkar, H. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, pp. 5628–5637, 2019.

Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Serfozo, R. Convergence of lebesgue integrals with varying measures. *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 380–402, 1982.

Stewart, J. Positive definite functions and generalizations, an historical survey. *The Rocky Mountain Journal of Mathematics*, 6(3):409–434, 1976.

Tammes, P. M. L. On the origin of number and arrangement of the places of exit on the surface of pollen-grains. *Recueil des travaux botaniques néerlandais*, 27(1):1–84, 1930.

Thomson, J. J. Xxiv. on the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 7(39):237–265, 1904.

Tian, Y. Contrastive multiview coding. GitHub repository https://github.com/HobbitLong/CMC/tree/58d06e9a82f7fea2e4af0a251726e9c6bf67c7c9, 2019.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

Torralba, A., Fergus, R., and Freeman, W. T. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970, 2008.

Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S., and Lucic, M. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Wang, F., Xiang, X., Cheng, J., and Yuille, A. L. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1041–1049, 2017.

Wang, S. and Manning, C. D. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2*, pp. 90–94. Association for Computational Linguistics, 2012.

Wu, M., Zhuang, C., Yamins, D., and Goodman, N. On the importance of views in unsupervised representation learning. 2020.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

Xu, J. and Durrett, G. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4503–4513, 2018.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *arXiv preprint arXiv:1506.06724*, 2015.

A. Proofs and Additional Theoretical Analysis

In this section, we present proofs for propositions and theorems in main paper Sections 4.1.2 and 4.2.

The propositions in Section 4.1.2 illustrate the deep relations between the Gaussian kernel $G_t \colon \mathcal{S}^d \times \mathcal{S}^d \to \mathbb{R}$ and the uniform distribution on the unit hypersphere $\mathcal{S}^d$. As we will show below in Section A.1, these properties directly follow well-known results on strictly positive definite kernels.

In Section A.2, we present a proof for Theorem 1. Theorem 1 describes the asymptotic behavior of $\mathcal{L}_{\mathsf{contrastive}}$ as the number of negative samples M approaches infinity. The theorem is strongly related to empirical contrastive learning, given an error term (deviation from the limit) decaying in $\mathcal{O}(M^{-1/2})$ and that empirical practices often use a large number of negatives (e.g., $M = 65536$ in He et al. (2019)) based on the observation that using more negatives consistently leads to better representation quality (Wu et al., 2018; Tian et al., 2019; He et al., 2019). Our proof further reveals connections between $\mathcal{L}_{\mathsf{contrastive}}$ and $\mathcal{L}_{\mathsf{uniform}}$ which is defined via the Gaussian kernel.

Finally, also in Section A.2, we present a weaker result on the setting where only a single negative is used in $\mathcal{L}_{\mathsf{contrastive}}$ (i.e., $M = 1$).

A.1. Proofs for Section 4.1.2 and Properties of $\mathcal{L}_{\mathsf{uniform}}$

To prove Propositions 1 and 2, we utilize the *strict positive definiteness* (Bochner, 1992; Stewart, 1976) of the Gaussian kernel G_t:

$$G_t(u, v) \triangleq e^{-t\|u-v\|_2^2} = e^{2t \cdot u^\top v - 2t}, \quad t > 0.$$

From there, we apply a known result about such kernels, from which the two propositions directly follow.

Definition (Strict positive definiteness (Bochner, 1992; Stewart, 1976))**.** A symmetric and lower semi-continuous kernel K on $A \times A$ (where A is infinite and compact) is called strictly positive definite if for every finite signed Borel measure μ supported on A whose energy

$$I_K[\mu] \triangleq \int_{\mathcal{S}^d} \int_{\mathcal{S}^d} K(u, v) \, \mathrm{d}\mu(v) \, \mathrm{d}\mu(u)$$

is well defined, we have $I_K[\mu] \geq 0$, where equality holds only if $\mu \equiv 0$ on the σ-algebra of Borel subsets of A.

Definition. Let $\mathcal{M}(\mathcal{S}^d)$ be the set of Borel probability measures on $\mathcal{S}^d$.

We are now in the place to apply the following two well-known results, which we present by restating Proposition 4.4.1, Theorem 6.2.1 and Corollary 6.2.2 of Borodachov et al. (2019) in weaker forms. We refer readers to Borodachov et al. (2019) for their proofs.

Lemma 1 (Strict positive definiteness of G_t)**.** *For $t > 0$, the Gaussian kernel $G_t(u, v) \triangleq e^{-t\|u-v\|_2^2} = e^{2t \cdot u^\top v - 2t}$ is strictly positive definite on $\mathcal{S}^d \times \mathcal{S}^d$.*

Lemma 2 (Strictly positive definite kernels on $\mathcal{S}^d$)**.** *Consider kernel $K_f \colon \mathcal{S}^d \times \mathcal{S}^d \to (-\infty, +\infty]$ of the form,*

$$K_f(u, v) \triangleq f(\|u - v\|_2^2). \tag{3}$$

If K_f is strictly positive definite on $\mathcal{S}^d \times \mathcal{S}^d$ and $I_{K_f}[\sigma_d]$ is finite, then σ_d is the unique measure (on Borel subsets of $\mathcal{S}^d$) in the solution of $\min_{\mu \in \mathcal{M}(\mathcal{S}^d)} I_{K_f}[\mu]$, and the normalized counting measures associated with any K_f-energy minimizing sequence of N-point configurations on $\mathcal{S}^d$ converges weak* to σ_d.

In particular, this conclusion holds whenever f has the property that $-f'(t)$ is strictly completely monotone on $(0, 4]$ and $I_{K_f}[\sigma_d]$ is finite.

We now recall Propositions 1 and 2.

Proposition 1. *σ_d is the unique solution (on Borel subsets of $\mathcal{S}^d$) of*

$$\min_{\mu \in \mathcal{M}(\mathcal{S}^d)} I_{G_t}[\mu] = \min_{\mu \in \mathcal{M}(\mathcal{S}^d)} \int_{\mathcal{S}^d} \int_{\mathcal{S}^d} G_t(u, v) \, \mathrm{d}\mu(v) \, \mathrm{d}\mu(u). \tag{4}$$

Proof of Proposition 1. This is a direct consequence of Lemmas 1 and 2. □

Proposition 2. *For each $N > 0$, the N point minimizer of the average pairwise potential is*

$$\mathbf{u}_N^* = \operatorname*{arg\,min}_{u_1,u_2,\ldots,u_N \in \mathcal{S}^d} \sum_{1 \le i < j \le N} G_t(u_i, u_j).$$

The normalized counting measures associated with the $\{\mathbf{u}_N^*\}_{N=1}^{\infty}$ sequence converge weak to σ_d.*

Proof of Proposition 2. This is a direct consequence of Lemmas 1 and 2. □

A.1.1. More Properties of $\mathcal{L}_{\text{uniform}}$

Range of $\mathcal{L}_{\text{uniform}}$. It's not obvious what the optimal value of $\mathcal{L}_{\text{uniform}}$ is. In the following proposition, we characterize the exact range of the expected Gaussian potential and how it evolves as dimensionality increases. The situation for $\mathcal{L}_{\text{uniform}}$ directly follows as a corollary.

Proposition 3 (Range of the expected pairwise Gaussian potential G_t)**.** *For $t > 0$, the expected pairwise Gaussian potential w.r.t. Borel probability measure $\mu \in \mathcal{M}(\mathcal{S}^d)$*

$$I_{G_t}[\mu] = \int_{\mathcal{S}^d} \int_{\mathcal{S}^d} G_t(u, v) \, \mathrm{d}\mu(v) \, \mathrm{d}\mu(u)$$

has range $[e^{-2t} \, {}_0F_1(; \frac{d+1}{2}; t^2), 1]$, where ${}_0F_1$ is the confluent hypergeometric limit function defined as

$${}_0F_1(; \alpha; z) \triangleq \sum_{n=0}^{\infty} \frac{z^n}{(\alpha)_n n!}, \tag{5}$$

where we have used the Pochhammer symbol $(a)_n = \begin{cases} 1 & \text{if } n = 0 \\ a(a+1)(n+2)\ldots(a+n-1) & \text{if } n \ge 1. \end{cases}$

We have

- *The minimum $e^{-2t} \, {}_0F_1(; \frac{d+1}{2}; t^2)$ is achieved iff $\mu = \sigma_d$ (on Borel subsets of $\mathcal{S}^d$). Furthermore, this value strictly decreases as d increases, converging to e^{-2t} in the limit $d \to \infty$.*
- *The maximum is achieved iff μ is a Dirac delta distribution, i.e., $\mu = \delta_u$ (on Borel subsets of $\mathcal{S}^d$), for some $u \in \mathcal{S}^d$.*

Proof of Proposition 3.

- **Minimum.**

  We know from Proposition 1 that σ_d *uniquely* achieves the minimum, given by the following integral ratio

$$\begin{aligned} I_{G_t}[\sigma_d] &= \frac{\int_0^\pi e^{-t(2\sin\frac{\theta}{2})^2} \sin^{d-1}\theta \, \mathrm{d}\theta}{\int_0^\pi \sin^{d-1}\theta \, \mathrm{d}\theta} \\ &= \frac{\int_0^\pi e^{-2t(1-\cos\theta)} \sin^{d-1}\theta \, \mathrm{d}\theta}{\int_0^\pi \sin^{d-1}\theta \, \mathrm{d}\theta} \\ &= e^{-2t} \frac{\int_0^\pi e^{2t\cos\theta} \sin^{d-1}\theta \, \mathrm{d}\theta}{\int_0^\pi \sin^{d-1}\theta \, \mathrm{d}\theta}. \end{aligned}$$

  The denominator, with some trigonometric identities, can be more straightforwardly evaluated as

$$\int_0^\pi \sin^{d-1}\theta \, \mathrm{d}\theta = \sqrt{\pi} \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d+1}{2})}.$$

The numerator is

$$\begin{aligned}\int_0^\pi e^{2t\cos\theta}\sin^{d-1}\theta\,\mathrm{d}\theta &= -\int_0^\pi e^{2t\cos\theta}\sin^{d-2}\theta\cos'\theta\,\mathrm{d}\theta \\ &= \int_{-1}^1 e^{2ts}(1-s^2)^{d/2-1}\,\mathrm{d}s \\ &= \frac{\Gamma(\frac{d-1}{2}+\frac{1}{2})\sqrt{\pi}}{\Gamma(\frac{d-1}{2}+1)}\,{}_0F_1(;\frac{d-1}{2}+1;-\frac{1}{4}(-2it)^2) \\ &= \frac{\Gamma(\frac{d}{2})\sqrt{\pi}}{\Gamma(\frac{d+1}{2})}\,{}_0F_1(;\frac{d+1}{2};t^2).\end{aligned}$$

where we have used the following identity based on the Poisson formula for Bessel functions and the relationship between ${}_0F_1$ and Bessel functions:

$$\int_{-1}^1 e^{izs}(1-s^2)^{\nu-\frac{1}{2}}\,\mathrm{d}s = \frac{\Gamma(\nu+\frac{1}{2})\sqrt{\pi}}{(\frac{z}{2})^\nu}J_\nu(z) = \frac{\Gamma(\nu+\frac{1}{2})\sqrt{\pi}}{\Gamma(\nu+1)}\,{}_0F_1(;\nu+1;-\frac{1}{4}z^2).$$

Putting both together, we have

$$\begin{aligned}I_{G_t}[\sigma_d] &= e^{-2t}\frac{\int_0^\pi e^{2t\cos\theta}\sin^{d-1}\theta\,\mathrm{d}\theta}{\int_0^\pi \sin^{d-1}\theta\,\mathrm{d}\theta} \\ &= e^{-2t}\frac{\frac{\Gamma(\frac{d}{2})\sqrt{\pi}}{\Gamma(\frac{d+1}{2})}\,{}_0F_1(;\frac{d+1}{2};t^2)}{\sqrt{\pi}\frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d+1}{2})}} \\ &= e^{-2t}\,{}_0F_1(;\frac{d+1}{2};t^2) \\ &= e^{-2t}\sum_{n=0}^\infty \frac{t^{2n}}{(\frac{d+1}{2})_n n!},\end{aligned}$$

where we have used the definition of ${}_0F_1$ in Equation (5) to expand the formula.
Notice that each summand strictly decreases as $d\to\infty$. So must the total sum.
For the asymptotic behavior at $d\to\infty$, it only remains to show that

$$\lim_{d\to\infty}\sum_{n=0}^\infty \frac{t^{2n}}{(\frac{d+1}{2})_n n!} = 1. \tag{6}$$

For the purpose of applying the Dominated Convergence Theorem (DCT) (on the counting measure). We consider the following summable series

$$\sum_{n=0}^\infty \frac{t^{2n}}{n!} = e^{t^2},$$

with each term bounding the corresponding one in Equation (6):

$$\frac{t^{2n}}{n!} \geq \frac{t^{2n}}{(\frac{d+1}{2})_n n!}, \qquad \forall n\geq 0, d>0.$$

Thus,

$$\lim_{d\to\infty}\sum_{n=0}^\infty \frac{t^{2n}}{(\frac{d+1}{2})_n n!} = \sum_{n=0}^\infty \lim_{d\to\infty}\frac{t^{2n}}{(\frac{d+1}{2})_n n!} = 1+0+0+\cdots = 1.$$

Hence, the asymptotic lower range is e^{-2t}.

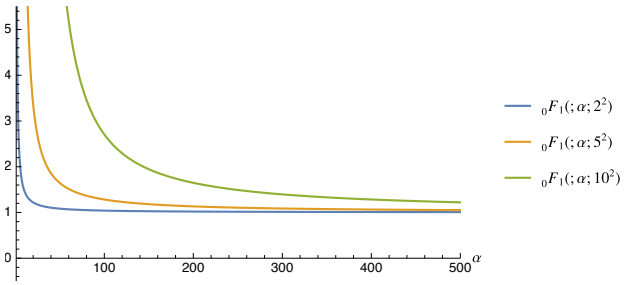


Figure 10: Asymptotic behavior of ${}_0F_1(;\alpha;z)$. For $z>0$, as α grows larger, the function converges to 1.

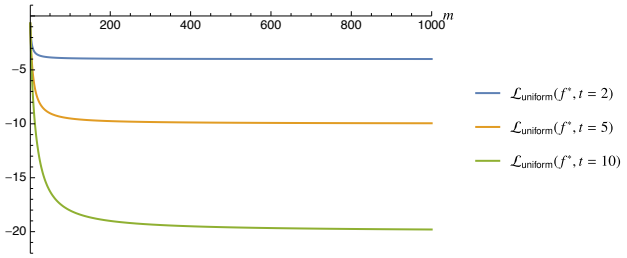


Figure 11: Asymptotic behavior of optimal $\mathcal{L}_{\text{uniform}}(f,t)$, attained by a perfectly uniform encoder f^*. As the feature dimension m grows larger, the value converges to $-2t$.

- **Maximum.**

  Obviously, Dirac delta distributions δ_u, $u \in \mathcal{S}^d$ would achieve a maximum of 1. We will now show that all Borel probability measures μ s.t. $I_{G_t}[\mu]=1$ are delta distributions.

  Suppose that such a μ is not a Dirac delta distribution. Then, we can take distinct $x, y \in \operatorname{supp}(\mu) \subseteq \mathcal{S}^d$, and open neighborhoods around x and v, $N_x, N_y \in \mathcal{S}^d$ such that they are small enough and disjoint:

$$N_x \triangleq \{u \in \mathcal{S}^d \colon \|u-x\|_2 < \frac{1}{3}\|x-y\|_2\}$$
$$N_y \triangleq \{u \in \mathcal{S}^d \colon \|u-y\|_2 < \frac{1}{3}\|x-y\|_2\}.$$

  Then,

$$\begin{aligned}
I_{G_t}[\mu] &= \int_{\mathcal{S}^d}\int_{\mathcal{S}^d} G_t(u,v)\,\mathrm{d}\mu(v)\,\mathrm{d}\mu(u) \\
&= \int_{\mathcal{S}^d}\int_{\mathcal{S}^d} e^{-t\|u-v\|_2^2}\,\mathrm{d}\mu(v)\,\mathrm{d}\mu(u) \\
&\le (1-2\mu(N_x)\mu(N_y))e^{-t\cdot 0} + 2\int_{N_x}\int_{N_y} e^{-t\|u-v\|_2^2}\,\mathrm{d}\mu(v)\,\mathrm{d}\mu(u) \\
&< 1-2\mu(N_x)\mu(N_y) + 2\mu(N_x)\mu(N_y)e^{-t(\|x-y\|_2/3)^2} \\
&= 1-2\mu(N_x)\mu(N_y)(1-e^{-\frac{t}{9}\|x-y\|_2^2}) \\
&< 1.
\end{aligned}$$

  Hence, only Dirac delta distributions attain the maximum.

□

Corollary 1 (Range of $\mathcal{L}_{\text{uniform}}$)**.** *For encoder $f\colon \mathbb{R}^n \to \mathcal{S}^{m-1}$, $\mathcal{L}_{\text{uniform}}(f;t) \in [-2t+\log {}_0F_1(;\frac{m}{2};t^2), 0]$, where the lower bound $-2t+\log {}_0F_1(;\frac{m}{2};t^2)$ is achieved only by perfectly uniform encoders f, and the upper bound 0 is achieved only by degenerate encoders that output a fixed feature vector almost surely.*

Furthermore, the lower bound strictly decreases as the output dimension m increases, attaining the following asymptotic value

$$\lim_{m\to\infty} -2t+\log {}_0F_1(;\frac{m}{2};t^2) = -2t. \tag{7}$$

Intuition for the optimal $\mathcal{L}_{\text{uniform}}$ value in high dimensions. If we ignore the $\log {}_0F_1(;\frac{m}{2};t^2)$ term, informally, the optimal value of $-2t$ roughly says that any pair of feature vectors on $\mathcal{S}^d$ has distance about $\sqrt{2}$, i.e., are nearly orthogonal to each other. Indeed, vectors of high dimensions are usually nearly orthogonal, which is also consistent with the asymptotic result in Equation (7).

Figures 10 and 11 visualize how ${}_0F_1$ and the optimal $\mathcal{L}_{\text{uniform}}$ (given by perfectly uniform encoders) evolve.

Lower bound of $\mathcal{L}_{\text{uniform}}$ estimates. In practice, when $\mathcal{L}_{\text{uniform}}$ calculated using expectation over (a batch of) empirical samples $\{x_i\}_{i=1}^{B}$, $B > 1$, the range in Corollary 1 is indeed valid, since it bounds over all distributions:

$$\hat{\mathcal{L}}^{(1)}_{\text{uniform}} \triangleq \log \frac{1}{B^2} \sum_{i=1}^{B} \sum_{j=1}^{B} e^{-t\|f(x_i)-f(x_j)\|^2} > -2t + \log {}_0F_1(; \frac{m}{2}; t^2). \tag{8}$$

However, often $\mathcal{L}_{\text{uniform}}$ is empirically estimated without considering distances between a vector and itself (e.g., in Figure 6 and in our experiment settings as described in Appendix B):

$$\hat{\mathcal{L}}^{(2)}_{\text{uniform}} \triangleq \log \frac{1}{B(B-1)} \sum_{i=1}^{B} \sum_{j \in \{1,\dots,B\} \setminus \{i\}} e^{-t\|f(x_i)-f(x_j)\|^2}. \tag{9}$$

While both quantities converge to the correct value in the limit, the lower bound is not always true for this one, because it is *not* the expected pairwise Gaussian kernel based on some distribution. Note the following relation:

$$\hat{\mathcal{L}}^{(2)}_{\text{uniform}} = \log \left(\frac{B \cdot \exp(\hat{\mathcal{L}}^{(1)}_{\text{uniform}}) - 1}{B-1} \right).$$

We can derive a valid lower bound using Equation 8: for ${}_0F_1(; \frac{m}{2}; t^2) > \frac{e^{2t}}{B}$,

$$\hat{\mathcal{L}}^{(2)}_{\text{uniform}} > \log \left(\frac{B \cdot \exp(-2t + \log {}_0F_1(; \frac{m}{2}; t^2)) - 1}{B-1} \right) = \log \left(\frac{Be^{-2t} {}_0F_1(; \frac{m}{2}; t^2) - 1}{B-1} \right).$$

Since this approaches fails for cases that ${}_0F_1(; \frac{m}{2}; t^2) \leq \frac{e^{2t}}{B}$, we can combine it with the naive lower bound $-4t$, and have

$$\hat{\mathcal{L}}^{(2)}_{\text{uniform}} > \begin{cases} \max(-4t, \log \left(\frac{Be^{-2t} {}_0F_1(; \frac{m}{2}; t^2) - 1}{B-1} \right)) & \text{if } {}_0F_1(; \frac{m}{2}; t^2) > \frac{e^{2t}}{B} \\ -4t & \text{otherwise.} \end{cases}$$

Non-negative versions of $\mathcal{L}_{\text{uniform}}$ for practical uses. By definition, $\mathcal{L}_{\text{uniform}}$ always non-positive. As shown above, different $\mathcal{L}_{\text{uniform}}$ empirical estimates may admit different lower bounds. However, in our experience, for reasonably large batch sizes, adding an offset of $2t$ often ensures a non-negative loss that is near zero at optimum. When output dimensionality m is low, it might be useful to add an additional offset of $-\log {}_0F_1(; \frac{m}{2}; t^2)$, which can be computed with the help of the SciPy package function `scipy.special.hyp0f1(m/2, t**2)` (Virtanen et al., 2020).

A.2. Proofs and Additional Results for Section 4.2

The following lemma directly follows Theorem 3.3 and Remarks 3.4 (b)(i) of Serfozo (1982). We refer readers to Serfozo (1982) for its proof.

Lemma 3. *Let A be a compact second countable Hausdorff space. Suppose*

1. *$\{\mu_n\}_{n=1}^{\infty}$ is a sequence of finite and positive Borel measures supported on A that converges weak* to some finite and positive Borel measure μ (which is same as vague convergence since A is compact);*
2. *$\{f_n\}_{n=1}^{\infty}$ is a sequence of Borel measurable functions that converges continuously to a Borel measurable f;*
3. *$\{f_n\}_n$ are uniformly bounded over A.*

Then, we have the following convergence:

$$\lim_{n\to\infty} \int_{x\in A} f_n(x)\, \mathrm{d}\mu_n(x) = \int_{x\in A} f(x)\, \mathrm{d}\mu(x).$$

We now recall Theorem 1.

Theorem 1 (Asymptotics of $\mathcal{L}_{\mathsf{contrastive}}$)**.** *For fixed $\tau > 0$, as the number of negative samples $M \to \infty$, the (normalized) contrastive loss converges to*

$$\begin{aligned}
&\lim_{M\to\infty} \mathcal{L}_{\mathsf{contrastive}}(f;\tau,M) - \log M \\
&\quad= \lim_{M\to\infty} \mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\mathsf{pos}} \\ \{x_i^-\}_{i=1}^M \overset{i.i.d.}{\sim} p_{\mathsf{data}}}} \left[-\log \frac{e^{f(x)^\mathsf{T} f(y)/\tau}}{e^{f(x)^\mathsf{T} f(y)/\tau} + \sum_i e^{f(x_i^-)^\mathsf{T} f(y)/\tau}} \right] - \log M \\
&\quad= -\frac{1}{\tau} \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{pos}}} \left[f(x)^\mathsf{T} f(y) \right] + \mathop{\mathbb{E}}_{x\sim p_{\mathsf{data}}} \left[\log \mathop{\mathbb{E}}_{x^-\sim p_{\mathsf{data}}} \left[e^{f(x^-)^\mathsf{T} f(x)/\tau} \right] \right].
\end{aligned} \tag{2}$$

We have the following results:

1. *The first term is minimized iff f is perfectly aligned.*
2. *If perfectly uniform encoders exist, they form the exact minimizers of the second term.*
3. *For the convergence in Equation (2), the absolute deviation from the limit (i.e., the error term) decays in $\mathcal{O}(M^{-1/2})$.*

Proof of Theorem 1. We first show the convergence stated in Equation (2) along with its speed (result 3), and then the relations between the two limiting terms and the alignment and uniformity properties (results 1 and 2).

- **Proof of the convergence in Equation (2) and the $\mathcal{O}(M^{-1/2})$ decay rate of its error term (result 3).**

  Note that for any $x, y \in \mathbb{R}^n$ and $\{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{data}}$, we have

$$\lim_{M\to\infty} \log \left(\frac{1}{M} e^{f(x)^\mathsf{T} f(y)/\tau} + \frac{1}{M} \sum_{i=1}^M e^{f(x_i^-)^\mathsf{T} f(x)/\tau} \right) = \log \mathop{\mathbb{E}}_{x^-\sim p_{\mathsf{data}}} \left[e^{f(x^-)^\mathsf{T} f(x)/\tau} \right] \qquad \text{almost surely,} \tag{10}$$

  by the strong law of large numbers (SLLN) and the Continuous Mapping Theorem.
  Then, we can derive

$$\begin{aligned}
&\lim_{M\to\infty} \mathcal{L}_{\mathsf{contrastive}}(f;\tau,M) - \log M \\
&\quad= \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{pos}}} \left[-f(x)^\mathsf{T} f(y)/\tau \right] + \lim_{M\to\infty} \mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\mathsf{pos}} \\ \{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{data}}}} \left[\log \left(\frac{1}{M} e^{f(x)^\mathsf{T} f(y)/\tau} + \frac{1}{M} \sum_{i=1}^M e^{f(x_i^-)^\mathsf{T} f(x)/\tau} \right) \right] \\
&\quad= \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{pos}}} \left[-f(x)^\mathsf{T} f(y)/\tau \right] + \mathbb{E} \left[\lim_{M\to\infty} \log \left(\frac{1}{M} e^{f(x)^\mathsf{T} f(y)/\tau} + \frac{1}{M} \sum_{i=1}^M e^{f(x_i^-)^\mathsf{T} f(x)/\tau} \right) \right] \\
&\quad= -\frac{1}{\tau} \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathsf{pos}}} \left[f(x)^\mathsf{T} f(y) \right] + \mathop{\mathbb{E}}_{x\sim p_{\mathsf{data}}} \left[\log \mathop{\mathbb{E}}_{x^-\sim p_{\mathsf{data}}} \left[e^{f(x^-)^\mathsf{T} f(x)/\tau} \right] \right],
\end{aligned}$$

  where we justify the switching of expectation and limit by the convergence stated in Equation (10), the boundedness of $e^{u^\mathsf{T} v/\tau}$ (where $u, v \in \mathcal{S}^d, \tau > 0$), and the Dominated Convergence Theorem (DCT).

For convergence speed, we have

$$
\begin{aligned}
&\left|\left(\lim_{M\to\infty}\mathcal{L}_{\mathsf{contrastive}}(f;\tau,M)-\log M\right)-\left(\mathcal{L}_{\mathsf{contrastive}}(f;\tau,M)-\log M\right)\right| \\
&\quad=\left|\mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\mathsf{pos}}\\ \{x_i^-\}_{i=1}^M\overset{\text{i.i.d.}}{\sim}p_{\mathsf{data}}}}\left[\log\mathop{\mathbb{E}}_{x^-\sim p_{\mathsf{data}}}\left[e^{f(x^-)^\top f(x)/\tau}\right]-\log\left(\frac{1}{M}e^{f(x)^\top f(y)/\tau}+\frac{1}{M}\sum_{i=1}^M e^{f(x_i^-)^\top f(x)/\tau}\right)\right]\right| \\
&\quad\le\mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\mathsf{pos}}\\ \{x_i^-\}_{i=1}^M\overset{\text{i.i.d.}}{\sim}p_{\mathsf{data}}}}\left[\left|\log\mathop{\mathbb{E}}_{x^-\sim p_{\mathsf{data}}}\left[e^{f(x^-)^\top f(x)/\tau}\right]-\log\left(\frac{1}{M}e^{f(x)^\top f(y)/\tau}+\frac{1}{M}\sum_{i=1}^M e^{f(x_i^-)^\top f(x)/\tau}\right)\right|\right] \\
&\quad\le e^{1/\tau}\mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\mathsf{pos}}\\ \{x_i^-\}_{i=1}^M\overset{\text{i.i.d.}}{\sim}p_{\mathsf{data}}}}\left[\left|\mathop{\mathbb{E}}_{x^-\sim p_{\mathsf{data}}}\left[e^{f(x^-)^\top f(x)/\tau}\right]-\left(\frac{1}{M}e^{f(x)^\top f(y)/\tau}+\frac{1}{M}\sum_{i=1}^M e^{f(x_i^-)^\top f(x)/\tau}\right)\right|\right] \\
&\quad\le\frac{1}{M}e^{2/\tau}+e^{1/\tau}\mathop{\mathbb{E}}_{x,\{x_i^-\}_{i=1}^M\overset{\text{i.i.d.}}{\sim}p_{\mathsf{data}}}\left[\left|\mathop{\mathbb{E}}_{x^-\sim p_{\mathsf{data}}}\left[e^{f(x^-)^\top f(x)/\tau}\right]-\frac{1}{M}\sum_{i=1}^M e^{f(x_i^-)^\top f(x)/\tau}\right|\right] \\
&\quad=\frac{1}{M}e^{2/\tau}+\mathcal{O}(M^{-1/2}),
\end{aligned}
\tag{11}
$$

where the first inequality follows the Intermediate Value Theorem and the $e^{1/\tau}$ upper bound on the absolute derivative of $\log$ between the two points, and the last equality follows the Berry-Esseen Theorem given the bounded support of $e^{f(x_i^-)^\top f(x)/\tau}$ as following: for i.i.d. random variables Y_i with bounded support $\subset[-a,a]$, zero mean and $\sigma_Y^2\le a^2$ variance, we have

$$
\begin{aligned}
\mathbb{E}\left[\left|\frac{1}{M}\sum_{i=1}^M Y_i\right|\right]&=\frac{\sigma_Y}{\sqrt{M}}\mathbb{E}\left[\left|\frac{1}{\sqrt{M}\sigma_Y}\sum_{i=1}^M Y_i\right|\right] \\
&=\frac{\sigma_Y}{\sqrt{M}}\int_0^{\frac{a\sqrt{M}}{\sigma_Y}}\mathbb{P}\left[\left|\frac{1}{\sqrt{M}\sigma_Y}\sum_{i=1}^M Y_i\right|>x\right]\mathrm{d}x \\
&\le\frac{\sigma_Y}{\sqrt{M}}\int_0^{\frac{a\sqrt{M}}{\sigma_Y}}\mathbb{P}\left[|\mathcal{N}(0,1)|>x\right]+\frac{C_a}{\sqrt{M}}\,\mathrm{d}x && \text{(Berry-Esseen)} \\
&\le\frac{\sigma_Y}{\sqrt{M}}\left(\frac{aC_a}{\sigma_Y}+\int_0^\infty\mathbb{P}\left[|\mathcal{N}(0,1)|>x\right]\mathrm{d}x\right) \\
&=\frac{\sigma_Y}{\sqrt{M}}\left(\frac{aC_a}{\sigma_Y}+\mathbb{E}\left[|\mathcal{N}(0,1)|\right]\right) \\
&\le\frac{C_a}{\sqrt{M}}+\frac{a}{\sqrt{M}}\mathbb{E}\left[|\mathcal{N}(0,1)|\right] \\
&=\mathcal{O}(M^{-1/2}),
\end{aligned}
$$

where the constant C_a only depends on a (which controls both the second and the third moment).

- **Proof of result 1: The first term is minimized iff f is perfectly aligned.**
  Note that for $u,v\in\mathcal{S}^d$,

$$\|u-v\|_2^2=2-2\cdot u^Tv.$$

  Then the result follows directly the definition of perfect alignment, and the existence of perfectly aligned encoders (e.g., an encoder that maps every input to the same output vector).

- **Proof of result 2: If perfectly uniform encoders exist, they form the exact minimizers of the second term.**
  For simplicity, we define the following notation:

Definition. $\forall \mu \in \mathcal{M}(\mathcal{S}^d)$, $u \in \mathcal{S}^d$, we define the continuous and Borel measurable function

$$U_\mu(u) \triangleq \int_{\mathcal{S}^d} e^{u^\top v/\tau} \,\mathrm{d}\mu(v). \tag{12}$$

with its range bounded in $[e^{-1/\tau}, e^{1/\tau}]$.

Then the second term can be equivalently written as

$$\mathop{\mathbb{E}}_{x\sim p_{\mathsf{data}}}\left[\log \mathop{\mathbb{E}}_{x^-\sim p_{\mathsf{data}}}\left[e^{f(x^-)^\top f(x)/\tau}\right]\right] = \mathop{\mathbb{E}}_{x\sim p_{\mathsf{data}}}\left[\log U_{p_{\mathsf{data}}\circ f^{-1}}(f(x))\right],$$

where $p_{\mathsf{data}} \circ f^{-1} \in \mathcal{M}(\mathcal{S}^d)$ is the probability measure of features, i.e., the pushforward measure of p_{data} via f.
We now consider the following relaxed problem, where the minimization is taken over $\mathcal{M}(\mathcal{S}^d)$, all possible Borel probability measures on the hypersphere $\mathcal{S}^d$:

$$\min_{\mu\in\mathcal{M}(\mathcal{S}^d)} \int_{\mathcal{S}^d} \log U_\mu(u) \,\mathrm{d}\mu(u). \tag{13}$$

Our strategy is to show that the unique minimizer of Equation (13) is σ_d, from which the result 2 directly follows. The rest of the proof is structured in three parts.

1. **We show that minimizers of Equation (13) exist, i.e., the above infimum is attained for some $\mu \in \mathcal{M}(\mathcal{S}^d)$.**
   Let $\{\mu_m\}_{m=1}^\infty$ be a sequence in $\mathcal{M}(\mathcal{S}^d)$ such that the infimum of Equation (13) is reached in the limit:

   $$\lim_{m\to\infty} \int_{\mathcal{S}^d} \log U_{\mu_m}(u) \,\mathrm{d}\mu_m(u) = \inf_{\mu\in\mathcal{M}(\mathcal{S}^d)} \int_{\mathcal{S}^d} \log U_\mu(u) \,\mathrm{d}\mu(u).$$

   From the Helly's Selection Theorem, let μ^* denote some weak* cluster point of this sequence. Then μ_m converges weak* to μ^* along a subsequence $m \in \mathcal{N} \in \mathbb{N}$. For simplicity and with a slight abuse of notation, we denote this convergent (sub)sequence of measures by $\{\mu_n\}_{n=1}^\infty$.
   We want to show that μ^* attains the limit (and thus the infimum), i.e.,

   $$\int_{\mathcal{S}^d} \log U_{\mu^*}(u) \,\mathrm{d}\mu^*(u) = \lim_{n\to\infty} \int_{\mathcal{S}^d} \log U_{\mu_n}(u) \,\mathrm{d}\mu_n(u). \tag{14}$$

   In view of Lemma 3, since $\mathcal{S}^d$ is a compact second countable Hausdorff space and $\{\log U_{\mu_n}\}_n$ is uniformly bounded over $\mathcal{S}^d$, it remains to prove that $\{\log U_{\mu_n}\}_n$ is continuously convergent to $\log U_{\mu^*}$.
   Consider any convergent sequence of points $\{x_n\}_{n=1}^\infty \in \mathbb{R}^{d+1}$ s.t. $x_n \to x$ where $x \in \mathcal{S}^d$.
   Let $\delta_n = x_n - x$. By simply expanding U_{μ_n} and μ_{μ^*}, we have

   $$e^{-\|\delta_n\|/\tau} U_{\mu_n}(x) \le U_{\mu_n}(x_n) \le e^{\|\delta_n\|/\tau} U_{\mu_n}(x).$$

   Since both the upper and the lower bound converge to $U_{\mu^*}(x)$ (by the weak * convergence of $\{\mu_n\}_n$ to μ^*), $U_{\mu_n}(x_n)$ must as well. We have proved the continuous convergence of $\{\log U_{\mu_n}\}_n$ to $\log U_{\mu^*}$.
   Therefore, the limit in Equation (14) holds. The infimum is thus attained at μ^*:

   $$\lim_{n\to\infty} \int_u \log U_{\mu_n}(u) \,\mathrm{d}\mu_n = \int_u \log U_{\mu^*}(u) \,\mathrm{d}\mu^*.$$

2. **We show that U_{μ^*} is constant μ^*-almost surely for any minimizer μ^* of Equation (13).**
   Let μ^* be any solution of Equation (13):

   $$\mu^* \in \mathop{\arg\min}_{\mu\in\mathcal{M}(\mathcal{S}^d)} \int_u \log U_\mu(u) \,\mathrm{d}\mu.$$

   Consider the Borel sets where μ^* has positive measure: $\mathcal{T} \triangleq \{T \in \mathcal{B}(\mathcal{S}^d) \colon \mu^*(T) > 0\}$. For any $T \in \mathcal{T}$, let μ_T^* denote the conditional distribution of μ^* on T, i.e., $\forall A \in \mathcal{B}(\mathcal{S}^d)$,

   $$\mu_T^*(A) = \frac{\mu^*(A \cap T)}{\mu^*(T)}.$$

Note that for any such $T \in \mathcal{T}$, the mixture $(1-\alpha)\mu^* + \alpha\mu_T^*$ is a valid probability distribution (i.e., in $\mathcal{M}(\mathcal{S}^d)$) for $\alpha \in (-\mu^*(T), 1)$, an open interval containing 0.
By the first variation, we must have

$$\begin{aligned}
0 &= \frac{\partial}{\partial\alpha}\int_{\mathcal{S}^d} \log U_{(1-\alpha)\mu^*+\alpha\mu_T^*}(u)\,\mathrm{d}((1-\alpha)\mu^*+\alpha\mu_T^*)(u)\bigg|_{\alpha=0} \\
&= \frac{\partial}{\partial\alpha}(1-\alpha)\int_{\mathcal{S}^d} \log U_{(1-\alpha)\mu^*+\alpha\mu_T^*}(u)\,\mathrm{d}\mu^*(u)\bigg|_{\alpha=0} + \frac{\partial}{\partial\alpha}\alpha\int_{\mathcal{S}^d} \log U_{(1-\alpha)\mu^*+\alpha\mu_T^*}(u)\,\mathrm{d}\mu_T^*(u)\bigg|_{\alpha=0} \\
&= -\int_{\mathcal{S}^d} \log U_{(1-\alpha)\mu^*+\alpha\mu_T^*}(u)\,\mathrm{d}\mu^*(u)\bigg|_{\alpha=0} + \frac{\partial}{\partial\alpha}\int_{\mathcal{S}^d} \log U_{(1-\alpha)\mu^*+\alpha\mu_T^*}(u)\,\mathrm{d}\mu^*(u)\bigg|_{\alpha=0} \\
&\qquad + \int_{\mathcal{S}^d} \log U_{(1-\alpha)\mu^*+\alpha\mu_T^*}(u)\,\mathrm{d}\mu_T^*(u)\bigg|_{\alpha=0} + 0\cdot\frac{\partial}{\partial\alpha}\int_{\mathcal{S}^d} \log U_{(1-\alpha)\mu^*+\alpha\mu_T^*}(u)\,\mathrm{d}\mu_T^*(u)\bigg|_{\alpha=0} \\
&= -\int_{\mathcal{S}^d} \log U_{\mu^*}(u)\,\mathrm{d}\mu^*(u) + \int_{\mathcal{S}^d} \frac{U_{\mu_T^*}(u) - U_{\mu^*}(u)}{U_{\mu^*}(u)}\,\mathrm{d}\mu^*(u) \\
&\qquad + \int_{\mathcal{S}^d} \log U_{\mu^*}(u)\,\mathrm{d}\mu_T^*(u) + 0\cdot\int_{\mathcal{S}^d} \frac{U_{\mu_T^*}(u) - U_{\mu^*}(u)}{U_{\mu^*}(u)}\,\mathrm{d}\mu_T^*(u) \\
&= \int_{\mathcal{S}^d} \frac{U_{\mu_T^*}(u)}{U_{\mu^*}(u)}\,\mathrm{d}\mu^*(u) + \int_{\mathcal{S}^d} \log U_{\mu^*}(u)\,\mathrm{d}(\mu_T^* - \mu^*)(u) - 1, \qquad (15)
\end{aligned}$$

where the Leibniz rule along with the boundedness of U_{μ^*} and $U_{\mu_{T_n}^*}$ together justify the exchanges of integration and differentiation.
Let $\{T_n\}_{n=1}^\infty$ be a sequence of sets in $\mathcal{T}$ such that

$$\lim_{n\to\infty}\int_{\mathcal{S}^d} U_{\mu^*}(u)\,\mathrm{d}\mu_{T_n}^*(u) = \sup_{T\in\mathcal{T}}\int_{\mathcal{S}^d} U_{\mu^*}(u)\,\mathrm{d}\mu_T^*(u) \triangleq U^*,$$

where the supremum must exist since U_{μ^*} is bounded above.
Because U_{μ^*} is a continuous and Borel measurable function, we have $\{u\colon U_{\mu^*}(u) > U^*\} \in \mathcal{B}(\mathcal{S}^d)$ and thus

$$\begin{aligned}
&\mu^*(\{u\colon U_{\mu^*}(u) > U^*\}) = 0, \\
&\mu_{T_n}^*(\{u\colon U_{\mu^*}(u) > U^*\}) = 0, \qquad && \forall n = 1, 2, \ldots,
\end{aligned}$$

otherwise $\{u\colon U_{\mu^*}(u) > U^*\} \in \mathcal{T}$, contradicting the definition of U^* as the supremum.
Asymptotically, U_{μ^*} is constant $\mu_{T_n}^*$-almost surely:

$$\begin{aligned}
&\int_{\mathcal{S}^d}\left|U_{\mu^*}(u) - \int_{\mathcal{S}^d} U_{\mu^*}(u')\,\mathrm{d}\mu_{T_n}^*(u')\right|\mathrm{d}\mu_{T_n}^*(u) \\
&\qquad = 2\int_{\mathcal{S}^d}\max\left(0,\ U_{\mu^*}(u) - \int_{\mathcal{S}^d} U_{\mu^*}(u')\,\mathrm{d}\mu_{T_n}^*(u')\right)\mathrm{d}\mu_{T_n}^*(u) \\
&\qquad \le 2(U^* - \int_{\mathcal{S}^d} U_{\mu^*}(u)\,\mathrm{d}\mu_{T_n}^*(u)) \\
&\qquad \to 0, && \text{as } n\to\infty,
\end{aligned}$$

where the inequality follows the boundedness of U_{μ^*} and that $\mu_{T_n}^*(\{u\colon U_{\mu^*}(u) > U^*\}) = 0$.
Therefore, given the continuity of $\log$ and the boundedness of U_{μ^*}, we have

$$\lim_{n\to\infty}\int_{\mathcal{S}^d} \log U_{\mu^*}(u)\,\mathrm{d}\mu_{T_n}^* = \log U^*.$$

Equation (15) gives that $\forall n = 1, 2, \ldots,$

$$\begin{aligned}
1 &= \int_{\mathcal{S}^d} \frac{U_{\mu^*_{T_n}}(u)}{U_{\mu^*}(u)} \,\mathrm{d}\mu^* + \int_{\mathcal{S}^d} \log U_{\mu^*}(u) \,\mathrm{d}(\mu^*_{T_n} - \mu^*) \\
&\geq \frac{1}{U^*} \int_{\mathcal{S}^d} U_{\mu^*_{T_n}}(u) \,\mathrm{d}\mu^*(u) + \int_{\mathcal{S}^d} \log U_{\mu^*}(u) \,\mathrm{d}\mu^*_{T_n} - \int_{\mathcal{S}^d} \log U_{\mu^*}(u) \,\mathrm{d}\mu^* \\
&= \frac{1}{U^*} \int_{\mathcal{S}^d} U_{\mu^*}(u) \,\mathrm{d}\mu^*_{T_n}(u) + \int_{\mathcal{S}^d} \log U_{\mu^*}(u) \,\mathrm{d}\mu^*_{T_n} - \int_{\mathcal{S}^d} \log U_{\mu^*}(u) \,\mathrm{d}\mu^*,
\end{aligned}$$

where the inequality follows the boundedness of $\frac{U_{\mu^*_{T_n}}}{U_{\mu^*}}$ and that $\mu^*(\{u \colon U_{\mu^*}(u) > U^*\}) = 0$.
Taking the limit of $n \to \infty$ on both sides, we have

$$\begin{aligned}
1 = \lim_{n\to\infty} 1 &\geq \frac{1}{U^*} \lim_{n\to\infty} \int_{\mathcal{S}^d} U_{\mu^*}(u) \,\mathrm{d}\mu^*_{T_n}(u) + \lim_{n\to\infty} \int_{\mathcal{S}^d} \log U_{\mu^*}(u) \,\mathrm{d}\mu^*_{T_n}(u) - \int_{\mathcal{S}^d} \log U_{\mu^*}(u) \,\mathrm{d}\mu^*(u) \\
&= 1 + \log U^* - \int_{\mathcal{S}^d} \log U_{\mu^*}(u) \,\mathrm{d}\mu^*(u) \\
&\geq 1 + \log U^* - \log \int_{\mathcal{S}^d} U_{\mu^*}(u) \,\mathrm{d}\mu^*(u) \\
&\geq 1,
\end{aligned}$$

where the last inequality holds because the supremum taken over $\mathcal{T} \supset \{\mathcal{S}^d\}$.
Since $1 = 1$, all inequalities must be equalities. In particular,

$$\int_{\mathcal{S}^d} \log U_{\mu^*}(u) \,\mathrm{d}\mu^*(u) = \log \int_{\mathcal{S}^d} U_{\mu^*}(u) \,\mathrm{d}\mu^*(u).$$

That is, for any solution μ^* of Equation (13), U_{μ^*} must be constant μ^*-almost surely.

3. **We show that σ_d is the unique minimizer of the relaxed problem in Equation (13).**
Let $S \subset \mathcal{M}(\mathcal{S}^d)$ be the set of measures where the above property holds:

$$S \triangleq \left\{\mu \in \mathcal{M}(\mathcal{S}^d) \colon U_\mu \text{ is constant } \mu\text{-almost surely}\right\}.$$

The problem in Equation (13) is thus equivalent to minimizing over S:

$$\begin{aligned}
\operatorname*{arg\,min}_{\mu\in\mathcal{M}(\mathcal{S}^d)} \int_{\mathcal{S}^d} \log U_\mu(u) \,\mathrm{d}\mu(u) &= \operatorname*{arg\,min}_{\mu\in S} \int_{\mathcal{S}^d} \log U_\mu(u) \,\mathrm{d}\mu(u) \\
&= \operatorname*{arg\,min}_{\mu\in S} \log \int_{\mathcal{S}^d} U_\mu(u) \,\mathrm{d}\mu(u) \\
&= \operatorname*{arg\,min}_{\mu\in S} \log \int_{\mathcal{S}^d} \int_{\mathcal{S}^d} e^{u^\intercal v/\tau} \,\mathrm{d}\mu(v) \,\mathrm{d}\mu(u) \\
&= \operatorname*{arg\,min}_{\mu\in S} \left(\frac{1}{\tau} + \log \int_{\mathcal{S}^d} \int_{\mathcal{S}^d} e^{-\frac{1}{2\tau}\|u-v\|^2} \,\mathrm{d}\mu(v) \,\mathrm{d}\mu(u)\right) \\
&= \operatorname*{arg\,min}_{\mu\in S} \int_{\mathcal{S}^d} \int_{\mathcal{S}^d} G_{\frac{1}{2\tau}}(u, v) \,\mathrm{d}\mu(v) \,\mathrm{d}\mu(u).
\end{aligned}$$

By Proposition 1 and $\tau > 0$, we know that the uniform distribution σ_d is the unique solution to

$$\operatorname*{arg\,min}_{\mu\in\mathcal{M}(\mathcal{S}^d)} \int_{\mathcal{S}^d} \int_{\mathcal{S}^d} G_{\frac{1}{2\tau}}(u, v) \,\mathrm{d}\mu(v) \,\mathrm{d}\mu(u). \tag{16}$$

Since $\sigma_d \in S$, it must also be the unique solution to Equation (13).

Finally, if perfectly uniform encoders exist, σ_d is realizable, and they are the exact encoders that realize it. Hence, in such cases, they are the exact minimizers of

$$\min_f \mathop{\mathbb{E}}_{x\sim p_{\mathsf{data}}}\left[\log \mathop{\mathbb{E}}_{x^-\sim p_{\mathsf{data}}}\left[e^{f(x^-)^\mathsf{T} f(x)/\tau}\right]\right].$$

□

Relation between Theorem 1, $\mathcal{L}_{\mathsf{align}}$ and $\mathcal{L}_{\mathsf{uniform}}$. The first term of Equation (2) is equivalent with $\mathcal{L}_{\mathsf{align}}$ when $\alpha = 2$, up to a constant and a scaling. In the above proof, we showed that the second term favors uniformity, via the feature distribution that minimizes the pairwise Gaussian kernel (see Equation (16)):

$$\arg\min_{\mu\in\mathcal{M}(\mathcal{S}^d)} \int_{\mathcal{S}^d}\int_{\mathcal{S}^d} G_{\frac{1}{2\tau}}(u,v)\,\mathrm{d}\mu(v)\,\mathrm{d}\mu(u), \tag{17}$$

which can be alternatively viewed as the relaxed problem of optimizing for the uniformity loss $\mathcal{L}_{\mathsf{uniform}}$:

$$\arg\min_f \mathcal{L}_{\mathsf{uniform}}(f;\frac{1}{2\tau}) = \arg\min_f \mathbb{E}_{x,y\overset{\text{i.i.d.}}{\sim} p_{\mathsf{data}}}\left[G_{\frac{1}{2\tau}}(f(x),f(y))\right]. \tag{18}$$

The relaxation comes from the observation that Equation (17) minimizes over all feature distributions on $\mathcal{S}^d$, while Equation (18) only considers the realizable ones.

Relation between Equation (13) and minimizing average pairwise Gaussian potential (i.e., minimizing $\mathcal{L}_{\mathsf{uniform}}$). In view of the Proposition 1 and the proof of Theorem 1, we know that the uniform distribution σ_d is the unique minimizer of both of the following problems:

$$\{\sigma_d\} = \min_{\mu\in\mathcal{M}(\mathcal{S}^d)} \log\int_{\mathcal{S}^d}\int_{\mathcal{S}^d} e^{u^\mathsf{T} v/\tau}\,\mathrm{d}\mu(v)\,\mathrm{d}\mu(u),$$
$$\{\sigma_d\} = \min_{\mu\in\mathcal{M}(\mathcal{S}^d)} \int_{\mathcal{S}^d}\log\int_{\mathcal{S}^d} e^{u^\mathsf{T} v/\tau}\,\mathrm{d}\mu(v)\,\mathrm{d}\mu(u).$$

So pushing the $\log$ inside the outer integral doesn't change the solution. However, if we push the $\log$ all the way inside the inner integral, the problem becomes equivalent with minimizing the norm of the mean, i.e.,

$$\min_{\mu\in\mathcal{M}(\mathcal{S}^d)} \mathbb{E}_{U\sim\mu}[U]^\mathsf{T}\,\mathbb{E}_{U\sim\mu}[U],$$

which is minimized for any distribution with mean being the all-zeros vector 0, e.g., $\frac{1}{2}\delta_u + \frac{1}{2}\delta_{-u}$ for any $u\in\mathcal{S}^d$ (where δ_u is the Dirac delta distribution at u s.t. $\delta_u(S) = \mathbb{1}_S(u), \forall S\in\mathcal{B}(\mathcal{S}^d)$). Therefore, the location of the $\log$ is important.

Theorem 2 (Single negative sample)**.** *If perfectly aligned and uniform encoders exist, they form the exact minimizers of the contrastive loss $\mathcal{L}_{\mathsf{contrastive}}(f;\tau,M)$ for fixed $\tau>0$ and $M=1$.*

Proof of Theorem 2. Since $M=1$, we have

$$\begin{aligned}
\mathcal{L}_{\mathsf{contrastive}}(f;\tau,1) &= \mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\mathsf{pos}}\\ x^-\sim p_{\mathsf{data}}}}\left[-\frac{1}{\tau}f(x)^\mathsf{T} f(y) + \log\left(e^{f(x)^\mathsf{T} f(y)/\tau} + e^{f(x^-)^\mathsf{T} f(x)/\tau}\right)\right] \\
&\geq \mathop{\mathbb{E}}_{\substack{x\sim p_{\mathsf{data}}\\ x^-\sim p_{\mathsf{data}}}}\left[-\frac{1}{\tau} + \log\left(e^{1/\tau} + e^{f(x^-)^\mathsf{T} f(x)/\tau}\right)\right] \qquad (19)\\
&\geq -\frac{1}{\tau} + \min_{\mu\in\mathcal{M}(\mathcal{S}^d)} \int_{\mathcal{S}^d}\int_{\mathcal{S}^d} \log\left(e^{1/\tau} + e^{u^\mathsf{T} v/\tau}\right)\mathrm{d}\mu(u)\,\mathrm{d}\mu(v) \qquad (20)\\
&= -\frac{1}{\tau} + \min_{\mu\in\mathcal{M}(\mathcal{S}^d)} \int_{\mathcal{S}^d}\int_{\mathcal{S}^d} \log\left(e^{1/\tau} + e^{(2-\|u-v\|_2^2)/(2\tau)}\right)\mathrm{d}\mu(u)\,\mathrm{d}\mu(v).
\end{aligned}$$

By the definition of perfect alignment, the equality in Equation (19) is satisfied iff f is perfectly aligned.

Consider the function $f\colon (0,4] \to \mathbb{R}_+$ defined as

$$f(t) = \log(e^{\frac{1}{\tau}} + e^{\frac{2-t}{2\tau}}).$$

It has the following properties:

- $-f'(t) = \frac{1}{2\tau}\frac{e^{-\frac{t}{2\tau}}}{1+e^{-\frac{t}{2\tau}}} = \frac{1}{2\tau}(1-(1+e^{-\frac{t}{2\tau}})^{-1})$ is strictly completely monotone on $(0,+\infty)$: $\forall t \in (0,+\infty)$,

$$\frac{1}{2\tau}(1-(1+e^{-\frac{t}{2\tau}})^{-1}) > 0$$
$$(-1)^n \frac{\mathrm{d}^n}{\mathrm{d}t^n}\frac{1}{2\tau}(1-(1+e^{-\frac{t}{2\tau}})^{-1}) = \frac{n!}{(2\tau)^{n+1}}(1+e^{-\frac{t}{2\tau}})^{-(n+1)} > 0, \qquad n = 1,2,\dots.$$

- f is bounded on $(0,4]$.

In view of Lemma 2, we have that the equality in Equation (20) is satisfied iff the feature distribution induced by f (i.e., the pushforward measure $p_{\mathsf{data}} \circ f^{-1}$) is σ_d, that is, in other words, f is perfectly uniform.

Therefore,

$$\mathcal{L}_{\mathsf{contrastive}}(f;\tau,1) \geq -\frac{1}{\tau} + \int_{\mathcal{S}^d}\int_{\mathcal{S}^d} \log\left(e^{1/\tau} + e^{u^\intercal v/\tau}\right) \mathrm{d}\sigma_d(u)\,\mathrm{d}\sigma_d(v) = \text{constant independent of } f,$$

where equality is satisfied iff f is perfectly aligned and uniform. This concludes the proof. □

Difference between conditions of Theorems 1 and 2. We remark that the statement in Theorem 2 is weaker than the previous Theorem 1. Theorem 2 is conditioned on the existence perfectly aligned and uniform encoders. It only shows that $\mathcal{L}_{\mathsf{contrastive}}(f;\tau,M=1)$ favors alignment under the condition that perfect uniformity is realizable, and vice versa. In Theorem 1, $\mathcal{L}_{\mathsf{contrastive}}$ decomposes into two terms, each favoring alignment and uniformity. Therefore, the decomposition in Theorem 1 is exempof t from this constraint.

B. Experiment Details

All experiments are performed on 1-4 NVIDIA Titan Xp, Titan X PASCAL, Titan RTX, or 2080 Ti GPUs.

B.1. CIFAR-10, STL-10 and NYU-DEPTH-V2 Experiments

For CIFAR-10, STL-10 and NYU-DEPTH-V2 experiments, we use the following settings, unless otherwise stated in Tables 8 and 9 below:

- Standard data augmentation procedures are used for generating positive pairs, including resizing, cropping, horizontal flipping, color jittering, and random grayscale conversion. This follows prior empirical work in contrastive representation learning (Wu et al., 2018; Tian et al., 2019; Hjelm et al., 2018; Bachman et al., 2019).
- Neural network architectures follow the corresponding experiments on these datasets in Tian et al. (2019). For NYU-DEPTH-V2 evaluation, the architecture of the depth prediction CNN is described in Table 6.
- We use minibatch stochastic gradient descent (SGD) with 0.9 momentum and 0.0001 weight decay.
- We use linearly scaled learning rate (0.12 per 256 batch size) (Goyal et al., 2017).
  - CIFAR-10 and STL-10: Optimization is done over 200 epochs, with learning rate decayed by a factor of 0.1 at epochs 155, 170, and 185.
  - NYU-DEPTH-V2: Optimization is done over 400 epochs, with learning rate decayed by a factor of 0.1 at epochs 310, 340, and 370.
- Encoders are optimized over the training split. For evaluation, we freeze the encoder, and train classifiers / depth predictors on the training set samples, and test on the validation split.

Operator	Input Spatial Shape	Input #Channel	Kernel Size	Stride	Padding	Output Spatial Shape	Output #Channel
Input	$[h_{\text{in}}, w_{\text{in}}]$	c_{in}	—	—	—	$[h_{\text{in}}, w_{\text{in}}]$	c_{in}
Conv. Transpose + BN + ReLU	$[h_{\text{in}}, w_{\text{in}}]$	c_{in}	3	2	1	$[2h_{\text{in}}, 2w_{\text{in}}]$	$\lfloor c_{\text{in}}/2 \rfloor$
Conv. Transpose + BN + ReLU	$[2h_{\text{in}}, 2w_{\text{in}}]$	$\lfloor c_{\text{in}}/2 \rfloor$	3	2	1	$[4h_{\text{in}}, 4w_{\text{in}}]$	$\lfloor c_{\text{in}}/4 \rfloor$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Conv. Transpose + BN + ReLU	$[h_{\text{out}}/2, w_{\text{out}}/2]$	$\lfloor c_{\text{in}}/2^{n-1} \rfloor$	3	2	1	$[h_{\text{out}}, w_{\text{out}}]$	$\lfloor c_{\text{in}}/2^{n} \rfloor$
Conv.	$[h_{\text{out}}, w_{\text{out}}]$	$\lfloor c_{\text{in}}/2^{n} \rfloor$	3	1	1	$[h_{\text{out}}, w_{\text{out}}]$	1

Table 6: NYU-DEPTH-V2 CNN depth predictor architecture. Each Conv. Transpose+BN+ReLU block increases the spatial shape by a factor of 2, where BN denotes Batch Normalization (Ioffe & Szegedy, 2015). A sequence of such blocks computes a tensor of the correct spatial shape, from an input containing intermediate activations of a CNN encoder (which downsamples the input RGB image by a power of 2). A final convolution at the end computes the single-channel depth prediction.

- CIFAR-10 and STL-10: We use standard train-val split. Linear classifiers are trained with Adam (Kingma & Ba, 2014) over 100 epochs, with $\beta_1 = 0.5, \beta_2 = 0.999, \epsilon = 10^{-8}$, 128 batch size, and an initial learning rate of 0.001, decayed by a factor of 0.2 at epochs 60 and 80.
- NYU-DEPTH-V2: We use the train-val split on the 1449 labeled images from Nathan Silberman & Fergus (2012). Depth predictors are trained with Adam (Kingma & Ba, 2014) over 120 epochs, with $\beta_1 = 0.5, \beta_2 = 0.999, \epsilon = 10^{-8}$, 128 batch size, and an initial learning rate of 0.003, decayed by a factor of 0.2 at epochs 70, 90, 100, and 110.

At each SGD iteration, a minibatch of K positive pairs is sampled $\{(x_i, y_i)\}_{i=1}^K$, and the three losses for this minibatch are calculated as following:

- $\mathcal{L}_{\text{contrastive}}$: For each x_i, the sample contrastive loss is taken with the positive being y_i, and the negatives being $\{y_j\}_{j \neq i}$. For each y_i, the sample loss is computed similarly. The minibatch loss is calculated by aggregating these $2K$ terms:

$$\frac{1}{2K} \sum_{i=1}^{K} \log \frac{e^{f(x_i)^\intercal f(y_i)/\tau}}{\sum_{j=1}^{K} e^{f(x_i)^\intercal f(y_j)/\tau}} + \frac{1}{2K} \sum_{i=1}^{K} \log \frac{e^{f(x_i)^\intercal f(y_i)/\tau}}{\sum_{j=1}^{K} e^{f(x_j)^\intercal f(y_i)/\tau}}.$$

  This calculation follows empirical practices and is similar to Oord et al. (2018); Hénaff et al. (2019), and *end-to-end* in He et al. (2019).

- $\mathcal{L}_{\text{align}}$: The minibatch alignment loss is straightforwardly computed as

$$\frac{1}{K} \sum_{i=1}^{K} \|f(x_i) - f(y_i)\|_2^\alpha.$$

- $\mathcal{L}_{\text{uniform}}$: The minibatch uniform loss is calculated by considering each pair of $\{x_i\}_i$ and $\{y_i\}_i$:

$$\frac{1}{2} \log \left(\frac{2}{K(K-1)} \sum_{i \neq j} e^{-t\|f(x_i) - f(x_j)\|_2^2} \right) + \frac{1}{2} \log \left(\frac{2}{K(K-1)} \sum_{i \neq j} e^{-t\|f(y_i) - f(y_j)\|_2^2} \right).$$

Tables 8 and 9 below describe the full specifications of all 306 STL-10 and 64 NYU-DEPTH-V2 encoders. These experiment results are visualized in main paper Figure 5, showing a clear connection between representation quality and $\mathcal{L}_{\text{align}}$ & $\mathcal{L}_{\text{uniform}}$ metrics.

B.2. IMAGENET and IMAGENET-100 with Momentum Contrast (MoCo) Variants

MoCo and MoCo v2 with $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$. At each SGD iteration, let

- K be the minibatch size,
- $\{f(x_i)_i\}_{i=1}^K$ be the batched query features encoded by the current up-to-date encoder f (i.e., q in Algorithm 1 of He et al. (2019)),

IMAGENET-100 Classes									
n02869837	n01749939	n02488291	n02107142	n13037406	n02091831	n04517823	n04589890	n03062245	n01773797
n01735189	n07831146	n07753275	n03085013	n04485082	n02105505	n01983481	n02788148	n03530642	n04435653
n02086910	n02859443	n13040303	n03594734	n02085620	n02099849	n01558993	n04493381	n02109047	n04111531
n02877765	n04429376	n02009229	n01978455	n02106550	n01820546	n01692333	n07714571	n02974003	n02114855
n03785016	n03764736	n03775546	n02087046	n07836838	n04099969	n04592741	n03891251	n02701002	n03379051
n02259212	n07715103	n03947888	n04026417	n02326432	n03637318	n01980166	n02113799	n02086240	n03903868
n02483362	n04127249	n02089973	n03017168	n02093428	n02804414	n02396427	n04418357	n02172182	n01729322
n02113978	n03787032	n02089867	n02119022	n03777754	n04238763	n02231487	n03032252	n02138441	n02104029
n03837869	n03494278	n04136333	n03794056	n03492542	n02018207	n04067472	n03930630	n03584829	n02123045
n04229816	n02100583	n03642806	n04336792	n03259280	n02116738	n02108089	n03424325	n01855672	n02090622

Table 7: 100 randomly selected IMAGENET classes forming the IMAGENET-100 subset. These classes are the same as the ones used by Tian et al. (2019).

- $\{f_{\mathsf{EMA}}(y_i)\}_{i=1}^K$ be the batched key features encoded by the exponential moving average encoder f_{EMA} (i.e., $\mathtt{k}$ in Algorithm 1 of He et al. (2019)),
- $\{\mathtt{queue}_j\}_{j=1}^N$ be the feature queue, where N is the queue size.

$\mathcal{L}_{\mathsf{align}}$ and $\mathcal{L}_{\mathsf{uniform}}$ for this minibatch are calculated as following:

- $\mathcal{L}_{\mathsf{align}}$: The minibatch alignment loss is computed as disparity between features from the two encoders:

$$\frac{1}{K}\sum_{i=1}^{K}\|f(x_i)-f_{\mathsf{EMA}}(y_i)\|_2^{\alpha}.$$

- $\mathcal{L}_{\mathsf{uniform}}$: We experiment with two forms of $\mathcal{L}_{\mathsf{uniform}}$:
  1. Only computing pairwise distance between $\{f(x_i)\}_i$ and $\{\mathtt{queue}_j\}_j$:

$$\log\left(\frac{1}{NK}\sum_{i=1}^{K}\sum_{j=1}^{N}e^{-t\left\|f(x_i)-\mathtt{queue}_j\right\|_2^2}\right). \tag{21}$$

  2. Also computing pairwise distance inside $\{f(x_i)\}_i$:

$$\log\left(\frac{2}{2NK+K(K-1)}\sum_{i=1}^{K}\sum_{j=1}^{N}e^{-t\left\|f(x_i)-\mathtt{queue}_j\right\|_2^2}+\frac{2}{2NK+K(K-1)}\sum_{i\neq j}e^{-t\|f(x_i)-f(x_j)\|_2^2}\right). \tag{22}$$

B.2.1. IMAGENET-100 WITH MOCO

IMAGENET-100 details. We use the same IMAGENET-100 sampled by Tian et al. (2019), containing the 100 randomly selected classes listed in Table 7.

MoCo settings. Our MoCo experiment settings below mostly follow He et al. (2019) and the unofficial implementation by Tian (2019), because the official implementation was not released at the time of performing these analyses:

- Standard data augmentation procedures are used for generating positive pairs, including resizing, cropping, horizontal flipping, color jittering, and random grayscale conversion, following Tian (2019).
- Encoder architecture is ResNet50 (He et al., 2016).
- We use minibatch stochastic gradient descent (SGD) with 128 batch size, 0.03 initial learning rate, 0.9 momentum and 0.0001 weight decay.
- Optimization is done over 240 epochs, with learning rate decayed by a factor of 0.1 at epochs 120, 160, and 200.
- We use 0.999 exponential moving average factor, following He et al. (2019).

- For evaluation, we freeze the encoder, and train a linear classifier on the training set samples, and test on the validation split. Linear classifiers are trained with minibatch SGD over 60 epochs, with 256 batch size, and an initial learning rate of 10, decayed by a factor of 0.2 at epochs 30, 40, and 50.

Table 10 below describes the full specifications of all 45 IMAGENET-100 encoders. These experiment results are visualized in main paper Figure 9a, showing a clear connection between representation quality and $\mathcal{L}_{\text{align}}$ & $\mathcal{L}_{\text{uniform}}$ metrics.

B.2.2. IMAGENET WITH MOCO V2

MoCo v2 settings. Our MoCo v2 experiment settings directly follow Chen et al. (2020b) and the official implementation (Chen et al., 2020c):

- Standard data augmentation procedures are used for generating positive pairs, including resizing, cropping, horizontal flipping, color jittering, random grayscale conversion, and random Gaussian blurring, following Chen et al. (2020c).
- Encoder architecture is ResNet50 (He et al., 2016).
- We use minibatch stochastic gradient descent (SGD) with 256 batch size, 0.03 initial learning rate, 0.9 momentum and 0.0001 weight decay.
- Optimization is done over 200 epochs, with learning rate decayed by a factor of 0.1 at epochs 120 and 160.
- We use 0.999 exponential moving average factor, 65536 queue size, 128 feature dimensions.
- For evaluation, we freeze the encoder, and train a linear classifier on the training set samples, and test on the validation split. Linear classifiers are trained with minibatch SGD over 100 epochs, with 256 batch size, and an initial learning rate of 30, decayed by a factor of 0.1 at epochs 60 and 80.

Unlike the MoCo experiments on IMAGENET-100, which were based on unofficial implementations for reasons stated in Sec. B.2.1, the MoCo v2 experiments on full IMAGENET were based on the official implementation by Chen et al. (2020c). We provide a reference implementation that can fully reproduce the results in Table 5 at `https://github.com/SsnL/moco_align_uniform`, where we also provide a model checkpoint (trained using $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$) of 67.694% validation top1 accuracy.

B.3. BOOKCORPUS with Quick-Thought Vectors Variants

BOOKCORPUS details. Since the original BOOKCORPUS dataset (Zhu et al., 2015) is not distributed anymore, we use the unofficial code by Kobayashi (2019) to recreate our copy. Our copy ended up containing 52,799,513 training sentences and 50,000 validation sentences, compared to the original copy used by Quick-Thought Vectors (Logeswaran & Lee, 2018), which contains 45,786,400 training sentences and 50,000 validation sentences.

Quick-Thought Vectors with $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$. With Quick-Thought Vectors, the positive pairs are the neighboring sentences. At each optimization iteration, let

- $\{x_i\}_{i=1}^K$ be the K *consecutive* sentences forming this minibatch, where K be the minibatch size,
- f and g be the two RNN sentence encoders.

The original Quick-Thought Vectors (Logeswaran & Lee, 2018) does not $l2$-normalize on encoder outputs during training the encoder. Here we describe the calculation of $\mathcal{L}_{\text{contrastive}}$, $\mathcal{L}_{\text{align}}$, and $\mathcal{L}_{\text{uniform}}$ for $l2$-normalized encoders, in our modified Quick-Thought Vectors method. Note that this does not affect evaluation since features are $l2$-normalized before using in downstream tasks, following the original Quick-Thought Vectors (Logeswaran & Lee, 2018). For a minibatch, these losses are calculated as following:

- $\mathcal{L}_{\text{contrastive}}$ with temperature:

$$\begin{aligned}
&\frac{1}{K}\,\texttt{cross_entropy}(\texttt{softmax}(\{f(x_1)^\top g(x_j)\}_j), \{0, 1, 0, \dots, 0\}) \\
&\quad + \frac{1}{K}\sum_{i=2}^{K-1} \texttt{cross_entropy}(\texttt{softmax}(\{f(x_i)^\top g(x_j)\}_j), \{\underbrace{0, \dots, 0}_{(i-2)\ 0\text{'s}}, \frac{1}{2}, 0, \frac{1}{2}, \underbrace{0, \dots, 0}_{(K-i-1)\ 0\text{'s}}\}) + \\
&\quad + \frac{1}{K}\,\texttt{cross_entropy}(\texttt{softmax}(\{f(x_K)^\top g(x_j)\}_j), \{0, \dots, 1, 0\}).
\end{aligned}$$

This is almost identical with the original contrastive loss used by Quick-Thought Vectors, except that this does not additionally manually masks out the entries $f(x_i)^\intercal g(x_i)$ with zeros, which is unnecessary with $l2$-normalization.

- $\mathcal{L}_{\mathsf{align}}$: The minibatch alignment loss is computed as disparity between features from the two encoders encoding neighboring sentences (assuming $K >= 2$):

$$\frac{1}{K}\|f(x_1) - g(x_2)\|_2^\alpha + \frac{1}{2K}\sum_{i=2}^{K-2}(\|f(x_{i-1}) - g(x_i)\|_2^\alpha + \|f(x_i) - g(x_{i+1})\|_2^\alpha) + \frac{1}{K}\|f(x_{K-1}) - g(x_K)\|_2^\alpha.$$

- $\mathcal{L}_{\mathsf{uniform}}$: We combine the uniformity losses for each of f and g by summing them (instead of averaging since f and g are two different encoders):

$$\frac{2}{K(K-1)}\sum_{i\neq j} e^{-t\|f(x_i)-f(x_j)\|_2^2} + \frac{2}{K(K-1)}\sum_{i\neq j} e^{-t\|g(x_i)-g(x_j)\|_2^2}.$$

Our experiment settings below mostly the official implementation by Logeswaran & Lee (2018):

- Sentence encoder architecture is bi-directional Gated Recurrent Unit (GRU) (Cho et al., 2014) with inputs from a 620-dimensional word embedding trained jointly from scratch.
- We use Adam (Kingma & Ba, 2014) with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$, 400 batch size, 0.0005 constant learning rate, and 0.5 gradient norm clipping.
- Optimization is done during 1 epoch over the training data.
- For evaluation on a binary classification task, we freeze the encoder, and fit a logistic classifier with $l2$ regularization on the encoder outputs. A 10-fold cross validation is performed to determine the regularization strength among $\{1, 2^{-1}, \ldots, 2^{-8}\}$, following Kiros et al. (2015) and Logeswaran & Lee (2018). The classifier is finally tested on the validation split.

Table 11 below describes the full specifications of all 108 BOOKCORPUS encoders along with 6 settings that lead to training instability (i.e., `NaN` occurring). These experiment results are visualized in main paper Figure 9b, showing a clear connection between representation quality and $\mathcal{L}_{\mathsf{align}}$ & $\mathcal{L}_{\mathsf{uniform}}$ metrics. For the unnormalized encoders, the features are normalized before calculated $\mathcal{L}_{\mathsf{align}}$ and $\mathcal{L}_{\mathsf{uniform}}$ metrics, since they are nonetheless still normalized before being used in downstream tasks (Logeswaran & Lee, 2018).

Table 8: Experiment specifications for all 306 STL-10 encoders. We report the encoder representation quality measured by accuracy of linear and k-nearest neighbor (k-NN) with $k = 5$ classifiers on either encoder outputs or fc7 activations, via both a 5-fold cross validation of the training set and the held out validation set.
For encoder initialization, "rand" refers to standard network initialization, and symbols denote finetuning from a pretrained encoder, obtained via the experiment row marked with the same symbol. Initial learning rates (LRs) are usually either fixed as 0.12 or computed via a linear scaling (0.12 per 256 batch size). Dimensionality (abbreviated as "Dim.") shows the ambient dimension of the output features, i.e., they live on the unit hypersphere of one less dimension. The last three rows show encoders that are used to initialize finetuning, but are not part of the 285 encoders plotted in main paper Figure 3, due to their unusual batch size of 786. Their accuracy and $\mathcal{L}_{\mathsf{align}}$ & $\mathcal{L}_{\mathsf{uniform}}$ metrics follow the same trend shown in Figure 5a.

	Losses								Training Set 5-Fold Cross Val. Accuracy ↑				Validation Set Accuracy ↑			
	$\mathcal{L}_{\mathsf{contrastive}}$	$\mathcal{L}_{\mathsf{align}}$	$\mathcal{L}_{\mathsf{uniform}}$	Init.	Epochs	Batch Size	Initial LR	Dim.	Output + Linear	Output + 5-NN	fc7 + Linear	fc7 + 5-NN	Output + Linear	Output + 5-NN	fc7 + Linear	fc7 + 5-NN
	—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	2	0.0009375	128	—	—	—	—	19.31%	22.56%	47.58%	35.30%
	—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	3	0.00140625	128	—	—	—	—	43.97%	42.89%	56.89%	47.63%
	—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	4	0.001875	128	—	—	—	—	53.96%	52.89%	62.86%	55.06%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	200	16	0.0075	128	—	—	—	—	70.46%	70.54%	75.54%	69.63%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	16	0.0075	128	—	—	—	—	69.59%	70.04%	76.23%	68.38%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	16	0.0075	128	—	—	—	—	74.68%	74.34%	79.06%	73.68%
	—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	16	0.0075	128	—	—	—	—	74.75%	73.00%	77.84%	71.70%
	—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	16	0.0075	128	—	—	—	—	73.93%	74.09%	79.25%	73.38%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	16	0.12	128	—	—	—	—	67.30%	66.36%	71.53%	66.38%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	16	0.12	128	—	—	—	—	71.93%	71.24%	75.49%	69.89%
	—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	16	0.12	128	—	—	—	—	71.85%	70.21%	74.65%	69.88%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	200	32	0.015	128	—	—	—	—	71.80%	72.04%	77.29%	70.74%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	32	0.015	128	—	—	—	—	73.39%	73.39%	79.43%	73.85%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	32	0.015	128	—	—	—	—	78.04%	76.60%	82.23%	76.04%
	—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	32	0.015	128	—	—	—	—	78.71%	76.45%	81.66%	76.25%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	32	0.12	128	—	—	—	—	70.43%	69.66%	74.95%	69.69%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	32	0.12	128	—	—	—	—	75.40%	73.70%	78.56%	73.21%
	—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	32	0.12	128	—	—	—	—	75.83%	73.95%	78.48%	73.55%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	64	0.03	128	—	—	—	—	74.59%	74.48%	80.64%	75.52%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	64	0.03	128	—	—	—	—	79.25%	77.84%	82.84%	76.53%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	64	0.12	128	—	—	—	—	77.80%	75.75%	81.45%	75.49%
	—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	64	0.12	128	—	—	—	—	78.66%	76.19%	81.40%	75.30%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	64	0.03	512	—	—	—	—	80.44%	78.05%	83.04%	77.29%
	—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	64	0.03	1024	—	—	—	—	81.48%	78.49%	82.88%	77.11%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	64	0.03	1024	—	—	—	—	80.81%	77.80%	83.18%	77.15%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	200	128	0.06	128	—	—	—	—	73.14%	73.73%	79.90%	72.58%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	128	0.06	128	—	—	—	—	75.26%	74.88%	80.98%	75.36%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	128	0.06	128	—	—	—	—	79.55%	78.09%	83.39%	76.96%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	200	128	0.12	128	—	—	—	—	73.11%	73.84%	78.44%	72.11%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	128	0.12	128	—	—	—	—	75.65%	74.80%	80.74%	74.58%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.687)$	—	—	rand	200	128	0.12	128	—	—	—	—	74.13%	73.14%	79.81%	74.10%
	—	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	128	0.12	128	—	—	—	—	79.05%	76.61%	81.77%	73.83%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	128	0.12	128	—	—	—	—	79.74%	77.78%	82.70%	75.23%
	—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	128	0.12	128	—	—	—	—	80.19%	77.91%	82.75%	75.91%
	—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	256	0.12	64	—	—	—	—	78.40%	78.26%	83.46%	76.25%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	200	256	0.12	128	—	—	—	—	75.23%	75.86%	80.64%	73.56%
♡	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	256	0.12	128	—	—	—	—	76.09%	75.81%	81.49%	75.52%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.6)$	—	—	rand	200	256	0.12	128	—	—	—	—	75.61%	74.56%	81.09%	75.36%
	—	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	256	0.12	128	—	—	—	—	79.94%	77.95%	82.66%	73.65%
	—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	256	0.12	128	—	—	—	—	80.54%	78.55%	83.54%	76.81%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	256	0.12	128	—	—	—	—	80.76%	78.57%	84.24%	76.60%
△	—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	256	0.12	128	—	—	—	—	81.29%	78.49%	83.55%	74.08%
	—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	256	0.12	256	—	—	—	—	81.79%	79.13%	84.11%	76.60%
	—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	256	0.12	256	—	—	—	—	81.48%	79.61%	83.86%	76.79%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	256	0.12	256	—	—	—	—	80.95%	78.74%	83.69%	77.11%
	—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	256	0.12	512	—	—	—	—	81.33%	78.76%	83.81%	76.88%
	—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	360	0.16875	8192	—	—	—	—	82.49%	78.96%	83.86%	76.68%
	—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	512	0.24	4096	—	—	—	—	82.34%	78.84%	84.06%	75.74%
	$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	200	768	0.36	2	—	—	—	—	29.46%	25.50%	59.95%	52.83%

$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	768	0.36	2	—	—	—	—	30.66%	25.39%	48.61%	42.49%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	2	—	—	—	—	27.85%	26.04%	49.29%	43.10%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	2	—	—	—	—	29.05%	23.94%	45.39%	38.48%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	200	768	0.36	3	—	—	—	—	39.59%	39.66%	63.24%	56.64%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	768	0.36	3	—	—	—	—	42.29%	39.70%	68.35%	59.82%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	3	—	—	—	—	41.10%	39.63%	65.64%	56.04%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	3	—	—	—	—	41.40%	41.45%	67.88%	58.78%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	200	768	0.36	4	—	—	—	—	46.94%	47.08%	64.35%	58.10%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	768	0.36	4	—	—	—	—	53.39%	55.41%	73.93%	67.89%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	4	—	—	—	—	47.19%	51.69%	70.00%	62.36%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	200	768	0.36	16	—	—	—	—	64.20%	68.73%	75.66%	69.55%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	768	0.36	16	—	—	—	—	71.93%	73.54%	80.53%	74.66%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	16	—	—	—	—	65.41%	70.41%	77.18%	70.55%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	16	—	—	—	—	70.25%	74.99%	81.59%	74.52%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	32	—	—	—	—	70.30%	73.50%	79.63%	72.21%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	32	—	—	—	—	73.65%	76.93%	82.81%	75.19%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2.5)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	32	—	—	—	—	73.71%	77.40%	82.93%	75.86%
—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	64	—	—	—	—	77.33%	78.35%	84.00%	76.63%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	64	—	—	—	—	77.94%	78.23%	83.51%	76.59%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.005)$	—	—	rand	200	768	0.36	128	67.88%	70.15%	74.64%	68.19%	68.14%	71.13%	75.14%	68.88%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.01)$	—	—	rand	200	768	0.36	128	69.63%	70.62%	75.68%	68.99%	69.86%	70.98%	76.13%	69.65%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	200	768	0.36	128	75.01%	75.11%	80.93%	73.20%	75.46%	75.58%	81.34%	73.93%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.08)$	—	—	rand	200	768	0.36	128	76.12%	76.06%	81.72%	73.95%	76.58%	76.79%	81.81%	74.43%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.09)$	—	—	rand	200	768	0.36	128	77.15%	77.15%	82.52%	73.96%	77.74%	77.46%	83.23%	74.81%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	—	—	rand	200	768	0.36	128	77.55%	77.40%	82.93%	74.29%	77.83%	77.81%	83.39%	75.19%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.11)$	—	—	rand	200	768	0.36	128	78.48%	78.20%	83.29%	74.99%	79.01%	78.73%	83.73%	75.60%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.125)$	—	—	rand	200	768	0.36	128	79.05%	78.06%	83.30%	74.53%	79.59%	78.55%	84.09%	75.55%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.13)$	—	—	rand	200	768	0.36	128	79.46%	78.55%	**83.98%**	75.16%	79.80%	78.60%	**84.45%**	75.98%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.15)$	—	—	rand	200	768	0.36	128	79.81%	78.47%	83.62%	74.64%	80.16%	78.99%	84.19%	75.20%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.16)$	—	—	rand	200	768	0.36	128	79.54%	78.38%	83.35%	74.42%	80.04%	78.68%	83.88%	75.06%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.175)$	—	—	rand	200	768	0.36	128	79.74%	78.20%	83.56%	74.80%	80.29%	78.49%	83.96%	75.81%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.19)$	—	—	rand	200	768	0.36	128	80.14%	78.30%	83.52%	75.39%	80.46%	78.75%	83.89%	76.33%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.2)$	—	—	rand	200	768	0.36	128	79.64%	77.80%	83.37%	75.07%	79.99%	77.96%	83.73%	75.98%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.25)$	—	—	rand	200	768	0.36	128	79.27%	77.24%	82.70%	75.33%	79.50%	77.49%	83.10%	76.31%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.3)$	—	—	rand	200	768	0.36	128	78.79%	77.01%	82.58%	75.16%	78.98%	77.18%	82.84%	75.74%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	200	768	0.36	128	76.57%	75.30%	81.18%	75.30%	76.66%	75.61%	81.61%	75.71%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.75)$	—	—	rand	200	768	0.36	128	74.59%	73.41%	79.72%	74.27%	74.63%	73.52%	80.18%	75.01%
$\mathcal{L}_{\mathsf{c}}(\tau{=}1)$	—	—	rand	200	768	0.36	128	72.88%	72.14%	79.16%	74.08%	73.00%	72.31%	79.54%	74.61%
$\mathcal{L}_{\mathsf{c}}(\tau{=}2)$	—	—	rand	200	768	0.36	128	67.79%	67.15%	77.04%	71.65%	67.13%	66.77%	77.35%	71.84%
★ $\mathcal{L}_{\mathsf{c}}(\tau{=}2.5)$	—	—	rand	200	768	0.36	128	66.11%	65.30%	75.80%	70.59%	65.33%	65.30%	76.31%	70.93%
$\mathcal{L}_{\mathsf{c}}(\tau{=}5)$	—	—	rand	200	768	0.36	128	55.56%	55.74%	70.29%	65.25%	55.75%	55.83%	70.75%	65.58%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	rand	200	768	0.36	128	75.13%	75.59%	81.52%	73.55%	75.59%	76.26%	82.10%	74.33%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	rand	200	768	0.36	128	77.76%	78.02%	83.28%	74.56%	78.04%	78.44%	83.73%	75.33%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	rand	200	768	0.36	128	74.86%	73.92%	80.16%	74.55%	74.96%	73.93%	80.63%	75.13%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	rand	200	768	0.36	128	74.69%	74.10%	80.53%	74.77%	74.80%	74.28%	80.91%	75.31%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	rand	200	768	0.36	128	73.31%	72.84%	79.82%	73.73%	73.54%	72.94%	80.26%	74.58%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.77%	75.98%	81.50%	73.48%	76.11%	76.45%	82.08%	74.00%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.17%	77.61%	83.04%	74.54%	78.64%	78.10%	83.26%	75.45%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	77.73%	76.23%	81.96%	75.10%	77.98%	76.60%	82.38%	75.45%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.93%	75.55%	81.45%	73.18%	76.13%	76.00%	81.95%	74.11%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	77.98%	77.18%	82.77%	74.12%	78.38%	77.79%	83.51%	74.99%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.69%	76.99%	82.57%	75.12%	79.03%	77.38%	82.93%	75.46%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.71%	75.22%	80.94%	72.80%	76.05%	75.60%	81.56%	73.46%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.38%	77.85%	82.87%	74.36%	78.84%	78.54%	83.10%	74.73%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	79.72%	77.94%	83.03%	75.32%	80.04%	78.24%	83.28%	75.66%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	76.19%	75.62%	81.15%	73.09%	76.90%	76.21%	81.61%	74.48%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.59%	78.02%	83.18%	74.63%	78.68%	78.48%	83.76%	75.49%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.25%	78.32%	83.35%	74.26%	80.43%	78.71%	83.76%	75.44%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	76.31%	75.78%	81.59%	72.79%	76.69%	76.33%	82.23%	73.63%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.55%	77.94%	83.21%	74.67%	79.03%	78.45%	83.75%	75.71%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	79.93%	78.25%	82.92%	75.22%	80.30%	78.54%	83.34%	76.04%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	**80.84%**	78.87%	83.72%	75.56%	81.06%	79.05%	84.14%	76.48%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	$2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	77.49%	76.15%	80.99%	74.41%	78.09%	76.83%	81.63%	75.11%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	rand	200	768	0.36	128	75.40%	75.53%	81.53%	73.91%	75.74%	76.19%	82.00%	74.63%

$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	rand	200	768	0.36	128	77.70%	77.70%	83.39%	75.27%	78.06%	78.26%	83.93%	76.21%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	rand	200	768	0.36	128	73.86%	73.12%	80.08%	74.54%	74.05%	73.18%	80.53%	75.14%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	76.12%	76.22%	81.75%	73.68%	76.46%	76.75%	82.36%	74.44%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.40%	78.01%	83.39%	75.21%	78.83%	78.30%	83.74%	75.84%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.35%	76.49%	82.02%	75.60%	78.60%	77.18%	82.65%	76.19%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	76.59%	75.74%	81.48%	73.59%	77.20%	76.43%	82.03%	74.36%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.85%	77.43%	82.98%	74.87%	79.20%	77.95%	83.29%	75.60%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	79.53%	77.56%	82.84%	75.19%	79.71%	77.95%	83.19%	76.08%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	77.07%	76.49%	81.78%	73.10%	77.44%	76.98%	82.33%	73.85%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.55%	78.04%	83.20%	74.30%	78.91%	78.38%	83.81%	75.18%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.47%	78.36%	83.42%	75.82%	80.88%	78.51%	83.83%	76.65%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	76.30%	76.43%	81.72%	73.35%	76.56%	77.11%	82.11%	74.00%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.71%	78.00%	83.35%	74.46%	79.29%	78.44%	83.81%	75.45%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.51%	78.99%	83.57%	75.47%	80.95%	79.44%	83.98%	76.45%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.48%	76.10%	81.47%	72.97%	75.80%	76.86%	82.06%	73.81%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	77.78%	78.07%	83.23%	74.51%	78.38%	78.46%	83.89%	75.49%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.04%	76.18%	81.89%	73.67%	78.43%	76.44%	82.33%	74.44%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	rand	200	768	0.36	128	10.00%	10.36%	11.07%	14.20%	10.00%	9.40%	12.53%	14.27%
—	$0.9875 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.025 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	9.90%	11.04%	13.72%	10.00%	10.94%	13.03%	13.64%
—	$0.975 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.05 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	9.98%	10.65%	14.29%	10.00%	9.75%	12.11%	14.77%
—	$0.9 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	10.08%	10.10%	13.62%	10.00%	9.95%	10.00%	13.49%
—	$0.95 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	10.51%	10.15%	13.27%	10.00%	9.85%	10.00%	11.99%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	9.93%	10.39%	14.38%	10.00%	10.26%	10.00%	14.03%
—	$0.56 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.12 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.93%	75.10%	80.88%	74.87%	75.99%	75.41%	81.40%	75.66%
—	$0.88 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.12 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	10.13%	10.00%	12.89%	10.00%	11.18%	10.03%	12.43%
—	$0.9375 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.125 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	10.52%	10.42%	13.71%	10.00%	9.14%	10.05%	14.26%
—	$0.57 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.14 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	76.35%	75.51%	81.07%	75.27%	76.55%	75.86%	81.69%	75.70%
—	$0.86 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.14 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	9.07%	10.33%	14.24%	10.00%	9.91%	10.73%	15.08%
—	$0.855 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.145 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	10.67%	10.30%	14.11%	10.00%	9.35%	11.70%	13.30%
—	$0.85 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.15 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	10.17%	10.00%	12.97%	10.00%	10.05%	10.00%	13.16%
—	$0.925 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.15 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	9.79%	10.10%	13.11%	10.00%	9.73%	10.11%	12.91%
—	$0.845 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.155 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	74.56%	74.06%	80.10%	74.93%	74.99%	74.39%	80.44%	75.83%
—	$0.58 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.16 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	77.03%	76.34%	81.25%	75.26%	77.33%	76.76%	81.80%	75.89%
—	$0.84 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.16 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	74.49%	74.03%	80.30%	74.72%	74.73%	74.10%	80.70%	75.13%
—	$0.9125 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.175 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	9.41%	10.39%	13.64%	10.00%	10.14%	10.10%	14.14%
—	$0.59 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.18 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	77.25%	76.38%	81.39%	75.41%	77.65%	77.06%	81.68%	76.19%
—	$0.82 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.18 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	76.09%	75.10%	80.99%	75.63%	76.45%	75.48%	81.45%	76.48%
—	$0.91 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.18 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.11%	74.63%	80.50%	75.28%	75.40%	75.04%	80.85%	75.83%
—	$0.9075 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.185 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.29%	74.83%	80.64%	75.04%	75.69%	75.41%	80.93%	75.65%
—	$0.905 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.19 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.69%	74.61%	80.80%	74.98%	75.99%	74.95%	81.21%	75.59%
—	$0.9025 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.195 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.81%	74.93%	80.75%	74.66%	76.06%	75.29%	81.16%	75.14%
—	$0.8 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	76.52%	75.96%	81.05%	75.38%	76.75%	76.24%	81.29%	75.83%
—	$0.9 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.92%	75.02%	80.85%	75.36%	76.15%	75.29%	81.15%	76.24%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.14%	74.29%	80.39%	74.76%	75.46%	74.44%	80.64%	75.34%
—	$0.7 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.3 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.61%	77.00%	82.14%	75.73%	78.94%	77.50%	82.26%	76.34%
—	$0.6 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	79.36%	77.80%	82.63%	75.55%	79.60%	77.93%	82.86%	76.63%
—	$0.8 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	79.24%	77.52%	82.44%	75.23%	79.65%	77.89%	82.69%	75.71%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.45%	77.09%	82.30%	75.38%	78.85%	77.53%	82.86%	76.02%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.03%	78.47%	83.12%	75.14%	80.39%	78.70%	83.56%	75.70%
—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	79.72%	77.30%	82.69%	75.44%	79.96%	77.55%	83.35%	76.14%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	79.09%	77.50%	82.80%	75.46%	79.27%	77.96%	83.10%	76.45%
—	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.23%	78.67%	83.49%	75.61%	80.45%	78.83%	84.01%	76.61%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.37%	78.82%	83.05%	75.54%	80.48%	79.11%	83.33%	76.50%
—	$0.7 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.29%	78.16%	83.40%	75.59%	80.59%	78.66%	83.83%	76.24%
—	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.7 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.16%	78.91%	83.39%	**76.21%**	80.58%	79.51%	83.78%	77.03%
—	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	74.67%	78.15%	82.53%	75.83%	75.13%	78.63%	83.03%	76.45%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.59%	78.73%	83.73%	76.05%	81.08%	79.10%	84.04%	76.88%
—	$0.6 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.29%	78.74%	83.53%	75.75%	80.65%	78.89%	83.89%	76.86%
—	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.9 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	69.77%	75.72%	80.55%	73.38%	70.29%	76.13%	80.88%	74.14%
—	$0.08 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.92 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	67.65%	73.97%	79.35%	71.86%	68.04%	74.90%	79.84%	72.50%
—	$0.96 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.92 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.74%	78.71%	83.49%	76.14%	81.08%	79.26%	83.95%	77.26%
—	$0.06 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.94 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	66.88%	73.81%	79.21%	72.32%	67.46%	74.68%	79.56%	73.09%
—	$0.97 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.94 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.28%	78.45%	83.51%	75.68%	80.63%	78.63%	83.83%	76.33%
—	$0.04 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.96 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	63.89%	70.80%	76.33%	69.55%	64.21%	71.49%	77.10%	70.38%

—	$0.98 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.96 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.76%	78.69%	83.97%	75.63%	81.15%	78.89%	84.43%	76.78%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.975 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	79.94%	78.45%	83.34%	75.23%	80.44%	78.86%	83.65%	75.83%
—	$0.02 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.98 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	56.39%	63.06%	69.48%	62.85%	56.78%	63.90%	69.80%	63.82%
—	$0.99 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.98 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.24%	78.90%	83.34%	74.89%	80.45%	79.40%	83.76%	75.55%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.98 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.29%	78.64%	83.46%	75.23%	80.77%	78.84%	83.96%	75.90%
—	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	20.62%	15.96%	24.52%	16.13%	20.50%	16.14%	24.64%	16.24%
—	$0.0025 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	36.14%	33.19%	46.82%	35.22%	36.28%	33.76%	47.04%	36.05%
—	$0.005 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	48.38%	49.74%	59.67%	49.55%	48.69%	50.41%	59.81%	50.40%
—	$0.0125 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	51.31%	57.94%	64.95%	57.49%	51.80%	58.75%	65.40%	58.01%
—	$0.025 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	46.13%	51.81%	58.51%	51.30%	46.61%	52.65%	59.03%	51.99%
—	$0.025 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	57.34%	62.50%	69.09%	61.76%	57.89%	63.43%	69.58%	62.51%
—	$0.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	70.80%	75.24%	80.59%	72.59%	71.40%	75.54%	81.20%	73.36%
—	$0.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	76.14%	78.45%	82.97%	75.90%	76.83%	78.88%	83.51%	76.74%
—	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	76.72%	78.01%	83.26%	75.61%	77.30%	78.43%	83.79%	76.25%
—	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	78.71%	77.76%	83.13%	75.42%	79.36%	78.01%	83.64%	76.24%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.41%	**79.18%**	83.85%	75.54%	80.03%	79.35%	84.20%	76.84%
—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.54%	78.84%	83.61%	75.26%	80.89%	79.29%	84.23%	76.28%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.32%	78.90%	83.48%	74.97%	80.76%	79.23%	83.75%	76.15%
—	$1.025 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.37%	78.69%	83.48%	75.78%	80.74%	79.06%	84.00%	76.56%
—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	80.50%	78.41%	83.54%	75.89%	80.84%	78.65%	83.95%	76.56%
—	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	75.37%	73.62%	78.88%	71.55%	75.78%	73.83%	79.15%	72.35%
—	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	72.69%	75.62%	80.67%	73.49%	73.14%	75.99%	81.49%	74.20%
—	$0.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	70.61%	73.50%	78.53%	71.85%	71.03%	74.10%	79.13%	72.50%
—	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	67.35%	70.98%	76.84%	69.13%	67.69%	71.64%	77.40%	69.91%
—	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	64.43%	68.89%	74.24%	68.15%	65.01%	69.34%	74.70%	68.80%
—	$0.0875 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.825 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	63.38%	68.83%	73.56%	67.33%	64.05%	69.76%	73.91%	68.14%
—	$0.075 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.85 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	63.02%	69.32%	74.49%	68.22%	63.44%	69.91%	75.05%	69.06%
—	$0.0625 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.875 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	58.73%	64.37%	70.93%	63.74%	59.23%	65.14%	71.54%	64.69%
—	$0.05 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.9 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	57.61%	64.13%	69.13%	63.09%	58.03%	65.09%	69.43%	64.09%
—	$0.025 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.95 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	50.89%	57.70%	63.93%	57.83%	51.46%	58.39%	64.45%	58.34%
—	$0.0125 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.975 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	44.71%	50.89%	57.75%	51.21%	45.14%	51.99%	57.98%	52.11%
—	—	$2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	21.99%	19.46%	28.94%	20.10%	21.91%	19.75%	29.65%	20.76%
—	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	63.63%	70.70%	75.85%	69.41%	64.14%	71.43%	76.50%	69.99%
—	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	66.52%	72.89%	77.66%	70.98%	67.16%	73.52%	78.19%	71.79%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$2.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$3 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%
—	—	$5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	19.61%	14.29%	21.70%	14.97%	19.64%	14.19%	21.61%	15.58%
—	$0.05 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	50.49%	55.71%	61.45%	55.15%	50.91%	56.71%	61.58%	56.19%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	128	10.00%	10.00%	10.00%	10.01%	10.00%	10.00%	10.00%	10.00%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	256	—	—	—	—	82.10%	79.45%	84.15%	77.10%
—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	256	—	—	—	—	81.53%	79.03%	83.54%	76.35%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	256	—	—	—	—	81.33%	79.06%	84.03%	75.89%
—	$0.025 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	512	—	—	—	—	75.76%	72.75%	78.29%	71.04%
—	$0.375 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	512	—	—	—	—	82.33%	79.18%	83.91%	76.44%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	512	—	—	—	—	82.55%	79.64%	84.29%	75.74%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	512	—	—	—	—	82.04%	78.79%	83.98%	76.50%
—	$0.025 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	1024	—	—	—	—	76.39%	72.45%	78.23%	70.59%
—	$0.05 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	1024	—	—	—	—	79.68%	75.43%	80.81%	73.45%
—	$0.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	1024	—	—	—	—	**83.03%**	79.63%	84.15%	76.10%
—	$0.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	1024	—	—	—	—	82.85%	79.44%	83.91%	75.35%
—	$0.375 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	1024	—	—	—	—	82.63%	79.33%	83.69%	76.09%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	1024	—	—	—	—	82.85%	**79.75%**	83.85%	76.81%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	1024	—	—	—	—	81.89%	79.09%	84.03%	75.51%
—	$0.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	768	0.36	1536	—	—	—	—	82.93%	79.55%	84.00%	75.81%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	1024	0.48	512	—	—	—	—	82.20%	79.36%	83.69%	75.73%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	1024	0.48	512	—	—	—	—	81.66%	79.03%	83.88%	75.49%
—	$0.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	1024	0.48	1024	—	—	—	—	82.40%	78.98%	83.34%	75.85%
—	$0.375 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	1024	0.48	1024	—	—	—	—	82.74%	79.48%	83.70%	76.59%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	200	1024	0.48	1024	—	—	—	—	82.51%	79.11%	83.46%	74.94%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	△	12	256	0.12	128	—	—	—	—	79.31%	77.45%	83.34%	76.60%
—	$5e-05 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	♣	12	256	0.12	128	—	—	—	—	64.11%	62.45%	77.96%	68.56%
—	$0.0005 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	♣	12	256	0.12	128	—	—	—	—	63.90%	62.40%	77.81%	68.55%

—	$0.005 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	♣	12	256	0.12	128	—	—	—	—	61.53%	61.66%	76.83%	66.68%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	♣	12	256	0.12	128	—	—	—	—	10.36%	23.01%	49.19%	39.39%
—	—	$0.01 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	♣	12	256	0.12	128	—	—	—	—	44.75%	41.79%	55.59%	38.59%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.01 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	♣	12	256	0.12	128	—	—	—	—	10.03%	32.81%	57.95%	41.53%
—	—	$0.1 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	♣	12	256	0.12	128	—	—	—	—	54.78%	54.05%	65.77%	50.13%
—	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♣	12	256	0.12	128	—	—	—	—	55.74%	52.03%	63.90%	50.59%
—	$0.005 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♣	12	256	0.12	128	—	—	—	—	57.85%	55.18%	65.64%	53.33%
—	$0.05 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♣	12	256	0.12	128	—	—	—	—	68.46%	66.07%	72.88%	64.65%
—	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♡	12	256	0.12	128	—	—	—	—	77.63%	76.65%	81.75%	75.95%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♠	12	256	0.12	128	—	—	—	—	70.00%	68.21%	74.15%	66.77%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♡	12	256	0.12	128	—	—	—	—	77.73%	76.33%	81.61%	76.00%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♢	12	256	0.12	128	—	—	—	—	74.23%	72.89%	79.01%	71.46%
—	$0.625 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♢	12	256	0.12	128	—	—	—	—	74.40%	72.84%	79.29%	71.41%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♡	12	256	0.12	128	—	—	—	—	76.48%	75.86%	81.04%	75.43%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♢	12	256	0.12	128	—	—	—	—	73.13%	72.24%	78.33%	71.15%
—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♡	12	256	0.12	128	—	—	—	—	76.80%	75.75%	81.00%	75.11%
—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♢	12	256	0.12	128	—	—	—	—	73.11%	71.73%	78.23%	71.79%
—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♠	12	256	0.12	128	—	—	—	—	69.10%	67.21%	74.19%	66.25%
—	$1.875 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	♢	12	256	0.12	128	—	—	—	—	72.63%	71.08%	77.79%	70.98%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	□	12	768	0.36	128	—	—	—	—	75.34%	74.00%	81.09%	73.23%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	★	12	768	0.36	128	—	—	—	—	65.60%	64.25%	70.73%	64.79%
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	★	12	768	0.36	128	—	—	—	—	69.64%	67.70%	74.89%	68.74%
$0.25 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	★	12	768	0.36	128	—	—	—	—	69.11%	68.34%	74.30%	69.30%
$0.05 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	★	12	768	0.36	128	—	—	—	—	70.43%	69.70%	76.08%	71.31%
$0.025 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	□	12	768	0.36	128	—	—	—	—	80.27%	78.65%	83.93%	77.00%
$0.025 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	★	12	768	0.36	128	—	—	—	—	70.00%	68.74%	76.24%	71.86%
$0.01 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	□	12	768	0.36	128	—	—	—	—	80.46%	78.88%	83.64%	**77.38%**
$0.01 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	★	12	768	0.36	128	—	—	—	—	68.13%	67.38%	75.63%	71.28%
—	$0.00025 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	★	12	768	0.36	128	—	—	—	—	65.94%	64.33%	75.14%	70.90%
—	$0.0005 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	★	12	768	0.36	128	—	—	—	—	64.88%	63.18%	74.78%	70.88%
—	$0.0005 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	★	12	768	0.36	128	—	—	—	—	64.89%	63.53%	74.76%	70.89%
—	$0.001 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	★	12	768	0.36	128	—	—	—	—	62.65%	61.93%	74.31%	70.36%
—	$0.0025 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	★	12	768	0.36	128	—	—	—	—	59.18%	60.09%	72.98%	69.41%
—	$0.005 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	★	12	768	0.36	128	—	—	—	—	52.18%	55.06%	71.40%	67.10%
—	$0.005 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	★	12	768	0.36	128	—	—	—	—	52.86%	55.95%	71.63%	67.76%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—	★	12	768	0.36	128	—	—	—	—	10.00%	17.42%	36.69%	34.94%
—	—	$0.0001 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	60.32%	59.49%	70.65%	64.70%
—	—	$0.0005 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	44.34%	43.41%	61.06%	53.97%
—	$0.0005 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.0005 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	66.14%	66.13%	75.29%	70.20%
—	—	$0.001 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	41.61%	40.73%	56.91%	48.24%
—	$0.001 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.001 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	66.23%	66.55%	75.16%	70.25%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.001 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	10.00%	17.79%	35.06%	34.11%
—	$0.002 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.002 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	66.35%	67.07%	74.50%	70.33%
—	—	$0.01 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	44.64%	41.55%	50.75%	42.90%
—	$0.01 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.01 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	71.54%	70.71%	75.45%	70.43%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.01 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	10.00%	18.05%	32.93%	31.53%
—	$0.03 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.02 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	72.13%	71.86%	76.33%	71.78%
—	$0.025 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.025 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	73.40%	72.58%	76.44%	72.09%
—	$0.0375 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.025 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	72.54%	71.56%	76.14%	71.89%
—	$0.05 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.05 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	73.94%	72.63%	77.05%	72.36%
—	—	$0.1 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	54.51%	48.40%	60.60%	49.00%
—	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	73.30%	72.21%	76.54%	72.13%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	67.45%	67.03%	74.04%	68.73%
—	$0.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.25 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	73.09%	71.66%	76.80%	71.16%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	72.18%	71.56%	76.38%	70.93%
—	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	39.45%	35.56%	47.18%	35.60%
—	$0.0005 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	43.58%	38.19%	49.38%	38.64%
—	$0.005 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	50.10%	47.36%	56.66%	48.73%
—	$0.05 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	65.65%	66.15%	71.48%	66.10%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	70.34%	70.04%	74.88%	68.76%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	70.84%	69.88%	75.61%	69.34%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	66.83%	65.59%	72.09%	65.30%
—	$1.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	65.18%	62.32%	69.77%	62.31%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	63.21%	61.86%	68.66%	60.80%

	—	$2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	★	12	768	0.36	128	—	—	—	—	61.93%	60.78%	68.54%	60.18%
♢	$\mathcal{L}_{\mathsf{c}}(\tau{=}1)$	—	—	rand	200	786	0.12	128	—	—	—	—	70.35%	70.11%	80.41%	73.15%
♣	$\mathcal{L}_{\mathsf{c}}(\tau{=}2)$	—	—	rand	200	786	0.12	128	—	—	—	—	64.19%	62.38%	78.11%	68.77%
♠	$\mathcal{L}_{\mathsf{c}}(\tau{=}3)$	—	—	rand	200	786	0.12	128	—	—	—	—	55.04%	53.94%	74.95%	64.04%

Table 9: Experiment specifications for all 64 NYU-DEPTH-V2 encoders. We report the encoder representation quality measured by mean squared error (MSE) of a CNN depth predictor trained on conv5 or conv4 activations, via both a 5-fold cross validation of the training set and the held out validation set.
All encoders in this table use standard network initialization (denoted as "rand"). Dimensionality (abbreviated as "Dim.") shows the ambient dimension of the output features, i.e., they live on the unit hypersphere of one less dimension.

Losses			Init.	Epochs	Batch Size	Initial LR	Dim.	Training Set 5-Fold Cross Val. MSE ↓		Validation Set MSE ↓	
$\mathcal{L}_{\mathsf{contrastive}}$	$\mathcal{L}_{\mathsf{align}}$	$\mathcal{L}_{\mathsf{uniform}}$						conv5	conv4	conv5	conv4
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7405	0.7979	0.7378	0.7969
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.25)$	—	—	rand	400	128	0.06	128	0.7188	0.7747	0.7259	0.7761
—	$4.375 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.8039	0.8297	0.8032	0.8281
—	$3.625 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7290	0.7775	0.7303	0.7749
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7121	0.7689	0.7191	0.7725
—	$3.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7270	0.7741	0.7260	0.7772
$\mathcal{L}_{\mathsf{c}}(\tau{=}4)$	—	—	rand	400	128	0.06	128	0.7592	0.8195	0.7598	0.8175
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.3333 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7165	0.7697	0.7215	0.7693
—	$2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7300	0.7669	0.7226	0.7699
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.05)$	—	—	rand	400	128	0.06	128	0.7170	0.7672	0.7206	0.7637
$\mathcal{L}_{\mathsf{c}}(\tau{=}1)$	—	—	rand	400	128	0.06	128	0.7505	0.7958	0.7560	0.7965
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$7.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.8188	0.8556	0.8302	0.8590
—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7237	0.7788	0.7224	0.7806
—	$4.625 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.8692	0.8820	0.8724	0.8840
—	$3.375 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7663	0.7935	0.7691	0.7938
—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	**0.7008**	0.7621	**0.7014**	0.7592
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.25 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7293	0.7997	0.7313	0.8013
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	400	128	0.06	128	0.7079	**0.7468**	0.7105	**0.7460**
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.005)$	—	—	rand	400	128	0.06	128	0.7608	0.8109	0.7633	0.8149
—	$4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7721	0.8195	0.7737	0.8190
—	$1.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7231	0.7810	0.7193	0.7889
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7044	0.7714	0.7047	0.7718
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7329	0.7751	0.7454	0.7786
—	$2.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7295	0.7747	0.7304	0.7785
—	$4.125 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7497	0.8129	0.7478	0.8128
—	$0.125 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$2.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.8109	0.8535	0.8092	0.8523
—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7509	0.7892	0.7324	0.7926
—	$3.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7514	0.8005	0.7531	0.8003
—	$2.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7360	0.7706	0.7413	0.7747
—	$4.875 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.8699	0.8882	0.8717	0.8918
—	$3.125 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7203	0.7713	0.7138	0.7682
—	$1.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7261	0.7744	0.7259	0.7715
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—	rand	400	128	0.06	128	0.7334	0.7743	0.7293	0.7701
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2857 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7456	0.8070	0.7423	0.8030
—	$2.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7289	0.7591	0.7250	0.7597
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$3 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7819	0.8352	0.7808	0.8314
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$10 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.8422	0.8896	0.8430	0.8857
—	$3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7203	0.7642	0.7160	0.7643
—	$3.875 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7477	0.7980	0.7476	0.7960
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.4)$	—	—	rand	400	128	0.06	128	0.7181	0.7628	0.7163	0.7651
—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7670	0.8225	0.7700	0.8224
—	$1.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7311	0.7922	0.7265	0.7942
—	$1.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7323	0.7900	0.7297	0.7884
—	$4.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7592	0.8350	0.7585	0.8297
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7909	0.8517	0.7891	0.8526
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—	rand	400	128	0.06	128	0.7068	0.7594	0.7028	0.7624
—	$3.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7352	0.7853	0.7294	0.7817
—	$3.125 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7152	0.7661	0.7060	0.7667
—	$3.625 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7420	0.7925	0.7505	0.7970
—	$5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.8072	0.8631	0.8084	0.8617
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	—	—	rand	400	128	0.06	128	0.7074	0.7539	0.7124	0.7491
—	$1.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7255	0.7793	0.7199	0.7765
—	$7.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.8160	0.8512	0.8131	0.8505
—	$4.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.8102	0.8633	0.8084	0.8721

—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$2.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7696	0.8208	0.7669	0.8141
—	$2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7209	0.7839	0.7370	0.7867
$0.5 \cdot \mathcal{L}_{\mathsf{c}}(\tau{=}0.1)$	—	—	rand	400	128	0.06	128	0.7062	0.7586	0.7024	0.7575
$\mathcal{L}_{\mathsf{c}}(\tau{=}10)$	—	—	rand	400	128	0.06	128	0.7860	0.8375	0.7850	0.8335
—	$3.375 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7236	0.7703	0.7230	0.7728
—	$0.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7596	0.8122	0.7574	0.8107
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.3)$	—	—	rand	400	128	0.06	128	0.7337	0.7653	0.7361	0.7640
$\mathcal{L}_{\mathsf{c}}(\tau{=}5)$	—	—	rand	400	128	0.06	128	0.7801	0.8278	0.7715	0.8355
—	$3.25 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.7495	0.7903	0.7503	0.7941
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	rand	400	128	0.06	128	0.8062	0.8597	0.8042	0.8608

Table 10: Experiment specifications for all 45 IMAGENET-100 ResNet50 encoders trained using methods based on Momentum Contrast (MoCo) (He et al., 2019). We report the encoder representation quality measured by accuracy of a linear classifier on penultimate layer activations, via both a 3-fold cross validation of the training set and the held out validation set.
All encoders in this table use standard network initialization (denoted as "rand"). Dimensionality (abbreviated as "Dim.") shows the ambient dimension of the output features, i.e., they live on the unit hypersphere of one less dimension.
For $\mathcal{L}_{\mathsf{uniform}}$, the "Intra-batch" column denotes whether $\mathcal{L}_{\mathsf{uniform}}$ calculation includes pairwise distances within batch in addition to distances w.r.t. to the queue (i.e., Equation (22) vs. Equation (21)).

Losses				Init.	Epochs	Batch Size	Queue Size	Initial LR	Dim.	Training Set 3-Fold Cross Val. Accuracy ↑		Validation Set Accuracy ↑	
$\mathcal{L}_{\mathsf{contrastive}}$	$\mathcal{L}_{\mathsf{align}}$	$\mathcal{L}_{\mathsf{uniform}}$ Form	$\mathcal{L}_{\mathsf{uniform}}$ Intra-batch							top1	top5	top1	top5
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.01)$	—	—		rand	240	128	16384	0.03	128	62.45%	85.64%	64.14%	86.12%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.07)$	—	—		rand	240	128	16384	0.03	128	71.68%	91.00%	72.80%	91.64%
$\mathcal{L}_{\mathsf{c}}(\tau{=}0.5)$	—	—		rand	240	128	16384	0.03	128	68.56%	91.21%	69.98%	91.80%
$\mathcal{L}_{\mathsf{c}}(\tau{=}1)$	—	—		rand	240	128	16384	0.03	128	62.19%	87.73%	64.06%	88.32%
$\mathcal{L}_{\mathsf{c}}(\tau{=}2)$	—	—		rand	240	128	16384	0.03	128	53.62%	83.03%	55.46%	84.18%
$\mathcal{L}_{\mathsf{c}}(\tau{=}5)$	—	—		rand	240	128	16384	0.03	128	37.52%	68.93%	39.00%	70.86%
—	$2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	—		rand	240	128	16384	0.03	128	1.03%	5.12%	1.22%	5.42%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.125 \cdot \mathcal{L}_{\mathsf{u}}(t{=}8)$	✓	rand	240	128	16384	0.03	128	65.89%	88.28%	67.42%	88.96%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.15 \cdot \mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	240	128	16384	0.03	128	67.51%	88.95%	68.90%	89.68%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.17 \cdot \mathcal{L}_{\mathsf{u}}(t{=}6)$	✓	rand	240	128	16384	0.03	128	67.90%	89.83%	69.18%	90.76%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	240	128	16384	0.03	128	69.27%	90.08%	70.46%	90.86%
—	$1.8 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	1.00%	4.94%	1.00%	5.00%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.25 \cdot \mathcal{L}_{\mathsf{u}}(t{=}4)$	✓	rand	240	128	16384	0.03	128	69.77%	90.57%	70.70%	91.14%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.33 \cdot \mathcal{L}_{\mathsf{u}}(t{=}3)$	✓	rand	240	128	16384	0.03	128	70.67%	91.14%	71.86%	91.58%
—	$1.6 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	67.34%	90.27%	69.16%	91.00%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✗	rand	240	128	16384	0.03	128	70.91%	91.38%	72.34%	91.86%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	71.03%	91.61%	71.90%	92.06%
—	$1.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	71.11%	91.69%	72.06%	92.28%
—	$1.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	71.76%	91.51%	72.78%	91.90%
—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	70.23%	91.01%	71.40%	91.36%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}1)$	✓	rand	240	128	16384	0.03	128	68.07%	90.66%	69.54%	91.14%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✗	rand	240	128	16384	0.03	128	69.59%	90.67%	70.64%	91.28%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	70.45%	91.25%	71.48%	91.72%
—	$1.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	72.39%	91.71%	73.80%	92.22%
—	$2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✗	rand	240	128	16384	0.03	128	72.19%	**92.35%**	73.30%	92.74%
—	$2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✗	rand	240	128	32768	0.03	128	72.41%	92.08%	73.54%	92.74%
—	$2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	72.69%	92.21%	73.74%	**92.80%**
—	$2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	32768	0.03	128	72.65%	92.09%	73.68%	92.46%
—	$2.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✗	rand	240	128	16384	0.03	128	71.77%	91.99%	73.00%	92.14%
—	$2.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	72.31%	91.99%	73.50%	92.38%
—	$3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	72.03%	92.09%	73.48%	92.56%
—	$3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}3)$	✓	rand	240	128	16384	0.03	128	**73.49%**	92.24%	**74.60%**	92.74%
—	$4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}4)$	✓	rand	240	128	16384	0.03	128	72.93%	92.03%	74.30%	92.54%
—	$5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	240	128	16384	0.03	128	71.96%	91.67%	73.04%	92.28%
—	$6 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}6)$	✓	rand	240	128	16384	0.03	128	70.49%	90.63%	72.02%	91.24%
—	$7 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	240	128	16384	0.03	128	70.66%	90.83%	72.32%	91.86%
—	$8 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}8)$	✓	rand	240	128	16384	0.03	128	69.47%	90.33%	70.86%	91.26%
—	$0.8 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	70.45%	90.72%	71.22%	91.06%
—	$0.6 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	69.03%	90.53%	70.44%	90.92%
—	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	67.04%	89.24%	68.32%	89.76%
—	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$1.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	66.71%	88.93%	68.10%	89.48%
—	—	$2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	2.43%	9.97%	2.92%	10.56%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	58.43%	84.67%	60.36%	85.02%
—	$2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✗	rand	240	128	32768	0.03	128	69.68%	91.13%	70.80%	91.80%
—	$2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	240	128	16384	0.03	128	69.62%	90.77%	70.92%	91.42%

Table 11: Experiment specifications for all 108 BOOKCORPUS recurrent encoders trained using methods based on Quick-Thought Vectors (Logeswaran & Lee, 2018). We report the encoder representation quality measured by accuracy of logistic classifiers on encoder outputs for the Movie Review Sentence Polarity (MR) and Customer Product Sentiment (CR) binary classification tasks, via both a 5-fold cross validation of the training set (of the downstream task) and the held out validation set (of the downstream task).
All encoders in this table use standard network initialization (denoted as "rand"). Dimensionality (abbreviated as "Dim.") shows the ambient dimension of the output features, i.e., features from $l2$-normalized encoders live on the unit hypersphere of one less dimension. Regardless of whether the encoder is $l2$-normalized (indicated in "Normalization" column), the features are always normalized before being used for downstream tasks, following Logeswaran & Lee (2018).
The only unnormalized encoder is obtained using the unmodified Quick-Thought Vectors algorithm. 6 configurations that suffer from training instability (i.e., NaN occurring) are also reported.

Losses			Normalization	Init.	Epochs	Batch Size	Initial LR	Dim.	Training Set 5-Fold Cross Val. Accuracy ↑		Validation Set Accuracy ↑	
$\mathcal{L}_{\text{contrastive}}$	$\mathcal{L}_{\text{align}}$	$\mathcal{L}_{\text{uniform}}$							MR	CR	MR	CR
$\mathcal{L}_{\text{c}}(\tau{=}1)$	—	—	✗	rand	1	400	0.0005	1200	76.33%	81.90%	77.23%	83.07%
$\mathcal{L}_{\text{c}}(\tau{=}0.005)$	—	—	✓	rand	1	400	0.0005	1200	74.97%	82.94%	76.85%	82.54%
$\mathcal{L}_{\text{c}}(\tau{=}0.01)$	—	—	✓	rand	1	400	0.0005	1200	75.02%	82.20%	75.54%	82.28%
$\mathcal{L}_{\text{c}}(\tau{=}0.05)$	—	—	✓	rand	1	400	0.0005	1200	75.48%	**83.64%**	**77.69%**	83.86%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	—	—	✓	rand	1	400	0.0005	1200	**76.37%**	83.32%	77.51%	82.28%
$\mathcal{L}_{\text{c}}(\tau{=}0.1)$	—	—	✓	rand	1	400	0.0005	1200	75.82%	81.90%	74.79%	83.86%
$\mathcal{L}_{\text{c}}(\tau{=}0.2)$	—	—	✓	rand	1	400	0.0005	1200	74.33%	81.08%	75.63%	80.16%
$\mathcal{L}_{\text{c}}(\tau{=}0.25)$	—	—	✓	rand	1	400	0.0005	1200	72.33%	79.49%	71.51%	78.84%
$\mathcal{L}_{\text{c}}(\tau{=}0.3)$	—	—	✓	rand	1	400	0.0005	1200	72.85%	78.54%	73.57%	79.10%
$\mathcal{L}_{\text{c}}(\tau{=}0.4)$	—	—	✓	rand	1	400	0.0005	1200	69.72%	77.28%	67.85%	77.51%
$\mathcal{L}_{\text{c}}(\tau{=}0.5)$	—	—	✓	rand	1	400	0.0005	1200	68.97%	76.27%	68.98%	74.07%
$\mathcal{L}_{\text{c}}(\tau{=}0.6)$	—	—	✓	rand	1	400	0.0005	1200	68.61%	75.48%	68.88%	73.81%
$\mathcal{L}_{\text{c}}(\tau{=}0.7)$	—	—	✓	rand	1	400	0.0005	1200	67.89%	74.01%	67.76%	76.46%
$\mathcal{L}_{\text{c}}(\tau{=}0.8)$	—	—	✓	rand	1	400	0.0005	1200	67.02%	74.77%	66.07%	74.34%
$\mathcal{L}_{\text{c}}(\tau{=}0.9)$	—	—	✓	rand	1	400	0.0005	1200	66.78%	74.01%	65.32%	72.75%
$\mathcal{L}_{\text{c}}(\tau{=}1)$	—	—	✓	rand	1	400	0.0005	1200	66.67%	74.12%	65.79%	74.34%
$\mathcal{L}_{\text{c}}(\tau{=}1.5)$	—	—	✓	rand	1	400	0.0005	1200	63.92%	70.47%	65.42%	75.93%
$\mathcal{L}_{\text{c}}(\tau{=}2)$	—	—	✓	rand	1	400	0.0005	1200	63.97%	72.06%	62.79%	71.69%
$\mathcal{L}_{\text{c}}(\tau{=}5)$	—	—	✓	rand	1	400	0.0005	1200	62.21%	69.50%	62.98%	73.54%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	$\mathcal{L}_{\text{a}}(\alpha{=}2)$	—	✓	rand	1	400	0.0005	1200	69.16%	73.39%	68.13%	72.75%
$\mathcal{L}_{\text{c}}(\tau{=}1)$	$\mathcal{L}_{\text{a}}(\alpha{=}2)$	—	✓	rand	1	400	0.0005	1200	49.68%	63.81%	49.77%	63.49%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	$0.9 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.1 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	71.26%	77.90%	71.42%	76.72%
$\mathcal{L}_{\text{c}}(\tau{=}1)$	$0.9 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.1 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	51.26%	63.78%	52.01%	63.49%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	$0.8 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	76.25%	83.05%	76.48%	83.33%
$\mathcal{L}_{\text{c}}(\tau{=}1)$	$0.8 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	71.33%	79.31%	70.48%	78.31%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	$0.7 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.3 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	75.67%	81.20%	74.60%	81.48%
$\mathcal{L}_{\text{c}}(\tau{=}1)$	$0.7 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.3 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	71.59%	78.72%	73.66%	78.84%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	$0.6 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	75.06%	82.23%	74.41%	81.48%
$\mathcal{L}_{\text{c}}(\tau{=}1)$	$0.6 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	70.53%	78.43%	68.88%	75.93%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	$0.5 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	74.45%	81.61%	74.51%	**84.66%**
$\mathcal{L}_{\text{c}}(\tau{=}1)$	$0.5 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	66.06%	72.97%	63.64%	73.02%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	$0.4 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	73.23%	80.61%	74.32%	82.54%
$\mathcal{L}_{\text{c}}(\tau{=}1)$	$0.4 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	57.75%	67.55%	57.92%	69.84%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	$0.3 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.7 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	72.99%	79.46%	74.88%	77.25%
$\mathcal{L}_{\text{c}}(\tau{=}1)$	$0.3 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.7 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	56.96%	64.31%	55.30%	65.34%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	$0.2 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	71.94%	79.43%	70.95%	78.04%
$\mathcal{L}_{\text{c}}(\tau{=}1)$	$0.2 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	54.90%	64.22%	55.11%	63.76%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	$0.1 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.9 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	70.53%	78.25%	69.82%	78.57%
$\mathcal{L}_{\text{c}}(\tau{=}1)$	$0.1 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.9 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	55.56%	64.90%	53.98%	65.08%
$\mathcal{L}_{\text{c}}(\tau{=}0.075)$	—	$\mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	70.13%	77.66%	70.67%	77.25%
$\mathcal{L}_{\text{c}}(\tau{=}1)$	—	$\mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	54.76%	63.45%	53.98%	64.81%
—	$\mathcal{L}_{\text{a}}(\alpha{=}2)$	—	✓	rand	1	400	0.0005	1200	49.85%	63.81%	50.05%	63.49%
—	$\mathcal{L}_{\text{a}}(\alpha{=}2)$	—	✓	rand	1	400	0.0005	1200	50.02%	63.81%	49.30%	63.49%
—	$\mathcal{L}_{\text{a}}(\alpha{=}2)$	—	✓	rand	1	400	0.0005	1200	50.04%	63.81%	49.95%	63.49%
—	$0.9091 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.0909 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	49.67%	63.81%	49.86%	63.49%
—	$0.9 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.1 \cdot \mathcal{L}_{\text{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	49.71%	63.81%	49.77%	63.49%
—	$0.9 \cdot \mathcal{L}_{\text{a}}(\alpha{=}2)$	$0.1 \cdot \mathcal{L}_{\text{u}}(t{=}5)$	✓	rand	1	400	0.0005	1200	73.42%	81.23%	73.76%	80.95%

—	$0.9 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1 \cdot \mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	1	400	0.0005	1200	70.59%	78.57%	71.60%	77.51%
—	$0.8889 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1111 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	50.14%	63.81%	49.86%	63.49%
—	$0.875 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.125 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	50.33%	63.98%	49.86%	63.49%
—	$0.875 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.125 \cdot \mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	1	400	0.0005	1200	64.70%	72.71%	64.10%	71.69%
—	$0.8571 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1429 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	59.80%	66.52%	59.51%	67.72%
—	$0.8333 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.1667 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	68.42%	76.07%	68.60%	75.13%
—	$0.8333 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1667 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	66.69%	73.09%	67.95%	71.69%
—	$0.833 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.167 \cdot \mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	1	400	0.0005	1200	54.35%	64.49%	56.33%	63.49%
—	$0.8298 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.1702 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	67.38%	74.68%	67.29%	73.81%
—	$0.8298 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1702 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	66.24%	73.33%	64.76%	77.25%
—	$0.8261 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.1739 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	65.91%	75.27%	66.82%	74.07%
—	$0.8261 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1739 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	67.65%	73.56%	67.95%	72.49%
—	$0.8222 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.1778 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	66.73%	75.13%	67.85%	73.54%
—	$0.8222 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1778 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	69.33%	73.42%	69.54%	74.60%
—	$0.8182 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.1818 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	66.17%	74.36%	65.70%	74.34%
—	$0.8182 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1818 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	69.61%	75.51%	70.10%	75.40%
—	$0.814 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.186 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	63.43%	72.74%	63.82%	73.28%
—	$0.814 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.186 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	71.32%	77.72%	70.85%	77.25%
—	$0.8095 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.1905 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	63.47%	72.33%	63.82%	73.28%
—	$0.8095 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1905 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	71.33%	77.19%	71.13%	75.40%
—	$0.8049 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.1951 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	61.17%	70.79%	61.01%	73.54%
—	$0.8049 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.1951 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	72.04%	77.93%	73.38%	77.51%
—	$0.8 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	60.91%	69.41%	59.14%	70.37%
—	$0.8 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	72.60%	80.34%	73.48%	79.89%
—	$0.8 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	1	400	0.0005	1200	54.82%	63.19%	51.64%	64.02%
—	$0.8 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.2 \cdot \mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	1	400	0.0005	1200	53.67%	63.90%	57.92%	65.61%
—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.25 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	55.29%	63.63%	55.11%	70.11%
—	$0.75 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.25 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	72.60%	80.72%	71.88%	79.63%
—	$0.7 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.3 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	54.24%	63.87%	55.01%	68.52%
—	$0.7 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.3 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	71.80%	78.93%	73.76%	77.78%
—	$0.7 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.3 \cdot \mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	1	400	0.0005	1200	55.34%	62.07%	53.51%	63.23%
—	$0.7 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.3 \cdot \mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	1	400	0.0005	1200	54.22%	64.28%	55.20%	60.85%
—	$0.6667 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.3333 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	55.42%	63.25%	54.83%	68.78%
—	$0.6667 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.3333 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	68.49%	76.48%	67.20%	74.60%
—	$0.6 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	54.86%	63.63%	55.30%	67.46%
—	$0.6 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	60.60%	69.35%	61.29%	68.25%
—	$0.6 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	1	400	0.0005	1200	54.64%	63.96%	56.61%	62.43%
—	$0.6 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.4 \cdot \mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	1	400	0.0005	1200	55.28%	63.63%	55.20%	63.76%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	53.61%	64.40%	52.86%	66.14%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	55.42%	64.75%	55.76%	66.40%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	1	400	0.0005	1200	55.49%	63.16%	55.39%	64.29%
—	$0.5 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.5 \cdot \mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	1	400	0.0005	1200	56.06%	63.90%	57.73%	64.81%
—	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	54.27%	64.37%	54.45%	63.49%
—	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	55.22%	63.69%	57.73%	67.72%
—	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	1	400	0.0005	1200	53.26%	63.57%	53.70%	65.87%
—	$0.4 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.6 \cdot \mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	1	400	0.0005	1200	54.53%	63.66%	53.14%	64.55%
—	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.7 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	54.75%	63.43%	53.42%	64.02%
—	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.7 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	53.64%	63.84%	54.64%	62.70%
—	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.7 \cdot \mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	1	400	0.0005	1200	55.13%	63.81%	55.39%	64.81%
—	$0.3 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.7 \cdot \mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	1	400	0.0005	1200	56.56%	63.87%	56.04%	66.67%
—	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	53.86%	64.04%	54.83%	69.31%
—	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	53.73%	65.34%	53.98%	64.55%
—	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	1	400	0.0005	1200	54.76%	64.37%	55.76%	65.87%
—	$0.2 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.8 \cdot \mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	1	400	0.0005	1200	54.86%	63.51%	53.89%	66.40%
—	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.9 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	54.60%	65.72%	56.42%	68.52%
—	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.9 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	54.60%	64.90%	57.26%	60.85%
—	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.9 \cdot \mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	1	400	0.0005	1200	56.23%	63.66%	55.48%	66.14%
—	$0.1 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}2)$	$0.9 \cdot \mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	1	400	0.0005	1200	54.65%	65.22%	55.95%	64.02%
—	—	$\mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	55.02%	62.69%	57.36%	67.72%
—	—	$\mathcal{L}_{\mathsf{u}}(t{=}5)$	✓	rand	1	400	0.0005	1200	54.95%	64.04%	56.04%	64.02%
—	—	$\mathcal{L}_{\mathsf{u}}(t{=}7)$	✓	rand	1	400	0.0005	1200	54.55%	63.48%	56.33%	63.49%
—	$\mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	—	✓	rand	1	400	0.0005	1200	NaN occurred			
—	$0.9091 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.0909 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	NaN occurred			
—	$0.9 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.1 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	rand	1	400	0.0005	1200	NaN occurred			

—	$0.8889 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.1111 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	`rand`	1	400	0.0005	1200	`NaN` occurred
—	$0.875 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.125 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	`rand`	1	400	0.0005	1200	`NaN` occurred
—	$0.8571 \cdot \mathcal{L}_{\mathsf{a}}(\alpha{=}1)$	$0.1429 \cdot \mathcal{L}_{\mathsf{u}}(t{=}2)$	✓	`rand`	1	400	0.0005	1200	`NaN` occurred