On Discriminative Probabilistic Modeling for Self-Supervised Representation Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the discriminative probabilistic modeling problem over a continuous domain for (multimodal) self-supervised representation learning. To address the challenge of computing the integral in the partition function for each anchor data, we leverage the multiple importance sampling (MIS) technique for robust Monte Carlo integration, which can recover the InfoNCE-based contrastive loss as a special case. Within this probabilistic modeling framework, we reveal the limitation of current InfoNCE-based contrastive loss for self-supervised representation learning and derive insights for developing better approaches by reducing the error of Monte Carlo integration. To this end, we propose a novel non-parametric method for approximating the sum of conditional densities required by MIS through optimization, yielding a new contrastive objective for self-supervised representation learning. Moreover, we design an efficient algorithm for solving the proposed objective. Experimental results on bimodal contrastive representation learning demonstrate the overall superior performance of our approach on downstream tasks.

1 Introduction

Self-supervised learning (SSL) of large models has emerged as a prominent paradigm for building artificial intelligence (AI) systems [1]. Although self-supervision differs from human supervision, SSL and supervised learning share similarities. For instance, many successful self-supervised learning models (e.g., CLIP [2]) still use the softmax function and cross-entropy loss to define their objective functions, similar to traditional multi-class classification in supervised learning. The key difference is that self-supervised learning focuses on **predicting relevant data instead of relevant labels**.

Discriminative probabilistic modeling (DPM) uses a parameterized model to capture the **conditional** probability $\Pr(\mathbf{a}|\mathbf{o})$ of a target $\mathbf{a} \in \mathcal{A}$ given an input data point $\mathbf{o}$, which is a fundamental supervised learning approach. For example, logistic regression for multi-class classification (MCC) uses $\Pr(\mathbf{a}|\mathbf{o})$ to define the probability of a label $\mathbf{a}$ given data $\mathbf{o}$, whose maximum likelihood estimation (MLE) yields the cross-entropy (CE) loss. Similarly, DPM approaches such as ListNet [3] have been used for learning to rank (L2R) to model the probability of a candidate $\mathbf{a}$ in a list given a query $\mathbf{o}$. In these supervised learning problems, the target $\mathbf{a}$ is from a finite set $\mathcal{A}$ (e.g. class labels or candidate list).

What if the target $\mathbf{a}$ in DPM is from a continuous domain $\mathcal{A}$? This is particularly useful for modeling the prediction task of self-supervised representation learning. Considering that each underlying object in the real world generates various forms of observational data, such as images, texts, and audio, DPM is a natural choice to model the probability of observing a data point from a continuous domain (e.g., the space of natural images, audio, or the continuous input embedding space of texts) given an "anchor" data point. The anchor data may come from a different modality.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

However, solving DPM over a continuous domain is deemed as a challenging task (c.f. Section 1.3 in [4]). Compared to the probabilistic modeling over discrete and finite sets, such as in traditional supervised learning tasks like MCC and L2R, the DPM problem over a continuous domain (real vector space) necessitates computing the partition function (i.e., the normalizing constant) for each anchor. This involves an integration over an underlying continuous space, rather than a finite summation. In this work, we study DPM over a continuous domain for self-supervised representation learning by investigating a computational framework of robust Monte Carlo integration of the partition functions based on multiple importance sampling (MIS) [5]. Related works are discussed in detail in Appendix A.

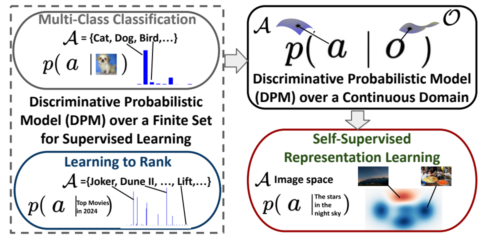

Figure 1: Discriminative probabilistic modeling for supervised learning and self-supervised representation learning.

The multiple importance sampling (MIS) approach [5, 6] was originally introduced to address the glossy highlights problem for image rendering in computer graphics, which involves computing several integrals of the form $g(r,s) = \int_{\mathcal{X}} f(\mathbf{x}; r, s)\mu(d\mathbf{x})$ corresponding to variations in light size s and surface glossiness r. For Monte Carlo integration of $g(r,s)$, importance sampling based on a sample from a single distribution may lead to a large variance under some light size/surface glossiness. To address this issue, the MIS approach constructs an unbiased estimator $\sum_{j=1}^n \omega^{(j)}(\mathbf{x}_j)\frac{f(\mathbf{x}_j;r,s)}{p_j(\mathbf{x}_j)}$ by combining samples $\mathbf{x}_1 \dots, \mathbf{x}_n$ from different strategies (distributions) $p_1, \dots, p_n$, where $\boldsymbol{\omega} = (\omega^{(1)}, \dots, \omega^{(n)})$ is a weighting function satisfies that $\sum_{i=1}^n \omega^{(j)}(\mathbf{x}) = 1$ whenever $f(\mathbf{x}; r, s) \neq 0$ and $\omega^{(j)}(\mathbf{x}) = 0$ whenever $p_i(\mathbf{x}) = 0$. In particular, [5] proposed the "balance heuristic" $\omega^{(j)}(\mathbf{x}) = \frac{p_j(\mathbf{x})}{\sum_{j'=1}^n p_{j'}(\mathbf{x})}$, $\forall j \in [n], \mathbf{x} \in \mathcal{X}$ and proved that this choice of $\boldsymbol{\omega}$ is near-optimal in terms of variance among all possible weighting functions. Empirically, MIS combined with the balance heuristic leads to improved rendering performance compared to importance sampling using a single distribution.

2 DPM over a Continuous Domain

When choosing $\mathcal{O}$ as the anchor space, we model the probability density $p(\mathbf{a} \mid \mathbf{o})$ of an object $\mathbf{a} \in \mathcal{A}$ given an anchor object $\mathbf{o} \in \mathcal{O}$ by the following DPM parameterized by $\mathbf{w}$.

$$p_{\mathbf{w}}(\mathbf{a} \mid \mathbf{o}) = \frac{\exp(e_{\mathbf{w}}(\mathbf{o}, \mathbf{a})/\tau)}{\int_{\mathcal{A}} \exp(e_{\mathbf{w}}(\mathbf{o}, \mathbf{a}')/\tau)\mu(d\mathbf{a}')}, \tag{1}$$

where $\tau > 0$ is a temperature parameter for flexibility, $e_{\mathbf{w}} : \mathcal{O} \times \mathcal{A} \to \mathbb{R}$ is a parameterized prediction function, which could be based on a "two-tower" model, like the one in SimCLR [7], or a "one-tower" model, similar to the one used in BERT [8]. We assume that $\exp(e_{\mathbf{w}}(\mathbf{o}, \mathbf{a})/\tau)$ is Lebesgue-integrable for $\mathbf{w} \in \mathcal{W}$, $\mathcal{W} \subset \mathbb{R}^d$. Here $p_{\mathbf{w}}(\mathbf{a} \mid \mathbf{o})$ is a valid probability density function because $\int_{\mathcal{A}} p_{\mathbf{w}}(\mathbf{a} \mid \mathbf{o})\mu(d\mathbf{a}) = 1$. Given a sample $\{(\mathbf{o}_1, \mathbf{a}_1), \dots, (\mathbf{o}_n, \mathbf{a}_n)\}$ from the joint distribution $p_{\mathbf{o},\mathbf{a}}$, the maximum likelihood estimation (MLE) is done by:

$$\min_{\mathbf{w}} \left\{ -\frac{1}{n} \sum_{i=1}^n \tau \log \frac{\exp(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_i)/\tau)}{\int_{\mathcal{A}} \exp(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}')/\tau)\mu(d\mathbf{a}')} \right\}. \tag{2}$$

Remark 1. *Learning the DPM $p_{\hat{\mathbf{w}}_*}$ via MLE for self-supervised pretraining naturally provides some performance guarantees for downstream discriminative tasks. Suppose that the true conditional density function is parameterized by some $\mathbf{w}_* \in \mathcal{W}$, i.e., $p = p_{\mathbf{w}_*}$ and $p_{\mathbf{w}_*}(\mathbf{a} \mid \mathbf{o}) = \frac{\exp(e_{\mathbf{w}_*}(\mathbf{o},\mathbf{a})/\tau)}{\int_{\mathcal{A}} \exp(e_{\mathbf{w}_*}(\mathbf{o},\mathbf{a}')/\tau)\mu(d\mathbf{a}')}$ for any $\mathbf{o} \in \mathcal{O}, \mathbf{a} \in \mathcal{A}$. Then, the maximum likelihood estimator $\hat{\mathbf{w}}_* = \arg\max_{\mathbf{w} \in \mathcal{W}} \frac{1}{n}\sum_{i=1}^n \log p_{\mathbf{w}}(\mathbf{a}_i \mid \mathbf{o}_i)$ with the sample $\{(\mathbf{o}_i, \mathbf{a}_i)\}_{i=1}^n$ converges in probability to $\mathbf{w}_*$ under some mild assumptions (see Theorem 2.1 in [9]). Due to the continuous mapping theorem, the learned model satisfies $e_{\hat{\mathbf{w}}_*}(\mathbf{o}, \mathbf{a}) \xrightarrow{p} e_{\mathbf{w}_*}(\mathbf{o}, \mathbf{a})$ if the parameterized models $e_{\mathbf{w}}$ has measure-zero discontinuity points on $\mathcal{W}$, which naturally provides a statistical guarantee for cross-modality retrieval. In Appendix E, we also discuss the performance of DPM on downstream classification tasks.*

When choosing $\mathcal{A}$ as the anchor space, we can also model the probability density of an object $\mathbf{o} \in \mathcal{O}$ given an anchor $\mathbf{a} \in \mathcal{A}$ by the parameterized model $p_{\mathbf{w}}(\mathbf{o} \mid \mathbf{a}) = \frac{\exp(e_{\mathbf{w}}(\mathbf{o},\mathbf{a})/\tau)}{\int_{\mathcal{O}} \exp(e_{\mathbf{w}}(\mathbf{o}',\mathbf{a})/\tau)\mu(d\mathbf{o}')}$ similar to (1).

Based on a sample $\{(\mathbf{o}_1, \mathbf{a}_1), \dots, (\mathbf{o}_n, \mathbf{a}_n)\}$ from the joint distribution $p_{\mathbf{o},\mathbf{a}}$, we can simultaneously model $p_{\mathbf{w}}(\mathbf{a} \mid \mathbf{o})$ and $p_{\mathbf{w}}(\mathbf{o} \mid \mathbf{a})$ via the objective below, which resembles the symmetric loss in [2].

$$\min_{\mathbf{w}} -\frac{1}{n}\sum_{i=1}^{n}\left(\tau \log \frac{\exp(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_i)/\tau)}{\int_{\mathcal{A}} \exp(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}')/\tau)\mu(d\mathbf{a}')} + \tau \log \frac{\exp(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_i)/\tau)}{\int_{\mathcal{O}} \exp(e_{\mathbf{w}}(\mathbf{o}', \mathbf{a}_i)/\tau)\mu(d\mathbf{o}')}\right).$$

2.1 An MIS-based Empirical Risk for Maximum Likelihood Estimation

For simplicity, let us focus on the case where $\mathcal{O}$ is the anchor space. The main challenge of MLE in (2) based on the sample $\{(\mathbf{o}_1, \mathbf{a}_1), \dots, (\mathbf{o}_n, \mathbf{a}_n)\}$ lies in computing the integral $g(\mathbf{w}; \mathbf{o}_i, \mathcal{A}) \coloneqq \int_{\mathcal{A}} \exp\left(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}')/\tau\right)\mu(d\mathbf{a}')$ for each $i \in [n]$, which is infeasible unless $\mathcal{A}$ is finite and sufficiently small. For the importance sampling method for Monte Carlo integration, it is difficult, if not impossible, to select a single instrumental distribution that works well for all integrals $g(\mathbf{w}; \mathbf{o}_i, \mathcal{A})$, $i \in [n]$. Moreover, drawing additional samples from q to construct an unbiased estimator of $g(\mathbf{w}; \mathbf{o}_i, \mathcal{A})$ leads to extra costs. Recall that we have a sample $\mathbf{a}_j$ drawn from the distribution $p_{\cdot|\mathbf{o}_j}$ for each anchor $\mathbf{o}_j$, $j = 1, 2, \dots, n$. Thus, we employ the MIS method with balance heuristic [5] to construct the estimator $\hat{g}(\mathbf{w}; \mathbf{o}_i, \hat{\mathbf{A}}) = \sum_{j=1}^{n} \frac{1}{\sum_{j'=1}^{n} p(\mathbf{a}_j|\mathbf{o}_{j'})} \exp\left(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j)/\tau\right)$ of $g(\mathbf{w}; \mathbf{o}_i, \mathcal{A}) = \int_{\mathcal{A}} \exp\left(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}')/\tau\right)\mu(d\mathbf{a}')$ by combining samples $\mathbf{a}_1, \dots, \mathbf{a}_n$ from n distributions $p_{\cdot|\mathbf{o}_1}, \dots, p_{\cdot|\mathbf{o}_n}$. In Appendix D, we show the unbiasedness of the estimator $\hat{g}(\mathbf{w}; \mathbf{o}_i, \hat{\mathbf{A}})$ and explain why we choose the balance heuristic over other possible weighting functions for MIS.

However, a remaining issue prevents us from using the MIS-based estimator $\hat{g}(\mathbf{w}; \mathbf{o}_i, \hat{\mathbf{A}})$. Unlike the rendering problem considered in [5], we do not have access to the conditional probability densities $p(\mathbf{a}_j \mid \mathbf{o}_{j'})$, $j, j' \in [n]$. Thus, there is a need for a cheap approximation $\tilde{q}^{(j)}$ of the sum of conditional densities $q^{(j)} \coloneqq \sum_{j'=1}^{n} p(\mathbf{a}_j \mid \mathbf{o}_{j'})$, $\forall j \in [n]$. It is worth noting that $q^{(j)}$ can be viewed as a measure of **popularity** of $\mathbf{a}_j$ on the dataset $\{(\mathbf{o}_i, \mathbf{a}_i)\}_{i=1}^{n}$. With a general approximation $\tilde{\mathbf{q}} = (\tilde{q}^{(1)}, \dots, \tilde{q}^{(n)})^\top$ of $\mathbf{q} = (q^{(1)}, \dots, q^{(n)})^\top$, the MLE objective in (2) with MIS can be written as

$$\hat{\mathcal{L}}(\mathbf{w}; \hat{\mathbf{O}}, \hat{\mathbf{A}}) = -\frac{1}{n}\sum_{i=1}^{n} \tau \log \frac{\exp(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_i)/\tau)}{\tilde{g}(\mathbf{w}; \mathbf{o}_i, \hat{\mathbf{A}})}, \quad \tilde{g}(\mathbf{w}; \mathbf{o}_i, \hat{\mathbf{A}}) = \sum_{j=1}^{n} \frac{\exp\left(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j)/\tau\right)}{\tilde{q}^{(j)}}. \quad (3)$$

Remark 2. *If we simply choose the uniform approximation* $\tilde{q}^{(j)} = \sum_{j'=1}^{n} \frac{1}{\mu(\mathcal{A})} = \frac{n}{\mu(\mathcal{A})}$, *minimizing* $\hat{\mathcal{L}}(\mathbf{w}; \hat{\mathbf{O}}, \hat{\mathbf{A}})$ *in* (3) *is equivalent to minimizing the InfoNCE-based loss in [10] (also see Appendix A).*

2.2 Non-parametric Method for Approximating the Measure of Popularity

In Appendix C, we show that simply choosing a uniform $\tilde{\mathbf{q}}$ in the InfoNCE-based loss to approximate the measure of popularity $\mathbf{q}$ (i.e. the sum of conditional densities) leads to a non-diminishing term in generalization error. In this section, we aim to find a way to approximate the measure of popularity $\mathbf{q}$[1]. For brevity, we denote $e(\cdot,\cdot) = e_{\mathbf{w}_*}(\cdot,\cdot)$ that corresponds to the real conditional density $p(\mathbf{a} \mid \mathbf{o}) = p_{\mathbf{w}_*}(\mathbf{a} \mid \mathbf{o}) = \frac{\exp(e_{\mathbf{w}_*}(\mathbf{o},\mathbf{a})/\tau)}{\int_{\mathcal{A}} \exp(e_{\mathbf{w}_*}(\mathbf{o},\mathbf{a}')/\tau)\mu(d\mathbf{a}')}$. Thus, for any $j \in [n]$ we have

$$q^{(j)} = \sum_{j'=1}^{n} p(\mathbf{a}_j \mid \mathbf{o}_{j'}) = \sum_{j'=1}^{n} \frac{\exp(e(\mathbf{o}_{j'}, \mathbf{a}_j)/\tau)}{\int_{\mathcal{A}} \exp(e(\mathbf{o}_{j'}, \mathbf{a})/\tau)\mu(d\mathbf{a})} \overset{\diamond}{\approx} \sum_{j'=1}^{n} \frac{\exp(e(\mathbf{o}_{j'}, \mathbf{a}_j)/\tau)}{\sum_{i'=1}^{n} \frac{1}{q^{(i')}} \exp(e(\mathbf{o}_{j'}, \mathbf{a}_{i'})/\tau)}, \quad (4)$$

where the last step $\diamond$ is due to the MIS-based Monte Carlo integration and becomes an equality when $n \to \infty$ (See Prop. 1 in Appendix D). Since the expression in (4) is implicit, we propose a non-parametric method to approximate $\mathbf{q}$ by solving the following convex optimization problem.

$$\min_{\boldsymbol{\zeta} \in \mathbb{R}^n} \left\{ -\frac{1}{n}\sum_{i=1}^{n} \tau \log\left(\frac{\exp(e(\mathbf{o}_i, \mathbf{a}_i)/\tau)}{\sum_{j=1}^{n} \exp((e(\mathbf{o}_i, \mathbf{a}_j) - \zeta^{(j)})/\tau)}\right) + \frac{1}{n}\sum_{j=1}^{n} \zeta^{(j)} \right\}. \quad (5)$$

The following theorem characterizes the set of optima of (5) and its relationship to $\mathbf{q}$.

[1]Note that our goal is neither estimating the sum of probability densities $q(\mathbf{a}) = \sum_{j'=1}^{n} p(\mathbf{a} \mid \mathbf{o}_{j'})$ for any $\mathbf{a} \in \mathcal{A}$ nor estimating the conditional density $p(\mathbf{a} \mid \mathbf{o})$ in general for any $\mathbf{o} \in \mathcal{O}, \mathbf{a} \in \mathcal{A}$.

Theorem 1. *Any optimal solution* $\boldsymbol{\zeta}_*$ *to* (5) *satisfies the following implicit expression*

$$\exp(\zeta_*^{(j)}/\tau) = \sum_{j'=1}^{n} \frac{\exp(e(\mathbf{o}_{j'}, \mathbf{a}_j)/\tau)}{\sum_{i'=1}^{n} \exp((e(\mathbf{o}_{j'}, \mathbf{a}_{i'}) - \zeta_*^{(i')})/\tau)}, \quad \forall j \in [n]. \tag{6}$$

Moreover, the optimal solutions are on a line $\boldsymbol{\zeta}_* = z\mathbf{1}_n + \mathbf{b}_*$ *for any* $z \in \mathbb{R}$ *and a unique* $\mathbf{b}_* \in \mathbb{R}^n$*, i.e., the optimal solution* $\boldsymbol{\zeta}_*$ *is unique up to an additive scalar* z*. Additionally, the true* $\mathbf{q}$ *in* (4) *can be approximated as* $q^{(j)} \approx \tilde{q}^{(j)} = \frac{\exp(\zeta_*^{(j)}/\tau)}{Z}$*,* $\forall j \in [n]$*, where* $Z = \exp(z/\tau) > 0$.

Remark 3. *Theorem 1 shows that we can find an approximation* $\tilde{\mathbf{q}}$ *of* $\mathbf{q}$ *by solving the convex optimization problem in* (5) *(up to a constant scaling factor* Z*). Note that there is no need to know the value of* Z *for empirical risk minimization. If we plug* $\tilde{\mathbf{q}}' = Z\tilde{\mathbf{q}} = \exp(\boldsymbol{\zeta}_*/\tau)$ *into* (3)*, the empirical risk becomes* $\hat{\mathcal{L}}(\mathbf{w}; \hat{\mathbf{O}}, \hat{\mathbf{A}}) - z$ *and does not change the empirical risk minimizer* $\hat{\mathbf{w}}_*$.

Appendix B provides a synthetic experiment to show the effectiveness of our non-parametric method.

2.3 Application to Self-Supervised Representation Learning

By substituting the $\tilde{\mathbf{q}}$ from the non-parametric method described in Section 2.2 into the empirical risk of DPM in (3), the empirical risk minimization (ERM) problem becomes

$$\min_{\mathbf{w}\in\mathcal{W}} \hat{\mathcal{L}}(\mathbf{w}; \hat{\mathbf{O}}, \hat{\mathbf{A}}), \quad \hat{\mathcal{L}}(\mathbf{w}; \hat{\mathbf{O}}, \hat{\mathbf{A}}) \coloneqq -\frac{1}{n}\sum_{i=1}^{n} \tau \log\left(\sum_{j=1}^{n} \frac{\exp(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_i)/\tau)}{\exp((e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j) - \zeta_*^{(j)})/\tau)}\right),$$

where $\boldsymbol{\zeta}_*$ is solved from (5). Since the true similarity function $e : \mathcal{O} \times \mathcal{A} \to [-c, c]$ in (5) is unknown, we replace $e(\cdot, \cdot)$ by the parametric model $e_{\mathbf{w}}(\cdot, \cdot)$ to reach the following joint minimization problem.

$$\min_{\mathbf{w}\in\mathcal{W}, \boldsymbol{\zeta}\in\mathbb{R}^n} \left\{-\frac{1}{n}\sum_{i=1}^{n} \tau \log\left(\frac{\exp(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_i)/\tau)}{\sum_{j=1}^{n} \exp((e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j) - \zeta^{(j)})/\tau)}\right) + \frac{1}{n}\sum_{j=1}^{n} \zeta^{(j)}\right\}. \tag{7}$$

A straightforward approach for solving the above problem is taking an alternating algorithm: optimizing over ζ with fixed $\mathbf{w}$, then optimizing over $\mathbf{w}$ with fixed ζ. However, this is costly as both $\mathbf{w}$ and ζ are high-dimensional variables. Next, we propose an efficient gradient-based algorithm NUCLR described in Appendix G to minimize the loss in (7) by formulating the problem as a finite-sum coupled compositional optimization (FCCO) problem [11].

3 Experiments on Bimodal Representation Learning

We apply our algorithm to bimodal self-supervised representation learning on the CC3M [12] and CC12M [13] datasets. Detailed settings of our experiments can be found in Appendix H. We compare the testing performance of our method on downstream tasks with CLIP [2], SigLIP [14], CyCLIP [15], and SogCLR [10]. Compared to those baselines, our NUCLR achieves overall superior performance.

Table 1: A comparison of test performance. The best result in each column is highlighted in **black**.

Dataset	Algorithm	MSCOCO	Flickr30k	CIFAR100	ImageNet1k	Mean
CC3M	CLIP	24.23 ± 0.14	46.33 ± 0.76	33.94 ± 0.87	35.91 ± 0.33	35.10 ± 0.22
	SigLIP	23.21 ± 0.14	44.95 ± 0.45	35.70 ± 0.84	37.53 ± 0.09	35.35 ± 0.31
	CyCLIP	24.47 ± 0.25	47.10 ± 0.83	37.27 ± 0.61	36.63 ± 0.04	36.37 ± 0.42
	SogCLR	28.54 ± 0.25	52.20 ± 0.64	35.50 ± 1.71	40.40 ± 0.12	39.16 ± 0.33
	NUCLR (Ours)	**29.55 ± 0.26**	**53.55 ± 0.22**	**37.45 ± 0.45**	**40.49 ± 0.30**	**40.26 ± 0.19**
CC12M	CLIP	30.30 ± 0.15	55.21 ± 0.45	25.35 ± 0.64	44.28 ± 0.22	38.79 ± 0.30
	SigLIP	30.13 ± 0.45	55.40 ± 0.32	26.60 ± 1.89	46.12 ± 0.12	39.56 ± 0.68
	CyCLIP	30.35 ± 0.24	54.63 ± 0.20	26.71 ± 2.09	44.94 ± 0.02	39.15 ± 0.50
	SogCLR	33.91 ± 0.26	59.28 ± 0.07	26.10 ± 0.88	**49.82 ± 0.14**	42.28 ± 0.27
	NUCLR (Ours)	**34.36 ± 0.13**	**60.45 ± 0.03**	**28.16 ± 1.35**	**49.82 ± 0.23**	**43.20 ± 0.39**

References

[1] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

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

[3] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

[4] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[5] Eric Veach and Leonidas J Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, 1995.

[6] Eric Veach. *Robust Monte Carlo methods for light transport simulation*. Stanford University, 1998.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Whitney K Newey and Daniel McFadden. Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245, 1994.

[10] Zhuoning Yuan, Yuexin Wu, Zi-Hao Qiu, Xianzhi Du, Lijun Zhang, Denny Zhou, and Tianbao Yang. Provable stochastic optimization for global contrastive learning: Small batch does not harm performance. In *International Conference on Machine Learning*, pages 25760–25782. PMLR, 2022.

[11] Bokun Wang and Tianbao Yang. Finite-sum coupled compositional stochastic optimization: Theory and applications. In *International Conference on Machine Learning*, pages 23292–23317. PMLR, 2022.

[12] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[13] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021.

[14] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

[15] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. *Advances in Neural Information Processing Systems*, 35:6704–6719, 2022.

[16] Ilyes Khemakhem, Ricardo Monti, Diederik Kingma, and Aapo Hyvarinen. Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. *Advances in Neural Information Processing Systems*, 33:12768–12778, 2020.

[17] Duy-Nguyen Ta, Eric Cousineau, Huihua Zhao, and Siyuan Feng. Conditional energy-based models for implicit policies: The gap between theory and practice. *arXiv preprint arXiv:2207.05824*, 2022.

[18] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.

[19] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

[20] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2019.

[21] Yifei Wang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. A unified contrastive energy-based model for understanding the generative ability of adversarial training. *arXiv preprint arXiv:2203.13455*, 2022.

[22] Beomsu Kim and Jong Chul Ye. Energy-based contrastive learning of visual representations. *Advances in Neural Information Processing Systems*, 35:4358–4369, 2022.

[23] Alice Bizeul, Bernhard Schölkopf, and Carl Allen. A probabilistic model to explain self-supervised representation learning. *arXiv preprint arXiv:2402.01399*, 2024.

[24] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

[25] Yunwen Lei, Tianbao Yang, Yiming Ying, and Ding-Xuan Zhou. Generalization analysis for contrastive representation learning. In *International Conference on Machine Learning*, pages 19200–19227, 2023.

[26] Chung-Yiu Yau, Hoi-To Wai, Parameswaran Raman, Soumajyoti Sarkar, and Mingyi Hong. Emc^2: Efficient mcmc negative sampling for contrastive learning with global convergence. *arXiv preprint arXiv:2404.10575*, 2024.

[27] Hiroki Waida, Yuichiro Wada, Léo Andéol, Takumi Nakagawa, Yuhui Zhang, and Takafumi Kanamori. Towards understanding the mechanism of contrastive learning via similarity structure: A theoretical analysis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 709–727. Springer, 2023.

[28] Lan V Truong. On rademacher complexity-based generalization bounds for deep learning. *arXiv preprint arXiv:2208.04284*, 2022.

[29] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[30] Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, and Jaz Kandola. The perceptron algorithm with uneven margins. In *ICML*, volume 2, pages 379–386, 2002.

[31] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *International Conference on Machine Learning*, pages 507–516. PMLR, 2016.

[32] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[33] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

[34] Zeju Li, Konstantinos Kamnitsas, and Ben Glocker. Overfitting of neural nets under class imbalance: Analysis and improvements for segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22*, pages 402–410. Springer, 2019.

[35] Benjin Zhu, Junqiang Huang, Zeming Li, Xiangyu Zhang, and Jian Sun. Eqco: Equivalent rules for self-supervised contrastive learning. *arXiv preprint arXiv:2010.01929*, 2020.

[36] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Delving into inter-image invariance for unsupervised visual representations. *International Journal of Computer Vision*, 130(12):2994–3013, 2022.

[37] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[40] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 and CIFAR-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6:1, 2009.

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

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[43] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[44] Zi-Hao Qiu, Quanqi Hu, Zhuoning Yuan, Denny Zhou, Lijun Zhang, and Tianbao Yang. Not all semantics are created equal: Contrastive self-supervised learning with automatic temperature individualization. *arXiv preprint arXiv:2305.11965*, 2023.

[45] Siladittya Manna, Soumitri Chattopadhyay, Rakesh Dey, Saumik Bhattacharya, and Umapada Pal. Dystress: Dynamically scaled temperature in self-supervised contrastive learning. *arXiv preprint arXiv:2308.01140*, 2023.

[46] Zi-Hao Qiu, Siqi Guo, Mao Xu, Tuo Zhao, Lijun Zhang, and Tianbao Yang. To cool or not to cool? temperature network meets large foundation models via dro. *arXiv preprint arXiv:2404.04575*, 2024.

[47] Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17, 2016.

[48] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford university press, 2013.

[49] Victor De la Pena and Evarist Giné. *Decoupling: from dependence to independence*. Springer Science & Business Media, 2012.

[50] Mehryar Mohri and Andres Munoz Medina. Learning theory and algorithms for revenue optimization in second price auctions with reserve. In *International conference on machine learning*, pages 262–270. PMLR, 2014.

[51] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[52] Rui Ray Zhang, Xingwu Liu, Yuyi Wang, and Liwei Wang. Mcdiarmid-type inequalities for graph-dependent variables and stability bounds. *Advances in Neural Information Processing Systems*, 32, 2019.

[53] Stéphan Clémencon, Gábor Lugosi, and Nicolas Vayatis. Ranking and empirical minimization of u-statistics. *The Annals of Statistics*, pages 844–874, 2008.

[54] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299. PMLR, 2018.

A Related Work

Probabilistic Models for Self-Supervised Representation Learning: Discriminative probabilistic models learn the *conditional* probability mass/density function $p(\mathbf{y} \mid \mathbf{x})$ of $\mathbf{y}$ given data $\mathbf{x}$. Recently, some works have focused on modeling the conditional probability density function $p(\mathbf{y} \mid \mathbf{x})$ for the unsupervised representation learning task, where both $\mathbf{x}$ and $\mathbf{y}$ may belong to uncountable spaces. [16] studied the identifiability (i.e., the learned representations are unique up to a linear transformation) of DPM and showed its connection to nonlinear ICA models. [17] improved the Langevin MCMC method to handle the partition function in DPM for learning implicit representations of behavior-cloned policies in robotics. By discarding the partition function, [18] and [19] proposed the energy-based models I-JEPA and V-JEPA to learn visual representations by predicting the relevance between data representations. Although the high-level concept of JEPA is similar to our work in that both aim to predict the relevance between data representations, our approach is grounded in discriminative probabilistic modeling, whereas JEPA is an energy-based model that omits the partition function. Consequently, JEPA lacks some statistical guarantees of probabilistic models, such as the convergence of the maximum likelihood estimator, which have implications for performance on downstream tasks (See Section 2.1). Furthermore, JEPA is designed specifically for the visual modality whereas our algorithm applies to multimodality.

Besides, a discriminative model $p(\mathbf{y} \mid \mathbf{x})$ and a generative model $p(\mathbf{x})$ can be connected by modeling the joint distribution $p(\mathbf{x}, \mathbf{y})$. Hybrid models [20, 21, 22, 23] simultaneously perform discriminative and generative modeling, while our work focuses on learning the conditional density for downstream discriminative tasks. Although the generative component in hybrid models might offer some benefits for representation learning, such as achieving reasonably good performance with small batch size, [22] have pointed out that current hybrid models significantly increase the computational burden and are difficult to apply to large-scale datasets such as ImageNet1k due to the expensive inner loops of SGLD. In contrast, our method achieves good performance with a small batch size using techniques based on the finite-sum coupled compositional optimization (FCCO) [11, 10], which only introduces marginal computational overhead even on large-scale datasets. Furthermore, it is mentioned in [23] that hybrid models like SimVAE face difficulties scaling to large-scale, complex datasets, as "learning representations for complex data distributions under a generative regime remains a challenge compared to discriminative approaches."

Theory of Contrastive Learning: The InfoNCE loss is the most widely used objective function in contrastive learning [7, 2]. Given a dataset of pairs $\{(\mathbf{o}_i, \mathbf{a}_i)\}_{i=1}^n$ from two views or modalities, the InfoNCE loss contrasts each positive data with k negative data in the sampled batch. Both empirical observations [7, 2, 10] and theoretical analysis [10] demonstrate that algorithms based on InfoNCE perform well only when the batch size is sufficiently large (e.g. 32,768 for CLIP training), which demands a lot of computational resources. Besides, several works analyze the generalization error of InfoNCE [24, 25]. However, these analyses have a critical limitation: the generalization error increases with k, contradicting practical observations.

To address the issue of large batch size of InfoNCE, [10] studied the global contrastive loss (GCL), which can be expressed as $-\frac{1}{n}\sum_{i=1}^n \log \frac{\exp(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_i)/\tau)}{\sum_{j=1}^n \exp(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j)/\tau)}$, which can be viewed as a variant of InfoNCE loss that contrasts each positive data with all negative data. By formulating the minimization of GCL as a finite-sum compositional optimization (FCCO) problem [11], they developed the SogCLR algorithm, which converges to a neighborhood of GCL's stationary point even with small batch sizes (e.g., 256). Using an MCMC-based negative sampling approach, [26] introduced the EMC^2 algorithm, which converges to the stationary point of GCL with a small batch size. However, EMC^2 appears to perform worse than SogCLR on larger datasets such as ImageNet1k. Besides, [27] established the generalization bound of the kernel contrastive loss (KCL), which is a lower bound of GCL when the kernel is bilinear.

B Synthetic Experiment

We design a synthetic experiment to verify the effectiveness of our non-parametric method in Section 2.2. Consider anchor data space and $\mathcal{O} = \left\{(x, y) \mid x^2 + y^2 \le 1, x \in [-1, 1], y \in [0, 1]\right\}$ and contrast data space $\mathcal{A} = \{(x, y) \mid x \in [0, 1], y \in [0, 1]\}$. Let $\mathbf{o}$ be uniformly distributed on $\mathcal{O}$ and the conditional density of an $\mathbf{a} \in \mathcal{A}$ given $\mathbf{o} \in \mathcal{O}$ is $p(\mathbf{a} \mid \mathbf{o}) = \frac{\exp(e(\mathbf{o},\mathbf{a})/\tau)}{\int_{\mathcal{A}} \exp(e(\mathbf{o},\mathbf{a})/\tau)\mu(d\mathbf{a})}$, where $\tau = 0.2$ and $e(\mathbf{o}, \mathbf{a}) \coloneqq \mathbf{o}^\top \mathbf{a}$. In this problem, $\int_{\mathcal{A}} \exp(e(\mathbf{o}, \mathbf{a})/\tau)\mu(d\mathbf{a})$ can be exactly computed.

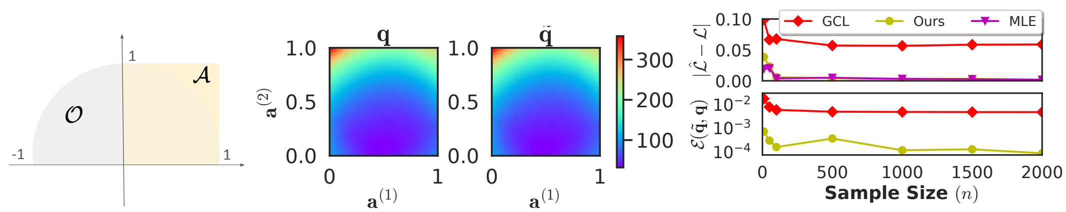

Figure 2: Left: Illustration of spaces $\mathcal{O}$ and $\mathcal{A}$; **Middle:** RBF interpolated heatmaps of the true $\mathbf{q}$ and our estimated $\tilde{\mathbf{q}}$ on data $\{\mathbf{a}_j\}_{j=1}^n$ when $n = 100$; **Right:** Comparing our non-parametric method's and GCL's generalization error $|\hat{\mathcal{L}}(\hat{\mathbf{O}}, \hat{\mathbf{A}}) - \mathcal{L}|$ and error term $\mathcal{E}(\tilde{\mathbf{q}}, \mathbf{q}, \hat{\mathbf{O}}, \hat{\mathbf{A}})$ in Theorem 2 across various n. "MLE" refers to the MLE objective in (2) with the exact partition function.

We construct a dataset $\{(\mathbf{o}_i, \mathbf{a}_i)\}_{i=1}^n$ as follows: First, we uniformly sample $\mathbf{o}_1, \ldots, \mathbf{o}_n$ from $\mathcal{O}$; Then, we sample each $\mathbf{a}_i$ from $p_{\cdot|\mathbf{o}_i}$ using rejection sampling. The ground-truth $\mathbf{q}$ can be computed as $q^{(j)} = \sum_{j'=1}^n p(\mathbf{a}_j \mid \mathbf{o}_{j'})$ using the analytic expression of $p(\mathbf{a} \mid \mathbf{o})$. To solve the convex minimization problem in (5), we initialize $\boldsymbol{\zeta}_0 = \mathbf{0}_n$ and obtain $\boldsymbol{\zeta}_*$ by running gradient descent until the gradient norm is below 10^{-15}, yielding $\tilde{\mathbf{q}}' = \exp(\boldsymbol{\zeta}_*/\tau)$. We approximate the true risk $\mathcal{L} = \mathbf{E}_{\mathbf{o},\mathbf{a}}[-\tau \log p(\mathbf{a} \mid \mathbf{o})]$ using the exact expression of $p(\mathbf{a} \mid \mathbf{o})$ on $N = 50,000$ sampled pairs. Besides, we estimate Z by $\frac{\max_j \exp(\zeta_*^{(j)}/\tau)}{\max_j q^{(j)}}$ to obtain $\tilde{\mathbf{q}} = \frac{\tilde{\mathbf{q}}'}{Z}$. It is worth noting that computing the true risk $\mathcal{L}$ and the constant Z is only for generating the plots in Figure 2, which is neither necessary nor feasible for the empirical risk minimization problem on high-dimensional real data.

As shown in the first two columns of Figure 2, our method effectively approximates the true $\mathbf{q}$ up to a constant Z. Moreover, the right column in Figure 2 confirms the result in Theorem 2 and Remark 2 that the uniform approximation of $\mathbf{q}$ in GCL results in a non-diminishing term in generalization error as n increases. In contrast, our method achieves a significantly smaller generalization error, which almost matches the MLE objective in (2) with the exact partition function.

C Finite-Sample Generalization Analysis

Corresponding to the empirical risk of MLE in 2, the true (expected) risk can be defined as

$$\mathcal{L}(\mathbf{w}) \coloneqq \mathbf{E}_{\mathbf{o},\mathbf{a}}\left[-\tau \log \frac{\exp(e_{\mathbf{w}}(\mathbf{o}, \mathbf{a})/\tau)}{\int_{\mathcal{A}} \exp(e_{\mathbf{w}}(\mathbf{o}, \mathbf{a}')/\tau)\mu(d\mathbf{a}')}\right]. \tag{8}$$

Next, we analyze the error between the empirical risk $\hat{\mathcal{L}}(\mathbf{w}; \hat{\mathbf{O}}, \hat{\mathbf{A}})$ in (3) with a *general* approximation $\tilde{\mathbf{q}}$ and the true risk $\mathcal{L}(\mathbf{w})$ in (8) for discriminative probabilistic modeling via MLE. This analysis provides (i) insights into the statistical error of GCL [10], and (ii) guidance on finding an approximation $\tilde{\mathbf{q}}$ better than the uniform one used by GCL as discussed in Remark 2. First, we state the necessary assumptions of our analysis.

Assumption 1. *There exist $c_1, c_2 > 0$ such that $\|\mathbf{o}\|_2 \le c_1$, $\|\mathbf{a}\|_2 \le c_2$ for any $\mathbf{o} \in \mathcal{O}, \mathbf{a} \in \mathcal{A}$.*

We focus on representation learning, where the prediction function $e_{\mathbf{w}}(\mathbf{o},\mathbf{a})$ is based on the inner product between the feature $e_1(\mathbf{w}_1;\mathbf{o})$ of $\mathbf{o}\in\mathcal{O}$ and the feature $e_2(\mathbf{w}_2;\mathbf{a})$ of $\mathbf{a}\in\mathcal{A}$, where $\mathbf{w}_1$ and $\mathbf{w}_2$ are the encoders+projection heads of the first and second views/modalities, respectively. In our theory, we consider the case that both $\mathbf{w}_1$ and $\mathbf{w}_2$ are L-layer neural networks[2] with positive-homogeneous and 1-Lipschitz continuous activation function $\sigma(\cdot)$ (e.g. ReLU).

Assumption 2. *Suppose that $e_1(\mathbf{w}_1;\mathbf{o})\in\mathbb{R}^{d_L}$, $e_2(\mathbf{w}_2;\mathbf{a})\in\mathbb{R}^{d_L}$ for some $d_L\geq 1$. Moreover, we have $\|e_1(\mathbf{w}_1;\mathbf{o})\|_2\leq\sqrt{c}$, $\|e_2(\mathbf{w}_2;\mathbf{a})\|_2\leq\sqrt{c}$ for some $c>0$ such that $e_{\mathbf{w}}(\mathbf{o},\mathbf{a})\in[-c,c]$.*

Based on the assumptions above, we provide a finite-sample generalization error bound between the empirical risk $\hat{\mathcal{L}}(\mathbf{w};\hat{\mathbf{O}},\hat{\mathbf{A}})$ in (3) and the true risk $\mathcal{L}(\mathbf{w})$ in (8).

Theorem 2. *Suppose that Assumptions* (1), (2) *hold. Consider the prediction function $e_{\mathbf{w}}$ parameterized by two branches of L-layer deep neural networks and an approximation $\tilde{\mathbf{q}}$ of $\mathbf{q}$, where $q^{(j)}=\sum_{j'=1}^n p(\mathbf{a}_j\mid\mathbf{o}_{j'})\geq\Omega(n)$ almost surely, $\forall j\in[n]$. With probability at least $1-\delta$, $\delta\in(0,1)$,*

$$\left|\hat{\mathcal{L}}(\mathbf{w};\hat{\mathbf{O}},\hat{\mathbf{A}})-\mathcal{L}(\mathbf{w})\right|\leq O\left(\frac{1}{n}+\sqrt{\frac{d_L}{n}}+\sqrt{\frac{\log(1/\delta)}{n}}+\mathcal{E}_{\mathbf{w}}(\tilde{\mathbf{q}},\mathbf{q};\hat{\mathbf{O}},\hat{\mathbf{A}})\right), \tag{9}$$

where $\mathcal{E}(\tilde{\mathbf{q}},\mathbf{q};\hat{\mathbf{O}},\hat{\mathbf{A}}):=\frac{1}{n}\sum_{i=1}^n\sum_{j=1}^n\left|\frac{1}{\tilde{q}^{(j)}}-\frac{1}{q^{(j)}}\right|\exp((e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_i)-c)/\tau)$ is an error term.

The proof can be found in Appendix I.

Remark 4. *(i) The global contrastive loss (GCL) with a uniform $\tilde{q}^{(j)}=\frac{n}{\mu(\mathcal{A})}$ leads to a* **non-diminishing** *error term $\mathcal{E}(\tilde{\mathbf{q}},\mathbf{q};\hat{\mathbf{O}},\hat{\mathbf{A}})$ when used as an objective for discriminative probabilistic modeling over a continuous domain; (ii) Moreover, the bias term $\mathcal{E}(\tilde{\mathbf{q}},\mathbf{q};\hat{\mathbf{O}},\hat{\mathbf{A}})$ vanishes when $\mathcal{A}$ is a finite set. Then, the result reproduces the classical result in the literature for supervised learning.*

D MIS with A General Weight Function for DPM

We consider the following MIS-based estimator with a size-m sample from each distribution $p_{\cdot|\mathbf{o}_j}$ and a general weight function $\boldsymbol{\omega}$ for the integral $g(\mathbf{w};\mathbf{o}_i,\mathcal{A})=\int_{\mathcal{A}}\exp(e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a})/\tau)\mu(d\mathbf{a})$. The estimator $\hat{g}(\mathbf{w};\mathbf{o}_i,\hat{\mathbf{A}})$ can be covered as a special case when $m=1$.

$$\hat{g}(\mathbf{w};\mathbf{o}_i,\hat{\mathbf{A}},\boldsymbol{\omega})=\sum_{j=1}^n\frac{1}{m}\sum_{l=1}^m\frac{\omega^{(j)}(\mathbf{a}_{j,l})}{p(\mathbf{a}_{j,l}\mid\mathbf{o}_j)}\exp\left(e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_j)/\tau\right),\quad\hat{\mathbf{A}}=\bigcup_{j=1}^n\{\mathbf{a}_{j,1},\dots,\mathbf{a}_{j,m}\}, \tag{10}$$

where $\boldsymbol{\omega}$ is a weighting function such that $\omega(\mathbf{a})$ is on a probability simplex, $\forall\mathbf{a}\in\mathcal{A}$. We denote $\hat{\mathbf{O}}:=\{\mathbf{o}_1,\dots,\mathbf{o}_n\}$, $\Xi_{i,j}(\boldsymbol{\omega},\mathbf{a}_{j,l}):=\frac{\omega^{(j)}(\mathbf{a}_{j,l})}{p(\mathbf{a}_{j,l}|\mathbf{o}_j)}\exp\left(e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_j)/\tau\right)$. We consider the "balance heuristic" $\omega_{\text{bl}}^{(j)}(\mathbf{a})=\frac{p(\mathbf{a}|\mathbf{o}_j)}{\sum_{j'=1}^n p(\mathbf{a}|\mathbf{o}_{j'})}$, $\forall\mathbf{a}\in\mathcal{A}$ and $\forall j\in[n]$ proposed in [5]. Proposition 1 shows the unbiasedness of estimator in (10) and justifies why we choose the balance heuristic.

Proposition 1. *For each $\boldsymbol{\omega}$, we have that $\hat{g}(\mathbf{w};\mathbf{o}_i,\hat{\mathbf{A}},\boldsymbol{\omega})$ is an unbiased estimator of the integral $g(\mathbf{w};\mathbf{o}_i,\mathcal{A})$; (ii) The balance heuristic $\boldsymbol{\omega}_{bl}$ minimizes $\frac{1}{m}\mathbf{E}[\sum_{j=1}^n\sum_{l=1}^m\Xi_{i,j}(\boldsymbol{\omega},\mathbf{a}_{j,l})^2\mid\hat{\mathbf{O}}]$ among all possible weighting functions for any i, where $\frac{1}{m}\mathbf{E}[\sum_{j=1}^n\sum_{l=1}^m\Xi_{i,j}(\boldsymbol{\omega},\mathbf{a}_{j,l})^2\mid\hat{\mathbf{O}}]$ is an upper bound of the variance $\mathrm{Var}[\hat{g}(\mathbf{w};\mathbf{o}_i,\hat{\mathbf{A}},\boldsymbol{\omega})\mid\hat{\mathbf{O}}]$; (iii) If $\sum_{j'=1}^n p(\mathbf{a}\mid\mathbf{o}_{j'})\geq\Omega(n)$ almost surely for any $\mathbf{a}\in\mathcal{A}$ and Assumptions 2 holds, the variance goes to zero when $n\to\infty$ or $m\to\infty$.*

Proof. Since for any $j\in[n]$ $\mathbf{a}_{j,1},\dots,\mathbf{a}_{j,m}$ are i.i.d. distributed, we have

$$\begin{aligned}
&\mathbf{E}\left[\hat{g}(\mathbf{w};\mathbf{o}_i,\hat{\mathbf{A}},\boldsymbol{\omega})\mid\hat{\mathbf{O}}\right]=\sum_{j=1}^n\mathbf{E}\left[\frac{\omega^{(j)}(\mathbf{a}_{j,1})}{p(\mathbf{a}_{j,1}\mid\hat{\mathbf{O}})}\exp\left(e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_{j,1})/\tau\right)\mid\hat{\mathbf{O}}\right]\\
&=\sum_{j=1}^n\int_{\mathcal{A}}\frac{\omega^{(j)}(\mathbf{a})}{p(\mathbf{a}\mid\mathbf{o}_j)}p(\mathbf{a}\mid\mathbf{o}_j)\exp\left(e_{\mathbf{w}}(O_i,\mathbf{a})/\tau\right)\mu(d\mathbf{a})\overset{\star}{=}\int_{\mathcal{A}}\sum_{j=1}^n\omega^{(j)}(\mathbf{a})\exp\left(e_{\mathbf{w}}(O_i,\mathbf{a})/\tau\right)\mu(d\mathbf{a})\\
&=\int_{\mathcal{A}}\exp\left(e_{\mathbf{w}}(O_i,\mathbf{a})/\tau\right)\mu(d\mathbf{a}),
\end{aligned} \tag{11}$$

[2] Our results could potentially be extended to other neural networks, such as ConvNets, using the corresponding Rademacher complexity bounds (See e.g., 28).

where $\star$ is due to Tonelli's theorem. We denote that $\Xi_{i,j}(\boldsymbol{\omega}, \mathbf{a}) \coloneqq \frac{\omega^{(j)}(\mathbf{a})}{p(\mathbf{a}|\mathbf{o}_j)} \exp\left(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a})/\tau\right)$. Since $\{\mathbf{a}_{j,l}\}_{j\in[n],l\in[m]}$ are mutually independent and for a specific j, $\mathbf{a}_{j,1}, \ldots, \mathbf{a}_{j,l}$ are also identically distributed, the variance of the estimator in (10) can be upper bounded as

$$\begin{aligned}\mathrm{Var}[\hat{g}(\mathbf{w}; \mathbf{o}_i, \hat{\mathbf{A}}, \boldsymbol{\omega}) \mid \hat{\mathbf{O}}] &= \frac{1}{m}\sum_{j=1}^{n} \mathbf{E}[\Xi_{i,j}(\boldsymbol{\omega}, \mathbf{a}_{j,1})^2 \mid \hat{\mathbf{O}}] - \frac{1}{m}\sum_{j=1}^{n} \mathbf{E}[\Xi_{i,j}(\omega^{(j)}, \mathbf{a}_{j,1}) \mid \hat{\mathbf{O}}]^2 \qquad (12)\\ &\leq \frac{1}{m}\sum_{j=1}^{n} \mathbf{E}[\Xi_{i,j}(\boldsymbol{\omega}, \mathbf{a}_{j,1})^2 \mid \hat{\mathbf{O}}] = \frac{1}{m}\sum_{j=1}^{n}\int_{\mathcal{A}} \frac{\omega^{(j)}(\mathbf{a})^2 \exp\left(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a})/\tau\right)^2}{p(\mathbf{a} \mid \mathbf{o}_j)} \mu(d\mathbf{a}).\end{aligned}$$

Due to Tonelli's theorem, we have

$$\sum_{j=1}^{n}\int_{\mathcal{A}} \frac{\omega^{(j)}(\mathbf{a})^2 \exp\left(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a})/\tau\right)^2}{p(\mathbf{a} \mid \mathbf{o}_j)} \mu(d\mathbf{a}) = \int_{\mathcal{A}}\sum_{j=1}^{n} \frac{\omega^{(j)}(\mathbf{a})^2 \exp\left(e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a})/\tau\right)^2}{p(\mathbf{a} \mid \mathbf{o}_j)} \mu(d\mathbf{a}).$$

We can instead minimize the variance upper bound at each $\mathbf{a}$ pointwise. Then, minimizing $\sum_{j=1}^{n} \frac{\omega^{(j)}(\mathbf{a})^2 \exp(e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a})/\tau)^2}{p(\mathbf{a}|\mathbf{o}_j)}$ subject to the simplex constraint leads to $\omega_{\mathrm{bl}}^{(j)}(\mathbf{a}) = \frac{p(\mathbf{a}|\mathbf{o}_j)}{\sum_{j'=1}^{n} p(\mathbf{a}|\mathbf{o}_{j'})}$. Plugging this into (12) and using Assumption 2 and $\sum_{j'=1}^{n} p(\mathbf{a} \mid \mathbf{o}_{j'}) \geq \Omega(n)$ a.s., we have

$$\mathrm{Var}[\hat{g}(\mathbf{w}; O_i, \hat{\mathbf{A}}, \boldsymbol{\omega}_{\mathrm{bl}}) \mid \hat{\mathbf{O}}] \leq \frac{1}{m}\sum_{j=1}^{n}\int_{\mathcal{A}} \frac{p(\mathbf{a} \mid O_j)\exp\left(e_{\mathbf{w}}(O_i, \mathbf{a})/\tau\right)^2}{(\sum_{j'=1}^{n} p(\mathbf{a} \mid O_{j'}))^2}\mu(d\mathbf{a}) = O\left(\frac{1}{mn}\right).$$

□

Interestingly, the minimizer $\boldsymbol{\omega}_{\mathrm{bl}}$ of $\frac{1}{m}\mathbf{E}[\sum_{j=1}^{n}\sum_{j=1}^{m} \Xi_{i,j}(\boldsymbol{\omega}, \mathbf{a}_{j,l})^2 \mid \hat{\mathbf{O}}]$ does not depend on $\mathbf{o}_i$. Plugging the balance heuristic $\boldsymbol{\omega}_{\mathrm{bl}}$ into (10), we can obtain the estimator $\hat{g}(\mathbf{w}; \mathbf{o}_i, \hat{\mathbf{A}})$ in the main paper.

E Performance of DPM on Downstream Zero-Shot Classification

Suppose that the true conditional density function $p(\mathbf{a} \mid \mathbf{o})$ is generated by some $\mathbf{w}_* \in \mathcal{W}$, i.e., $p(\mathbf{a} \mid \mathbf{o}) = p_{\mathbf{w}_*}(\mathbf{a} \mid \mathbf{o}) = \frac{\exp(e_{\mathbf{w}_*}(\mathbf{o},\mathbf{a})/\tau)}{\int_{\mathcal{A}} \exp(e_{\mathbf{w}_*}(\mathbf{o},\mathbf{a}')/\tau)\mu(d\mathbf{a}')}$. Then, the maximum likelihood estimator $\hat{\mathbf{w}}_* = \arg\max_{\mathbf{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^{n} \log p_{\mathbf{w}}(\mathbf{a}_i \mid \mathbf{o}_i)$ with the sample $\{(\mathbf{o}_i, \mathbf{a}_i)\}_{i=1}^{n}$ converges in probability to $\mathbf{w}_*$ under some mild assumptions (see Theorem 2.1 in [9]).

Let us consider the downstream multi-class classification problem with $K > 1$ distinct classes. The task is to predict the ground-truth label $y \in \{1, \ldots, K\}$ of a data point $\mathbf{o} \in \mathcal{O}$. Suppose that there are K subsets $\mathcal{A}_1, \ldots, \mathcal{A}_K$ of $\mathcal{A}$ and any $\mathbf{a} \in \mathcal{A}_k$ belongs to the k-th class. Moreover, assume that the ground-truth label $y(\mathbf{o})$ of data $\mathbf{o}$ is $y(\mathbf{o}) = \arg\max_{y\in[K]} \Pr(y \mid \mathbf{o})$. Given the model $\hat{\mathbf{w}}_*$ trained via MLE, the predicted label $s_{\hat{\mathbf{w}}_*}(\mathbf{o})$ of a data $\mathbf{o} \in \mathcal{O}$ can be obtained by the following 1-nearest neighbor (1-NN) classifier:

$$s_{\hat{\mathbf{w}}_*}(\mathbf{o}) = \arg\max_{k\in[K]} e_{\hat{\mathbf{w}}_*}(\mathbf{o}, \mathbf{a}_k),$$

where $\mathbf{a}_k \in \mathcal{A}$ is an example of the k-th class. For instance, the example $\mathbf{a}_k$ of the k-th class of the downstream image classification could be "a photo of {class_k}" when $\mathcal{O}$ is the image domain and $\mathcal{A}$ is the text domain [2]. Due to the monotonicity of the function $\exp(\cdot/\tau)$ and the expression of $p_{\mathbf{w}}$ in (1), we have $s_{\hat{\mathbf{w}}_*}(\mathbf{o}) = \arg\max_{k\in[K]} e_{\hat{\mathbf{w}}_*}(\mathbf{o}, \mathbf{a}_k) = \arg\max_{k\in[K]} p_{\hat{\mathbf{w}}}(\mathbf{a}_k \mid \mathbf{o})$. As long as the probability mass $\Pr(k \mid \mathbf{o})$ on class k is proportional to the probability density $p_{\mathbf{w}_*}(\mathbf{a}_k \mid \mathbf{o})$ on the example $\mathbf{a}_k$ of class k, the zero-one loss $\ell_{0/1}(\mathbf{o}, y(\mathbf{o}); \hat{\mathbf{w}}_*) = \mathbb{I}[s_{\hat{\mathbf{w}}_*}(\mathbf{o}) \neq y(\mathbf{o})]$ on the data-label pair $(\mathbf{o}, y(\mathbf{o}))$ of the downstream classification approaches zero when $\hat{\mathbf{w}}_* \xrightarrow{p} \mathbf{w}_*$.

F Proof of Theorem 1

Proof. The problem in (5) is equivalent to

$$\min_{\boldsymbol{\zeta}\in\mathbb{R}^n} \left\{\frac{1}{n}\sum_{i=1}^{n} \tau \log\left(\sum_{j=1}^{n} \exp((e(\mathbf{o}_i, \mathbf{a}_j) - \zeta^{(j)})/\tau)\right) + \frac{1}{n}\sum_{j=1}^{n} \zeta^{(j)}\right\}. \qquad (13)$$

We define that $\Phi(\boldsymbol{\zeta}) \coloneqq \frac{1}{n}\sum_{i=1}^n \tau \log\left(\sum_{j=1}^n \exp((e(\mathbf{o}_i, \mathbf{a}_j) - \zeta^{(j)})/\tau)\right) + \frac{1}{n}\sum_{j=1}^n \zeta^{(j)}$. Due to the first-order optimality condition, setting $\frac{\partial}{\partial \zeta^{(j)}}\Phi(\boldsymbol{\zeta})$ to 0 results in (6).

Due to the property of the log-sum-exp function and $e(\mathbf{o}_i, \mathbf{a}_j) \in [-c, c]$, we have

$$\Phi(\boldsymbol{\zeta}) \geq \frac{1}{n}\sum_{i=1}^n \max_{j\in[n]}\left\{e(\mathbf{o}_i, \mathbf{a}_j) - \zeta^{(j)}\right\} + \frac{1}{n}\sum_{j=1}^n \zeta^{(j)} \geq -c - \min_{j\in[n]} \zeta^{(j)} + \frac{1}{n}\sum_{j=1}^n \zeta^{(j)} \geq -c.$$

Thus, the function $\Phi(\boldsymbol{\zeta})$ is proper convex. Recall that the log-sum-exp function is affine on the diagonal and parallel lines $\boldsymbol{\zeta} = z\mathbf{1}_n + \mathbf{b}$, where $\mathbf{b} \in \mathbb{R}^n$ and $z \in \mathbb{R}$. Thus, $\Phi(\boldsymbol{\zeta})$ is affine on $\boldsymbol{\zeta} = z\mathbf{1}_n + \mathbf{b}$. On each line $\boldsymbol{\zeta} = z\mathbf{1}_n + \mathbf{b}$ with a specific $\mathbf{b} \in \mathbb{R}^n$ and varying $z \in \mathbb{R}$, we have

$$\begin{aligned}
\Phi(\boldsymbol{\zeta}) &= \frac{1}{n}\sum_{i=1}^n \tau \log\left(\sum_{j=1}^n \exp((e(\mathbf{o}_i, \mathbf{a}_j) - z + b^{(j)})/\tau)\right) + z + \frac{1}{n}\sum_{j=1}^n b^{(j)} \\
&= \frac{1}{n}\sum_{i=1}^n \tau \log\left(\exp(-z/\tau)\sum_{j=1}^n \exp((e(\mathbf{o}_i, \mathbf{a}_j) - b^{(j)})/\tau)\right) + z + \frac{1}{n}\sum_{j=1}^n b^{(j)} \\
&= \frac{1}{n}\sum_{i=1}^n \tau \log\left(\sum_{j=1}^n \exp((e(\mathbf{o}_i, \mathbf{a}_j) - b^{(j)})/\tau)\right) + \frac{1}{n}\sum_{j=1}^n b^{(j)}.
\end{aligned}$$

Note that the expression on the R.H.S is fixed when z varies, i.e., $\Phi(\boldsymbol{\zeta})$ has zero directional derivatives along each of diagonal and parallel lines $\boldsymbol{\zeta} = z\mathbf{1}_n + \mathbf{b}$. Recall that the log-sum-exp function is strictly convex along any direction other than the diagonal and parallel lines $\boldsymbol{\zeta} = \mathbf{1}_n + \mathbf{b}$. Since a sum of strictly convex functions is strictly convex and $\frac{1}{n}\sum_{j=1}^n \zeta^{(j)}$ is affine, $\Phi(\boldsymbol{\zeta})$ is also strictly convex along any direction other than the diagonal and parallel lines $\boldsymbol{\zeta} = z\mathbf{1}_n + \mathbf{b}$.

Note that each $\boldsymbol{\zeta} \in \mathbb{R}^n$ is uniquely located on a line $\boldsymbol{\zeta} = z\mathbf{1}_n + \mathbf{b}$ for some specific $\mathbf{b}$ and the function values $\Phi(\boldsymbol{\zeta})$ of different points on the same line $\boldsymbol{\zeta} = z\mathbf{1}_n + \mathbf{b}$ are the same. Thus, if $\boldsymbol{\zeta}_*$ is a minimum of $\Phi(\boldsymbol{\zeta})$, then any point on the line $\zeta = z\mathbf{1}_n + \mathbf{b}_*$ is a minimum of $\Phi(\boldsymbol{\zeta})$, where $\mathbf{b}_*$ is uniquely determined by $\boldsymbol{\zeta}_*$. Since the set of minima of a convex function is convex, there may exist an uncountably infinite number of consecutive lines parallel to the diagonal such that each point on those lines is a minimum of $\Phi(\boldsymbol{\zeta})$. However, we can rule out such a possibility since $\Phi(\boldsymbol{\zeta})$ is strictly convex in any direction other than $\boldsymbol{\zeta} = z\mathbf{1}_n + \mathbf{b}$ such that points on two consecutive lines parallel to the diagonal cannot be minimums simultaneously. Thus, there exists a unique $\mathbf{b}_* \in \mathbb{R}^n$ such that any point on the line $\boldsymbol{\zeta} = z\mathbf{1}_n + \mathbf{b}_*$ is a minimum of $\Phi(\boldsymbol{\zeta})$, i.e., the minimum of $\Phi(\boldsymbol{\zeta})$ is unique up to an arbitrary scalar additive term $z \in \mathbb{R}$. Finally, notice that $\tau \log \mathbf{q}$ is approximately on this line of minima $\boldsymbol{\zeta} = z\mathbf{1}_n + \mathbf{b}_*$ by comparing (4) and (6).

□

G NUCLR for Self-Supervised Representation Learning

The problem in (7) can be formulated as a finite-sum compositional optimization problem [11].

$$\begin{aligned}
&\min_{\mathbf{w},\boldsymbol{\zeta}} \hat{\mathcal{L}}(\mathbf{w}, \boldsymbol{\zeta}) = \frac{1}{n}\sum_{i=1}^n \log\left(\frac{1}{n-1}\exp(-\zeta^{(i)}/\tau) + g_i(\mathbf{w}, \boldsymbol{\zeta})\right) + \frac{1}{n}\sum_{j=1}^n \zeta^{(j)}, \\
&g_i(\mathbf{w}, \boldsymbol{\zeta}) = \frac{1}{n-1}\sum_{j\in\mathcal{S}_i^-} \exp((e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j) - e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j) - \zeta^{(j)})/\tau), \quad \mathcal{S}_i^- \coloneqq \{1, , \dots, n\}\backslash\{i\}.
\end{aligned}$$

In each iteration, we first sample a mini-batch of pairs $\{(\mathbf{o}_i, \mathbf{s}_i)\}_{i\in\mathcal{B}}$. Based on the sampled mini-batch, we can construct unbiased estimators $\tilde{g}_i(\mathbf{w}, \boldsymbol{\zeta}; \mathcal{B})$, $\nabla_{\mathbf{w}}\tilde{g}_i(\mathbf{w}, \boldsymbol{\zeta}; \mathcal{B})$, $\frac{\partial}{\partial \zeta^{(j)}}\tilde{g}_i(\mathbf{w}, \boldsymbol{\zeta}; \mathcal{B})$ of $g_i(\mathbf{w}, \boldsymbol{\zeta})$, $\nabla_{\mathbf{w}} g_i(\mathbf{w}, \boldsymbol{\zeta})$, and $\frac{\partial}{\partial \zeta^{(j)}} g_i(\mathbf{w}, \boldsymbol{\zeta})$. However, directly combining these unbiased estimators does not lead to unbiased estimators of $\nabla_{\mathbf{w}}\hat{\mathcal{L}}(\mathbf{w}, \boldsymbol{\zeta})$ and $\nabla_{\zeta^{(j)}}\hat{\mathcal{L}}(\mathbf{w}, \boldsymbol{\zeta})$ because the problem is compositional. Consequently, the resulting algorithm requires a large batch size $|\mathcal{B}|$ to converge.

Motivated by the SOX algorithm [11] for general FCCO problems and the SogCLR algorithm [10] for GCL, we propose NUCLR (Algorithm 1) to minimize the loss in (7). First, we keep track of an exponential moving average (EMA) estimator $u^{(i)}$ of $g_i(\mathbf{w}, \boldsymbol{\zeta})$ for each $i \in [n]$ as in Step 5 in Algorithm 1 to resolve the large batch issue. Based on $\{u^{(i)}\}_{i\in\mathcal{B}}$, the stochastic estimator of $\nabla_{\mathbf{w}}\hat{\mathcal{L}}(\mathbf{w}, \boldsymbol{\zeta})$ can be computed as in Step 6 in Algorithm 1. Then, we can update the model parameter $\mathbf{w}$ based on an optimizer, e.g., AdamW. Next, we update the auxiliary variable $\boldsymbol{\zeta}$ based on the mini-batch $\mathcal{B}$ and the EMA estimators $\{u^{(i)}\}_{i\in\mathcal{B}}$. To efficiently update the n-dimensional variable $\boldsymbol{\zeta}$, we adopt the randomized block coordinate approach [29]: We only update those $\zeta^{(j)}, j \in \mathcal{B}$ for one step by a gradient-based optimizer while keeping $\zeta^{(j)}, j \notin \mathcal{B}$ unchanged. Based on $\{u^{(i)}\}_{i\in\mathcal{B}}$, the stochastic estimator of the partial derivatives $\frac{\partial}{\partial \zeta^{(j)}}\hat{\mathcal{L}}(\mathbf{w}, \boldsymbol{\zeta})$ for any j in the minibatch $\mathcal{B}$ can be computed as in Step 9 in Algorithm 1.

Algorithm 1 NUCLR Algorithm for Self-Supervised Representation Learning

1: Initialize $\mathbf{w}_0, \mathbf{u}_0, \boldsymbol{\zeta} = \zeta_0 \mathbf{1}_n$ and set up $\xi_0 > \zeta_0, \eta, \gamma$
2: **for** $t = 0, 1 \ldots, T-1$ **do**
3: Sample $\mathcal{B}_t \subset \{1, \ldots, n\}$
4: Compute $\Sigma_t^{(i,j)} = e_{\mathbf{w}_t}(\mathbf{o}_i, \mathbf{a}_j) - e_{\mathbf{w}_t}(\mathbf{o}_i, \mathbf{a}_i)$ for $i, j \in \mathcal{B}_t$
5: Update $u_{t+1}^{(i)} = \begin{cases} (1-\gamma)u_t^{(i)} + \gamma \frac{1}{B-1}\sum_{j\in\mathcal{B}_t\setminus\{i\}} \exp((\Sigma_t^{(i,j)} - \zeta_t^{(j)})/\tau), & i \in \mathcal{B}_t \\ u_t^{(i)}, & i \notin \mathcal{B}_t \end{cases}$
6: Compute $\hat{G}(\mathbf{w}_t) = \frac{1}{B}\sum_{i\in\mathcal{B}_t} \frac{1}{u_{t+1}^{(i)} + \frac{1}{n-1}\exp(-\xi_t/\tau)} \left(\frac{1}{B-1}\sum_{j\in\mathcal{B}_t\setminus\{i\}} \exp((\Sigma_t^{(i,j)} - \zeta_t^{(j)})/\tau)\nabla_{\mathbf{w}}\Sigma_t^{(i,j)}\right)$
7: Update $\mathbf{w}_{t+1}$ by a momentum or adaptive method with $\hat{G}(\mathbf{w}_t)$ as the gradient estimator
8: Compute $\hat{G}(\zeta_t^{(j)}) = -\frac{1}{n-1}\frac{1}{B}\sum_{i\in\mathcal{B}_t} \frac{1}{u_{t+1}^{(i)} + \frac{1}{n-1}\exp(-\zeta_t^{(i)}/\tau)} \exp((\Sigma_t^{(i,j)} - \zeta_t^{(j)})/\tau) + \frac{1}{n}$ for $j \in \mathcal{B}_t$
9: Update $\zeta_{t+1}^{(j)} = \begin{cases} \zeta_t^{(j)} - \eta\hat{G}(\zeta_t^{(j)}), & j \in \mathcal{B}_t \\ \zeta_t^{(j)}, & j \notin \mathcal{B}_t \end{cases}$
10: Update $\xi_{t+1} = \max\{\xi_0, \max_{j\in[n]} \zeta_{t+1}^{(j)}\}$
11: **end for**

Computational and Memory Overheads of NUCLR: Compared to the $O(Bd)$ per-iteration computational cost of the SimCLR/CLIP algorithm [7, 2], our proposed NUCLR leads to a computational overhead $O(B)$ similar to SogCLR [10] for updating the scalars $\{u^{(i)}\}_{i\in\mathcal{B}_t}$ and $\{\zeta^{(i)}\}_{i\in\mathcal{B}_t}$. This extra $O(B)$ cost can be ignored since d is extremely large in modern deep neural networks. Owing to the moving average estimator $\mathbf{u}$, our NUCLR does not require a huge batch size B for good performance, unlike SimCLR/CLIP. Thus, NUCLR is also more memory-efficient, making it suitable for environments with limited GPU resources, similar to SogCLR. NUCLR needs to store one extra n-dimensional vector $\boldsymbol{\zeta}$. Maintaining $\boldsymbol{\zeta}$ in GPU only requires less than 100MB for 12 million data points, which is negligible compared to the GPU memory required for backpropagation. Moreover, we may instead maintain the vector $\boldsymbol{\zeta}$ in CPU and only transfer those needed $\{\zeta^{(j)}\}_{j\in\mathcal{B}_t}$ to GPU in each iteration. The overhead can be further reduced by overlapping communication and computation.

Freeze period of $\boldsymbol{\zeta}$: At the beginning of training when $\mathbf{w}$ is far from $\mathbf{w}_*$, then the optimal $\boldsymbol{\zeta}$ in (7) may be far from the optimal solution to (5). So the learned ζ values at the earlier iterations may not be accurate enough, which could hurt the learning. To mitigate this issue, we freeze $\boldsymbol{\zeta}$ in the first T_0 iterations, where T_0 is much smaller than the total number of iterations T.

Downweighting the Positive Pairs: In (7), the denominator $\sum_{j=1}^{n}\exp((e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j) - \zeta^{(j)})/\tau)$ in the log-likelihood can be seen as the weighted variant $\sum_{j=1}^{n}\exp(-\zeta^{(j)}/\tau)\exp((e_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j))/\tau)$ of the standard term in GCL, where $\exp(-\zeta^{(j)}/\tau)$ can be viewed as the "strength" of pushing $\mathbf{a}_j$ away from $\mathbf{o}_i$. In each iteration of our algorithm, the gradient w.r.t. $\mathbf{w}$ is computed using the current value of the auxiliary variable $\boldsymbol{\zeta} \in \mathbb{R}^n$, whose all coordinates are updated from the same initialized value $\zeta_0 \in \mathbb{R}$. Consequently, we assign almost the same weight to the positive pair $(\mathbf{o}_i, \mathbf{a}_i)$ and negative pairs $\{(\mathbf{o}_i, \mathbf{a}_j)\}_{j\neq i}$ at the beginning of training, which may slow down the learning process. To address this issue, we introduce a scalar $\xi_t = \|\boldsymbol{\zeta}_t\|_\infty$ to reduce the weight of positive pair $(\mathbf{o}_i, \mathbf{a}_i)$ from $\exp(-\zeta_t^{(i)}/\tau)$ to $\exp(-\xi_t/\tau)$, which prevents the positive pair has a larger weight than negative pairs. The value of ξ_t is updated at the end of each iteration. It is worth noting that the value of ξ_t is *adaptively* updated in Algorithm 1 and there is no need to tune it as a hyperparameter.

Margin Interpretation of NUCLR: Cross-entropy and contrastive losses with an additive margin $m > 0$ have been widely studied in the literature [30, 31, 32, 33, 34, 35], which can be viewed as a smooth version of the hinge loss to separate the matching (positive) pair $(\mathbf{o}_i, \mathbf{a}_i)$ from negative pairs $\{(\mathbf{o}_i, \mathbf{a}_j) \mid \mathbf{a}_j \neq \mathbf{a}_i, \mathbf{a}_j \in \mathcal{A}\}$. In supervised learning tasks such as face verification and multi-class classification, using a relatively large margin has been shown to be beneficial [32, 33]. However, the "false negative" issue is more pronounced in self-supervised learning. Determining the appropriate margin becomes more difficult, as aggressively and uniformly pushing away all positive and negative pairs may hurt the performance [36]. As shown in Line 7 of Algorithm 1, our NUCLR algorithm adopts an *individualized* margin $-\zeta^{(j)}$ for each negative data $\mathbf{a}_j$ when updating the model parameter $\mathbf{w}$. Rather than relying on an expensive grid search for individualized margins, our method learns them in a principled way. Intuitively, the margin between $(\mathbf{o}_i, \mathbf{a}_i)$ and $(\mathbf{o}_i, \mathbf{a}_j)$ should be smaller when $\mathbf{a}_j$ is popular, as it is more likely to be a false negative. We observe that $\zeta^{(j)}$ can also serve as a measure of the popularity since $\tilde{q}^{(j)} \propto \exp(\zeta^{(j)}/\tau)$ when $\zeta^{(j)}$ is optimized. As a result, NUCLR can help *tolerate* potential false negatives because the margin $-\zeta^{(j)}$ between pairs $(\mathbf{o}_i, \mathbf{a}_i)$ and $(\mathbf{o}_i, \mathbf{a}_j)$ is smaller when the popularity proxy $\zeta^{(j)}$ is larger.

H Detailed Settings of Experiments on Bimodal Representation Learning

The training set of CC3M contains $n = 2,723,200$ image-text pairs, while that of CC12M contains $n = 9,184,256$ image-text pairs. We evaluate the performance of trained models on downstream zero-shot image-text retrieval and image classification tasks. Retrieval performance is evaluated on the test splits of the Flickr30k [37] and MSCOCO [38] datasets, in terms of the mean Recall@1 score for image-to-text and text-to-image retrievals. The top-1 classification accuracy is evaluated on the ImageNet1k [39] and CIFAR100 [40] datasets. We compare our proposed NUCLR algorithm with baselines CLIP [2], SigLIP [14], DCL [41],CyCLIP [15], and SogCLR [10].

We focus on the limited-resource setting: All experiments utilize distributed data-parallel (DDP) training on two NVIDIA A100 GPUs with 40GB memory and the total batch size B in each iteration is 512. Besides, we use ResNet-50 as the vision encoder and DistilBert as the text encoder. The output embedding of each encoder is projected by a linear layer into a 256-dimensional feature representation for computing the losses. We run each algorithm 3 times with different random seeds and each run contains 30 epochs. We tune the hyperparameters of all algorithms based on the performance on the validation splits. The optimizer for the model parameter $\mathbf{w}$ is AdamW [42] with a weight decay of 0.02 and a cosine annealing learning rate schedule [43]. For all algorithms, we choose a fixed temperature parameter τ tuned within $\{0.005, 0.01, 0.03, 0.05\}$. It is worth noting that both our algorithm and the baselines have the option to set the temperature τ as a learnable parameter or utilize some more sophisticated strategies [44, 45, 46]. However, we do not explore that in this paper. For SogCLR and our algorithm NUCLR, we set $\gamma = 0.8$. For our NUCLR, we select $\zeta_0 = -0.05$ on the CC3M dataset and $\zeta_0 = 0$ on the CC12M dataset. Besides, we freeze $\boldsymbol{\zeta}$ in the first 5 epochs.

I Proof of Theorem 2

The structure of our proof is as follows:

- Section I.1 presents necessary lemmas for our generalization analysis;
- Section I.2 decomposes the generalization error into two parts, which are handled by Section I.3 and Section I.4, respectively;
- Section I.5 provides bounds for Rademacher complexities of function classes parameterized by deep neural networks.

The main theorem can be proved by combining (15), (16), (17), (18), (19), (22), (25), (26).

I.1 Lemmas

The following two lemmas provide contraction lemmas on Rademacher complexities. Lemma 1 considers the class of real-valued functions, and Lemma 2 considers the class of vector-valued functions [47, 25]. Let ϵ_i and $\epsilon_{i,j}$ be Rademacher variables.

Lemma 1 (Contraction Lemma, Thm 11.6 in [48]). *Let $\tau : \mathbb{R}_+ \mapsto \mathbb{R}_+$ be convex and nondecreasing. Suppose $\psi : \mathbb{R} \mapsto \mathbb{R}$ is contractive ($|\psi(t) - \psi(\tilde{t})| \le G|t - \tilde{t}|$) and $\psi(0) = 0$. Then for any $\tilde{\mathcal{F}}$ we have*

$$\mathbf{E}_{\boldsymbol{\epsilon}}\tau\Big(\sup_{f\in\tilde{\mathcal{F}}}\sum_{i=1}^{n}\epsilon_i\psi\big(f(x_i)\big)\Big) \le \mathbf{E}_{\boldsymbol{\epsilon}}\tau\Big(G\sup_{f\in\tilde{\mathcal{F}}}\sum_{i=1}^{n}\epsilon_i f(x_i)\Big).$$

We say that a function $\psi : \mathbb{R}^d \to \mathbb{R}$ is G-Lipschitz continuous w.r.t. $\|\cdot\|_2$ if $|\psi(x) - \psi(\mathbf{x})| \le G\|\mathbf{x} - \mathbf{x}'\|_2$ for a $G > 0$ and any $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$.

Lemma 2. *Let $\mathcal{F}$ be a class of bounded functions $f : \mathcal{Z} \mapsto \mathbb{R}^d$ which contains the zero function. Let $\tau : \mathbb{R}_+ \to \mathbb{R}_+$ be a continuous, non-decreasing, and convex function. Assume $\tilde{g}_1, \ldots, \tilde{g}_n : \mathbb{R}^d \to \mathbb{R}$ are G-Lipschitz continuous w.r.t. $\|\cdot\|_2$ and satisfy $\tilde{g}_i(\mathbf{0}) = 0$. Then*

$$\mathbf{E}_{\boldsymbol{\epsilon}\sim\{\pm1\}^n}\tau\Big(\sup_{f\in\mathcal{F}}\sum_{i=1}^{n}\epsilon_i\tilde{g}_i(f(\mathbf{x}_i))\Big) \le \mathbf{E}_{\boldsymbol{\epsilon}\sim\{\pm1\}^{nd}}\tau\Big(G\sqrt{2}\sup_{f\in\mathcal{F}}\sum_{i=1}^{n}\sum_{j=1}^{d}\epsilon_{i,j}f_j(\mathbf{x}_i)\Big). \tag{14}$$

The following lemma estimates the moment generation function of a Rademacher chaos variable of order 2 [49].

Lemma 3. *Let $\epsilon_i, i \in [n]$ be independent Rademacher variables. Let $a_{i,j} \in \mathbb{R}, i,j \in [n]$. Then for $Z = \sum_{1\le i<j\le n}\epsilon_i\epsilon_j a_{ij}$ we have*

$$\mathbf{E}_{\boldsymbol{\epsilon}}\exp\Big(|Z|/(4es)\Big) \le 2, \quad \textit{where } s^2 := \sum_{1\le i<j\le n}a_{i,j}^2.$$

The following lemma is a version of Talagrand's contraction lemma.

Lemma 4 (Lemma 8 in [50]). *Let $\mathcal{H}$ be a hypothesis set of functions mapping $\mathcal{X}$ to $\mathbb{R}$ and ψ is G-Lipschitz functions for some $G > 0$. Then, for any sample S of n points $x_1, \ldots, x_n \in \mathcal{X}$, the following inequality holds.*

$$\frac{1}{n}\mathbf{E}_{\epsilon_{1:n}}\left[\sup_{h\in\mathcal{H}}\sum_{i=1}^{n}\epsilon_i\psi(h(x_i))\right] \le \frac{G}{n}\mathbf{E}_{\epsilon_{1:n}}\left[\sup_{h\in\mathcal{H}}\sum_{i=1}^{n}\epsilon_i h(x_i)\right].$$

I.2 Error Decomposition

Considering $\log_e x \le x - 1$ for any $x > 0$, we have

$$\begin{aligned}
&\hat{\mathcal{L}}(\mathbf{w}; \hat{\mathbf{O}}, \hat{\mathbf{A}}) - \mathcal{L}(\mathbf{w}) \\
&= \mathbf{E}_{\mathbf{o},\mathbf{a}}[e_{\mathbf{w}}(\mathbf{o},\mathbf{a})] - \frac{1}{n}\sum_{i=1}^{n}e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_i) + \frac{1}{n}\sum_{i=1}^{n}\tau\log(\tilde{g}(\mathbf{w};\mathbf{o}_i,\hat{\mathbf{A}})) - \mathbf{E}\left[\tau\log g(\mathbf{w};\mathbf{o},\mathcal{A})\right] \\
&= \mathbf{E}_{\mathbf{o},\mathbf{a}}[e_{\mathbf{w}}(\mathbf{o},\mathbf{a})] - \frac{1}{n}\sum_{i=1}^{n}e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_i) + \frac{1}{n}\sum_{i=1}^{n}\mathbf{E}_{\mathbf{o}}\left[\tau\log\frac{\sum_{j=1}^{n}\frac{1}{\tilde{q}^{(j)}}\exp((e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_j)-c)/\tau)}{\int_{\mathcal{A}}\exp((e_{\mathbf{w}}(\mathbf{o},\mathbf{a})-c)/\tau)\mu(d\mathbf{a})}\right] \\
&\le \underbrace{\mathbf{E}_{\mathbf{o},\mathbf{a}}[e_{\mathbf{w}}(\mathbf{o},\mathbf{a})] - \frac{1}{n}\sum_{i=1}^{n}e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_i)}_{\text{I}} + \underbrace{\frac{\underline{C}}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{\tilde{q}^{(j)}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_j)) - \underline{C}\mathbf{E}_{\mathbf{o}}\left[\int_{\mathcal{A}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a}))\mu(d\mathbf{a})\right]}_{\text{II}},
\end{aligned} \tag{15}$$

where we define $\bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a}) \coloneqq \frac{e_{\mathbf{w}}(\mathbf{o},\mathbf{a})-c}{\tau} \in [-2c/\tau, 0]$ such that $\exp(\bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a})) \in [\exp(-2c/\tau), 1]$. Besides, and $\overline{C} \coloneqq \sup_{\mathbf{o}\in\mathcal{O}}\frac{\tau}{\int_{\mathcal{A}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a}))\mu(d\mathbf{a})}$. Due to Assumption 2, $\overline{C} \le \frac{\tau\exp(2c/\tau)}{\mu(\mathcal{A})} < \infty$. In practice, $\overline{C}$ could be much smaller than the worst-case value $\frac{\tau\exp(2c/\tau)}{\mu(\mathcal{A})}$. Similarly, we have

$$\begin{aligned}
\mathcal{L}(\mathbf{w}) - \hat{\mathcal{L}}(\mathbf{w}; \hat{\mathbf{O}}, \hat{\mathbf{A}}) &\le \frac{1}{n}\sum_{i=1}^{n}e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_i) - \mathbf{E}[e_{\mathbf{w}}(\mathbf{o},\mathbf{a})] \\
&\quad + \overline{C}'\mathbf{E}\left[\int_{\mathcal{A}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a}))\mu(d\mathbf{a})\right] - \frac{\overline{C}'}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{\tilde{q}^{(j)}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_j)),
\end{aligned} \tag{16}$$

where $\overline{C}' = \frac{\tau\|\tilde{q}\|_\infty}{n}\exp(2c/\tau)$.

I.3 Bounding Term I

Define the function class $\mathcal{E} \coloneqq \{(\mathbf{o},\mathbf{a}) \mapsto e_{\mathbf{w}}(\mathbf{o},\mathbf{a}) \mid \mathbf{w} \in \mathcal{W}\}$. Since $(\mathbf{o}_1,\mathbf{a}_1),\dots,(\mathbf{o}_n,\mathbf{a}_n)$ are i.i.d. and Assumption 1 ($e_{\mathbf{w}}(\mathbf{o},\mathbf{a}) \in [-c,c]$ for any $\mathbf{w} \in \mathcal{W}$), we can apply the McDiarmid's inequality to $\mathbf{E}_{\mathbf{o},\mathbf{a}}[e_{\mathbf{w}}(\mathbf{o},\mathbf{a})] - \frac{1}{n}\sum_{i=1}^n e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_i)$ and utilize the symmeterization argument following Theorem 3.3 in [51]. With probability at least $1-\frac{\delta}{4}$,

$$\mathbf{E}_{\mathbf{o},\mathbf{a}}[e_{\mathbf{w}}(\mathbf{o},\mathbf{a})] \le \frac{1}{n}\sum_{i=1}^n e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_i) + 2\mathfrak{R}_n(\mathcal{E}) + 6c\sqrt{\frac{\log(8/\delta)}{2n}},$$

where $\mathfrak{R}_n(\mathcal{E}) \coloneqq \mathbf{E}_{\hat{\mathbf{O}},\hat{\mathbf{A}}}[\hat{\mathfrak{R}}_n^+(\mathcal{E})]$, $\hat{\mathfrak{R}}_n^+(\mathcal{E}) \coloneqq \mathbf{E}_{\epsilon_{1:n}}\left[\sup_{e\in\mathcal{E}} \frac{1}{n}\sum_{i=1}^n \epsilon_i e(\mathbf{o}_i,\mathbf{a}_i)\right]$ is the empirical Rademacher complexity of $\mathcal{E}$ on the sample $\hat{\mathbf{O}} \times \hat{\mathbf{A}}$, and $\epsilon_1,\dots,\epsilon_n$ are Rademacher random variables. Similarly, we can also apply McDiarmid's inequality to $\frac{1}{n}\sum_{i=1}^n e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_i) - \mathbf{E}_{\mathbf{o},\mathbf{a}}[e_{\mathbf{w}}(\mathbf{o},\mathbf{a})]$ and then use the symmetrization argument. With probability at least $1-\frac{\delta}{4}$,

$$\frac{1}{n}\sum_{i=1}^n e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_i) \le \mathbf{E}_{\mathbf{o},\mathbf{a}}[e_{\mathbf{w}}(\mathbf{o},\mathbf{a})] + 2\mathfrak{R}_n(\mathcal{E}) + 6c\sqrt{\frac{\log(8/\delta)}{2n}},$$

Thus, with probability at least $1-\frac{\delta}{2}$, we have

$$\left|\frac{1}{n}\sum_{i=1}^n e_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_i) - \mathbf{E}_{\mathbf{o},\mathbf{a}}[e_{\mathbf{w}}(\mathbf{o},\mathbf{a})]\right| \le 2\mathfrak{R}_n(\mathcal{E}) + 6c\sqrt{\frac{\log(8/\delta)}{2n}}. \tag{17}$$

I.4 Bounding Term II

We decompose the term II in (15) as follows.

$$\begin{aligned}
\mathrm{II} &= \frac{1}{n}\sum_{i=1}^n\sum_{j=1}^n \frac{1}{\tilde{q}^{(j)}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_j)) - \mathbf{E}_{\mathbf{o}}\left[\int_{\mathcal{A}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a}))\mu(d\mathbf{a})\right] \\
&= \underbrace{\frac{1}{n}\sum_{i=1}^n\sum_{j=1}^n\left(\frac{1}{\tilde{q}^{(j)}} - \frac{1}{q^{(j)}}\right)\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_j))}_{\mathrm{II.a}} + \underbrace{\frac{1}{n}\sum_{i=1}^n\sum_{j=1}^n \frac{1}{q^{(j)}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_j)) - \mathbf{E}_{\mathbf{o}}\left[\int_{\mathcal{A}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a}))\mu(d\mathbf{a})\right]}_{\mathrm{II.b}}.
\end{aligned} \tag{18}$$

Thus, we have $|\mathrm{II}| \le |\mathrm{II.a}| + |\mathrm{II.b}|$.

Since $\exp(\bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a})) = \exp((e_{\mathbf{w}}(\mathbf{o},\mathbf{a}) - c)/\tau) \le 1$ for any $\mathbf{o} \in \mathcal{O}, \mathbf{a} \in \mathcal{A}$, we have

$$|\mathrm{II.a}| \le \frac{1}{n}\sum_{i=1}^n\sum_{j=1}^n\left|\frac{1}{\tilde{q}^{(j)}} - \frac{1}{q^{(j)}}\right|\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_j)) \le \sum_{j=1}^n\left|\frac{1}{\tilde{q}^{(j)}} - \frac{1}{q^{(j)}}\right|. \tag{19}$$

We define $\Psi(\hat{\mathbf{O}},\hat{\mathbf{A}}) \coloneqq= \sup_{\mathbf{w}}\left\{\frac{1}{n}\sum_{i=1}^n\sum_{j=1}^n \frac{1}{q^{(j)}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i,\mathbf{a}_j)) - \mathbf{E}_{\mathbf{o}}\left[\int_{\mathcal{A}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a}))\mu(d\mathbf{a})\right]\right\}$. We denote that $\hat{\mathbf{O}}_\ell = (\hat{\mathbf{O}}\backslash\{\mathbf{o}_\ell\}) \cup \{\mathbf{o}'_\ell\}$, $\hat{\mathbf{A}}_\ell = (\hat{\mathbf{A}}\backslash\{\mathbf{a}_\ell\}) \cup \{\mathbf{a}'_\ell\}$, where $(\mathbf{o}'_1,\mathbf{a}'_1),\dots,(\mathbf{o}'_n,\mathbf{a}'_n)$ are i.i.d. to $(\mathbf{o}_1,\mathbf{a}_1),\dots,(\mathbf{o}_n,\mathbf{a}_n)$. We denote that $q(\mathbf{a};\hat{\mathbf{O}}) \coloneqq \sum_{\mathbf{o}\in\hat{\mathbf{O}}}^n p(\mathbf{a} \mid \mathbf{o})$ such that $q^{(j)} = q(\mathbf{a}_j;\hat{\mathbf{O}})$. If $q^{(j)} = \sum_{j'=1}^n p(\mathbf{a}_j \mid \mathbf{o}_{j'}) \ge \Omega(n)$ almost surely, we have

$$\begin{aligned}
|\Psi(\hat{\mathbf{O}},\hat{\mathbf{A}}) - \Psi(\hat{\mathbf{O}}_\ell,\hat{\mathbf{A}})| &= \left|\sup_{\mathbf{w}}\frac{1}{n}\sum_{j=1}^n\frac{1}{q^{(j)}}\exp(\bar{e}_{\mathbf{w}}(O_\ell,A_j)) - \sup_{\mathbf{w}}\frac{1}{n}\sum_{j=1}^n\frac{1}{q(A_j;\hat{\mathbf{O}}_\ell)}\exp(\bar{e}_{\mathbf{w}}(O'_\ell,A_j))\right| \le O(1/n), \\
|\Psi(\hat{\mathbf{O}},\hat{\mathbf{A}}) - \Psi(\hat{\mathbf{O}},\hat{\mathbf{A}}_\ell)| &= \left|\sup_{\mathbf{w}}\frac{1}{n}\sum_{i=1}^n\frac{1}{q(A_\ell;\hat{\mathbf{O}})}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i,A_\ell)) - \sup_{\mathbf{w}}\frac{1}{n}\sum_{i=1}^n\frac{1}{q(A'_\ell;\hat{\mathbf{O}})}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i,A'_\ell))\right| \le O(1/n).
\end{aligned}$$

Since $\mathbf{o}_i$ and A_j are mutually dependent only when $i = j$, we then apply the McDiarmid-Type inequalities for graph-dependent variables (Theorem 3.6 in [52]) to the term II.b and –II.b. With probability at least $1-\frac{\delta}{4}$, $\delta \in (0,1)$, we have

$$\mathrm{II.b} \le \mathbf{E}\left[\sup_{\mathbf{w}}\mathrm{II.b}\right] + O\left(\sqrt{\frac{10\log(4/\delta)}{n}}\right). \tag{20}$$

Similarly, with probability at least $1 - \frac{\delta}{4}$, $\delta \in (0,1)$, we have

$$-\text{II.b} \le \mathbf{E}\left[\sup_{\mathbf{w}} \{-\text{II.b}\}\right] + O\left(\sqrt{\frac{10\log(4/\delta)}{n}}\right). \tag{21}$$

Let $(\mathbf{o}_1', \mathbf{a}_1'), \ldots, (\mathbf{o}_n', \mathbf{a}_n')$ be a virtual sample i.i.d. to $(\mathbf{o}_1, \mathbf{a}_1), \ldots, (\mathbf{o}_n, \mathbf{a}_n)$. Denote that $\hat{\mathbf{O}}' \coloneqq \{\mathbf{o}_1', \ldots, \mathbf{o}_n'\}$, $\hat{\mathbf{A}}' \coloneqq \{\mathbf{a}_1', \ldots, \mathbf{a}_n'\}$. Due to (11), we have

$$\mathbf{E}_{\mathbf{o}}\left[\int_{\mathcal{A}} \exp(\bar{e}_{\mathbf{w}}(\mathbf{o}, \mathbf{a}))\mu(d\mathbf{a})\right] = \mathbf{E}_{\hat{\mathbf{O}}', \hat{\mathbf{A}}'}\left[\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1}{q(\mathbf{a}_j'; \hat{\mathbf{O}}')} \exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i', \mathbf{a}_j'))\right].$$

We can rewrite and decompose the $\mathbf{E}\left[\sup_{\mathbf{w}} \text{II.b}\right]$ term as

$$\begin{aligned}
\mathbf{E}\left[\sup_{\mathbf{w}} \text{II.b}\right] &= \mathbf{E}\left[\sup_{\mathbf{w}}\left\{\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{q^{(j)}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j)) - \mathbf{E}_{\mathbf{o}}\left[\int_{\mathcal{A}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}, \mathbf{a}))\mu(d\mathbf{a})\right]\right\}\right] \\
&= \mathbf{E}\left[\sup_{\mathbf{w}}\left\{\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{q^{(j)}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j)) - \mathbf{E}_{\hat{\mathbf{O}}', \hat{\mathbf{A}}'}\left[\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{q(\mathbf{a}_j'; \hat{\mathbf{O}}')}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i', \mathbf{a}_j'))\right]\right\}\right] \\
&\le \mathbf{E}_{\hat{\mathbf{O}}, \hat{\mathbf{A}}, \hat{\mathbf{O}}', \hat{\mathbf{A}}'}\left[\sup_{\mathbf{w}}\left\{\frac{1}{n}\sum_{i=1}^{n}\frac{1}{q(\mathbf{a}_i; \hat{\mathbf{O}})}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_i)) - \frac{1}{n}\sum_{i=1}^{n}\frac{1}{q(\mathbf{a}_i'; \hat{\mathbf{O}}')}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i', \mathbf{a}_i'))\right\}\right] \\
&\quad + \mathbf{E}_{\hat{\mathbf{O}}, \hat{\mathbf{A}}, \hat{\mathbf{O}}', \hat{\mathbf{A}}'}\left[\sup_{\mathbf{w}}\left\{\frac{1}{n}\sum_{i=1}^{n}\sum_{j\ne i}\frac{1}{q^{(j)}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j)) - \frac{1}{n}\sum_{i=1}^{n}\sum_{j\ne i}\frac{1}{q(\mathbf{a}_j'; \hat{\mathbf{O}}')}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i', \mathbf{a}_j'))\right\}\right] \\
&\le O(1/n) + \mathbf{E}\left[\sup_{\mathbf{w}}\left\{\frac{1}{n}\sum_{i=1}^{n}\sum_{j\ne i}\frac{1}{q^{(j)}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j)) - \frac{1}{n}\sum_{i=1}^{n}\sum_{j\ne i}\frac{1}{q(\mathbf{a}_j'; \hat{\mathbf{O}}')}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i', \mathbf{a}_j'))\right\}\right],
\end{aligned}$$

the last step is due to the assumption $q(\mathbf{a}_i; \hat{\mathbf{O}}) = \sum_{j'=1}^{n} p(\mathbf{a}_i \mid \mathbf{o}_{j'}) \ge \Omega(n)$. Next, we adapt the proof technique in Theorem 6 of [27]. W.l.o.g., we assume that n is even (If n is odd, we can apply the following analysis to the first $n-1$ terms in the summation, where $n-1$ is even. The last term in the summation is a $O(1/n)$ term, which does not change the result). Suppose that S_n is the set of all permutations (the symmetric group of degree n). Then, for each $s \in S$, pairs $(\mathbf{o}_{s(2i-1)}), \mathbf{a}_{s(2i)})$ $(i = 1, \ldots, n/2)$ are mutually independent. Consider the alternative expression of a U-statistics of order 2 (See Appendix 1 in [53]).

$$\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\ne i}\frac{1}{q^{(j)}}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_i, \mathbf{a}_j)) = \frac{1}{n!(n/2)}\sum_{s\in S_n}\sum_{i=1}^{n/2}\frac{1}{q(\mathbf{a}_{s(2i)}; \hat{\mathbf{O}})}\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{s(2i-1)}, \mathbf{a}_{s(2i)})).$$

It then follows that

$$\begin{aligned}
\mathbf{E}\left[\sup_{\mathbf{w}} \text{II.b}\right] &\le O(1/n) + \frac{n-1}{n/2}\mathbf{E}\left[\sup_{\mathbf{w}}\frac{1}{n!}\sum_{s\in S_n}\sum_{i=1}^{n/2}\left(\frac{\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{s(2i-1)}, \mathbf{a}_{s(2i)}))}{q(\mathbf{a}_{s(2i)}; \hat{\mathbf{O}})} - \frac{\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{s(2i-1)}', \mathbf{a}_{s(2i)}'))}{q(\mathbf{a}_{s(2i)}'; \hat{\mathbf{O}}')}\right)\right] \\
&\le O(1/n) + \frac{n-1}{n/2}\frac{1}{n!}\sum_{s\in S_n}\mathbf{E}\left[\sup_{\mathbf{w}}\sum_{i=1}^{n/2}\left(\frac{\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{s(2i-1)}, \mathbf{a}_{s(2i)}))}{q(\mathbf{a}_{s(2i)}; \hat{\mathbf{O}})} - \frac{\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{s(2i-1)}', \mathbf{a}_{s(2i)}'))}{q(\mathbf{a}_{s(2i)}'; \hat{\mathbf{O}}')}\right)\right] \\
&= O(1/n) + \frac{n-1}{n/2}\mathbf{E}\left[\sup_{\mathbf{w}}\sum_{i=1}^{n/2}\left(\frac{\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{2i-1}, \mathbf{a}_{2i}))}{q(\mathbf{a}_{2i}; \hat{\mathbf{O}})} - \frac{\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{2i-1}', \mathbf{a}_{2i}'))}{q(\mathbf{a}_{2i}'; \hat{\mathbf{O}}')}\right)\right] \\
&= O(1/n) + \frac{n-1}{n/2}\mathbf{E}\left[\sup_{\mathbf{w}}\sum_{i=1}^{n/2}\epsilon_i\left(\frac{\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{2i-1}, \mathbf{a}_{2i}))}{q(\mathbf{a}_{2i}; \hat{\mathbf{O}})} - \frac{\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{2i-1}', \mathbf{a}_{2i}'))}{q(\mathbf{a}_{2i}'; \hat{\mathbf{O}}')}\right)\right] \\
&\le O(1/n) + \frac{2(n-1)}{n/2}\mathbf{E}\left[\sup_{\mathbf{w}}\sum_{i=1}^{n/2}\frac{\epsilon_i\exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{2i-1}, \mathbf{a}_{2i}))}{q(\mathbf{a}_{2i}; \hat{\mathbf{O}})}\right],
\end{aligned}$$

where we have used the symmetry between the permutations in S_n and $(\mathbf{o}_i, \mathbf{a}_i), (\mathbf{o}'_i, \mathbf{a}'_i)$. By Lemma 4 and the assumption $q(\mathbf{a}_{2i}; \hat{\mathbf{O}}) = \sum_{j'=1}^{n} p(\mathbf{a}_{2i} \mid \mathbf{o}_{j'}) \geq \Omega(n)$, we further get

$$\mathbf{E}\left[\sup_{\mathbf{w}} \text{II.b}\right] \leq O(1/n) + O(1/n)\mathbf{E}\left[\sup_{\mathbf{w}} \sum_{i=1}^{n/2} \epsilon_i \exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{2i-1}, \mathbf{a}_{2i}))\right].$$

Define the function class $\bar{\mathcal{G}} = \{(\mathbf{o}, \mathbf{a}) \mapsto \exp(\bar{e}_{\mathbf{w}}(\mathbf{o}, \mathbf{a})) \mid \mathbf{w} \in \mathcal{W}\}$. Then, we define the following empirical Rademacher complexity

$$\hat{\mathfrak{R}}^-_{n/2}(\bar{\mathcal{G}}; s) \coloneqq \frac{2}{n}\mathbf{E}_{\epsilon_{1:n/2}}\left[\sup_{\mathbf{w}} \sum_{i=1}^{n/2} \epsilon_i \exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{s(2i-1)}, \mathbf{a}_{s(2i)}))\right].$$

We further define the Rademacher complexity $\mathfrak{R}^-_{n/2}(\bar{\mathcal{G}}) \coloneqq \max_{s \in S_n} \mathbf{E}_{\hat{\mathbf{O}}, \hat{\mathbf{A}}}[\hat{\mathfrak{R}}^-_{n/2}(\bar{\mathcal{G}}; s)]$. We can also apply the symmetrization argument above to bound $\mathbf{E}[\sup_{\mathbf{w}}\{-\text{II.b}\}]$. Due to Assumption 1, we can bound the II.b term as: With probability $1 - \frac{\delta}{2}$, $\delta \in (0, 1)$, we have

$$|\text{II.b}| \leq O(1)\hat{\mathfrak{R}}^-_{n/2}(\bar{\mathcal{G}}; s) + O\left(\frac{1}{n} + \sqrt{\frac{10\log(4/\delta)}{n}}\right). \tag{22}$$

I.5 Bounding Rademacher Complexities

We consider the specific similarity function:

$$e_{\mathbf{w}}(\mathbf{o}, \mathbf{a}) = e_1(\mathbf{w}_1; \mathbf{o})^\top e_2(\mathbf{w}_2; \mathbf{a}).$$

We consider L-layer neural networks

$$e_1(\mathbf{w}_1; \mathbf{o}) \in \mathcal{F}_{1,L} = \{\mathbf{o} \to \sigma(W_{1,L}\sigma(W_{1,L-1} \ldots \sigma(W_{1,1}\mathbf{o}))) : \|W_{1,l}\|_F \leq B_l\},$$
$$e_2(\mathbf{w}_2; \mathbf{a}) \in \mathcal{F}_{2,L} = \{\mathbf{a} \to \sigma(W_{2,L}\sigma(W_{2,L-1} \ldots \sigma(W_{2,1}\mathbf{a}))) : \|W_{2,l}\|_F \leq B_l\}.$$

Suppose that $W_{1,l} \in \mathbb{R}^{d_{1,l} \times d_{1,l-1}}$, $W_{2,l} \in \mathbb{R}^{d_{2,l} \times d_{2,l-1}}$ and $d_{1,0} = d_1$, $d_{2,0} = d_2$, $d_{1,L} = d_{2,L} = d_L$. Define $W_l^\top = (W_l^{(1)}, \ldots, W_1^{(d_l)})$, where $W_l^{(\iota)}$ is the ι-th row of matrix W_l. The following results are adaptions of the results in [54].

I.5.1 Bounding $\mathfrak{R}_n(\mathcal{E})$

Define $h : \mathbb{R}^{2d} \to \mathbb{R}$ as $h(\mathbf{y}) = \mathbf{y}_1^\top \mathbf{y}_2$, where $\mathbf{y} = \begin{pmatrix}\mathbf{y}_1 \\ \mathbf{y}_2\end{pmatrix}$ and $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^d$. It is clear that $e_{\mathbf{w}}(\mathbf{o}, \mathbf{a}) = h(e_1(\mathbf{w}_1, \mathbf{o}), e_w(\mathbf{w}_2, \mathbf{a}))$. Due to Assumption 2, we have $\|e_1(\mathbf{w}_1, \mathbf{o})\|_2 \leq \sqrt{c}$ and $\|e_w(\mathbf{w}_2, \mathbf{a})\| \leq \sqrt{c}$. For any $\mathbf{y} = \begin{pmatrix}\mathbf{y}_1 \\ \mathbf{y}_2\end{pmatrix}, \mathbf{y}' = \begin{pmatrix}\mathbf{y}'_1 \\ \mathbf{y}'_2\end{pmatrix}$ and $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_1, \mathbf{y}'_2 \in [0, \sqrt{c}]^d$, we have

$$(h(\mathbf{y}) - h(\mathbf{y}'))^2 \leq 2(\mathbf{y}_1^\top(\mathbf{y}_2 - \mathbf{y}'_2))^2 + 2((\mathbf{y}_1 - \mathbf{y}'_1)^\top \mathbf{y}'_2)^2 \leq 2c\|\mathbf{y} - \mathbf{y}'\|_2^2,$$

where we have used $(a+b)^2 \leq 2a^2 + 2b^2$ and the decomposition $\mathbf{y}_1^\top \mathbf{y}_2 - (\mathbf{y}'_1)^\top \mathbf{y}'_2 = \mathbf{y}_1^\top(\mathbf{y}_2 - \mathbf{y}'_2) + (\mathbf{y}_1 - \mathbf{y}'_1)^\top \mathbf{y}'_2$. Thus, we can conclude that h is $\sqrt{2c}$-Lipschitz continuous to $\mathbf{y}$ and apply Lemma 2 to the function $e_{\mathbf{w}}(\mathbf{o}, \mathbf{a}) = h(e_1(\mathbf{w}_1, \mathbf{o}), e_w(\mathbf{w}_2, \mathbf{a}))$.

$$\begin{aligned}
\hat{\mathfrak{R}}^+_n(\mathcal{E}) &= \mathbf{E}_{\epsilon_{1:n}}\left[\sup_{e \in \mathcal{E}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i e(\mathbf{o}_i, \mathbf{a}_i)\right] \leq \frac{\sqrt{2c}}{n}\mathbf{E}_{\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2 \in \{\pm 1\}^{nd_L}}\left[\sup_{\mathbf{w}} \sum_{i=1}^{n}\sum_{\iota=1}^{d_L}\left(\epsilon_1^{(i,\iota)} e_1^{(\iota)}(\mathbf{w}_1, \mathbf{o}_i) + \epsilon_2^{(i,\iota)} e_2^{(\iota)}(\mathbf{w}_2, \mathbf{a}_i)\right)\right] \\
&\leq \frac{\sqrt{2c}}{n}\mathbf{E}_{\boldsymbol{\epsilon}_1 \in \{\pm 1\}^{nd_L}}\left[\sup_{\mathbf{w}} \sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \epsilon_1^{(i,\iota)} e_1^{(\iota)}(\mathbf{w}_1, \mathbf{o}_i)\right] + \frac{\sqrt{2c}}{n}\mathbf{E}_{\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2 \in \{\pm 1\}^{nd_L}}\left[\sup_{\mathbf{w}} \sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \epsilon_2^{(i,\iota)} e_2^{(\iota)}(\mathbf{w}_2, \mathbf{a}_i)\right] \\
&= \frac{\sqrt{2c}}{n}\mathbf{E}_{\boldsymbol{\epsilon}_1 \in \{\pm 1\}^{nd_L}}\left[\sup_{W_{1,L}, f_{1,L-1} \in \mathcal{F}_{1,L-1}} \sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \epsilon_1^{(i,\iota)} \sigma(f_{1,L-1}(\mathbf{o}_i)^\top W_{1,L}^{(\iota)})\right] \\
&\quad + \frac{\sqrt{2c}}{n}\mathbf{E}_{\boldsymbol{\epsilon}_2 \in \{\pm 1\}^{nd_L}}\left[\sup_{W_{2,L}, f_{2,L-1} \in \mathcal{F}_{2,L-1}} \sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \epsilon_2^{(i,\iota)} \sigma(f_{2,L-1}(\mathbf{a}_i)^\top W_{2,L}^{(\iota)})\right].
\end{aligned}$$

For simplicity, we can only consider one of the terms above and neglect the index of embedding networks (1 or 2). Let $\mathbf{x}_i$ be one of $\mathbf{o}_i$ and $\mathbf{a}_i$. Cauchy-Schwarz and $(\sup x)^2 \le \sup x^2$ imply

$$\mathbf{E}_{\boldsymbol{\epsilon}\in\{\pm1\}^{nd_L}}\left[\sup_{W_L, f\in\mathcal{F}_{L-1}} \sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\big(f(\mathbf{x}_i)^\top W_L^{(\iota)}\big)\right] \le \left(\mathbf{E}_{\boldsymbol{\epsilon}\in\{\pm1\}^{nd_L}}\left[\left(\sup_{W_L, f\in\mathcal{F}_{L-1}} \sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\big(f(\mathbf{x}_i)^\top W_L^{(\iota)}\big)\right)^2\right]\right)^{\frac{1}{2}}$$
$$\le \left(\mathbf{E}_{\boldsymbol{\epsilon}\in\{\pm1\}^{nd_L}}\left[\sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\big(f(\mathbf{x}_i)^\top W_L^{(\iota)}\big)\right)^2\right]\right)^{\frac{1}{2}}. \tag{23}$$

For a $\lambda > 0$, Jensen's inequality implies that

$$\mathbf{E}_{\boldsymbol{\epsilon}\in\{\pm1\}^{nd_L}}\left[\sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\big(f(\mathbf{x}_i)^\top W_L^{(\iota)}\big)\right)^2\right] = \frac{1}{\lambda}\log\exp\left(\lambda\mathbf{E}_{\boldsymbol{\epsilon}}\left[\sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\big(f(\mathbf{x}_i)^\top W_L^{(\iota)}\big)\right)^2\right]\right)$$
$$\le \frac{1}{\lambda}\log\left(\mathbf{E}_{\boldsymbol{\epsilon}}\exp\left(\lambda\sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\big(f(\mathbf{x}_i)^\top W_L^{(\iota)}\big)\right)^2\right)\right). \tag{24}$$

We utilize the following facts: (i) $\sup_x x^2 \le \max\{(\sup_x x)^2, (\sup_x(-x))^2\}$ and for a Rademacher random variable ϵ, we have $\epsilon, -\epsilon$ are i.i.d.; (ii) Lemma 1 with $\tau(t) = \exp(\lambda t^2)$ and σ is 1-Lipschitz; (iii) $(\sup x)^2 \le \sup x^2$; (iv) $\|W_l\|_F \le B_l$ for each $l \in [L]$:

$$\mathbf{E}_{\boldsymbol{\epsilon}}\exp\left(\lambda\sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\big(f(\mathbf{x}_i)^\top W_L^{(\iota)}\big)\right)^2\right)$$
$$\overset{\text{(i)}}{\le} 2\mathbf{E}_{\boldsymbol{\epsilon}}\exp\left(\lambda\left(\sup_{W_L, f\in\mathcal{F}_{L-1}} \sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\big(f(\mathbf{x}_i)^\top W_L^{(\iota)}\big)\right)^2\right)$$
$$\overset{\text{(ii)}}{\le} 2\mathbf{E}_{\boldsymbol{\epsilon}}\exp\left(\lambda\left(\sup_{W_L, f\in\mathcal{F}_{L-1}} \sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)} f(\mathbf{x}_i)^\top W_L^{(\iota)}\right)^2\right)$$
$$\overset{\text{(iii)}}{\le} 2\mathbf{E}_{\boldsymbol{\epsilon}}\exp\left(\lambda\sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)} f(\mathbf{x}_i)^\top W_L^{(\iota)}\right)^2\right)$$
$$\le 2\mathbf{E}_{\boldsymbol{\epsilon}}\exp\left(\lambda\sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{\iota=1}^{d_L}\left\|\sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)} f(\mathbf{x}_i)\right\|_2 \left\|W_L^{(\iota)}\right\|_2\right)^2\right)$$
$$\le 2\mathbf{E}_{\boldsymbol{\epsilon}}\exp\left(\lambda\sup_{W_L, f\in\mathcal{F}_{L-1}} \|W_L\|_F^2 \sum_{\iota=1}^{d_L}\left\|\sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)} f(\mathbf{x}_i)\right\|_2^2\right) \overset{\text{(iv)}}{\le} 2\mathbf{E}_{\boldsymbol{\epsilon}}\exp\left(\lambda B_L^2 \sup_{f\in\mathcal{F}_{L-1}} \sum_{\iota=1}^{d_L}\left\|\sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)} f(\mathbf{x}_i)\right\|_2^2\right)$$
$$= 2\mathbf{E}_{\boldsymbol{\epsilon}}\exp\left(\lambda B_L^2 \sup_{W_{L-1}, f\in\mathcal{F}_{L-2}} \sum_{\iota=1}^{d_L}\left\|\sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)} \sigma(W_{L-1}f(\mathbf{x}_i))\right\|_2^2\right).$$

Due to the positive-homogeneous property of the activation function $\sigma(\cdot)$, we have

$$\sum_{\iota=1}^{d_L}\left\|\sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)} \sigma(W_{L-1}f(\mathbf{x}_i))\right\|_2^2 = \sum_{\iota=1}^{d_L}\left\|\begin{pmatrix} \sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\big(f(\mathbf{x}_i)^\top W_{L-1}^{(1)}\big) \\ \vdots \\ \sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\big(f(\mathbf{x}_i)^\top W_{L-1}^{(d_{L-1})}\big)\end{pmatrix}\right\|_2^2$$
$$= \sum_{\iota=1}^{d_L}\sum_{r=1}^{d_{L-1}}\left(\sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\big(f(\mathbf{x}_i)^\top W_{L-1}^{(r)}\big)\right)^2 = \sum_{r=1}^{d_{L-1}}\left\|W_{L-1}^{(r)}\right\|_2^2 \sum_{\iota=1}^{d_L}\left(\sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\left(f(\mathbf{x}_i)^\top \frac{W_{L-1}^{(r)}}{\left\|W_{L-1}^{(r)}\right\|_2}\right)\right)^2$$
$$\le \|W_{L-1}\|_F^2 \max_{r\in[d_{L-1}]}\sum_{\iota=1}^{d_L}\left(\sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\left(f(\mathbf{x}_i)^\top \frac{W_{L-1}^{(r)}}{\left\|W_{L-1}^{(r)}\right\|_2}\right)\right)^2 \le B_{L-1}^2 \sup_{\mathbf{w}:\|\mathbf{w}\|_2\le 1}\sum_{\iota=1}^{d_L}\left(\sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)}\sigma\left(f(\mathbf{x}_i)^\top \mathbf{w}\right)\right)^2.$$

Thus, we can obtain

$$
\begin{aligned}
&\mathbf{E}_{\boldsymbol{\epsilon}} \exp\left(\lambda \sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)} \sigma(f(\mathbf{x}_i)^\top W_L^{(\iota)})\right)^2\right) \\
&\le 2\mathbf{E}_{\boldsymbol{\epsilon}} \exp\left(\lambda B_L^2 B_{L-1}^2 \sup_{\|\mathbf{w}\|_2\le 1, f\in\mathcal{F}_{L-2}} \sum_{\iota=1}^{d_L} \left(\sum_{i=1}^{n} \boldsymbol{\epsilon}^{(i,\iota)} \sigma(f(\mathbf{x}_i)^\top \mathbf{w})\right)^2\right) \\
&\le 2\mathbf{E}_{\epsilon_{1:n}} \exp\left(d_L \lambda B_L^2 B_{L-1}^2 \sup_{\|\mathbf{w}\|_2\le 1, f\in\mathcal{F}_{L-2}} \left(\sum_{i=1}^{n} \epsilon_i \sigma(f(\mathbf{x}_i)^\top \mathbf{w})\right)^2\right).
\end{aligned}
$$

Applying Lemma 1 with $\tau_\lambda(t) = \exp(d_L \lambda B_L^2 B_{L-1}^2 t^2)$ gives

$$
\begin{aligned}
&\mathbf{E}_{\boldsymbol{\epsilon}} \exp\left(\lambda \sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)} \sigma(f(\mathbf{x}_i)^\top W_L^{(\iota)})\right)^2\right) \le 2\mathbf{E}_{\epsilon_{1:n}}\left[\tau_\lambda\left(\sup_{\|\mathbf{w}\|_2\le 1, f\in\mathcal{F}_{L-2}} \left|\sum_{i=1}^{n} \epsilon_i \sigma(f(\mathbf{x}_i)^\top \mathbf{w})\right|\right)\right] \\
&\le 2\mathbf{E}_{\epsilon_{1:n}}\left[\tau_\lambda\left(\sup_{\|\mathbf{w}\|_2\le 1, f\in\mathcal{F}_{L-2}} \sum_{i=1}^{n} \epsilon_i \sigma(f(\mathbf{x}_i)^\top \mathbf{w})\right)\right] + 2\mathbf{E}_{\epsilon_{1:n}}\left[\tau_\lambda\left(\sup_{\|\mathbf{w}\|_2\le 1, f\in\mathcal{F}_{L-2}} -\sum_{i=1}^{n} \epsilon_i \sigma(f(\mathbf{x}_i)^\top \mathbf{w})\right)\right] \\
&= 4\mathbf{E}_{\epsilon_{1:n}}\left[\tau_\lambda\left(\sup_{\|\mathbf{w}\|_2\le 1, f\in\mathcal{F}_{L-2}} \sum_{i=1}^{n} \epsilon_i \sigma(f(\mathbf{x}_i)^\top \mathbf{w})\right)\right] \le 4\mathbf{E}_{\epsilon_{1:n}}\left[\tau_\lambda\left(\sup_{\|\mathbf{w}\|_2\le 1, f\in\mathcal{F}_{L-2}} \sum_{i=1}^{n} \epsilon_i f(\mathbf{x}_i)^\top \mathbf{w}\right)\right] \\
&\le 4\mathbf{E}_{\epsilon_{1:n}}\left[\tau_\lambda\left(\sup_{W_{L-2}, f\in\mathcal{F}_{L-3}} \left\|\sum_{i=1}^{n} \epsilon_i \sigma(W_{L-2} f(\mathbf{x}_i))\right\|_2\right)\right] \le 4\mathbf{E}_{\epsilon_{1:n}}\left[\tau_\lambda\left(B_{L-2} \sup_{\|\mathbf{w}\|_2\le 1, f\in\mathcal{F}_{L-3}} \left|\sum_{i=1}^{n} \epsilon_i f(\mathbf{x}_i)^\top \mathbf{w}\right|\right)\right],
\end{aligned}
$$

where in the last step we have used the positive-homogeneous property of $\sigma(\cdot)$ (e.g., analysis similar to handling the supremum over $W_L, f \in \mathcal{F}_{L-1}$). Applying the inequality above recursively over the layers leads to

$$
\mathbf{E}_{\boldsymbol{\epsilon}} \exp\left(\lambda \sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)} \sigma(f(\mathbf{x}_i)^\top W_L^{(\iota)})\right)^2\right) \le 2^L \mathbf{E}_{\epsilon_{1:n}}\left[\tau_\lambda\left(\prod_{l=1}^{L-2} B_l \left\|\sum_{i=1}^{n} \epsilon_i \mathbf{x}_i\right\|_2\right)\right].
$$

Plug the inequality above into (24).

$$
\mathbf{E}_{\boldsymbol{\epsilon}\in\{\pm1\}^{nd_L}}\left[\sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)} \sigma(f(\mathbf{x}_i)^\top W_L^{(\iota)})\right)^2\right] \le \frac{1}{\lambda}\log\left(2^L \mathbf{E}_{\epsilon_{1:n}} \exp\left(d_L \lambda \left(\prod_{l=1}^{L} B_l^2\right) \left\|\sum_{i=1}^{n} \epsilon_i \mathbf{x}_i\right\|_2^2\right)\right).
$$

Let $\tilde{\lambda} = d_L \lambda \left(\prod_{l=1}^{L} B_l^2\right)$ and choose $\lambda = \frac{1}{8esd_L\left(\prod_{l=1}^{L} B_l^2\right)}$, $s = \left(\sum_{1\le i\le \tilde{i}\le n} (\mathbf{x}_i^\top X_{\tilde{i}})^2\right)^{\frac{1}{2}}$. Then, $\tilde{\lambda} = 1/(8es)$ and we can apply Lemma 3 to show $\mathbf{E}_{\epsilon_{1:n}}\left[\exp\left(2\tilde{\lambda}\sum_{1\le i\le\tilde{i}\le n} \epsilon_i \epsilon_{\tilde{i}} \mathbf{x}_i^\top X_{\tilde{i}}\right)\right] \le 2$ such that

$$
\begin{aligned}
\mathbf{E}_{\epsilon_{1:n}} \exp\left(\tilde{\lambda}\left\|\sum_{i=1}^{n} \epsilon_i \mathbf{x}_i\right\|_2^2\right) &= \mathbf{E}_{\epsilon_{1:n}}\left[\exp\left(\tilde{\lambda}\sum_{i=1}^{n} \|\mathbf{x}_i\|_2^2 + 2\tilde{\lambda} \sum_{1\le i\le\tilde{i}\le n} \epsilon_i \epsilon_{\tilde{i}} \mathbf{x}_i^\top X_{\tilde{i}}\right)\right] \\
&= \exp\left(\tilde{\lambda}\sum_{i=1}^{n} \|\mathbf{x}_i\|_2^2\right) \mathbf{E}_{\epsilon_{1:n}}\left[\exp\left(2\tilde{\lambda} \sum_{1\le i\le\tilde{i}\le n} \epsilon_i \epsilon_{\tilde{i}} \mathbf{x}_i^\top X_{\tilde{i}}\right)\right] \le 2\exp\left(\tilde{\lambda}\sum_{i=1}^{n} \|\mathbf{x}_i\|_2^2\right).
\end{aligned}
$$

Since $\lambda = \frac{1}{8esd_L\left(\prod_{l=1}^{L} B_l^2\right)}$ and $s^2 \le \sum_{1\le i\le\tilde{i}\le n} \|\mathbf{x}_i\|_2^2 \|\mathbf{x}_{\tilde{i}}\|_2^2 \le \left(\sum_{i=1}^{n} \|\mathbf{x}_i\|_2^2\right)^2$, we can obtain

$$
\begin{aligned}
&\mathbf{E}_{\boldsymbol{\epsilon}\in\{\pm1\}^{nd_L}}\left[\sup_{W_L, f\in\mathcal{F}_{L-1}} \left(\sum_{i=1}^{n}\sum_{\iota=1}^{d_L} \boldsymbol{\epsilon}^{(i,\iota)} \sigma(f(\mathbf{x}_i)^\top W_L^{(\iota)})\right)^2\right] \le \frac{1}{\lambda}\log\left(2^{L+1}\exp\left(\tilde{\lambda}\sum_{i=1}^{n} \|\mathbf{x}_i\|_2^2\right)\right) \\
&= \frac{(L+1)\log 2}{\lambda} + d_L\left(\prod_{l=1}^{L} B_l^2\right)\sum_{i=1}^{n} \|\mathbf{x}_i\|_2^2 \le d_L\left(\prod_{l=1}^{L} B_l^2\right)(8(L+1)e\log 2 + 1)\sum_{i=1}^{n} \|\mathbf{x}_i\|_2^2.
\end{aligned}
$$

Due to (23), we can obtain

$$
\hat{\mathfrak{R}}_n^+(\mathcal{E}) = \mathbf{E}_{\epsilon_{1:n}}\left[\sup_{e\in\mathcal{E}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i e(\mathbf{o}_i, \mathbf{a}_i)\right] \le \frac{\sqrt{2c}}{\sqrt{n}}\sqrt{d_L\left(\prod_{l=1}^{L} B_l^2\right)(8(L+1)e\log 2 + 1)(c_1 + c_2)}. \quad (25)
$$

I.5.2 Bounding $\mathfrak{R}^-_{n/2}(\bar{\mathcal{G}})$

We define the dataset $\hat{\mathbf{D}}_s \coloneqq \{(O_{s(1)}, A_{s(2)}), \ldots, (O_{s(n-1)}, A_{s(n)})\}$. Consider $\mathcal{E} \coloneqq \{(\mathbf{o},\mathbf{a}) \mapsto e_{\mathbf{w}}(\mathbf{o},\mathbf{a}) \mid \mathbf{w} \in \mathcal{W}\}$ and the following two function classes

$$\bar{\mathcal{E}} \coloneqq \{(\mathbf{o},\mathbf{a}) \mapsto \bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a}) \mid \mathbf{w} \in \mathcal{W}\}, \quad \bar{\mathcal{G}} = \{(\mathbf{o},\mathbf{a}) \mapsto \exp(\bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a})) \mid \mathbf{w} \in \mathcal{W}\}.$$

The empirical Rademacher complexities of $\bar{\mathcal{E}}, \bar{\mathcal{G}}$ on $\hat{\mathbf{D}}_s$ can be defined as

$$\hat{\mathfrak{R}}^-_{n/2}(\bar{\mathcal{E}}; s) = \mathbf{E}_{\epsilon_{1:n/2}}\left[\frac{2}{n}\sup_{\mathbf{w}}\sum_{i=1}^{n/2}\epsilon_i \bar{e}_{\mathbf{w}}(\mathbf{o}_{s(2i-1)}, \mathbf{a}_{s(2i)})\right],$$
$$\hat{\mathfrak{R}}^-_{n/2}(\bar{\mathcal{G}}; s) = \mathbf{E}_{\epsilon_{1:n/2}}\left[\frac{2}{n}\sup_{\mathbf{w}}\sum_{i=1}^{n/2}\epsilon_i \exp(\bar{e}_{\mathbf{w}}(\mathbf{o}_{s(2i-1)}, \mathbf{a}_{s(2i)}))\right].$$

Note that $\exp(t)$ is 1-Lipschitz when $t \le 0$. Due to Lemma 4 and $\bar{e}_{\mathbf{w}}(\mathbf{o},\mathbf{a}) = (e_{\mathbf{w}}(\mathbf{o},\mathbf{a}) - c)/\tau$,

$$\hat{\mathfrak{R}}^-_{n/2}(\bar{\mathcal{G}}; s) \le \hat{\mathfrak{R}}^-_{n/2}(\bar{\mathcal{E}}; s) = \frac{1}{\tau}\hat{\mathfrak{R}}^-_{n/2}(\mathcal{E}; s). \tag{26}$$

Then, we can bound $\hat{\mathfrak{R}}^-_{n/2}(\mathcal{E}; s)$ in the way similar to bounding $\hat{\mathfrak{R}}^+_n(\mathcal{E})$ in Section I.5.1.