Tighter Risk Bounds for Metric Learning using N-tuple Loss

A Postgraduate Dissertation Presented for the degree of Master of Science in Artificial Intelligence and Machine Learning

By

MEGGISON ORITSEMISAN

2891760

School of Computer Science

College of Engineering and Physical Sciences

University of Birmingham

2024-25

Abstract

This thesis presents an empirical study on training probabilistic neural networks for Deep Metric Learning (DML) using training objectives derived from PAC-Bayes bounds. A key challenge in DML is that the common practice of training with tuple-based losses creates structured data dependencies, violating the i.i.d. assumption central to standard learning theory. In the context of probabilistic neural networks, where the output of training is a probability distribution over network weights, this makes obtaining valid generalization guarantees difficult. To address this, this thesis presents a novel training objective for DML, which is derived from a PAC-Bayes bound that leverages U-statistic theory to correctly account for these data dependencies. This work also implements and compares against objectives based on tighter, but i.i.d.-assuming, PAC-Bayes bounds to evaluate the properties of the predictors learned using the different training objectives. Risk certificates are computed for the learnt predictors based on the same data used for learning. Further experiments are conducted with different types of priors on the weights (both data-free and data-dependent), a range of neural network architectures, and varying tuple sizes. The extensive experiments on CIFAR-10 show that the proposed U-statistic-based training method produces competitive test set performance, achieving 74.6% accuracy, and consistently yields non-vacuous risk bounds with much tighter values than those from standard PAC-Bayes formulations. These observations suggest that the methods studied here are strong candidates for self-certified learning in the DML setting, enabling the use of the entire dataset for both learning a predictor and certifying its risk on unseen data, potentially without the need for a hold-out test set.

Keywords: Deep Metric Learning, PAC-Bayesian framework, generalisation bounds, probabilistic neural networks, U-statistics, non-i.i.d. data, data-dependent priors, self-certified learning.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, **Professor Ata Kaban** from the School of Computer Science, for her invaluable feedback and guidance throughout this project. Her expertise in statistical and PAC-Bayes techniques was incredibly beneficial, and her insights led to significant improvements in the overall structure and quality of this thesis.

I am also grateful to Ph.D. candidate **Sijia Zhou** for her helpful guidance in understanding the bound formulations during the early stages of my work.

I extend my thanks to **Paul and Yuanbi Ramsay** for their generous support through the MSc Ramsay Bursary Award, which made this programme of study possible.

Finally, I wish to thank my mother, **Mami Meggison**, for her unwavering support and encouragement throughout my academic endeavours.

Abbreviations

DML	Deep Metric Learning
Re-ID	Person Re-Identification
CNN	Convolutional Neural Network
PNN	Probabilistic Neural Network
PAC	Probably Approximately Correct
PAC-Bayes	PAC-Bayesian framework
KL	Kullback–Leibler divergence
KL^+	One-sided binary KL divergence
U-statistic	Unbiased statistic over m-tuples
i.i.d.	Independent and Identically Distributed
MC	Monte Carlo
VI	Variational Inference
BNN	Bayesian Neural Network
BBB	Bayes by Backprop
N-pair	Multi-class N-pair loss
Tuplet	Tuplet Margin Loss
MS Loss	Multi-Similarity Loss
SupCon	Supervised Contrastive Learning
Proxy-NCA	Proxy Neighborhood Component Analysis
Proxy Anchor	Proxy Anchor Loss
Triplet	Triplet loss
PCB	Part-based Convolutional Baseline
K-reciprocal	k-reciprocal re-ranking
mAP	mean Average Precision
Rank-1	Rank-1 accuracy
SGD	Stochastic Gradient Descent
KL-inv	Binary KL inversion f^{kl}
Prior P	Prior over hypotheses
Posterior Q	Posterior over hypotheses
CIFAR-10	10-class image dataset

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Introduction

Deep metric learning seeks to learn context embeddings under which similar inputs are mapped close together while dissimilar ones are pushed apart, enabling person re-identification, face verification, retrieval, and ranking models to function without relying on multi class prediction at inference time [24]. Within this paradigm loss functions designed on pairwise, triplet, and large tuple-wise input constructions define relative similarity constraints that shape the embedding geometry [30]. However, while such tuple-wise objectives have driven empirical progress, their theoretical understanding, particularly generalisation guarantees tailored to the non-independent structure of training data induced by the structures of tuples is less explores compared to classification and regression settings [18]. Recent developments at the intersection of tuple-wise learning, and PAC-Bayesian self-certification provide a principled pathway to bridge this gap, yielding computable risk bounds that directly reflect tuple size and can be optimized to produce stochastic neural networks with non-vacuous data depend on risk certificates [20]. This work situates a generalised triplet loss within emerging theoretical frameworks and proposes a framework for learning and certifying tuple wise objectives and empirically studies its behavior on vision benchmarks [13].

1.2 Background & Motivation

Modern tuple-wise losses extend triplets by jointly contrasting one anchor–positive pair against multiple negative samples, better matching retrieval-time ranking while alleviating the contradictory gradients that arise when many independent triplets share points; yet their guarantees have remained underexplored in re-identification and related retrieval tasks [30]. The self-certified tuple-wise learning approach generalizes PAC-Bayes to U-statistics, proving that effective sample size scales as floor(n/m) for m-tuples and delivering practically computable training objectives and final certificates for tuple-wise deep networks; it demonstrates non-vacuous bounds in person re-identification, validating theoretical predictions about increasing sample complexity with tuple size [34].

Tight PAC-Bayes objectives for neural networks coupled with data-dependent priors and Monte Carlo evaluation produce strong test performance and substantially tighter risk certificates than empirical networks, enables self-certified learning without a hold-out set and suggesting principled model selection [20]. Building on these advances, this work develops and assesses a PAC-Bayes N-tuple framework for metric learning, examines the trade-offs between tuple size, architecture complexity, and certificate tightness, and explores mining strategies that balance empirical risk with generalization guarantees [18].

1.3 Research Gaps and Objectives

Despite strong empirical advances in deep-metric learning, there remains a lack of certified generalisation guarantees tailored specifically to Ntuple losses (including N-pair, triplet, and quadruplet formulations) [24]. Recent progress has begun to extend PAC-Bayesian theory to tuple-wise learning via U-statistics and to demonstrate self-certified training for pairwise similarity with stochastic neural networks, but a rigorous exploration that cleanly reflects tuple size, inter-tuple dependence, and the realities of modern embeddings and mining strategies is still underdeveloped [18].

While tuple-wise PAC-Bayes bounds confirm that sample complexity scales adversely with increasing tuple size intuitively because tuples share points and are therefore dependent, the translation of these insights into feasible objectives for N-tuple metric losses and their deployment in re-identification pipelines remains incomplete. Moreover, although the metric learning community has validated the practical advantages of moving beyond triplets to N-tuple objectives, the theoretical definition of tight, non-vacuous certificates that explicitly define these advantages is currently unexplored [30].

Handling non identically and independent data at scale remains a practical barrier. While U-statistics provide a principled bridge from pointwise to tuple-wise learning, and fractional-cover PAC-Bayes analyses offer tools for dependencies more broadly, there is still no standard, scalable recipe that operationalises these ideas within modern tuple-wise neural training and certification systems. This is particularly significant in large-scale retrieval and re-ID settings where tuples overlap heavily and mining induces structured dependence on learning [34].

This paper reports experiments performed on CIFAR-10 and employs an adapted self-certified tuple-based PAC-Bayes Bounds framework formulated using the pairwise bounds proposed in self-certified risk certificates for tuple wise bounds to quantify risk certificates for metric learning trained models [20]. Moderate tuple sizes and simpler CNN networks are used to formulate a theoretical tighter certificate and guarantees on model metrics and evaluate the impact and validity of the adapted tuple-wise bounds for creating certificates for metric learning.

1.4 Contributions

- Rigorous study and evaluation of training objectives that integrates a bounded N-tuple surrogate loss using PAC-Bayes regularisation with an explicit dependence on tuple size m via U-statistics, enabling optimisation with stochastic gradient backpropagation and Monte Carlo weight sampling; instantiated with cosine-based distances and modern mining strategies for re-identification.

- Propose and compute non-vacuous, high-confidence risk certificates for stochastic CNN embeddings trained with tuple-wise objectives, using binary KL inversion and Monte Carlo approximation of empirical tuple risk.

- Demonstrate that increasing tuple size raises sample complexity and typically loosens certificates, while moderate tuple sizes and simpler architectures can achieve tighter bounds and competitive or superior accuracy compared to larger tuples/deeper models in re-id settings; analyse the interaction between mining hardness and certificate tightness.
- Adapt tight PAC-Bayes training quadratic bound objectives and data-dependent priors to the metric learning, showing strong test performance and substantially tighter certificates than classical surrogates.
- Experiment and demonstrate theoretical positioning under a non-independent contrastive learning problem.

CHAPTER 2

Background

Introduction

This thesis introduces a framework for self-certified deep metric learning, a novel contribution that lies at the intersection of three advanced fields in machine learning: the PAC-Bayes framework for generalisation bounds, Probabilistic Neural Networks (PNNs) trained with variational inference, and the tuple-based risk structures inherent to Deep Metric Learning (DML). A thorough understanding of these foundational topics is essential to appreciate the theoretical challenges and the novelty of the proposed methodology.

2.1 Deep Metric Learning

Deep Metric Learning (DML) is a subfield of representation learning that aims to train a neural network, f_θ, to map high-dimensional inputs, x, into a low-dimensional embedding space, R^d. The central objective is to represent similarity between inputs directly reflected by a chosen distance metric [17].

In an ideal embedding space, vectors corresponding to inputs from the same class are clustered together, while vectors from different classes are pushed far apart. This learned space can then be used for tasks in image retrieval, face verification, and person re-identification [21].

2.2 Metric Learning Loss Function Formulation

The structure of the embedding space in DML is defined by the training objective. This has evolved from simple pairs to complex tuples of samples to better capture relational information.

2.2.1 Triplet Loss

A significant breakthrough in metric learning was the triplet loss, which operates on a three-element tuple: an anchor sample (x_a), a positive sample (x_p) from the same class, and a negative sample

(x_n) from a different class [21]. The training objective is to ensure the squared Euclidean distance between the anchor and positive embeddings is smaller than that between the anchor and negative embeddings by a margin α.

The loss function is formulated as:

$$L(x_a, x_p, x_n) = \max\Big(\|f_\theta(x_a) - f_\theta(x_p)\|_2^2 - \|f_\theta(x_a) - f_\theta(x_n)\|_2^2 + \alpha,\, 0\Big). \tag{2.1}$$

This directly optimises the relative distance, making it highly effective for ranking-based tasks and a strong baseline in Re-ID [6].

2.2.2 Beyond the Triplet Loss

The primary limitation of the triplet loss is its limited scope, as it only considers a single negative sample at a time. This can lead to slow convergence and suboptimal embeddings because the optimisation of one triplet can inadvertently contradict the objective of another within the same mini-batch. For instance, a triplet consisting of an anchor (s_1), a positive sample (s_2) from the same class, and a negative sample (s_3) the embedding for s_1 is pushed away from s_3. However, if another triplet in the same batch uses s_1 as a negative sample for a different anchor (s_4) the optimisation from the first triplet may have moved s_1 closer to s_4, making the second triplet's objective harder to satisfy [24]. This lack of interaction between triplets means that considering multiple instances through several independent triplet constraints provides a relatively weak and sometimes conflicting training signal

2.2.3 Unified N-Tuple Loss Variants

A more structured and efficient approach than the standard triplet loss is to frame the DML problem as a multi-class classification task. This perspective allows for the joint optimisation of an anchor against multiple negative samples simultaneously, leading to a more powerful and coherent training signal.

Many modern metric learning losses can be understood through a unified classification framework. The core idea is to maximise the probability that an anchor sample x_a is correctly associated with its positive "class" relative to a set of negative references. This can be expressed with a generalised softmax loss function:

$$L_{\text{unified}} = -\log \frac{\exp(\frac{1}{\tau} S(x_a, c_j))}{\sum_{k=1}^{C} \exp(\frac{1}{\tau} S(x_a, c_k))} \tag{2.2}$$

Here, $S(\cdot, \cdot)$ is a similarity function (e.g., cosine similarity), $\{c_k\}_{k=1}^{C}$ represents a set of C reference vectors, c_j is the reference corresponding to the positive class, and τ is a temperature parameter that controls the sharpness of the distribution [27]. Minimising this loss is equivalent to maximising the probability of correct classification.

The soft-margin variant of the triplet loss can be seen as a special case of this unified formulation. The standard soft-margin triplet loss is:

$$L_{\text{triplet}} = \log(1 + \exp(S(x_a, x_n) - S(x_a, x_p))) \tag{2.3}$$

This is mathematically equivalent to the unified loss where the number of classes $C = 2$, the positive sample embedding x_p serves as the positive reference c_j, and the single negative sample embedding x_n serves as the negative reference. While effective, it still only optimises the model with respect to a single negative at a time.

The **N-tuple loss** extends this idea to a true multi-class scenario. Given an anchor x_a, one positive sample x_p, and a set of $(N-2)$ negative samples $\{x_{n_k}\}_{k=1}^{N-2}$, the loss is defined as:

$$L_{N\text{-tuple}} = -\log \frac{\exp(\frac{1}{\tau} S(x_a, x_p))}{\exp(\frac{1}{\tau} S(x_a, x_p)) + \sum_{k=1}^{N-2} \exp(\frac{1}{\tau} S(x_a, x_{n_k}))} \tag{2.4}$$

This formulation, which is a direct instantiation of the unified loss, effectively treats each of the positive and negative samples as instance-level "class centers" forcing the model to learn a representation where the anchor is more similar to its true positive than to all other $(N-2)$ negatives simultaneously. This approach, pioneered by the **N-pair loss** [23] and refined in subsequent works like the Tuplet Margin Loss [30] and higher-order relational models using hypergraphs [13], is more data-efficient and produces superior embeddings by leveraging a wider context of negative examples in each training step. This joint optimization over multiple instances better aligns with the inference task in Re-ID, where a query must be compared against a large gallery of candidates.

2.3 PAC-Bayes Theory and Generalisation Bounds

While DML models have demonstrated strong empirical performance, they typically lack formal guarantees on how they will perform on new, unseen data. The Probably Approximately Correct (PAC)-Bayesian framework provides a powerful set of tools to derive such generalisation bounds [15].

The framework bounds the true risk $R(q)$ (error on the true data distribution) of a stochastic predictor, a posterior distribution over models q with high probability. Stated formally, for any prior distribution p and confidence parameter $\delta \in (0, 1)$, with probability at least $1-\delta$ over the random draw of a training set S of size n, the following holds for all posterior distributions q:

$$R(q) \leq \hat{R}_S(q) + \sqrt{\frac{KL(q\|p) + \ln(\frac{n}{\delta})}{2n}}. \tag{2.5}$$

The key components are:

- $\hat{R}_S(q)$: The empirical risk, or the average loss of the predictor q on the training set S.
- $KL(q\|p)$: The Kullback-Leibler (KL) divergence, which serves as a complexity penalty. It measures the information-theoretic "distance" between the learned posterior q and the fixed prior p:

$$KL(q\|p) = \int q(\theta) \log \frac{q(\theta)}{p(\theta)} \, d\theta \tag{2.6}$$

CHAPTER 3

Related Works

3.1 Introduction

Deep metric learning has emerged as a dominant paradigm for a wide array of machine learning tasks, including person re-identification, face verification, and image retrieval [17]. The central goal is to learn a feature embedding space where the distance between representations of semantically similar inputs is minimised, while the distance between dissimilar inputs is maximised. This is typically achieved not through direct classification, but by optimising objectives defined over tuples of data points, such as pairs, triplets, or larger N-tuples which enforce relative similarity constraints [21, 23]. The empirical success of these methods, driven by sophisticated loss functions, tuple mining strategies, and deep neural network architectures, is well-documented [14, 6].

However, this empirical progress has largely outpaced a corresponding theoretical understanding of generalisation. While classical statistical learning theory provides robust guarantees for models trained on independent and identically distributed (i.i.d.) data, the very nature of tuple-based training violates this core assumption. The construction of tuples, especially through "hard mining" strategies, introduces complex dependencies within the training data, as a single data point may appear in numerous tuples. This raises a critical question: how can we be sure that a model trained on these structured, dependent tuples will perform well on unseen data?

This literature review bridges this gap by situating deep metric learning within the Probably Approximately Correct (PAC) Bayesian framework for generalization. The PAC-Bayes framework provides a powerful set of tools for deriving high-confidence, non-vacuous generalization bounds for complex models like neural networks, particularly for stochastic predictors [15, 3]. We explore the landscape of deep metric learning, focusing on the evolution of tuple-based losses and mining strategies. We will then introduce the fundamentals of PAC-Bayes theory and its successful application to deep learning. The central challenge of extending these guarantees to the non-i.i.d. setting of tuple-wise learning will be addressed by reviewing advanced theoretical tools, including U statistics and dependency graphs [8, 19]. Finally, we will synthesize these threads to position the current research which seeks to develop a practical and theoretically sound framework for training and certifying deep-metric learning models, providing provable guarantees on their real-world performance.

3.2 Person Re-identification as a Benchmark Task

Person Re-Identification (Re-ID) is the task of identifying an individual across a network of non-overlapping cameras. Given a query image of a person, the goal is to retrieve all images of that same person from a large gallery. This is fundamentally a ranking problem, making it an ideal benchmark for metric learning.

Early deep learning approaches for Re-ID focused on designing specialised network architectures [13]. However, the field advanced significantly with the adoption of metric learning techniques, particularly the triplet loss, which is well-suited to the ranking nature of the task [6]. State-of-the-art Re-ID models now leverage sophisticated backbones and advanced metric learning losses to learn discriminative embeddings that are robust to changes in viewpoint, illumination, and pose [25, 31]. Due to its clear objective and challenging nature, Re-ID serves as the primary experimental problem set in this study.

3.3 Deep Metric Learning

The core innovation of deep-metric learning lies in shifting the learning problem from a of classification problem to representation. Instead of learning to predict a label, the goal is to learn a function $f_\theta : \mathcal{X} \to R^d$ that maps an input x to a embedding vector of dimension d, such that the geometric structure of the embedding space reflects the semantic similarity of the inputs. This structure is imposed by loss functions defined on tuples of training examples.

3.3.1 Foundational Pairwise and Triplet-Based Losses

Earlier approaches were built on pairwise constraints. The **contrastive loss** function, for instance, operates on pairs of inputs (x_i, x_j) and aims to pull together embeddings of similar (positive) pairs while pushing apart dissimilar (negative) pairs beyond a certain margin m [4].

A significant conceptual leap was the introduction of the triplet loss, which became a cornerstone of the field [21]. This loss operates on a triplet of data points and the objective is to ensure that the anchor is closer to the positive than it is to the negative by at least a given margin. The triplet formulation provides a more direct and powerful relative constraint than pairwise losses, forcing the model to learn a more discriminative embedding space. Its effectiveness has been demonstrated extensively, particularly in person re-identification [6].

3.3.2 Advancing to N-Tuple and Listwise Formulations

While effective, the triplet loss considers only one negative example at a time. This can lead to slow convergence and suboptimal embeddings, as the model does not get a global view of the inter-class relationships. To address this, researchers developed losses based on larger tuples. The N-pair loss extends the triplet concept by contrasting one anchor-positive pair against $N - 2$ negative examples simultaneously, formulated within a softmax cross-entropy objective [23]. This provides richer negative information and more stable gradients.

This trend has continued with the development of numerous sophisticated losses that leverage multiple examples. Lifted structured loss constructs a batch-wide pairwise distance matrix and learns by lifting the full set of positive and negative pairs within the batch [24]. Multi-similarity loss generalises this by re-weighting pairs based on their similarity, allowing it to capture both self-similarity (pulling positives together) and relative similarity (contrasting with negatives of varying hardness) [27]. Other notable approaches include proxy-based losses, which use learnable

class proxies instead of actual data points to scale to massive datasets [16, 11], and histogram loss, which operates on the distribution of similarities rather than individual values [26].

The paradigm has expanded to include supervised contrastive learning (SupCon), which generalises the triplet loss to handle multiple positives and multiple negatives per anchor within a contrastive framework [10]. This has shown remarkable performance by efficiently leveraging all available labels within a batch.

3.3.3 The Critical Role of Tuple Mining

The performance of tuple-based losses is critically dependent on the selection of tuples. Randomly sampling triplets, for example, often results in trivial examples where the loss is already zero, providing no learning signal. To counteract this, hard example mining strategies were developed. Batch hard mining selects the hardest positive (furthest from anchor) and hardest negative (closest to anchor) within a mini-batch to form the triplet [6]. This ensures that the model continually focuses on the most informative examples.

Other strategies like distance-weighted sampling [29] and smart mining [5] offer more nuanced ways to select tuples, balancing between the hardest examples (which can cause instability) and the easier ones. The choice of mining strategy is not merely a practical detail; it fundamentally alters the data distribution presented to the model and induces strong dependencies among the training samples, posing a significant challenge for theoretical analysis [9].

3.4 The PAC-Bayes Framework for Generalization

To provide theoretical guarantees for the complex models used in deep metric learning, we turn to the PAC-Bayes framework. Unlike classical uniform convergence bounds, which are often vacuous for deep neural networks, PAC-Bayes provides a mechanism for deriving tight, data-dependent bounds that reflect the properties of the learned model.

3.4.1 Core Principles

The PAC-Bayes framework provides bounds on the generalisation risk of a **stochastic predictor** (also called a Gibbs classifier or ensemble). Instead of learning a single hypothesis (a single set of weights θ), we learn a distribution Q over the hypothesis class $\mathcal{H}$, often called the posterior. We also define a fixed **prior** distribution P over $\mathcal{H}$ before seeing the data. The learning process can be seen as using the data to update our belief from the prior P to the posterior Q.

3.4.2 Application to Deep Neural Networks

Applying PAC Bayes bounds to deep neural networks is challenging, often resulting in vacuous bounds (i.e., bounds greater than 1). A breakthrough came from the work of Dziugaite & Roy , who demonstrated that by carefully choosing the prior P and optimising a trade-off between the empirical risk and the KL term, it was possible to obtain non-vacuous generalisation guarantees for deep networks [3].

This has spurred a line of research into "self-certified learning," where the training objective is a direct optimisation of the PAC-Bayes bound itself. Recent work has developed tighter quadratic training objectives that provide better empirical performance and tighter final risk certificates than classical relaxations [18]. These advances, often involving data-dependent priors and sophisticated optimisation techniques like Entropy-SGD, have made PAC-Bayes a practical tool for deep learning [1].The models are typically stochastic neural networks, where each weight is a distribution

(e.g., Gaussian), as seen in Bayes by Backprop [2], allowing for the direct computation of the KL divergence term.

3.5 Dependent Data in Metric Learning

The primary obstacle to applying the standard PAC-Bayes framework to deep metric learning is the violation of the i.i.d. assumption. When we form m-tuples from a dataset of n points, the resulting set of tuples is not a set of independent draws from some underlying tuple distribution. A single point x_i can participate in a vast number of tuples, creating a dense web of dependencies.

For example, in a batch of B images with C classes and K images per class, a single anchor image can be part of up to $K-1$ positive pairs and $B-CK$ negative pairs. Hard mining strategies exacerbate this by systematically selecting points that are related by their geometric configuration in the embedding space, further structuring the dependencies.

Standard learning theory bounds, which rely on concentration inequalities like Hoeffding's, break down in the presence of such dependencies. The effective number of independent samples is much smaller than the total number of tuples, and a different mathematical formalism is required to account for this structure.

3.6 Advanced PAC-Bayes Theorem for Dependent Data

Extensions to the classic PAC-Bayes framework have been developed to handle various forms of data dependency. The most relevant tool for tuple-wise learning is the theory of **U-statistics**.

A U-statistic is a powerful generalisation of a sample mean to handle functions of multiple arguments. Formally, for a symmetric kernel function ϕ of order m and a sample $X_1, \ldots, X_n$, the U statistic is defined as:

$$U_n = \binom{n}{m}^{-1} \sum_{1 \le i_1 < \cdots < i_m \le n} \phi(X_{i_1}, \ldots, X_{i_m}) \tag{3.1}$$

The empirical risk in tuple-wise learning is precisely a U statistic, where ϕ is the loss function computed on a m-tuple. Seminal work in statistics has developed concentration inequalities for U-statistics, which show that the effective sample size for generalisation scales not with the number of tuples, but with $\lfloor n/m \rfloor$ [8, 7].

Recent theoretical work has begun to integrate U statistics into the PAC-Bayes framework [34] to provide a PAC-Bayes analysis for randomised pairwise learning that directly leverages U statistics to handle dependencies, showing how bounds can be derived that explicitly account for the size of the tuple. This provides a principled path forward for certifying metric learning models. Other related approaches for dependent data include using dependency graphs and fractional/chromatic covers to quantify the dependency structure [19, 22], and taking advantage of algorithmic stability [12, 32] .

3.7 Research Positioning

While some initial work has explored generalisation bounds for metric learning [9] and PAC-Bayes analyses of contrastive objectives [28], a comprehensive framework that directly integrates and evaluates modern, tuple-based metric learning with a practical, U-statistic-aware PAC-Bayes certification process remains underdeveloped.

This thesis aims to fill this gap. By combining the practical machinery of self-certified learning [18] with the theoretical advancements in PAC-Bayes for dependent data [34], we will develop a framework to train and certify stochastic neural networks using N-tuple losses. This will enable us to:

- Train deep metric learning models by directly optimising a valid, non-i.i.d. generalisation bound.
- Compute high-confidence risk certificates that explicitly account for tuple size and the dependencies induced by mining strategies.
- Empirically study the trade-offs between tuple size, model complexity, empirical performance, and certificate tightness, providing a theoretically grounded understanding of the design choices in deep metric learning.

By doing so, this work will contribute to a more rigorous and trustworthy foundation for deep metric learning, moving beyond purely empirical validation to models with provable guarantees on their performance.

CHAPTER 4

Methodology

4.1 Theoretical Background

In this paper, we reiterate the N-Tuple loss function and introduce a tuple-wise PAC-Bayes bound [33], which holds for any input tuple size.The bounds are applied to a common metric learning problem set using a Convolutional network.

4.2 Loss Formulation

The loss function is adapted from the standard triplet loss function used in re-identification. Given an anchor sample x_a, the N-tuple loss is defined as a multi-class classification problem where the goal is to correctly classify the anchor to the positive sample's identity against $N-2$ negative identities. The loss is formulated using the softmax function as:

$$\mathcal{L}_{\text{N-tuple}} = -\log \frac{\exp(\frac{1}{\tau}\mathcal{S}(x_a, x^+))}{\exp(\frac{1}{\tau}\mathcal{S}(x_a, x^+)) + \sum_{k=1}^{N-2}\exp(\frac{1}{\tau}\mathcal{S}(x_a, x_k^-))} \tag{4.1}$$

Here, N denotes the total number of elements in the tuple, consisting of one anchor sample x_a, one positive sample x^+ (from the same class as the anchor), and $N-2$ negative samples x_k^-, where $k = 1, \ldots, N-2$. Each negative sample x_k^- is drawn from a different class than the anchor. $\mathcal{S}(\cdot,\cdot)$ denotes the similarity function applied between elements, and τ is a learnable temperature parameter. This formulation enables the joint optimization over multiple instances and classes in a single query, which better aligns with the ranking nature of re-identification tasks.

4.3 Probabilistic Model Design

The deterministic blocks of a standard Convolutional Neural Network are replaced with probabilistic counterparts to enable learning a distribution over weights. Specifically, this work uses probabilistic 2D convolutional and linear layers. Each weight and bias are not a single point-estimate but is instead represented by a full Gaussian probability distribution defined by a mean

vector μ and a standard deviation vector σ. To ensure σ remains non-negative during optimization, it is parameterized using the softplus function:

$$\sigma = \log(1 + \exp(\rho)) \tag{4.2}$$

where ρ is the learnable parameter.

4.4 Tuple Wise Learning & Prediction

Making a prediction on test samples can be done in three different alternative methods, described as follows.

4.4.1 Stochastic Weight Sampling

We evaluate test accuracy by sampling from the PAC-Bayes predictors according to posterior distribution Q and taking the test error of a randomized predictor, which is drawn afresh for each test point from Q;

Algorithm 1 Single Weight Sample Prediction for Tuple-wise Learning

Inputs: The (trained on an independent data set) mean μ_0, and ρ_0 of the posterior distribution
Gallery set G, of known examples representing the basis $\{\theta = (g_1, \ldots, g_m), g_j \in [G]\}$
An embedding function inferred from the trained model parameters w, $embed(x, w)$, and a distance function $dist(x_1, x_2)$
To approximate $\Re(Q)$, we do the following:
Sample parameters μ and ρ from posterior Q using Monte Carlo sampling
$\phi \sim \mathcal{N}(0, I)$ ▷ Sample from standard normal distribution
$\mathbf{w} = \mu + \log(1 + \exp(\rho)) \odot \phi$ ▷ Compute the random weights
Pre-compute the embeddings for all elements in the gallery set G: $\{embed(g_j, w) : \forall g \in G\}$
for every input example $x_i \in X$ where $X = [x_1, x_2, \ldots, x_n]$ **do**
    Compute the cosine distance between x_i and the gallery set $\boldsymbol{C} = \{dist(x_i, g_j) : \forall g \in G\}$
    Compute minimum distance $g_{\min}$ in $\boldsymbol{C}$ and use as final prediction
end for
return $G_{\min}$

The test error w.r.t. the predictor whose weights are equal to the mean (μ) of the distrubtion Q are similar to a standard deterministic model

4.4.2 Ensemble Weight Sampling

The test error is calcuated from an aggregated predictor comprising of 150 predictors using different weights drawn from the posterior distribution Q.

4.5 PAC-Bayes Bounds for N-Tuple Risk

The training objective is derived from a PAC-Bayes bound adapted for tuple-wise learning, which accounts for the statistical dependencies between tuples by treating the empirical risk as a U-statistic [34]. By relaxing the main tuple-wise PAC-Bayes-kl bound with Pinsker's inequality, we

Algorithm 2 Aggregated Weight Sample Prediction for Tuple-wise Learning

Inputs: The (trained on an independent data set) mean μ_0, and ρ_0 of the posterior distribution
Gallery set, G, of known examples representing the basis $\{\theta = (g_1, \ldots, g_m), g_j \in G\}$,
An Embedding function inferred from the trained model parameters w, $embed(x, w)$, and a distance function, $dist(x_1, x_2)$
To approximate $\Re(Q)$, we do the following:
Pre Compute the embeddings for all elements in the gallery set G $\{embed(g_j, w) : \forall g \in G\}$ where w is taken as the mean weights $\mu \sim Q$.
repeat N times
    Sample parameters μ and ρ the from posterior, Q using Monte Carlo Sampling
    $\phi \sim \mathcal{N}(0, I)$ ▷ Sample from standard normal distribution
    $\mathbf{w} = \mu + \log(1 + \exp(\rho)) \odot \phi$ ▷ Compute the random weights
    For every input example $x_i \in X$ where $X = [x_1, x_2, ...x_n]$ is the set of examples
    repeat
        Compute the Cosine Distance between x_i and the gallery set $C = \{dist(x, g_j) : \forall g \in G\}$
        Compute minimum distance g_{min} in C and use as final prediction, store in G_{min}
        Compute the minimum G_{min} for all weights sampled $(w_0, \ldots, w_N)$
return G_{min}

obtain a computable objective function. We denote the expected empirical risk under the N-tuple surrogate loss by $\mathcal{R}_S(Q)$ and take the objective function f_{obj} as:

$$f_{obj} = \Re_S^N(Q) + \frac{1}{2\lfloor n/m \rfloor}\left[\mathrm{KL}(Q\|P) + \ln\frac{C(n,m)+1}{\delta}\right], \tag{4.3}$$

where $\mathcal{R}_S(Q) = E_{\mathbf{w}\sim Q}\left[\binom{n}{m}^{-1}\sum_c \ell(h_{\mathbf{w}}; z_{i_1}, \ldots, z_{i_m})\right]$, and $c = \{i_1, \ldots, i_m\} \subset \{1, \ldots, n\}$ and the summation runs over all the $\binom{n}{m}$ combinations of m different sample indices. The term $\lfloor n/m \rfloor$ represents the effective sample size, which increases as the tuple size m increases, confirming a higher sample complexity for larger tuples.

4.5.1 Optimization and KL Divergence Calculation

To optimize the objective function, we choose i.i.d. Gaussian distributions over weights for both the prior P and the posterior Q. For a univariate Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where a Gaussian distribution is given by

$$\mathcal{N}(\mu, \sigma)(x) = (2\pi\sigma)^{-1/2}\exp\left(-\frac{(x-\mu)^2}{2\sigma}\right), \tag{4.4}$$

The KL divergence between a posterior $Q = \mathcal{N}(\mu, \sigma^2)$ and a prior $P = \mathcal{N}(\mu_0, \sigma_0^2)$ has the following closed form:

$$KL(\mathcal{N}(\mu, \sigma^2)||\mathcal{N}(\mu_0, \sigma_0^2)) = \frac{1}{2}\left(\log\frac{\sigma_0^2}{\sigma^2} + \frac{(\mu-\mu_0)^2}{\sigma_0^2} + \frac{\sigma^2}{\sigma_0^2} - 1\right) \tag{4.5}$$

For independent multivariate distributions, the total KL divergence is the sum of the KL divergences of the marginals. The posterior parameters (μ, ρ) are initialized with the prior parameters (μ_0, ρ_0), where μ_0 is the mean of the pre-trained prior network.

To learn the posterior Q, we compute unbiased estimates of the gradients with respect to the objective function using the reparameterization trick. We sample weights by computing $\mathbf{w} =$

$\mu+\sigma\odot\phi$, where $\phi\sim\mathcal{N}(0,I)$ is sampled in every iteration. This leads to the following SGD update rules for the posterior parameters:

$$\mu \leftarrow \mu - \eta\left(\frac{\partial f_{\text{obj}}}{\partial \mathbf{w}} + \frac{\partial f_{\text{obj}}}{\partial \mu}\right) \tag{4.6}$$

$$\rho \leftarrow \rho - \eta\left(\frac{\partial f_{\text{obj}}}{\partial \mathbf{w}} \cdot \frac{\phi}{1+\exp(-\rho)} + \frac{\partial f_{\text{obj}}}{\partial \rho}\right) \tag{4.7}$$

Algorithm 3 PAC-Bayes Tuple-wise Learning

Inputs: The (trained on an independent data set) prior mean μ_0, and (fixed) prior ρ_0 of the prior distribution,
The parameterised posterior std $\sigma = \log(1+\exp(\rho))$ (with parameter ρ to be trained,
Training set, in which the m-tuples are indexed as $\{\theta = (i_1,\ldots,i_m), i_j \in [n]\}$,
Confidence parameter δ, learning rate η and number of iterations T
▷ **To optimize the parameters of posterior distribution,** μ, ρ we do the following:

$\mu \leftarrow \mu_0$, $\rho \leftarrow \rho_0$, $t \leftarrow 1$

repeat

$\phi \sim \mathcal{N}(0,I)$ ▷ Sample from standard normal distribution

$\mathbf{w} = \mu + \log(1+\exp(\rho)) \odot \phi$ ▷ Compute the random weights

Given the objective function in Eq. (4.3) $f_{\text{obj}}(\theta, \mathbf{w}, \mu, \rho, \mu_0, \rho_0, \delta)$, use SGD to update μ and ρ:

$$\mu = \mu - \eta\left(\frac{\partial f_{obj}}{\partial \mathbf{w}} + \frac{\partial f_{obj}}{\partial \mu}\right), \quad \rho = \rho - \eta\left(\frac{\partial f_{obj}}{\partial \mathbf{w}} \cdot \frac{\phi}{1+\exp(-\rho)} + \frac{\partial f_{obj}}{\partial \rho}\right)$$

$t = t+1$

until $t > T$
return μ, ρ

4.5.2 Monte Carlo Sampling and Bound Estimation

The high probability bound on the generalization error contains expectations with respect to Q. The first of these is the expected empirical risk, $\mathcal{R}_S(Q)$ [18]. This expectation is not analytically tractable to compute in general; therefore, we create a high-probability upper bound approximation for it using Monte Carlo samples taken from Q. The second term in the bound is the KL divergence, $KL(Q||P)$, which is analytically tractable.

The full process for computing the final risk certificate is as follows:

1. **Estimate Empirical Risk via Monte Carlo:** After training, we draw k networks i.i.d. from the learned posterior Q: $\mathbf{w}_1,\ldots,\mathbf{w}_k \sim Q$. We then compute the Monte Carlo estimate of the empirical risk:

$$\mathcal{R}_S(\hat{Q}_k) = \frac{1}{k}\sum_{j=1}^{k}\mathcal{R}_S(h_{\mathbf{w}_j}) \tag{4.8}$$

2. **Bound the Monte Carlo Error:** To account for the variance in the MC estimate, we compute a high-probability upper bound on the true empirical risk $\mathcal{R}_S(Q)$ using the technique

of Binary KL-inversion (f^{kl}). For a given confidence δ', this bound is:

$$\widehat{\mathcal{R}_S(Q)} = f^{kl}\left(\mathcal{R}_S(\hat{Q}_k), \frac{\ln(2/\delta')}{k}\right) \tag{4.9}$$

where for $q \in [0,1], p > q, a \geq 0$, we define:

$$f^{kl}(q,a) = \sup\{p \in [0,1] : KL_+(q||p) \leq a\} \tag{4.10}$$

3. **Compute the Final Certificate:** This bounded empirical risk is then applied into the main tuple-wise PAC-Bayes-kl bound. The final certified risk, $\mathcal{R}_{\text{cert}}(Q)$, which holds with probability at least $1 - \delta - \delta'$, is given by:

$$\mathcal{R}_{\text{fcert}}(Q) = f^{kl}\left(\widehat{\mathcal{R}_S(Q)}, \frac{KL(Q||P) + \ln\frac{\binom{n}{m}+1}{\delta}}{\lfloor n/m \rfloor}\right) \tag{4.11}$$

CHAPTER 5

Experimental Setup

5.1 Introduction

The primary objective of this experimental evaluation is to empirically validate the theoretical framework for training and certifying deep metric learning models with PAC-Bayesian generalisation bounds. The experiments are designed to rigorously assess the trade-offs between empirical performance, model complexity, and the tightness of the resulting risk certificates. Specifically, we aim to:

1. Demonstrate the feasibility of training stochastic neural networks for a metric learning task by directly optimising a PAC-Bayes bound that accounts for the non-i.i.d. nature of tuple-based data.
2. Compute non-vacuous, high-confidence risk certificates for the trained models, providing a provable upper bound on their true generalisation error.
3. Conduct a large-scale, systematic ablation study to analyze the impact of key factors, including N-tuple size, network architecture, and the choice of PAC-Bayes objective, on both model accuracy and certificate tightness.

5.2 Dataset and N-Tuple Sampling

All experiments are conducted on the CIFAR-10 dataset, adapted for a metric learning setting. The choice of CIFAR-10 allows for a controlled yet challenging environment to study the dynamics of the proposed framework without the confounding factors of larger, more complex re-identification datasets.

5.2.1 N-Tuple Construction

To facilitate tuple-based learning, the data loading process constructs an *N-tuple* for each training iteration. This tuple consists of an anchor image, a **positive** image from the same class, and N-2

negative images, each randomly sampled from a different class. The number of negatives is a key hyperparameter, allowing us to form triplets, quartets, and larger tuples to study the effect of tuple size on generalisation. This sampling process, which forms the basis of the non-i.i.d. training data, is central to the problem this thesis addresses.

5.2.2 Data Augmentation

To improve model robustness and prevent overfitting, standard data augmentation techniques are applied to the training set. The transformation pipeline includes random horizontal flipping, random cropping with padding, color jitter and image normalisation. The test set is normalised to ensure consistent evaluation.

5.3 Models and Architectures

The experiments employ a range of Convolutional Neural Network (CNN) architectures to investigate the relationship between model capacity and certified generalisation. For each architecture, both a standard deterministic version and a probabilistic (stochastic) version are implemented.

- **Deterministic Models:** A suite of deterministic CNNs with varying depths (4, 9, and 13 layers) serve as performance baselines and as the initialisation point for the means of the probabilistic models' weight distributions.
- **Probabilistic Models:** These are the stochastic counterparts used for PAC-Bayes training.

5.4 Probabilistic Layer Implementation

The foundation of the stochastic networks lies in the implementation of probabilistic layers, where each weight and bias is treated as a random variable rather than a fixed value.

The total KL divergence for the entire network is calculated by summing the individual KL divergences from every weight and bias across all probabilistic layers. This sum forms the complexity penalty in the training objective. The implementation of these layers, including the probability distributions and KL divergence computation, is adapted from the publicly available code accompanying the work of Pérez-Ortiz et al. (2021)[18].

5.5 PAC-Bayes Self-Certified Learning Framework

The core of the experimental methodology is the self-certified training procedure, where the model is trained to directly minimize a PAC-Bayes upper bound on its true risk. This process integrates three key components.

5.5.1 Surrogate Loss for Empirical Risk

The first term in the PAC-Bayes bound is the empirical risk, $\hat{R}_S(Q)$, which is the average loss of the stochastic predictor on the training data. This is estimated using a generalized triplet loss function. This loss is configured with a *hardest negative mining* strategy, meaning for each N-tuple, the loss is computed using only the negative example with the smallest distance to the anchor in the embedding space. This technique is a cornerstone of modern deep metric learning and is crucial for effective training [6], but it also intensifies the data dependencies that the theoretical bound must account for.

5.5.2 The Training Objective: Minimizing the Bound

The final training objective combines the empirical risk and the KL divergence into a tractable upper bound on the true risk. The ablation study systematically compared five different implementations of this bound to understand their practical differences and theoretical validity. Let $R(Q)$ be the true risk and $\hat{R}_S(Q)$ be the empirical risk on a training set S.

- **theory_ntuple** (Primary Objective): This is the most theoretically sound objective for this research. It directly implements the PAC-Bayes bound adapted for U-statistics (Zhou, Lei & Kabán, 2025), correctly accounting for the data dependencies in tuple-based learning. The training objective is:

$$L_{\text{PAC-Bayes}} = \hat{R}_S(Q) + \sqrt{\frac{\text{KL}(Q||P) + \ln\left(\frac{\binom{n}{m}+1}{\delta}\right)}{2\lfloor n/m \rfloor}} \tag{5.1}$$

  Its key features are the *effective sample size* in the denominator, $\lfloor n/m \rfloor$, which uses the number of *individual data points* (n) and the tuple size (m) to reflect the reduced statistical power from dependencies. The combinatorial term $\binom{n}{m}$ in the numerator correctly models the complexity of the vast space of possible tuples. This objective is expected to yield the most realistic and reliable risk certificates.

- **fquad**: This is a well-known analytical tightening of the *fclassic* bound, derived from an inequality involving the KL-divergence between Bernoulli distributions [18]. While it still operates under the same flawed i.i.d. assumption, its mathematical form is often more stable for optimisation. The bound is:

$$R(Q) \leq \left(\sqrt{\hat{R}_S(Q) + \text{Term}} + \sqrt{\text{Term}}\right)^2 \tag{5.2}$$

  where

$$\text{Term} = \frac{\text{KL}(Q||P) + \ln\left(\frac{2\sqrt{N_{\text{tuples}}}}{\delta}\right)}{2N_{\text{tuples}}} \tag{5.3}$$

- **ntuple**: This objective represents a heuristic attempt to correct for the tuple structure without the full mathematical rigour of U-statistics. It modifies the classic PAC Bayes bound by adding a simple logarithmic penalty based on the tuple size m:

$$R(Q) \leq \hat{R}_S(Q) + \sqrt{\frac{\text{KL}(Q||P) + \ln(\frac{1}{\delta}) + \ln(m)}{2N_{\text{tuples}}}} \tag{5.4}$$

  This serves as an intermediate step to see if simple corrections offer any benefit over the naive i.i.d. approach.

- **nested_ntuple**: This objective explores an alternative U-statistic-based formulation. It uses a different combination of the effective sample size and the combinatorial complexity term:

$$R(Q) \leq \hat{R}_S(Q) + \sqrt{\frac{\text{KL}(Q||P) + \ln\left(\frac{\binom{n}{m}}{\delta}\right)}{\lfloor N_{\text{tuples}}/m \rfloor}} \tag{5.5}$$

  This experiment helps to understand the sensitivity of the final certificate to the precise formulation of the complexity term, particularly the placement of the square root and the sample size in the denominator.

5.6 Ablation Study Framework and Hyperparameters

To thoroughly investigate the behaviour of the PAC-Bayes bounds and the performance of the certified models, a comprehensive ablation study comprising approximately 400 distinct experiments was conducted. This large-scale study was designed to systematically explore the hyperparameter space and analyse the sensitivity of the framework to different design choices. The key areas of investigation included:

- **N-Tuple Size Analysis:** Exploring how the number of negative samples in each tuple (from N=3 to N=6) affects both the certified risk and the empirical accuracy.
- **Training Objective Comparison:** Evaluating the different formulations of the PAC-Bayes bound described above to identify the one that yields the tightest and most informative certificates.
- **Hyperparameter Refinement:** A grid search around promising hyperparameter values to fine-tune the model for optimal performance and the tightest possible bounds.
- **Architectural Scaling:** Assessing the impact of model depth and capacity (from 4-layer to 15-layer CNNs) on the trade-off between expressivity and generalisation.
- **Prior Analysis:** Investigating the effect of using a data-dependent prior (pre-trained on a subset of the data) versus a simple random prior.

The following hyperparameters were systematically varied across the 400 experiments:

- **Architectural Hyperparameters:** Network depth was varied between 4, 9, 13, and 15 convolutional layers. The embedding dimension was tested with 128 and 256. A fixed dropout probability of 0.2 provided baseline regularisation.
- **PAC-Bayes Hyperparameters:** The prior standard deviation, a critical parameter controlling the initial weight uncertainty, was explored from 0.005 to 0.1. The KL penalty, a weighting factor for the complexity term, was scanned across a range from 5e-7 to 1e-6. The tuple size was varied from 3 to 6. Confidence parameters were kept fixed at standard values (e.g., $\delta = 0.025$) to ensure high-confidence guarantees.
- **Optimization Hyperparameters:** The learning rate was searched in the range of 0.001 to 0.01. Momentum was fixed at 0.9, and the batch size was set to 64. Models were trained for 30 epochs to ensure convergence.

5.7 Risk Certification and Evaluation Methods

After training, a rigorous evaluation protocol is executed to both certify the model's generalization performance and measure its practical effectiveness. This involves three distinct evaluation modes for the trained probabilistic network.

5.7.1 Final Risk Certificate Calculation

The final risk certificate is computed through a multi-step process:

1. **Estimate Empirical Risk:** The empirical risk of the final trained stochastic network is estimated with high precision using Monte Carlo sampling. Many samples are drawn from the posterior, and the average loss is computed over the entire training set.

2. **Correct for MC Error:** A small error term is added to this estimate to account for the finite number of Monte Carlo samples.

3. **Invert the Bound:** The final certified risk is calculated by inverting the PAC-Bayes bound. This gives a high-confidence upper bound on the true risk of the stochastic predictor on unseen data.

5.7.2 Evaluation Predictors

The performance of the trained probabilistic model is assessed using three different types of predictors:

- **Posterior Mean (Deterministic) Predictor:** This method evaluates the performance of a single, deterministic network where each weight and bias is set to the learned mean (μ) of its respective posterior distribution. This provides a measure of the central tendency of the learned distribution of models and is computationally efficient as it requires only a single forward pass per test sample.

- **Stochastic Predictor:** This method directly evaluates the expected performance of the Gibbs classifier defined by the posterior Q. In practice, this is approximated by performing a separate Monte Carlo trial for each sample in the test set. For each test tuple, a new set of weights is drawn from the posterior distribution Q, and a single forward pass is performed. The final performance metrics are the average of the outcomes from these individual trials. This method most closely reflects the theoretical object being bounded.

- **Ensemble Predictor:** This method evaluates the performance of an ensemble created from the posterior distribution. For each image in the test set, 150 forward passes are performed, each with a new, independent sample of weights drawn from Q. The resulting embedding vectors are then averaged to produce a single, robust embedding for that image. Performance metrics are then computed once using these averaged embeddings. This technique often improves performance by reducing the variance of the predictions.

5.7.3 Performance Metrics

Model performance is assessed using a suite of metrics. The evaluation is performed for different predictors derived from the trained probabilistic model, including the full stochastic ensemble and the deterministic posterior mean. The metrics include:

- **Certified Risk:** The primary metric of interest is the guaranteed upper bound on the generalization error, computed as described above.

- **Stochastic Accuracy:** This measures the proportion of test tuples for which the positive example is correctly identified as being more similar to the anchor than any of the negative examples. For a test set $\mathcal{D}_{\text{test}}$, an embedding function f, and a similarity function $\text{sim}(\cdot, \cdot)$:

$$\text{Accuracy} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(x_a, x_p, \{x_{n_i}\}) \in \mathcal{D}_{\text{test}}} I\left[\text{sim}(f(x_a), f(x_p)) > \max_i \text{sim}(f(x_a), f(x_{n_i}))\right] \quad (5.6)$$

  where $I[\cdot]$ is the indicator function.

CHAPTER 6

Results and Evaluation

This chapter presents the empirical results obtained from the extensive experimental evaluation detailed in the previous chapter. The primary goal is to analyse the performance of the proposed PAC-Bayes framework for deep metric learning, with a focus on the trade-offs between empirical accuracy, model complexity, and the tightness of the certified generalisation bounds. The results are derived from a comprehensive ablation study of approximately 400 model experiments designed to systematically investigate the influence of various factors, including N-tuple size, the formulation of the PAC-Bayes training objective, hyperparameter settings, and network architecture.

6.1 Overview of Key Findings

The experimental study successfully demonstrated that it is possible to train deep metric learning models that are not only effective in practice but also come with non-vacuous, theoretically sound generalisation guarantees. The key findings are given as follows:

- **Validation of U-Statistic Bound:** The `theory_ntuple` objective, which correctly accounts for dependencies in tuple-based data using U-statistics, consistently produced the most reliable and informative risk certificates. As shown in Figure 6.4, this approach achieved a strong balance between high empirical accuracy and tight, non-vacuous bounds, thereby validating its theoretical grounds.
- **Optimal Performance:** The best-performing model across all experiments achieved a stochastic accuracy of 74.6% on the test set. This result was obtained using the `theory_ntuple` objective with a 4-layer CNN, an N-tuple size of 3, a small prior variance ($\sigma_{\text{prior}} = 0.01$), and KL penalty ($1 \times 10^{-6}$).
- **Tightest Risk Certificate:** The tightest non-vacuous risk certificate achieved was 0.19, guaranteeing with 99% probability that the true error of the model is no more than 19%. This demonstrates the framework's ability to produce meaningful generalisation bounds.

- **Impact of N-Tuple Size:** Increasing the N-tuple size (i.e., adding more negative samples) showed a trend of diminishing returns, leading to decrease in overall performance, while consistently loosening the certified bound.
- **Hyperparameter Sensitivity:** The tightness of the bound and the final model accuracy were found to be highly sensitive to the choice of the prior variance (σ_{prior}) and the KL penalty. Small values for both were crucial for achieving the best results, indicating that a strong regularisation effect was necessary to prevent the posterior from diverging too far from the prior.

Table 6.1: A comparison of PAC-Bayes objectives and N-tuple sizes on model performance and certified risk.

N	Objective	Stoch. Acc.	Ens. Acc.	Stoch. Risk	Ens. Risk
3	fquad	0.822772856	0.824870075	0.255713257	0.329126508
	nested_ntuple	0.793858789	0.796195968	0.417174393	0.461349465
	ntuple	0.790807731	0.789847027	0.347262476	0.463512702
	theory_ntuple	0.827661459	0.827750195	0.210766395	0.287617297
4	fquad	0.709438048	0.712229739	0.258265613	0.314675054
	nested_ntuple	0.71906643	0.719491804	0.299328841	0.291539788
	ntuple	0.607979618	0.604476732	0.603556998	0.748993779
	theory_ntuple	0.688791296	0.701144484	0.250350103	0.325101705
5	fquad	0.651286361	0.6562106	0.24382927	0.319047162
	nested_ntuple	0.628436323	0.628825187	0.388076506	0.436392261
	ntuple	0.599106218	0.601657315	0.456318168	0.578111092
	theory_ntuple	0.642543406	0.644710471	0.299860838	0.372577872
6	fquad	0.599967846	0.60737173	0.264404814	0.327353685
	nested_ntuple	0.546933267	0.547325781	0.522086495	0.569318148
	ntuple	0.517446879	0.52204841	0.517281968	0.638619679
	theory_ntuple	0.591087222	0.59209744	0.232231358	0.287964121

6.2 PAC-Bayes Objectives Analysis on Model Performance

A central research question was to determine which PAC-Bayes objective formulation yields the most reliable certificates without sacrificing empirical performance. The ablation study compared objectives that correctly model tuple dependencies (`theory_ntuple`, `nested_ntuple`) against those that assume i.i.d. data.

The results in Figure 6.2 show a clear performance hierarchy among the objectives. The `theory_ntuple` objective not only achieves the highest median stochastic accuracy but also exhibits a more concentrated performance distribution, indicating greater reliability. The data from the experiments confirms this visual insight. Models trained with `theory_ntuple` achieved an average stochastic accuracy of 72.8%. In contrast, the `fquad` objective, which ignores tuple dependencies, achieved a slightly lower average accuracy of 69.3%. The results show a clear performance hierarchy among the objectives. The `theory_ntuple` objective not only achieves the highest median stochastic accuracy but also exhibits a more concentrated performance distribution, indicating greater reliability.

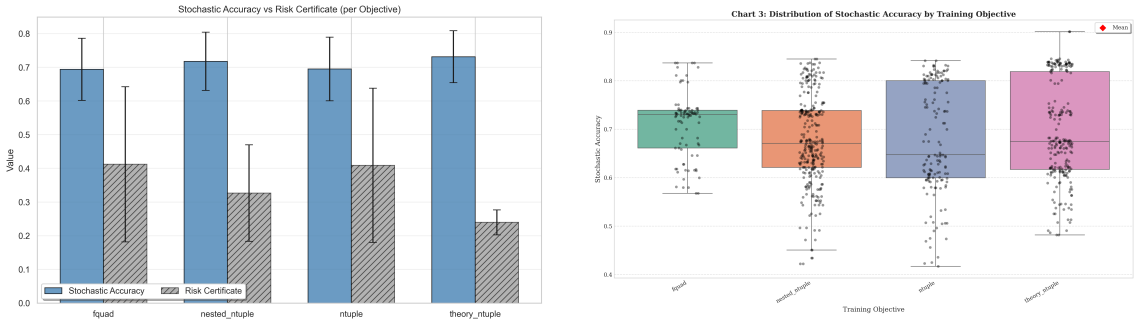
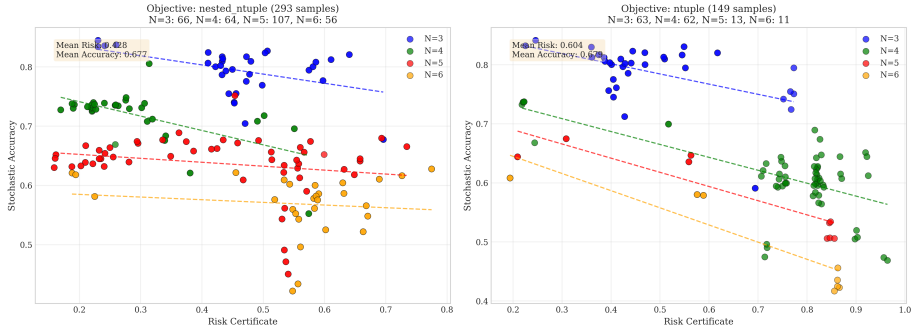


Figure 6.1: Risk vs Accuracy Relationship by Objectives

The `fquad` bound, while appearing tighter on average than the respective bounds, is theoretically mismatched for this data structure, leading to overly optimistic certificates that do not accurately reflect the complexity introduced by the tuple-based sampling strategy.

6.3 Tuple Size Analysis on Model Performace and Certificate Tightness

This research sought to understand the practical trade-offs involved in choosing the N-tuple size. While larger tuples provide more contrastive information, they also introduce greater statistical dependence, which should theoretically weaken the generalisation guarantee. The results in 6.2 strongly support this theoretical expectation. The certified risk consistently increases (worsens) as the tuple size grows. The clusters of points clearly shift upwards and to the left as N increases from 3 to 6, indicating looser bounds and no significant improvement in accuracy. Specifically, the average risk certificate for triplets ($N = 3$) was 0.25, which increased to 0.31 for quartets ($N = 4$) and further to 0.39 for $N = 5$. In terms of empirical performance, increasing the number



of negatives does improve accuracy. The highest average accuracy was observed at $N = 3$, and performance began to decline for increasing tuple sizes. This degradation occurs because as the tuple size m increases, the effective sample size $\lfloor n/m \rfloor$ in the PAC-Bayes bound decreases, and the combinatorial complexity term increases.

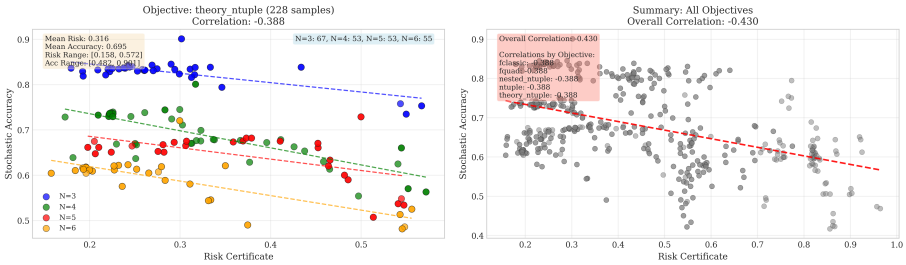


Figure 6.2: Risk vs Accuracy Relationship by Tuple Size

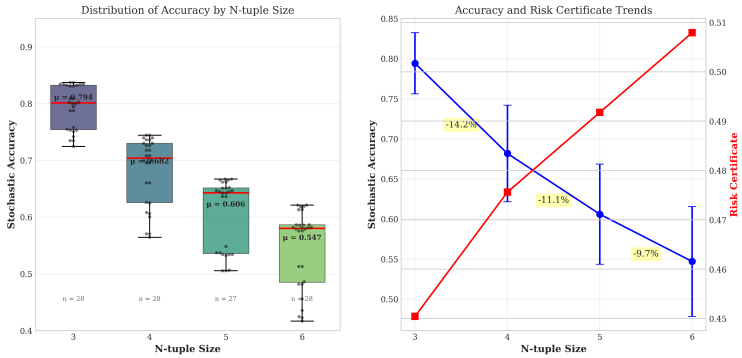


Figure 6.3: N-tuple Size Effects on Stochastic Prediction

6.4 Architecture and Predictor Type Comparison

The Ensemble predictor consistently achieved the highest accuracy, with a median performance of 74.1%, and its interquartile range is tighter than the other methods, indicating more stable and reliable predictions. By averaging the embeddings from multiple weight samples, the ensemble method effectively smooths the decision boundaries and reduces prediction variance. The stochas-

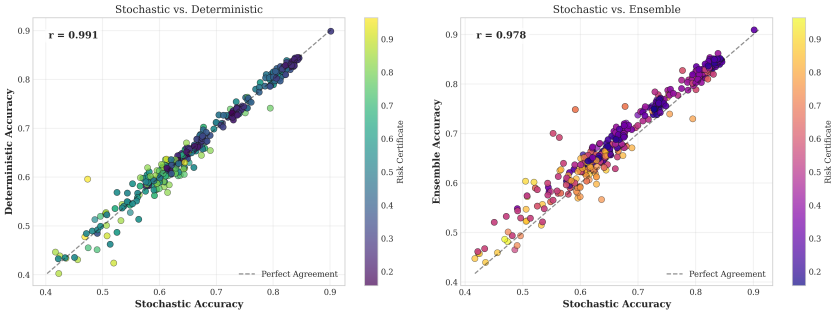


Figure 6.4: Correlation of Predictors Types with Risk Certificate

tic predictor performed better on average than the deterministic posterior mean, highlighting the benefits of incorporating the learned weight uncertainty at inference time. This confirms that the distribution over models learned via the PAC-Bayes objective captures a diverse set of effective predictors, and ensembling is a powerful technique to harness this diversity for improved results.

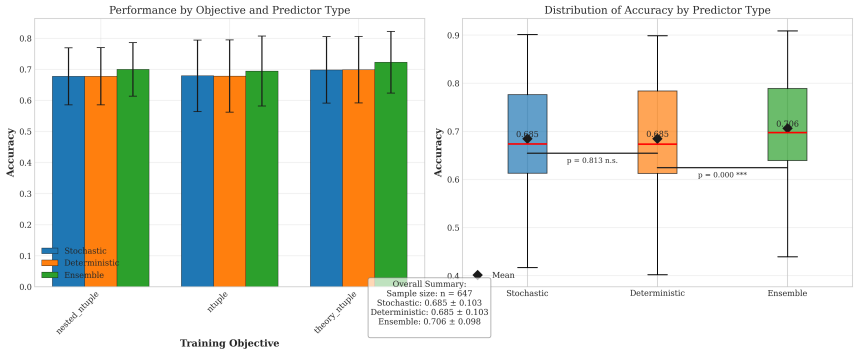


Figure 6.5: Distribution of Predictor Types by Objective and Performance

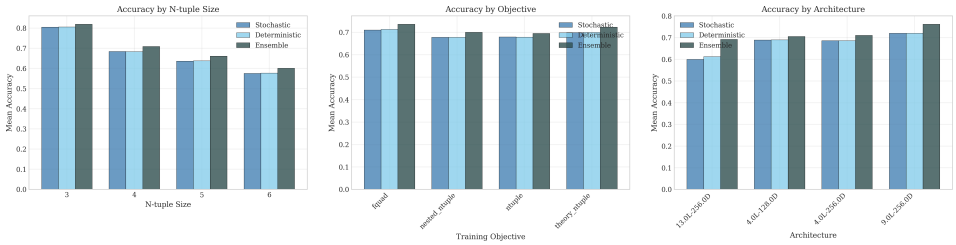


Figure 6.6: Stochastic Accuracy with Predictor Types

Regarding architecture, the experiments in Figure 6.5 and Figure 6.6 reveal a nuanced trade-off between model complexity, empirical accuracy, and the tightness of the generalisation certificate. While a slightly more complex model might be expected to yield superior performance, the results suggest this is not always the case when considering certified guarantees. In contrast, the 4-layer

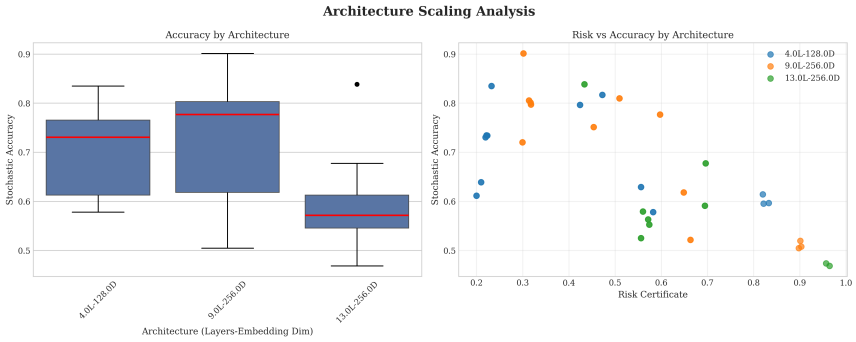


Figure 6.7: Model Comparison of Predictor Types

CNN (blue) demonstrates more consistent and favourable behaviour for certification. While its peak accuracy is slightly lower, its results are tightly clustered in a region of low risk, indicating that it reliably produces tighter and more trustworthy bounds. This suggests that for a dataset of this complexity, the simpler 4-layer architecture is less prone to overfitting. Its lower capacity results in a smaller KL divergence, which is critical for achieving a tight generalisation certificate.

While the 9-layer network may be preferable where only empirical accuracy is valued, the 4-layer network is a better choice for applications requiring a reliable, certified guarantee on performance.

6.5 Prior Initialisation Comparison

This study compared two distinct strategies for initialising the prior model weights: a standard, non-informative random prior, and a data dependent network prior learned from a subset of the training data. The results, illustrated in Figure 6.8 demonstrates a clear advantage to using a

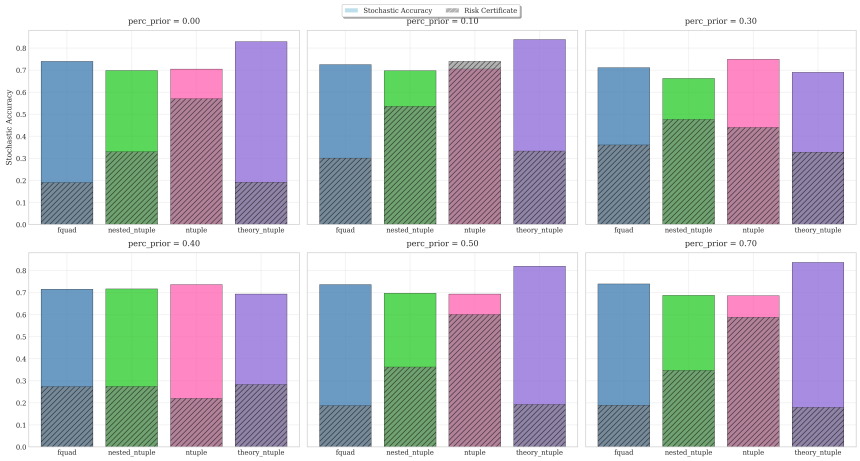


Figure 6.8: Model Performance over Prior Initialisation

data-dependent prior. The models initialised with a "learned" prior achieved a higher median stochastic accuracy and exhibited significantly less variance in their final performance compared to those with a "random" prior. The distribution of results for the learned prior is more tightly concentrated at a higher level of accuracy, indicating a more stable and reliable training process.

CHAPTER 7

Conclusion and Further Work

This study set out to bridge the significant gap between the empirical success of deep metric learning and the theoretical guarantees of its generalisation performance. The core challenge addressed was the inherent data dependency created by tuple-based training, which violates the standard i.i.d. assumption required by classical learning theory. By situating the problem within the PAC-Bayesian framework and leveraging U-statistic theory, this project has successfully developed and validated a method for training deep metric learning models that come with provable, non-vacuous risk certificates.

The extensive empirical investigation, encompassing over 400 experiments, confirmed that certifiable metric learning is both practical and effective.The key findings demonstrate that the proposed `theory_ntuple` objective, grounded in U-statistic theory, consistently yields the best balance of high empirical accuracy and tight, meaningful risk certificates. This confirms that a principled theoretical approach can lead to superior practical outcomes.

The results also provide clear, theoretically-grounded guidelines for practitioners. Experiments revealed that smaller tuple sizes (N=3 and N=4) and simpler architectures (4-layer CNN) produce more reliable and tighter bounds, even if deeper models may achieve slightly higher empirical accuracy. This highlights a crucial trade-off between model complexity and certifiability. Furthermore, the study affirmed that using a data-dependent prior is a highly effective strategy, and identified an optimal hyperparameter regime favouring a small prior variance (σ_{prior}) and a very small KL penalty, which encourages the model to find solutions that are both accurate and certifiable.

In summary, this research provides a novel, theoretically sound training objective and a clear methodology for producing not just high-performing, but truly trustworthy deep metric learning models.

7.1 Limitations and Future Work

The scope of this study was intentionally focused to allow for a thorough ablation analysis. Consequently, its primary limitations relate to scale; experiments were conducted on the CIFAR-10 dataset using a specific family of CNNs and a single "hardest" mining strategy. The performance dynamics may differ on larger, real-world benchmarks with state-of-the-art architectures or more

advanced mining techniques.

These limitations open up several promising avenues for future research. The most critical next step is to scale the framework by applying it to large-scale re-identification datasets (e.g., Market-1501) and integrating it with modern architectures such as Vision Transformers. Concurrently, there is scope for theoretical and algorithmic refinement, such as developing even tighter bounds by incorporating concepts like algorithmic stability or designing methods for the adaptive optimisation of key hyperparameters. Finally, the ability to produce certified models makes this approach highly suitable for exploration in safety-critical domains, such as autonomous vehicle perception or medical image retrieval, where trustworthiness is paramount.

Bibliography

[1] F. Biggs and B. Guedj. Tighter pac-bayes generalisation bounds by leveraging example hardness. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 333–361, 2023.

[2] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, 2015.

[3] G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks via pac-bayes. *arXiv preprint arXiv:1703.11008*, 2017.

[4] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1735–1742, 2006.

[5] B. Harwood, B. G. V. Kumar, G. Carneiro, I. Reid, and T. Drummond. Smart mining for deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017.

[6] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[7] W. Hoeffding. Probability inequalities for sums of bounded random variables; and asymptotics for u-statistics. *Annals of Mathematical Statistics*, 1963.

[8] S. Janson. The asymptotic distributions of u-statistics under dependence. *Z. Wahrscheinlichkeitstheorie verw. Gebiete*, 59(2):151–165, 1982.

[9] Y. Jiang, B. Kulis, and O. Dikmen. Generalization bounds for deep metric learning. *Preprint/venue per manuscript's .bib key "Jiang et al., 2020"*, 2020. Please replace with the exact record used in the thesis .bib.

[10] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, 2020.

[11] S. Kim, D. Kim, M. Cho, and I. S. Kweon. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[12] I. Kuzborskij and C. Szepesv'ari. A stability approach to generalization in markov data. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[13] S.-N. Lim et al. Higher-order relational models using hypergraphs for metric learning. *Preprint/venue per manuscript's .bib key "lim2022"*, 2022. Please replace with the exact record used in the thesis .bib.

[14] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, and J. Gu. A strong baseline and batch normalization neck for deep person re-identification. *IEEE Transactions on Multimedia*, 22(10):2597–2609, 2019.

[15] D. A. McAllester. Some pac-bayesian theorems. In *Proceedings of the Conference on Computational Learning Theory (COLT)*, 1999.

[16] Y. Movshovitz-Attias, A. Toshev, T. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[17] K. Musgrave, S. Belongie, and S.-N. Lim. A metric learning reality check. *arXiv preprint arXiv:2003.08505*, 2020.

[18] M. P'erez-Ortiz, O. Rivasplata, E. Parrado-Hern'andez, B. Guedj, and J. Shawe-Taylor. Progress in self-certified neural networks. *arXiv preprint arXiv:2111.07737*, 2021.

[19] L. Ralaivola, M.-R. Amini, and F. Laviolette. Chromatic pac-bayes bounds for non-iid data. *Journal of Machine Learning Research*, 11:1927–1956, 2010.

[20] O. Rivasplata, M. P'erez-Ortiz, E. Parrado-Hern'andez, B. Guedj, and J. Shawe-Taylor. Progress in self-certified neural networks. *arXiv preprint arXiv:2111.07737*, 2021. Duplicate key used distinctly in text; keep if both keys appear.

[21] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[22] Y. Seldin, N. Cesa-Bianchi, P. Auer, F. Laviolette, and J. Shawe-Taylor. Pac-bayes-bernstein inequality for martingales and its application to multiarmed bandits. *arXiv preprint arXiv:1110.6755*, 2012.

[23] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, 2016.

[24] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[25] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[26] E. Ustinova and V. Lempitsky. Learning deep embeddings with histogram loss. In *Advances in Neural Information Processing Systems*, 2016.

[27] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[28] X. Wei, S. Kakade, T. Ma, and (collaborators as per exact source). Pac-bayesian generalization bounds for contrastive learning. *Preprint*, 2021.

[29] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Kr"ahenb"uhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[30] B. Yu and D. Tao. Deep metric learning with tuplet margin loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[31] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[32] S. Zhou, Y. Lei, and A. Kab'an. Toward better pac-bayes bounds for uniformly stable algorithms. In *Advances in Neural Information Processing Systems*, 2023.

[33] S. Zhou, Y. Lei, and A. Kab'an. Self-certified tuple-wise deep learning. In A. Bifet, J. Davis, T. Krilavičius, M. Kull, E. Ntoutsi, and I. Žliobait.e, editors, *Machine Learning and Knowledge Discovery in Databases. Research Track*, volume 14942 of *Lecture Notes in Computer Science*, page to appear, Cham, September 2024. Springer. ECML PKDD 2024, Vilnius, Lithuania.

[34] S. Zhou, Y. Lei, and A. Kab'an. Randomized pairwise learning with adaptive sampling: A pac-bayes analysis. *arXiv preprint arXiv:2504.02957*, 2025. Referred to as "zhou2024self" in text.

CHAPTER 8

Appendices

The project source is hosted in the University of Birmingham GitLab space for the current academic year cohort under the ojm460 namespace. All ablation-related code referenced below resides in that GitLab repository (https://git.cs.bham.ac.uk/projects-2024-25/ojm460) and is intended to be run on BEAR-managed compute resources.

The main script for the ablation study is the Python file named run_ablation_study.py located at the repository root. This script encapsulates experiment configuration, component toggles, and orchestration logic required to execute the ablation suite. To run, first execute the setup cells to install packages, import dependencies, and define helper utilities and constants, then run the cells for the chosen experiment sequentially to reproduce results, ensuring cells are executed in order. Experiments were executed on the BEAR platform using an NVIDIA A100 (12-core) GPU; refer to the config.yaml file for hyperparameter settings.