University of Modena and Reggio Emilia
XXXVII cycle of the International Doctorate School in
Information and Communication Technologies

Advanced Discretization Techniques in the Era of Deep Learning

Riccardo Benaglia

Supervisor: Prof. Simone Calderara
Ph.D. Course Coordinator: Prof. Luigi Rovati

Modena, 2025

Review committee:

Prof. Lamberto Ballan, University of Padova
Daniele Bigoni, Ph.D., Accenture S.p.A.

Voler togliere il dubbio dalle nostre teste è come volere togliere l'aria ai nostri polmoni.

Abstract

Deep learning has transformed how we tackle complex tasks, but challenges persist, particularly in the areas of lifelong learning and generating reliable predictions in dynamic environments. This thesis investigates advanced discretization techniques aimed at addressing two crucial domains: Continual Learning (CL) and trajectory forecasting. Both present distinct challenges related to managing latent spaces and ensuring long-term adaptability.

Discretization techniques play a pivotal role in handling graph structures and latent space quantization. They simplify the management of complex, continuous data by structuring it in a more analyzable and model-friendly format. In graph structures, discretization helps capture relationships between entities, making these connections more interpretable and manageable. Latent space discretization, on the other hand, transforms continuous latent variables into discrete ones, improving the interpretability and efficiency of machine learning models. This is particularly advantageous in tasks like clustering, representation learning, and generative modeling, where clear, discrete categories within latent space allow models to generalize more effectively and produce more robust predictions.

In the first part, this thesis investigates the problem of catastrophic forgetting in Artificial Neural Networks (ANNs) during Continual Learning. Unlike biological intelligence, which integrates new knowledge throughout life without losing prior understanding, ANNs struggle when faced with a non-static training data distribution. CaSpeR-IL is introduced as a geometric regularizer that enhances the stability of rehearsal-based CL methods by enforcing spectral constraints on the latent space. Specifically, it mitigates the disruption caused by class interference during data replay, promoting a better partitioning of the latent space. This approach improves the state-of-the-art performance of CL models on standard benchmarks by maintaining more consistent predictions, even under memory constraints.

In the second part, the thesis addresses the challenge of trajectory forecasting, a key component in fields like video surveillance and sports

analytics. Forecasting the future movements of agents, such as basketball players interacting in real-time, requires a deep understanding of their intentions. Here, Vector Quantized Variational Autoencoders (VQ-VAEs) are exploited, utilizing a discrete latent space to prevent posterior collapse while capturing diverse future trajectories. The thesis proposes a novel adaptation mechanism through low-rank updates to the latent codebook, enabling instance-based customization of latent representations. This ensures that past motion patterns and contextual information dynamically shape the latent space, leading to more accurate and diverse trajectory predictions. It is empirically demonstrated that combining this approach with a diffusion-based predictive model achieves state-of-the-art performance on multiple trajectory forecasting benchmarks.

This work comprehensively studies discretization techniques in deep learning, showcasing their power in solving continual learning and trajectory forecasting challenges through geometric and latent space regularization strategies.

Sommario

Il deep learning ha trasformato il modo in cui affrontiamo compiti complessi, ma rimangono delle sfide, in particolare nell'apprendimento continuo e nella generazione di previsioni affidabili in ambienti dinamici. Questa tesi esplora tecniche avanzate di discretizzazione mirate ad affrontare due ambiti cruciali: il Continual Learning (CL) e la previsione di traiettorie di essere umani. Entrambi presentano sfide uniche legate alla gestione degli spazi latenti e alla capacità di adattamento a lungo termine.

Le tecniche di discretizzazione svolgono un ruolo fondamentale nella gestione delle strutture di grafi e nella quantizzazione degli spazi latenti. Queste semplificano la gestione di dati complessi e continui strutturandoli in un formato più analizzabile e adatto alla modellazione. Nelle strutture a grafo, la discretizzazione aiuta a catturare le relazioni tra le entità, rendendo tali connessioni più interpretabili e gestibili. La discretizzazione dello spazio latente, invece, trasforma le variabili latenti continue in discrete, migliorando l'interpretabilità e l'efficienza dei modelli di machine learning. Ciò è particolarmente vantaggioso in compiti come il clustering, l'apprendimento delle rappresentazioni e la modellazione generativa, dove categorie chiare e discrete all'interno dello spazio latente permettono ai modelli di generalizzare meglio e produrre previsioni più robuste.

Nella prima parte, la tesi indaga il problema del "catastrophic forgetting" nelle reti neurali artificiali (ANN) nel contesto del Continual Learning. A differenza delle connessioni bioligiche degli essere viventi, che integrano nuove conoscenze senza perdere quelle precedenti, le ANN faticano quando affrontano una distribuzione dei dati di addestramento non statica. Viene presentato CaSpeR-IL, un regolarizzatore geometrico che migliora la stabilità dei metodi di CL basati sul "rehearsal", imponendo vincoli spettrali sullo spazio latente. In particolare, Casper-IL mitiga l'interferenza tra classi durante la riproduzione dei dati, promuovendo un miglior partizionamento dello spazio latente. Questo approccio migliora le prestazioni di modelli "State Of The Art" di CL nei benchmark standard, mantenendo previsioni più consistenti anche con vincoli di memoria.

Nella seconda parte, la tesi affronta la sfida della previsione delle traiettorie di pedoni, un aspetto chiave in campi come la videosorveglianza

e l'analisi sportiva. Prevedere i movimenti futuri di agenti, come i giocatori di basket che interagiscono in tempo reale, richiede una profonda comprensione delle loro intenzioni. In questo caso, sono stati sfruttati i Vector Quantized Variational Autoencoders (VQ-VAE), che utilizzano uno spazio latente discreto per evitare il collasso della posterior (tipico dei VAE), catturando traiettorie future diversificate. La tesi propone un nuovo meccanismo di adattamento tramite aggiornamenti a bassa dimensionalità del codebook latente, permettendo una personalizzazione delle rappresentazioni latenti basata sui singoli casi. Ciò garantisce che i modelli di movimento passati e le informazioni contestuali modellino dinamicamente lo spazio latente, portando a previsioni più accurate e diversificate. Inoltre, viene mostrato come, combinando questo approccio con un diffusion model per la previsione dei codici discreti (output del processo di quantizzazione), si ottengono prestazioni SOTA su diversi benchmark di previsione delle traiettorie.

Questo lavoro studia in modo completo le tecniche di discretizzazione nel deep learning, dimostrando la loro efficacia nel risolvere le sfide del Continual Learning e della previsione di traiettorie tramite strategie di regolarizzazione geometrica e dello spazio latente.

Table of Contents

List of Abbreviations

ADE Average Displacement Error

C-VAE Conditional Variational Autoencoder

CaSpeR-IL Continual Spectral Regulariser for Incremental Learning

CL Continual Learning

Class-IL Class-Incremental Learning

CO^{2}L Contrastive Continual Learning

CSCCT Class-Incremental Learning with Cross-Space Clustering and Controlled Transfer

DDPM Denoising Diffusion Probabilistic Models

DER Dark Experience Replay

DER++ Dark Experience Replay++

DNN Deep Neural Network

Domain-IL Domain-Incremental Learning

ER Experience Replay

ER-ACE Experience Replay with Asymmetric Cross-Entropy

ER-RPC Experience Replay with Regular Polytope Classifier

FAA Final Average Accuracy

FAAF Final Average Adjusted Forgetting

FAF Final Average Forgetting

FDE Final Displacement Error

FT Finetuning

GAN Generative Adversarial Network

GCN Graph Convolutional Network

GDumb Greedy Sampler and Dumb Learner

GNN Graph Neural Network

iCaRL Incremental Classifier and Representation Learning

ID In-Distribution

JT Joint Training

LB-EBM Latent Belief Energy-Based Model

LGG Latent Geometry Graph

LLM Large Language Model

LRVQ Low-rank Adaptation for VQ-VAE

MID Motion Indeterminacy Diffusion

ML Machine Learning

MLE Maximum Likelihood Estimation

MSE Mean Squared Error

NBA NBA SportVU Dataset

NFL NFL Football Dataset

OOD Out-of-Distribution

PDF Probability Density Function

PECNet Predicted Endpoint Conditioned Network

PODNet Pooled Outputs Distillation Network

R-MNIST Rotated MNIST

RBM Rehearsal-Based CL Method

S-*mini*Img Sequential *mini*ImageNet

S-CIF10 Sequential CIFAR-10

S-CIF100 Sequential CIFAR-100

S-GAN Social-GAN

SCR Supervised Contrastive Replay

SDD Stanford Drone Dataset

SGD Stochastic Gradient Descent

SOTA state-of-the-art

Task-IL Task-Incremental Learning

VAE Variational Autoencoder

VQ Vector Quantization

VQ-VAE Vector Quantized Variational Autoencoder

X-DER eXtended Dark Experience Replay

X-DER $^{\text{RPC}}_{\text{future}}$ X-DER with RPC on future heads

Part I

Introduction

Chapter 1

Overview

1.1 A Philosophical and Scientific Perspective on Discretization

Discretization, the process of dividing continuous phenomena into distinct parts, resonates deeply with how humans understand and interact with the world. It is not just a computational tool but a reflection of a fundamental cognitive process that has been explored in philosophy, science, and mathematics for centuries.

Philosophers have long pondered how humans make sense of the continuous and often chaotic flow of reality. Immanuel Kant, in his seminal Critique of Pure Reason (1781), argued that the human mind actively imposes structure on sensory experiences to make them intelligible. According to Kant, this structuring process involves **segmenting the world into categories and concepts** – a form of mental discretization that transforms raw data into usable knowledge.

Similarly, René Descartes revolutionized mathematics and philosophy by introducing the Cartesian coordinate system in his 1637 work *La Géométrie*. By discretizing continuous space into measurable dimensions of x and y, Descartes laid the groundwork for modern analytical geometry. This act of **partitioning space into discrete points** not only enabled scientific progress but also reflected a deeper philosophical position: that understanding the infinite requires reducing it to finite, manageable parts.

In the sciences, discretization has been a pillar for both theoretical advancements and practical applications. For instance, Ludwig Boltzmann, one of the pioneers of statistical mechanics, demonstrated that macroscopic properties like temperature and pressure could be explained by the **discrete states of microscopic particles**. This paradigm shift highlighted how discretization allows scientists to bridge scales—from the invisible motions of atoms to observable physical phenomena.

Fig. 1.1: Joseph von Fraunhofer's 1814 solar spectrum, showing the discrete absorption lines that led to the development of quantum mechanics [12].

The emergence of quantum mechanics in the early 20th century transformed our understanding of the natural world. It revealed that energy levels—demonstrated by phenomena like the solar spectrum (see Fig. 1.1)—and other **particle properties** are **fundamentally quantized**, existing in distinct and **discrete states**. These groundbreaking discoveries underscore that discretization a fundamental characteristic of the universe's underlying structure.

In the modern era, Claude Shannon's 1948 work on information theory [101] underscored the power of **discretization in digital communication**. Shannon showed that by converting analog signals into discrete bits, information could be transmitted with high fidelity despite noise, proving that discretization is essential for both efficiency and reliability.

Beyond its philosophical and scientific significance, discretization has become central to data analysis and machine learning. By **discretizing continuous variables into intervals** or categories, machine learning models gain improved interpretability, enhanced robustness, and better alignment with certain data modalities. Techniques such as clustering-based discretization and entropy-based methods adaptively partition data to reveal meaningful structures, as seen in applications like histogram analysis and k-nearest neighbor (k-NN) graphs. In this thesis, we delve into the role of discretization in deep learning, exploring how it can enhance model performance in two distinct contexts: Continual Learning and Pedestrian Trajectory Generation.

When viewed through these philosophical and scientific lenses, discretization emerges as a **bridge between human cognition and machine**

intelligence. Humans, as Kant observed, impose discrete categories on the continuous flux of reality to understand it. Machines, similarly, use discretization techniques to structure complex, continuous data into actionable insights. Whether in partitioning latent spaces in machine learning or segmenting space-time in physics simulations, discretization exemplifies a **universal strategy for managing complexity**.

1.2 Contributions and Organisation

This thesis presents the candidate's research on discretization techniques applied to deep learning, conducted throughout their doctoral studies. Specifically, the candidate focused on exploring these techniques in two key areas: Continual Learning and Pedestrian Trajectory Generation. The contributions are organized as follows:

- **Part I** presents an **overview of discretization techniques for computer science and machine learning**:
  - **Chapter 1** highlights the fundamental role of discretization as both a cognitive and computational process, connecting human understanding, philosophical insights, and scientific advancements across disciplines.
  - **Chapter 2** presents a comprehensive survey of discretization techniques in machine learning, outlining the technical background required to understand the key methods utilized in this thesis.
- **Part II** comprises two chapters, each structured to include the general presentation of the task, the motivation behind the study, and the study itself:
  - **Chapter 3** begins with an introduction to Continual Learning (CL), followed by the presentation of Continual Spectral Regulariser for Incremental Learning (CaSpeR-IL). It addresses the challenge that Rehearsal-Based CL Methods (RBMs) may fail to produce a disentangled latent space when trained incrementally. This issue is mitigated by introducing a spectral-geometry motivated loss term, Continual Spectral Regulariser for Incremental Learning (CaSpeR-IL), leading to increased compactness in the latent space of state-of-the-art RBMs and improved performance.
  - **Chapter 4** starts with a background on trajectory forecasting, followed by the introduction of Low-rank Adaptation for VQ-VAE (LRVQ). It presents a novel approach to predicting future movements by integrating Vector Quantized Variational Autoencoders (VQ-VAEs) with low-rank adaptations, enhancing the flexibility

and accuracy of trajectory forecasts. This method achieves state-of-the-art performance on different standard benchmarks.

- Finally, **Part III** concludes the thesis by summarizing the presented results and outlining potential future developments and prospective research directions.

Chapter 2

Technical Background

Data discretization and discrete representation are essential techniques in data analysis and machine learning. While much of the early focus in machine learning revolved around continuous representations, such as embeddings or feature maps, the discrete paradigm has emerged as a complementary approach with distinct advantages. By discretizing continuous data into finite categories or leveraging predefined discrete structures, researchers can achieve improved interpretability, enhanced robustness, and better alignment with certain data modalities [15].

At its core, data discretization refers to the process of converting continuous variables into *discrete intervals* or *categories*. This transformation simplifies data and often reveals underlying patterns that might be obscured in the continuous domain. For instance, discretizing numerical ranges into bins, as seen in histogram-based analyses, allows for identifying trends and thresholds. Beyond simple binning, advanced techniques such as clustering-based discretization [107, 69] and entropy-based methods [25, 40] have provided robust ways to adaptively partition data. These methods tailor the discretization process to the data's inherent structure, preserving critical information while eliminating redundancy.

Discretization techniques, such as constructing *k-nearest neighbor* (k-NN) graphs, play a pivotal role in *spectral geometry* by enabling the analysis of continuous geometric structures through discrete representations. In a k-NN graph, each data point is connected to its k-closest neighbors based on a chosen distance metric, effectively capturing the local geometric relationships within the data. This discrete framework facilitates the application of **spectral methods**, including the computation of the Laplacian operator's eigenvalues and eigenfunctions, which are essential for tasks like dimensionality reduction, clustering, and manifold learning [66]. By leveraging k-NN graphs, spectral geometry can approximate the intrinsic geometry of data manifolds, allowing for efficient and robust analysis of complex, high-dimensional datasets. In Sec. 2.1, we

delve deeper into the principles and applications of spectral geometry, highlighting its transformative impact on modern machine learning and data analysis.

Recently, **discrete representations** have found widespread application in various domains of deep learning. In natural language processing (NLP), language data is inherently discrete, consisting of tokens, words, or characters. Tokenization processes, such as subword tokenization, further refine this discrete representation, enabling efficient modeling of textual data [15]. In computer vision, images can be discretized into patches, each represented as a vectorized token in modern transformer architectures [34]. In reinforcement learning and robotics, policies and actions are often represented as sequences of discrete steps, facilitating planning and decision-making [83].

The transition from continuous to discrete representations has particularly brought innovation in generative modeling. **Vector Quantized Variational Autoencoder (VQ-VAE)** exemplifies this shift [116]. By replacing the continuous latent space of conventional autoencoders with discrete codebooks, VQ-VAEs have demonstrated remarkable efficacy for the reconstruction and generation of high-quality data across modalities. This approach addresses challenges like *posterior collapse* in traditional Variational Autoencoders (VAEs) and introduces a structured latent space that is more interpretable and efficient for downstream tasks. In Sec. 2.2, we provide an in-depth overview of VQ-VAEs and their significance in modern machine learning, highlighting recent advancements and applications in generative modeling and representation learning.

2.1 Spectral geometry

Spectral geometry provides a framework for analyzing and processing geometric data through the eigendecomposition of operators such as the Laplacian. This approach leverages *intrinsic properties* of geometric structures, which are invariant under isometric transformations, making it particularly suitable for applications where robustness to deformations is critical.

Mathematically, let $\mathcal{M}$ be a Riemannian manifold, and consider its Laplace-Beltrami operator Δ. The eigenfunctions ϕ_i and eigenvalues λ_i of Δ are solutions to the equation:

$$\Delta\phi_i = -\lambda_i\phi_i, \tag{2.1}$$

where $\lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \cdots$ form a discrete spectrum. These eigenfunctions and eigenvalues encode geometric information about the manifold and are intrinsic, meaning they remain invariant under isometric transformations (*e.g.*, a rotation by $45°$) of $\mathcal{M}$.

For a graph $\mathcal{G}$ with vertices $v_1, \ldots, v_n$, the spectral decomposition satisfies:

$$Lu_i = \lambda_i u_i, \tag{2.2}$$

where L is the Laplacian operator, u_i are the eigenvectors and λ_i are the corresponding eigenvalues. The eigenvalues encode various properties of the graph, such as the number of connected components (indicated by the number of zero eigenvalues) and the presence of clusters.

The quality of the Laplacian and its spectral properties heavily depends on **how the graph $\mathcal{G}$ is constructed**. By discretizing continuous features into graph representations, spectral methods can effectively handle high-dimensional data while preserving its geometric and topological structure. This preprocessing step ensures that the graph encapsulates meaningful relationships, directly influencing the performance of spectral techniques applied on top of it.

The first thing to consider when constructing a graph is the choice of the *distance metric* $\text{dist}(v_i, v_j)$ (or, viceversa, the similarity metric) which determines the edges between vertices v_i and v_j. Different metrics can be used depending on the application, where the most common are the Euclidean distance and the cosine similarity.

Then, it is fundamental to choose a *graph construction method*. Some of the most common graph construction methods are:

- k-Nearest Neighbors (k-NN): the edges are created between a vertex and its k-nearest neighbors based on a similarity metric. The adjacency matrix A is defined as:

$$A_{ij} = \begin{cases} 1, & \text{if } v_j \in \text{k-NN}(v_i) \text{ or } v_i \in \text{k-NN}(v_j), \\ 0, & \text{otherwise,} \end{cases} \tag{2.3}$$

  where $\text{k-NN}(v_i)$ denotes the set of k-nearest neighbors of vertex v_i, determined using the chosen similarity metric;

- ϵ-Neighborhood Graph: the vertices are connected if their pairwise distance is less than a threshold ϵ. The adjacency matrix A is defined as:

$$A_{ij} = \begin{cases} 1, & \text{if } \text{dist}(v_i, v_j) < \epsilon, \\ 0, & \text{otherwise;} \end{cases} \tag{2.4}$$

- Fully Connected Graphs with Weighted Edges: all vertices are connected, and edge weights represent similarities or affinities between vertices. A common choice for the weights is the Gaussian (RBF) kernel, defined as:

$$A_{ij} = \exp\left(-\frac{\text{dist}(v_i, v_j)^2}{2\sigma^2}\right), \tag{2.5}$$

where σ is a scaling parameter controlling the width of the Gaussian kernel.

Finally, *various forms* of the Laplacian operator can be used, each with different properties. For example, the *unnormalized* Laplacian $L = D - A$, where D is the degree matrix, is widely used but sensitive to scale differences, while there are forms of *normalized* Laplacians, such as the symmetrically normalized Laplacian $L_{\text{sym}} = D^{-1/2}LD^{-1/2}$, that account for vertex degree variations, making it robust to uneven degree distributions. Specialized Laplacians may incorporate additional regularization terms or constraints tailored to specific applications.

The gap between consecutive eigenvalues, particularly the *spectral gap* $\lambda_1 - \lambda_0$, is an important quantity in applications such as clustering and partitioning. Larger spectral gaps indicate well-separated clusters, as formalized by Cheeger's inequality [66]:

$$\frac{\lambda_1}{2} \leq \phi \leq \sqrt{2\lambda_1}, \tag{2.6}$$

where ϕ is the conductance of the graph, a measure of the quality of the partition. The second eigenvalue λ_1 (the Fiedler value) is particularly significant in graph partitioning, as it captures the connectivity of the graph's components. This eigenvalue is greater than 0 if and only if $\mathcal{G}$ is a connected graph. This follows from the fundamental result that the number of times 0 appears as an eigenvalue in the Laplacian matrix corresponds to the **number of connected components** in the graph. The work in [66] reaffirms this principle and extends it by addressing a conjecture that relates the presence of k eigenvalues close to zero to the ability to partition the graph into k subsets, each forming a sparse cut.

Spectral Geometry in Machine Learning and *Isospectralization*. Spectral geometry also serves as a tool for compact and isometry-invariant representations of data. As shown in [6], the latent space of deep neural networks (DNNs) can be modeled as a Riemannian manifold. In this framework, latent vectors encode an extrinsic embedding of the manifold, which is not unique due to isometric equivalence. By focusing on intrinsic spectral properties, one avoids overfitting to specific realizations of the data manifold, promoting generalization and stability in downstream tasks.

For enhancing performance in self-supervised learning, the work by Tsai et al. [47] introduces the Spectral Contrastive Loss, a novel approach that integrates spectral decomposition with contrastive learning to enhance representation learning. This method constructs an augmentation graph where edges connect augmented views of the same data point, enabling the application of spectral techniques to capture intrinsic data structures.

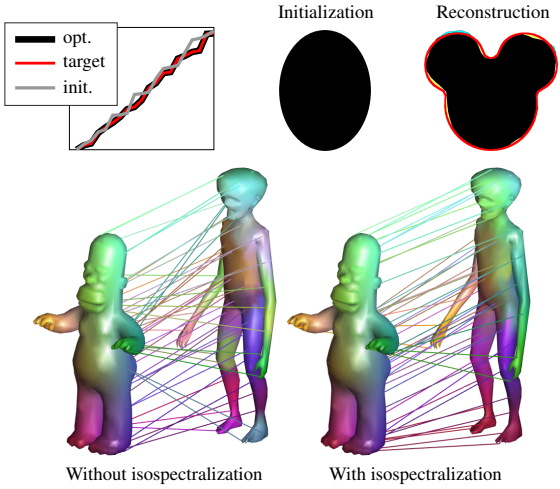


Fig. 2.1: Overall results obtained in [26], using isospectralization to deformable shape matching. *Top row*: recovering Mickey Mouse's shape from its first 20 Laplacian eigenvalues (red points in the leftmost plot) through deformation of an initial ellipsoid. The target embedding is outlined in red atop the reconstructed shape. *Bottom row*: as proposed in [26], aligning Laplacian eigenvalues (*i.e.*, *isospectralization*) serves as a preconditioning step for non-isometric deformable shape matching. The results of a baseline matching algorithm are shown before (left) and after (right) isospectralization, with corresponding points marked in matching colors.

Recent studies have proposed methods that leverage spectral learning with wild data to address both Out-of-Distribution (OOD) generalization and detection challenges. For instance, [120] utilizes a graph-based framework to model data relationships, enabling the derivation of provable error bounds for OOD tasks. Additionally, [36] has explored the impact of In-Distribution (ID) labels on OOD detection performance. By employing a graph-theoretic approach, this work analyzes the separability of ID data from OOD data through spectral decomposition, establishing conditions under which ID labels enhance OOD detection.

Recent advancements in Graph Neural Networks (GNNs) have led to the development of simplified architectures that balance efficiency and performance. For instance, the Simple Spectral Graph Convolution model [132] employs a modified Markov Diffusion Kernel to derive a variant of Graph Convolutional Network (GCN) that combines strengths of both spatial and spectral methods. This approach effectively captures both local and global node contexts, addressing issues like over smoothing and

high computational costs associated with deeper GCNs.

Finally, a special mention goes to *inverse* spectral techniques: these aim to directly manipulate a graph's spectral properties to achieve desired characteristics. By prescribing specific behaviors for certain eigenvalues, these methods seek to construct graphs whose Laplacian spectra align with the specified criteria.

In the geometry processing area, such approaches take the name of *isospectralization* techniques and have been recently used in diverse applications such as deformable shape matching [26] (see Fig. 2.1), shape exploration and reconstruction [77], shape modeling [82] and adversarial attacks on shapes [90]. In Chapter 3 we build upon these techniques to develop a novel isospectralization method that encourages the creation of well-separated latent embeddings to improve model classification accuracy and resilience against catastrophic forgetting in continual learning scenarios.

2.2 Discrete Representation Learning

In deep learning, significant research has focused on learning representations with continuous features [49, 117, 32, 48, 24]. However, in this chapter, we are particularly interested in **discrete representations** [79, 80], which may naturally align with various modalities. Language is inherently discrete and speech are often represented as a sequence of symbols (*i.e.*, tokenization is used as a preprocessing step before inputting sentences into a model [100, 62]). Images can also be described as sequences of patches [34]. Moreover, discrete representations are well-suited for complex reasoning, planning, and predictive learning. A significant advancement in discrete representation learning was made by van den Oord et al. [116], whose framework has inspired numerous developments in recent years.

Vector Quantized Variational Autoencoders (VQ-VAEs) provide a data-driven discretization framework by replacing the continuous latent space of conventional VAEs with a discrete set of codewords. This transformation addresses issues such as *posterior collapse*, a common challenge in generative modeling. Since their introduction [116], VQ-VAEs have demonstrated substantial potential in tasks like image and time-series generation [92], with subsequent research refining the two fundamental stages: the *codebook* learning and the *discrete prior* learning [60].

Standard VAEs [58] employ *i)* an **encoder** $\mathrm{E} \equiv \mathrm{E}(x|\theta_{\mathrm{E}})$ that, given input x, outputs a parametric posterior distribution $q(z|x)$ over latent variable z; *ii)* a **decoder** $\mathrm{G} \equiv \mathrm{G}(z|\theta_{\mathrm{G}})$ that provides the reconstruction of the input data as $p_{\theta_{\mathrm{G}}}(x|z)$. The posterior $q(z|x)$ is encouraged to conform to a standard Gaussian prior distribution $p(z)$, which could lead to over-regularized representations (*i.e.*, *posterior collapse*). VQ-VAEs extend VAEs

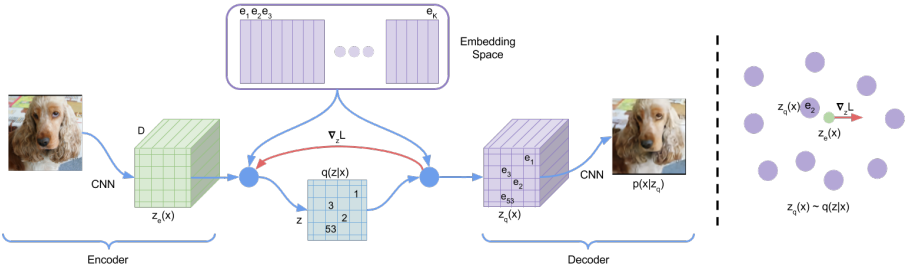


Fig. 2.2: Representation of VQ-VAE in the original work by van den Oord et al. [116]. The encoder maps the input sequence x to a sequence of latent vectors z, which are quantized to the nearest codeword in the codebook e. The decoder reconstructs the input from the quantized latent vectors. Our notation slightly differs from theirs: we represent the encoder as E (denoted as D in the original work), the decoder as G (CNN in their work), the codebook containing C elements (K in their notation), and the quantized latent vectors as z^q (denoted z_q in the original paper).

by employing discrete latent variables and Vector Quantization (VQ) [43]. In particular, both posterior and prior distributions are categorical, and their samples provide indices for a learned **embedding table** $e \in \mathbb{R}^{C \times D}$, which consists of C D-dimensional latent vectors.

Given the input $x \in \mathbb{R}^{T \times d}$, where T is the number of tokens and d is the input channel, the encoder provides a continuous representation $z \in \mathbb{R}^{T \times D}$, where $z_t \in \mathbb{R}^D$ with $t \in \{1, 2, \ldots, T\}$ and D indicates the dimension of the latent space. Then, the VQ-VAE characterizes the posterior as a joint distribution over T independent **categorical** variables $q(c_1, c_2, \ldots, c_T|x)$ (one for each latent). Following the vector quantization paradigm, each marginal $q(c_t|x)$ is determined by matching each element of the encoding sequence z_t with the **nearest** vector in the codebook e:

$$q(c_t|x) = \underbrace{\mathcal{C}(p_1, p_2, \ldots, p_C)}_{[0,\ldots,0,1,0,\ldots,0]} \text{ s.t. } p_c = \begin{cases} 1 & \text{if } c = \text{argmin}_{c' \in \{1,2,\ldots,C\}} \|z_t - e_{c'}\|_2^2 \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

where $\mathcal{C}(\cdot)$ represents the categorical distribution, which outputs a one-hot vector indicating the index of the selected codebook entry. Notably, the posterior distribution is *deterministic* and not stochastic as for VAEs: hence, we can *draw* a sample $z^q \equiv z^q(x)$ from the posterior distribution by **selecting** the corresponding rows of the codebook, as follows:

$$\begin{aligned} z^q &= [e_{c_1}, e_{c_2}, \ldots, e_{c_T}] \\ c_t \sim q(c_t|x) \implies c_t &= \text{argmax}\, q(c_t|x). \end{aligned} \tag{2.8}$$

The subsequent step regards the decoder G, which reconstructs $\hat{x}$ from

the sampled latent vector. During training, the first stage optimizes the following loss:

$$\mathcal{L}_{\text{FS}} = \underbrace{\log p_{\theta_{\text{G}}}(x|z^q)}_{\text{rec. error } \textit{e.g.}\text{, MSE}} + \sum\nolimits_t \underbrace{\|\texttt{sg}[z_t] - e_{c_t}\|^2}_{\text{embedding loss}} + \sum\nolimits_t \underbrace{\|z_t - \texttt{sg}[e_{c_t}])\|^2}_{\text{commitment loss}}, \tag{2.9}$$

where sg is a shortcut for the `stopgradient` operator, which stops back-propagation from that computational node backward. The second term encourages the quantized latent vectors to be as close as possible to the nearest codeword, while the third one encourages the encoder to be *committed* to the chosen codeword.

The goal of the second stage is to learn a parametric model $p_{\theta_p}(c_1, c_2, \ldots, c_T)$ – termed *categorical prior* – which allows to draw new samples from the latent space. During this phase, the modules of the VQ-VAE are no longer subject to learning. Given the trained encoder, each training example x is embedded into a sequence of indices, built by relating each latent vector to the nearest row of the codebook (as in Eq. 2.8). On top of that, the generative model targets the generating process $p(c_1, c_2, \ldots, c_T)$ of the discrete latent codes, and optimizes the following Maximum Likelihood Estimation (MLE) training objective:

$$\mathcal{L}_{\text{SS}} = \mathbb{E}_{\substack{c_1,\ldots,c_T \\ c_t \sim q(c_t|y)}} [-\log p_{\theta_p}(c_1, c_2, \ldots, c_T)]. \tag{2.10}$$

Vector Quantization and Beyond. As mentioned before, recent works tried to improve the representation and the generative capability of the VQ-VAE framework addressing respectively the codebook learning step and the categorical prior learning.

For example, SQ-VAE [110] addresses the issue of codebook collapse commonly observed in VQ-VAE, where only a fraction of the codebook's capacity is utilized. Replacing deterministic quantization with a pair of stochastic dequantization and quantization processes, SQ-VAE enhances codebook utilization without relying on conventional heuristics such as stop-gradient or exponential moving averages.

To create a more comprehensive codebook, [38] supplements the original training losses of VQ-VAE with adversarial training. This adversarial component encourages the generation of perceptually realistic image parts, ensuring that the discrete representations retain high-fidelity details. Subsequently, a Transformer is used to model the global composition of these discrete image components, enabling the synthesis of high-resolution images that exhibit both coherent global structures and fine local details.

Additionally, [131] introduces a regularized vector quantization framework designed to enhance tokenized image synthesis. In particular, the proposed framework includes two key components: *i*) a Prior Distribution Regularization, which quantifies the discrepancy between a prior token

distribution and the predicted token distribution, thereby effectively mitigating codebook collapse and enhancing codebook utilization; and *ii*) a Stochastic Mask Regularization, which incorporates stochasticity into the quantization process to address inference stage misalignment, ultimately improving the model's robustness.

Another notable work is the one of [65]. In this paper the authors address the limitations of traditional vector quantization methods, which often struggle with the rate-distortion trade-off when reducing sequence lengths for autoregressive models, combining Residual-Quantized Variational Autoencoder (RQ-VAE) with an RQ-Transformer. RQ-VAE effectively approximates feature maps with a fixed codebook size, representing images as stacked maps of discrete codes. Subsequently, the RQ-Transformer predicts the next set of quantized feature vectors, enabling efficient modeling of long-range interactions within the image data.

Finally, recent advancements regarding discrete prior learning involve not only architectural modifications, such as the Transformer architecture used in [38], but also a critical reevaluation of autoregression. Two notable works in this context are the Vector Quantized Diffusion Model introduced in [44] and improved in [112], which combines vector quantization with diffusion processes to enhance image generation quality and efficiency, and the Masked Generative Image Transformer (MaskGIT) presented in [18], which employs a bidirectional transformer decoder to enable non-sequential image generation, significantly accelerating the synthesis process.

We built upon these advancements to develop a novel approach to trajectory forecasting, as detailed in Chapter 4. In particular, our work enhances the representation capability of VQ-VAEs by integrating low-rank adaptations of the codebook, and makes use of a discrete diffusion model to learn the categorical prior.

Part II

Discretization Techinques for Deep Learning

Chapter 3

Latent Space Modelling via Geometric Constraints

3.1 Background and related work

Intelligent beings in nature continuously learn and adapt to changing environments by integrating new knowledge with prior understanding into a cohesive framework. In stark contrast, artificial neural networks (ANNs) tend to overfit the current data they are trained on, often resulting in the rapid degradation of previously learned information – a phenomenon termed *catastrophic forgetting* [78].

Continual Learning (CL) is a subfield of machine learning dedicated to developing methods that enable deep models to retain previously acquired knowledge while learning from new data [27].

While catastrophic forgetting can be studied in conjunction with any Machine Learning (ML) task (*e.g.*, segmentation [16, 125], detection [87, 56], generation [130], captioning [28], etc.), this chapter is focused on continual classification problems. Such a choice is in line with the majority of CL literature and allows us to highlight the key issues of incremental model operation with simple and easy-to-follow experiments.

In CL, a model f with parameters θ is trained on a sequence of T tasks $\{\mathcal{T}_0, \ldots, \mathcal{T}_{T-1}\}$. The i^{th} task consists of input-label pairs $\{x_i^{(n)}, y_i^{(n)}\}_{n=1}^{|\mathcal{T}_i|} \subset \mathcal{X}_i \times \mathcal{Y}_i$. While these data-points are *i.i.d.* (*i.e.*, independent and identically distributed) within $\mathcal{T}_i$, the overall training procedure does not abide by the *i.i.d.* assumption, as the data distribution changes between tasks. The objective of CL is minimizing the risk over all tasks:

$$\mathcal{L}_{\text{CL}} \triangleq \sum_{i=0}^{T-1} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{T}_i} \big[\mathcal{L}(f_\theta(x), y)\big], \tag{3.1}$$

where $\mathcal{L}$ indicates the loss associated with the classification task (*e.g.*, the categorical cross-entropy) given prediction $f_\theta(x)$ and ground-truth label y. While pursuing the optimal solution to Eq. 3.1 – $\theta^* = \text{argmin}_\theta \mathcal{L}_{\text{CL}}$ – the learner is not given free access to all data: only one task can be learned at any given time and with a limited number of observations. To prevent the performance deterioration on past data associated with catastrophic forgetting, CL models combine the optimization of the empirical risk on the current task $\mathcal{T}_c$ with a separate regularisation term $\mathcal{L}_R$:

$$\hat{\mathcal{L}}_{\text{CL}} \triangleq \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{T}_c} \big[\mathcal{L}(f_\theta(x), y)\big] + \mathcal{L}_{\text{R}}. \tag{3.2}$$

The additional term $\mathcal{L}_{\text{R}}$ can vary significantly for different models. This gives rise to distinct classes of CL approaches, which are presented in detail in Sec. 3.1.4.

3.1.1 Continual Learning Scenarios

The definition of the CL classification problem provided in the previous section is general enough to allow for the formulation of different experimental settings, with varying characteristics and degrees of complexity. A rigorous taxonomy of possible experimental CL designs was introduced by *Van de Ven et al.* [114], encompassing three *academic* scenarios that gave practitioners and researchers a common language for the description of CL experiments. This section will present a detailed description of these settings.

All three scenarios outlined in [114] define a supervised classification problem in which tasks are presented sequentially. During each task, the model only has access to a specific subset of the dataset, and transitions between tasks are explicitly signaled, allowing the model to take preparatory steps before moving on. Based on the type of decision function the model must learn, *Van de Ven et al.* categorize these scenarios as follows:

- Task-Incremental Learning (Task-IL): classification tasks are disjoint (*i.e.*,, $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for tasks $\mathcal{T}_i$ and $\mathcal{T}_j$). The model is informed of the task identity during evaluation, simplifying learning. Task identity $t \in \mathcal{J}$ is provided during inference, enabling a task-conditioned classification function $f : \mathcal{X} \times \mathcal{J} \to \mathcal{Y}$. This scenario is considered the easiest of the classical scenarios [39, 4] and is valuable for measuring forgetting without classifier bias from imbalanced data presentation [121];

- Domain-Incremental Learning (Domain-IL): all classes are presented in each task, but inputs undergo task-dependent transformations (*e.g.*,, rotations or pixel permutations). The task identity is not provided at test time, requiring the model to learn a task-independent

classification function $f : \mathcal{X} \to \mathcal{Y}$. Specialized datasets are often used in Domain-IL experiments;

- Class-Incremental Learning (Class-IL): similar to Task-IL, with disjoint classification tasks, but without task identity at test time. The model must learn a unified classification function $f : \mathcal{X} \to \mathcal{J} \times \mathcal{Y}$ that identifies both the source task and the corresponding class. This scenario introduces the challenge of incrementally learning a unified classifier [51], potentially leading to bias toward recently seen classes [121, 2]. It is considered the most challenging and representative of the three scenarios [39].

3.1.2 Continual Learning Benchmarks

This section provides an overview of the key characteristics of the CL benchmarks employed throughout this chapter. Unless otherwise specified, all experiments in subsequent chapters will adhere to these benchmark specifications, utilize Stochastic Gradient Descent (SGD) as the optimizer, and allocate 10% of the training data as a validation set for hyperparameter tuning. All benchmarks focus on image classification tasks. Below are detailed descriptions of the benchmarks:

- Sequential CIFAR-10 (S-CIF10) [129]: this benchmark is derived from the CIFAR-10 dataset [61] and consists of 5 tasks, each containing 2 disjoint classes. For each class, the dataset provides 5000 training images and 1000 test images, all formatted as 32×32 RGB;

- Sequential CIFAR-100 (S-CIF100) [129, 93, 22]: based on the CIFAR-100 dataset [61], this benchmark splits the 100 classes into 10 sequential tasks, each comprising 10 disjoint classes. Each class includes 500 training images and 100 test images, all in 32×32 RGB format;

- Sequential *mini*ImageNet (S-*mini*Img) [22, 37, 33]: this benchmark is created from the *mini*ImageNet dataset [118], a curated 100-class subset of the ImageNet dataset [31]. It is organized into 20 tasks, each containing 5 disjoint classes. For each class, the dataset provides 500 training images and 100 test images, all in 84×84 RGB format;

- Rotated MNIST (R-MNIST) [70]: in this benchmark, the learner needs to classify all MNIST [64] digits for 20 subsequent tasks, in which images are rotated by a random angle in the interval $[0, \pi[$ (different for each task).

3.1.3 Evaluation Metrics

As CL experiments span multiple tasks, several evaluation metrics have been proposed in the literature to express distinctive aspects of the learning dynamics. Let a_i^t denote the model accuracy on the i^{th} task after training on task $\mathcal{T}_t$, in this chapter these measures are used to evaluate the performance of a CL methods:

- **Final Average Accuracy (FAA)** assesses the final average performance on the overall joint classification problem after learning all tasks incrementally:

$$\text{FAA} \triangleq \frac{1}{T}\sum_{j=0}^{T-1} a_j^{T-1}. \tag{3.3}$$

  This measure provides a compact summary of the trade-off between learning a task in an incremental manner or doing so jointly by allowing a direct comparison with the *i.i.d.* baseline. For this reason, FAA is largely adopted in literature [70, 27, 4] and it is the main performance indicator used in this chapter;

- **Final Average Forgetting (FAF)** [19, 21, 20] measures the average performance degradation occurring on past tasks between their peak and final accuracy:

$$\text{FAF} \triangleq \frac{1}{T-1}\sum_{j=0}^{T-2} f_j, \quad \text{s.t.} \quad f_j = \max_{l\in\{0,\dots,T-2\}} a_j^l - a_j^{T-1}. \tag{3.4}$$

  This measure can also take on a negative value in the case of a model that improves its accuracy on past tasks over time (a phenomenon known as positive *backward transfer*);

- **Final Average Adjusted Forgetting (FAAF)** is an adjusted version of FAF (introduced for the first time in [9]). This variant aims at allowing easier comparison of the forgetting rate between models with different peak accuracy by focusing on performance degradation alone and excluding backward transfer:

$$\begin{aligned} \text{FAAF} &\triangleq \frac{1}{T}\sum_{j=0}^{T-1}\left[\frac{a_j^* - a_j^{T-2}}{a_j^l}\right]^+, \\ \text{s.t.} \quad a_j^l &= \max_{t\in\{j,\dots,T-1\}} a_j^t, \quad \forall j \in \{0,\dots,T-2\}, \end{aligned} \tag{3.5}$$

  where $[\cdot]^+$ indicates lower-bound clipping to zero.

3.1.4 State of the Art

In this section, we provide a review of some of the most prominent approaches to CL, which will serve as benchmarks for the experiments conducted later in this work. CL methods are typically grouped into three main categories [39, 27]: *architectural* methods are typically very effective in counteracting forgetting devoting distinguished sets of parameters to distinct tasks [74]; *regularization-based* methods do not alter the model's architecture, but instead condition its evolution using additional loss function terms to prevent forgetting previous tasks [59]; *rehearsal-based* methods (RBMs) operate by maintaining a fixed-size working memory of previously encountered exemplars, which are then used to prevent forgetting by either replaying them directly and/or using them as an additional source of regularisation [22]. Since our method was benchmarked exclusively against RBM methods, we provide a detailed list of these models below, highlighting their key characteristics. Furthermore, CL experiments often include both a lower and an upper bound for performance evaluation by incorporating two standard **baseline** approaches: **Finetuning (FT)** and **Joint Training (JT)**. FT involves training the Deep Neural Network (DNN) directly on the incoming data stream without applying any strategies to mitigate catastrophic forgetting. In contrast, JT trains the model on all available data simultaneously, thus avoiding forgetting entirely.

Rehearsal-Based Methods. The methods used to benchmark our approach are rehearsal-based methods. Below, we summarize the most prominent rehearsal-based methods in the CL literature:

- **Experience Replay (ER)** [91, 95]: a foundational rehearsal method that maintains an *i.i.d.* sample of past data in a memory buffer, typically managed via *reservoir sampling* [119]. Buffer data is interleaved with the incoming stream for joint optimization. Experience Replay (ER) remains a robust baseline and the foundation for many SOTA approaches;

- **Experience Replay with Regular Polytope Classifier (ER-RPC)** [88] extends ER with the *Regular Polytope Classifier*, which enforces uniform output space distribution among all classes to ensure consistent classification boundaries;

- **Incremental Classifier and Representation Learning (iCaRL)** [93] combines a self-knowledge distillation loss with a *nearest-mean-of-exemplars* classifier, enabling robust predictions even with small buffers and challenging benchmarks;

- **Experience Replay with Asymmetric Cross-Entropy (ER-ACE)** [14] addresses class imbalances in ER by separating cross-entropy contribu-

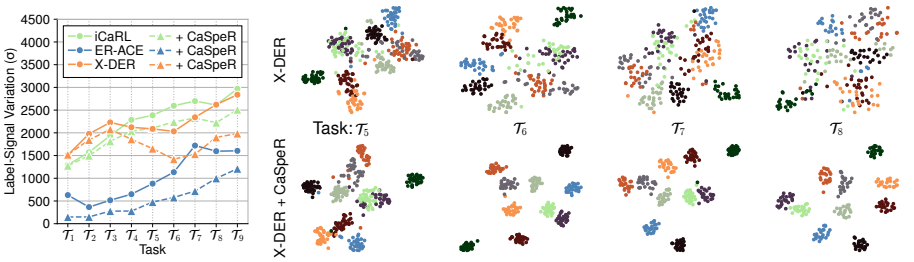
tions of buffer and stream classes, improving accuracy and reducing interference;

- **Pooled Outputs Distillation Network (PODNet)** [35] extends iCaRL by applying self-distillation across convolutional layers and supporting multi-modal representations;
- **Contrastive Continual Learning (CO^2L)** [17] replaces the cross-entropy objective with contrastive learning to reduce buffer-induced bias, requiring a separate linear classifier for inference;
- **Greedy Sampler and Dumb Learner (GDumb)** [89] stores data in the memory buffer without training and trains a new model on the buffer from scratch during evaluation, questioning recent CL advances;
- **Supervised Contrastive Replay (SCR)** [73] applies Supervised Contrastive Loss [57] to enforce consistency between two views of input data;
- **Dark Experience Replay++ (DER++)** [13] integrates rehearsal with knowledge distillation by aligning the network's logits with those sampled throughout the training trajectory, thereby promoting consistency with its past and mitigating catastrophic forgetting;
- **Class-Incremental Learning with Cross-Space Clustering and Controlled Transfer (CSCCT)** [7] combines latent-space clustering to maintain class boundaries with a controlled transfer objective to prevent negative transfer from unrelated classes;
- **eXtended Dark Experience Replay (X-DER)** [10] enhances Dark Experience Replay (DER) method by revising replay memory to incorporate new information and facilitating the learning of previously unseen classes. Specifically, in this chapter, we use as a competitor the more effective baseline that combines X-DER with the Regular Polytope Classifier [88].

3.2 Motivation

This chapter focuses on the evolution of the latent space for replayed data as tasks progress.

We observe that the learner faces challenges in separating latent projections of replay examples from different classes, making the downstream classifier susceptible to interference when the input distribution shifts or representations are perturbed. Drawing on the Riemannian nature of the latent space in DNNs [6], we address this issue by utilizing **spectral geometry** (Sec. 2.1). This approach enables manipulation of the



(a) (b)

Fig. 3.1: Illustrations of the alterations occurring in RBMs' latent spaces. (a) A quantitative evaluation measured as Label-Signal Variation (σ) within the LGG for buffer data-points – *lower is better*; (b) t-SNE embedding of the features computed by X-DER $^{\text{RPC}}_{\text{future}}$ for buffered examples in later tasks (top). Interference between classes is visibly reduced if CaSpeR-IL is applied (bottom). All experiments are carried out on S-CIF100, (a) has $m = 500$, (b) has $m = 2000$.

overall structure of network representations without imposing constraints on individual coordinates.

To analyze how the latent space evolves with the introduction of new tasks, we examine the Latent Geometry Graph (LGG) $\mathcal{G}$ after training on task $\mathcal{T}_i$ ($i \in \{1, ..., T-1\}$). The graph $\mathcal{G}$ is constructed by taking all replay examples of the memory buffer B with fixed size m, passing them through the model to extract pre-classifier features $\{z \triangleq f_{\theta^{pre-class}}(x) \mid x \in B\}$, and building a k-NN graph based on these features [63]. A measure of latent space sparsity with respect to class representations is provided by the Label-Signal Variation σ [63] on the adjacency matrix $A \in \mathbb{R}^{m \times m}$ of $\mathcal{G}$:

$$\sigma \triangleq \sum_{i=1}^{m} \sum_{j=1}^{m} \mathbb{1}_{y_i^b \neq y_j^b} a_{i,j}, \tag{3.6}$$

where $\mathbb{1}_{\cdot}$ is the indicator function, and $a_{i,j}$ represents the $(i,j)^{\text{th}}$ element of A, equal to 1 if the i^{th} element of B is in the k-NN set of the j^{th} element and 0 otherwise.

We evaluate this metric across three state-of-the-art (SOTA) RBMs and present the results in Fig. 3.1a. The results indicate a steadily increasing σ, suggesting that examples from different classes become more entangled in later tasks. This trend is also evident from the t-SNE embedding of points in B (Fig. 3.1b for X-DER with RPC on future heads (X-DER $^{\text{RPC}}_{\text{future}}$)), which shows decreasing distances between examples from different classes over time. To counteract these issues, we propose a novel loss term that promotes cohesive latent space structures for RBMs. Our

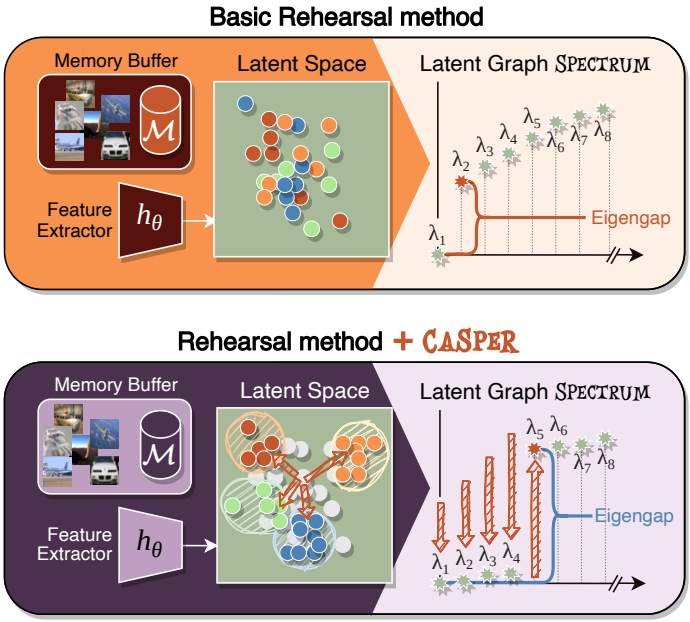


Fig. 3.2: An overview of the proposed CaSpeR-IL regulariser. RBMs struggle to separate the latent-space projections of replay data-points. Our proposal targets the spectrum of the latent geometry graph to induce a partitioning behavior by maximizing the *eigengap* for the number of seen classes.

approach, termed **Continual Spectral Regulariser for Incremental Learning (CaSpeR-IL)** (illustrated in Fig. 3.2), leverages graph-spectral theory (Sec. 2.1) to encourage the creation of well-separated latent embeddings. As demonstrated in Fig. 3.1, CaSpeR-IL can be effectively integrated with existing RBMs, improving classification accuracy and resilience against catastrophic forgetting.

3.3 Continual Spectral Regulariser for Incremental Learning

Our method builds upon the intuition that the latent spaces of DNNs bear a structure informative of the data space they are trained on [102]. By applying a geometric regularisation term, we seek to enforce a desirable structure for latent representations, *i.e.*, partitioning the vertices of $\mathcal{G}$ into well-separated subgraphs with high internal connectivity.

Alg. 3.1: CaSpeR-IL Loss Computation

Input: Memory buffer B of saved samples, with size m; batch-size b; Hyperparameter of number of classes to sample p.

$x^b \leftarrow \text{BalancedSampling}(B, p)$ ▷ Sampling from B

$z^b \leftarrow f_{\theta^{pre-class}}(x^b)$ ▷ Get features

$\mathcal{D} = \{D_{i,j} = \|z_i^b - z_j^b\| \text{ with } i, j = 1, ..., m\}$ ▷ Distance matrix

$A \leftarrow \text{k-NN}(\mathcal{D})$ ▷ Adjacency matrix

$D \leftarrow \text{diag}(\sum_i^m a_{1,i}, \sum_i^m a_{2,i}, ..., \sum_i^m a_{b,i})$ ▷ Degree matrix

$L \leftarrow I - D^{-1/2} A D^{-1/2}$ ▷ Normalized Laplacian

$\lambda \leftarrow \text{Eigenvalues}(L)$ ▷ Retrieve eigenvalues

$\mathcal{L}_{\text{CaSpeR}} \leftarrow -\lambda_{p+1} + \sum_{j=1}^{p} \lambda_j$ ▷ Eq. 3.7

Output: $\mathcal{L}_{\text{CaSpeR}}$

The loss computation is fully described in Alg. 3.1[1]. We take the examples in B and forward them through the network; their features are used to build a k-NN graph $\mathcal{G}$; following [63], we refer to it as the *latent geometry graph* (LGG). Then, we calculate the degree matrix D of $\mathcal{G}$ and we compute its normalized Laplacian as $L = I - D^{-1/2} A D^{-1/2}$, where A the adjacency matrix of $\mathcal{G}$ and I is the identity matrix. Finally, we compute the eigenvalues λ of L – that satisfy Eq. 2.2 – and sort them in ascending order.

Let g be the number of different classes within the buffer, we calculate our regularizing loss as:

$$\mathcal{L}_{\text{CaSpeR}} \triangleq -\lambda_{g+1} + \sum_{j=1}^{g} \lambda_j \,. \tag{3.7}$$

The proposed loss term is weighted through the hyperparameter ρ and added to the stream classification loss and the regularization term, specific to each CL model (Eq. 3.2). Overall, our method optimizes the following objective:

$$\underset{\theta}{\text{argmin}}\, \mathcal{L}(f_\theta(x), y) + \mathcal{L}_{\text{R}} + \rho\, \mathcal{L}_{\text{CaSpeR}} \,. \tag{3.8}$$

Through Eq. 3.7, we increase the eigengap $\lambda_{g+1} - \lambda_g$ while minimizing the first g eigenvalues. Since the number of eigenvalues close to zero corresponds to the number of loosely connected partitions within the graph [66], our loss indirectly encourages the points in the buffer to be clustered without strict supervision.

A wealth of results in spectral graph theory, originating with works such as [23, 104, 103], establishes the gap between neighboring Laplacian eigenvalues as a quantitative indicator of graph partitioning quality.

[1] Since our proposal relies on the availability in B of a minimum number of samples for each class, the *BalancedSampling* function extract b examples from the buffer belonging to p different classes uniformly.

Building on these insights, our approach shifts from the *forward* problem—determining the optimal partitioning for a given graph—to the *inverse* problem of constructing a graph that exhibits the desired partitioning properties (*i.e.*, the *isospectralization* techniques outlined in Sec. 2.1).

Motivated by the principle that the number of eigenvalues close to zero reflects the number of loosely connected partitions in the graph [66], our loss function promotes the clustering of points within the buffer. Notably, this clustering is achieved indirectly, without relying on explicit supervision.

Efficient Batch Operation

The application of CaSpeR-IL involves the computationally expensive step of constructing the complete LGG $\mathcal{G}$ at each forward pass by processing all replay examples stored in the memory buffer B. Since B is typically orders of magnitude larger than a batch of input examples, this step becomes infeasible in practice.

To address this challenge, we propose an efficient approximation to the original objective by operating on a sampled sub-graph $\mathcal{G}_\text{p} \subset \mathcal{G}$ instead of the full graph $\mathcal{G}$. This sub-graph is randomly selected to include p out of the g classes represented in the memory buffer. Further computational efficiency is achieved by extracting a smaller sub-sampled graph $\mathcal{G}_\text{p}^t \subset \mathcal{G}_\text{p}$, which includes t exemplars per class.

By performing this random sampling at each forward pass, we optimize a Monte Carlo approximation of the original objective in Eq. 3.7:

$$\mathcal{L}^*_\text{CaSpeR} \triangleq \mathop{\mathbb{E}}_{\mathcal{G}_\text{p} \subset \mathcal{G}} \left[\mathop{\mathbb{E}}_{\mathcal{G}_\text{p}^t \subset \mathcal{G}_\text{p}} \left[-\lambda_{p+1}^{\mathcal{G}_\text{p}^t} + \sum_{j=1}^{p} \lambda_j^{\mathcal{G}_\text{p}^t} \right] \right], \tag{3.9}$$

where $\lambda^{\mathcal{G}_\text{p}^t}$ are the eigenvalues of the Laplacian of $\mathcal{G}_\text{p}^t$. This formulation enforces the eigengap at p, as each $\mathcal{G}_\text{p}^t$ is constructed to include samples from exactly p communities within $\mathcal{G}$. This approximation significantly reduces computational costs while retaining the effectiveness of CaSpeR-IL.

3.4 Experiments

We present a detailed breakdown of Class-IL results in Tab. 3.1, evaluating the performance of CaSpeR-IL applied to five SOTA RBMs: ER-ACE, iCaRL, DER++, X-DER $^{\text{RPC}}_{\text{future}}$, and PODNet. The evaluation is conducted on S-CIF10, S-CIF100, and S-*mini*Img, employing standard metrics FAA and FAAF. For fairness, consistent hyperparameters are used across all methods: a batch size of 64 examples, with the same number sampled from the memory

Tab. 3.1: FAA (FAAF) on S-CIF100 for RBMs with and w/o CaSpeR-IL.

Class-IL	**S-CIF10**		**S-CIF100**		**S-*mini*Img**	
JT (UB)	87.08 (–)		63.11 (–)		52.76 (–)	
FT (LB)	19.53 (100.00)		8.38 (100.00)		3.87 (100.00)	
Buffer Size	500	1000	500	2000	2000	5000
ER-ACE	66.13 (21.76)	71.72 (14.88)	34.99 (51.41)	46.52 (34.60)	22.03 (49.04)	27.26 (29.99)
+ **CaSpeR-IL**	69.58 (20.56)	73.82 (14.11)	36.70 (46.61)	47.85 (33.86)	23.36 (47.90)	29.15 (28.36)
iCaRL	52.71 (22.69)	62.94 (21.64)	39.56 (32.73)	40.47 (31.24)	19.42 (36.89)	20.17 (33.23)
+ **CaSpeR-IL**	55.66 (20.56)	63.99 (21.05)	40.87 (32.31)	41.83 (25.55)	20.46 (35.90)	21.45 (32.26)
DER++	67.38 (26.77)	71.17 (25.12)	28.01 (57.56)	43.27 (34.94)	20.88 (74.48)	28.55 (61.03)
+ **CaSpeR-IL**	69.11 (26.18)	73.12 (23.43)	32.16 (53.41)	46.95 (30.08)	22.61 (71.01)	29.96 (57.60)
X-DER $^{\text{RPC}}_{\text{future}}$	63.23 (14.99)	65.72 (12.28)	35.89 (44.54)	46.37 (23.57)	24.80 (44.69)	30.98 (30.12)
+ **CaSpeR-IL**	65.56 (14.41)	67.84 (10.65)	38.23 (43.90)	48.11 (18.47)	26.24 (41.72)	31.63 (28.71)
PODNet	37.22 (40.49)	45.97 (39.49)	30.16 (54.49)	32.12 (46.73)	16.82 (52.32)	20.81 (46.50)
+ **CaSpeR-IL**	39.85 (39.51)	47.40 (38.90)	32.27 (48.32)	38.64 (35.65)	18.09 (50.33)	23.63 (45.08)

buffer. For S-CIF10 and S-CIF100, models are trained for 20 epochs per task without a learning rate scheduler, while for S-*mini*Img, training spans 50 epochs per task with a learning rate decay of 0.1 applied at epochs 35 and 45.

At a high level, CaSpeR-IL consistently improves FAA across all evaluated methods and datasets. However, a deeper analysis reveals several noteworthy trends.

First, the improvement in accuracy does not scale with the size of the memory buffer, contrasting with the typical behavior of replay regularization methods [17, 22]. This can likely be attributed to the geometric nature of our approach: spectral properties of graphs are known to be robust to coarsening [55], enabling CaSpeR-IL to remain effective without relying on large amounts of data.

Interestingly, most methods show comparable FAA gains in CL settings on S-CIF100, indicating that CaSpeR-IL promotes balanced responses for both stream and replay classes. This is further supported by a significant reduction in FAAF, demonstrating CaSpeR-IL's ability to mitigate learning biases [10].

On S-*mini*Img, while CaSpeR-IL still outperforms the baselines, the improvement in FAA is less pronounced. The mixed FAAF results in Class-IL suggest that our approach may be less effective when comparing classes learned across different tasks. This limitation might stem from our batch-based approximation, which samples only a subset of classes at each training step, potentially struggling with datasets featuring a

Tab. 3.2: Comparison with contrastive baselines. We report FAA and the average variance of same-class projections on the latent space.

Class-IL	**S-CIF100**			
Buffer Size	500		2000	
	FAA	Variance	FAA	Variance
SCR	31.18	2.2111	43.39	4.4439
ER-ACE	34.99	0.5313	46.52	0.5769
+ **CaSpeR-IL**	**36.70**	0.4926	**47.85**	0.5478
+ **CSCCT**	34.93	**0.3931**	45.91	**0.4290**
iCaRL	39.56	0.8381	40.47	0.8248
+ **CaSpeR-IL**	**40.57**	**0.8289**	**41.83**	**0.8057**
+ **CSCCT**	39.36	0.9167	40.87	1.0392
DER++	28.01	0.1283	43.27	0.1209
+ **CaSpeR-IL**	**32.16**	0.0964	**46.95**	0.1012
+ **CSCCT**	30.17	**0.0552**	44.27	**0.0857**
X-DER $^{\text{RPC}}_{\text{future}}$	35.89	0.2265	46.37	0.2523
+ **CaSpeR-IL**	**38.23**	0.2065	**48.11**	**0.2207**
+ **CSCCT**	36.23	**0.1974**	45.51	0.2242
PODNet	30.16	0.4229	32.12	0.7366
+ **CaSpeR-IL**	**32.27**	0.4197	**38.64**	0.5700
+ **CSCCT**	30.78	**0.1809**	33.59	**0.2577**

larger number of tasks.

Finally, PODNet stands out as an outlier, exhibiting lower FAA and higher FAAF compared to the other methods, indicating a tendency to overfit the current training data. Nonetheless, CaSpeR-IL positively impacts its training, providing a stabilizing effect that is particularly noticeable with larger memory buffers. This highlights the additional regularization benefits of CaSpeR-IL, facilitating model convergence. These observations align with our findings in Sec. 3.5.4, where we explore the application of CaSpeR-IL in settings with limited supervision.

Comparison with Contrastive Learning. The comprehensive study in [47] interprets contrastive learning as a parametric form of spectral clustering on the input augmentation graph, establishing a conceptual link to our approach. Given the alignment between the objectives of CaSpeR-IL and contrastive learning, we compare CaSpeR-IL with two existing contrastive continual learning baselines: SCR [57] and CSCCT [7].

We evaluate these methods on the S-CIF100 benchmark. SCR is a standalone model that extends Experience Replay, whereas CSCCT is a module designed to integrate with existing CL methods. As a direct competitor, CSCCT is implemented on the same baselines as CaSpeR-

Tab. 3.3: Class-IL FAA results (S-CIF100, $m = 2000$) of k-NN classifiers trained on top of the latent representations of B data.

k-NN Clsf (Class-IL)	w/o CaSpeR-IL 5-NN	w/o CaSpeR-IL 11-NN	w/ CaSpeR-IL 5-NN	w/ CaSpeR-IL 11-NN
ER-ACE	43.73	44.41	46.75+3.02	47.29+2.88
iCaRL	34.86	37.78	36.00+1.14	38.33+0.55
DER++	44.21	44.24	45.75+1.54	46.00+1.76
X-DER $^{\text{RPC}}_{\text{future}}$	43.44	44.62	49.47+6.03	49.49+4.87
PODNet	21.11	22.60	27.88+6.77	28.94+6.34

IL. While CSCCT exhibits similarities to CaSpeR-IL, it requires access to a snapshot of the past model during training and incorporates both streaming and memory data in its loss formulation. Results of this comparison are reported in Tab. 3.2. Additionally, to investigate the impact of different CL approaches on the latent space, we measure the average intra-class variance of latent projections at the end of training.

Our observations reveal several key insights. First, SCR demonstrates higher latent-space variance compared to other baselines. In contrast, both CSCCT and CaSpeR-IL effectively reduce the variance within the latent space. Interestingly, while CSCCT often achieves the lowest variance, CaSpeR-IL consistently outperforms it in terms of accuracy. These results suggest two important conclusions: *i)* intra-class variance in the latent space does not directly correlate with classification accuracy; *ii)* directly constraining individual latent coordinates, as done by CSCCT, may restrict the model's ability to rearrange data points, while spectral geometry, as employed by CaSpeR-IL, offers a more flexible clustering mechanism. This softer approach enables the latent space to organize itself into structures better suited for classification tasks, balancing flexibility and structure to enhance performance.

3.5 Analysis

In this section, we briefly present some additional experiments aimed at showing the geometric properties conferred to the model by CaSpeR-IL.

3.5.1 k-NN classification

Our first objective is to verify whether CaSpeR-IL effectively separates the latent embeddings of examples from different classes. While Fig. 3.1 already supports this assumption by analyzing the Label-Signal Variation

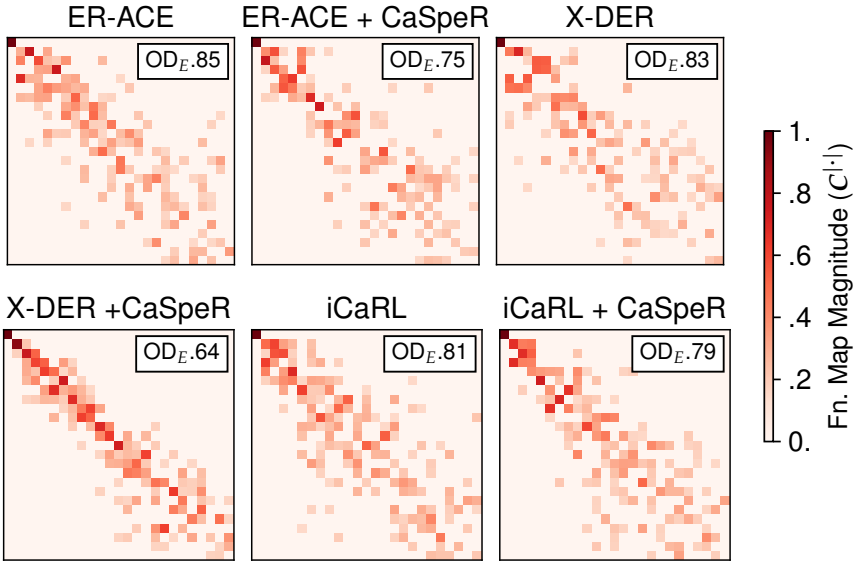


Fig. 3.3: For several RBMs with and without CaSpeR-IL, the functional map magnitude matrices $C^{|\cdot|}$ between the LGGs $\mathcal{G}_4$ and $\mathcal{G}_9$, computed on the test set of $\mathcal{T}_0, ..., \mathcal{T}_4$ after training up to $\mathcal{T}_4$ and $\mathcal{T}_9$ respectively (S-CIF100 - $m = 2000$). The closer $C^{|\cdot|}$ to the diagonal, the less geometric distortion between $\mathcal{G}_4$ and $\mathcal{G}_9$. We report the first 25 rows and columns of $C^{|\cdot|}$, focusing on smooth (low-frequency) matches [84] and apply a $C^{|\cdot|} > 0.15$ threshold to increase clarity.

(σ) on $\mathcal{G}$, we further investigate this aspect by training k-NN classifiers [122] on the latent representations z produced by the methods outlined in Sec. 3.4 and evaluating their FAA.

Tab. 3.3 reports the results for 5-NN and 11-NN classifiers, which utilize the latent-space projections of the final buffer B as their support set. The steady improvements observed with CaSpeR-IL in prior experiments extend to this classification setting as well. These results confirm that CaSpeR-IL is effective in disentangling the latent representations of different classes, further validating the efficacy of our proposed approach.

3.5.2 Latent Space Consistency

To further investigate the dynamics of the latent space across the evaluated models, we examine the emergence of distortions in the LGG. Specifically, for a given RBM, we compare $\mathcal{G}_4$ and $\mathcal{G}_9$—the LGGs constructed after training on $\mathcal{T}_4$ and $\mathcal{T}_9$, respectively—computed using the test set of tasks $\mathcal{T}_0, \ldots, \mathcal{T}_4$.

The comparison between $\mathcal{G}_4$ and $\mathcal{G}_9$ is formalized through the node-to-node bijection $\mathfrak{F} : \mathcal{G}_4 \to \mathcal{G}_9$, represented as a functional map matrix C [84] with elements:

$$c_{i,j} \triangleq \langle \phi_i^{\mathcal{G}_4}, \phi_j^{\mathcal{G}_9} \circ \mathfrak{F} \rangle, \tag{3.10}$$

where $\phi_i^{\mathcal{G}_4}$ is the i^{th} Laplacian eigenvector of $\mathcal{G}_4$ (similarly for $\mathcal{G}_9$), and $\circ$ denotes function composition. The matrix C encodes the similarity between the Laplacian eigenspaces of the two graphs. Ideally, if the latent space remains unchanged between $\mathcal{T}_4$ and $\mathcal{T}_9$ for previously learned classes, $\mathfrak{F}$ would be an *isomorphism*, and C would be diagonal [84]. In practice, $\mathfrak{F}$ is only approximately isomorphic, with better approximations resulting in C becoming sparse and funnel-shaped.

Fig. 3.3 shows the absolute functional map matrices $C^{|\cdot|} \triangleq \text{abs}(C)$ for ER-ACE, DER++, iCaRL, and X-DER $^{\text{RPC}}_{\text{future}}$ on S-CIF100, both with and without CaSpeR-IL. Notably, methods that gain the most from CaSpeR-IL (ER-ACE, X-DER $^{\text{RPC}}_{\text{future}}$) exhibit tighter functional map matrices, indicating reduced interference. This suggests that the partitioning behavior enforced by CaSpeR-IL helps preserve the geometric consistency of the LGG for previously learned classes during later tasks. In contrast, iCaRL shows only marginal improvements, likely due to its less discriminative training regime, which induces limited changes in the latent space structure.

To quantify the similarity of each $C^{|\cdot|}$ matrix to the identity matrix, we compute its off-diagonal energy [96]:

$$\text{OD}_E \triangleq \frac{1}{||C||_{\text{F}}^2} \sum_i \sum_{j \neq i} c_{i,j}^2, \tag{3.11}$$

where $||\cdot||_{\text{F}}$ denotes the Frobenius norm. The results show that CaSpeR-IL significantly reduces OD_E, indicating an increase in the diagonality of the functional matrices and further validating the effectiveness of CaSpeR-IL in maintaining geometric coherence in the latent space.

3.5.3 Different Incremental Benchmarks

We further evaluate the effectiveness of CaSpeR-IL in two additional *academic* CL scenarios: Task-IL and Domain-IL. For Task-IL, we assess the performance of the various SOTA methods introduced in Sec. 3.4, both with and without CaSpeR-IL, on the S-CIF100 and S-*mini*Img datasets. For Domain-IL, we perform a similar evaluation using the R-MNIST dataset [70].

The results in Tab. 3.4 indicate that CaSpeR-IL enhances the model's ability to learn and consolidate tasks independently in the Task-IL scenario. Similarly, the findings in Tab. 3.5 demonstrate that CaSpeR-IL provides measurable performance improvements even in the Domain-IL setting, highlighting its broad applicability.

Tab. 3.4: Task-IL results – FAA (FAAF) – for SOTA rehearsal CL methods, with and without CaSpeR-IL.

Task-IL	**S-CIF100**		**S-*mini*Img**	
Joint (UB)	88.81 (–)		87.39 (–)	
Finetune (LB)	30.10 (62.84)		24.05 (67.37)	
Buffer Size	500	2000	2000	5000
ER-ACE	73.86 (10.73)	80.69 (5.37)	69.05 (13.72)	72.78 (8.93)
+ **CaSpeR-IL**	75.14 (4.91)	81.57 (4.93)	69.59 (13.05)	74.14 (8.12)
iCaRL	78.38 (5.38)	78.47 (4.91)	70.35 (3.92)	70.44 (2.68)
+ **CaSpeR-IL**	79.31 (4.61)	79.43 (3.41)	71.19 (3.67)	71.93 (3.65)
DER++	70.55 (11.12)	78.60 (5.96)	69.78 (13.37)	73.81 (8.59)
+ **CaSpeR-IL**	73.25 (9.49)	80.78 (3.04)	70.97 (11.75)	75.18 (7.93)
X-DER $^{\text{RPC}}_{\text{future}}$	77.28 (2.43)	82.55 (0.92)	74.32 (4.95)	77.70 (3.71)
+ **CaSpeR-IL**	78.26 (5.47)	83.77 (0.27)	75.99 (3.88)	78.71 (2.32)
PODNet	67.37 (19.76)	69.63 (15.16)	60.60 (14.00)	66.15 (10.71)
+ **CaSpeR-IL**	70.81 (15.26)	71.90 (11.32)	64.84 (10.01)	70.85 (7.99)

Tab. 3.5: FAA values on Rotated MNIST.

Domain-IL	R-MNIST	
Joint (UB)	95.76	
Finetune (LB)	67.66	
Buffer Size	200	500
ER-ACE	83.45	86.86
+ **CaSpeR-IL**	86.19	88.11
DER++	89.91	92.00
+ **CaSpeR-IL**	90.96	93.21

Tab. 3.6: Class-IL FAA values on S-CIF100, with reduced amount of annotations (CSSL). Buffer size 2000.

	CSSL	
Labels %	0.8%	5%
ER-ACE	8.46	11.87
+ **CaSpeR-IL**	8.55	14.16
PsER-ACE	2.31	16.35
+ **CaSpeR-IL**	9.69	17.42

3.5.4 Continual Semi-supervised Learning

In a supervised CL setting, we integrate CaSpeR-IL into the processing of buffer data points, promoting the separation of all previously encountered classes in the latent space. Notably, our approach does not impose strict supervision requirements; instead of relying on labels for each node in the LGG, it only requires knowledge of the total number of classes g to be clustered (Eq. 3.7). Assuming the classes are equally distributed in the data, *balanced sampling* can be approximated by random sampling from the memory buffer. Consequently, CaSpeR-IL does not depend on individual example annotations, making it well-suited for semi-supervised scenarios where it can enhance accuracy and facilitate convergence.

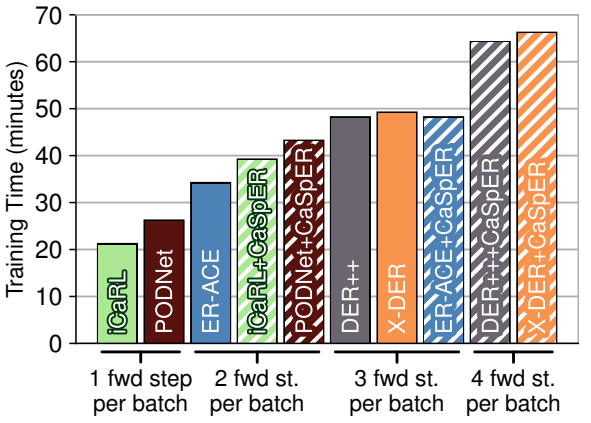


Fig. 3.4: Wallclock training time for the approaches evaluated in Sec. 3.4 benchmarked on an identical hardware setup (GPU NVIDIA V100). Training time grows linearly w.r.t. the number of per-batch forward steps.

[11] introduce Continual Semi-Supervised Learning (CSSL), a novel CL experimental benchmark where only a fraction of examples in the input stream are annotated. In Tab. 3.6, we present results for S-CIF100 under the CSSL setting, with only 0.8% or 5% of examples annotated. Traditional CL methods in this scenario, such as ER-ACE, often discard large amounts of unlabeled data, leading to significantly reduced performance compared to the fully supervised case. Alternatively, some methods employ *pseudo-labeling* (e.g., PsER-ACE) to annotate unlabeled examples using the in-training model. However, pseudo-labeling can backfire if the limited annotations are insufficient for the model to produce reliable predictions, as observed in the 0.8% labeled scenario.

To leverage the unlabeled examples, we extend CaSpeR-IL to data points from the input stream, setting k (the number of nearest neighbors for the adjacency matrix) equal to the number of classes in a given task. This approach leads to overall performance improvements across tested models and, critically, mitigates the failure cases observed when PseudoER-ACE is applied with sparse annotations. These results indicate that CaSpeR-IL effectively reduces the impact of noisy labels generated by *pseudo-labeling*, further demonstrating its utility in semi-supervised CL settings.

3.5.5 An analysis of training time

As outlined in Sec. 3.3, CaSpeR-IL involves non-negligible computations that might seem to have a significant impact on the overall training

time. In this section, we shed light on the matter of the efficiency of our approximated procedure by comparing the wall-clock time between models with and without CaSpeR-IL on identical run conditions.

In Fig. 3.4, we report the overall training time for the approaches evaluated in Sec. 3.4, specifically on S-CIF100, benchmarked on an identical hardware setup. We observe that the overhead of CaSpeR-IL is fundamentally dominated by the time it takes to perform an additional forward step of the corresponding batch through the model. For X-DER $^{\text{RPC}}_{\text{future}}$, which already performs 3 forward steps, we register a 33% increase in compute time; for ER-ACE, which performs 2, we get a 43% increase, and so on. We believe that such an overhead is comparable to the ones of other CL solutions (for instance, meaning that ER-ACE+CaSpeR-IL ends up having roughly the same compute time as X-DER $^{\text{RPC}}_{\text{future}}$ and DER++).

3.6 Conclusions

This chapter introduced CaSpeR-IL, a novel regularization method for RBMs that leverages principles from *spectral geometry*, a geometric and data discretization technique, to promote clustering behavior in the latent space of the memory buffer B. By harnessing spectral properties of the graph representation of latent embeddings, CaSpeR-IL facilitates a quantifiable disentanglement of latent projections associated with distinct classes, enabling improved interpretability and organization of the model's latent space.

A key advantage of CaSpeR-IL lies in its minimal reliance on supervision, as it does not require explicit annotations for individual examples but only knowledge of the total number of classes to cluster. This makes it particularly effective in scenarios with limited annotations, such as those explored in [11]. While previous methods aimed at coherent class representation in reduced-annotation CL scenarios rely on heuristic approaches, CaSpeR-IL introduces a principled geometric formulation to its learning objective. Furthermore, additional experiments presented in this chapter demonstrate that CaSpeR-IL not only improves accuracy but also enhances convergence in low-supervision settings.

The application of geometric constraints, such as those introduced by CaSpeR-IL, represents a promising research direction in the field of continual learning. Our findings suggest that latent-space entanglement is particularly pronounced in unsupervised continual learning scenarios [72, 41], where the lack of supervision results in weaker training signals. These settings provide a natural testbed for exploring spectral and broader geometric regularization techniques, paving the way for the development of models with enhanced robustness and structured latent representations.

Chapter 4

Trajectory Forecasting through Low-Rank Adaptation of Discrete Latent Codes

4.1 Background and related work

Trajectory forecasting is a critical task with diverse applications, including video surveillance [67], multi-object tracking [75, 30], behavioral analysis [97], and intrusion detection [109]. At its core, the objective is to predict the future paths of agents based on a limited number of observations of their current motions. This task becomes particularly challenging due to the need to account for various factors, such as interactions between pedestrians [53, 108, 81] and visual attributes of the environment in which they move [29].

Traditional approaches to trajectory prediction primarily focus on the past movements of the target agent [8], ignoring the influence of external factors. However, agent trajectories are often affected by the motions of others, whether to avoid collisions or perform coordinated actions. Early solutions incorporated social behavior modeling through hand-crafted rules, energy-based features, or discrete choice models [5, 86]. With the rise of machine learning, data-driven methods have become the standard, leveraging deep models to infer social interactions [3, 46]. Notably, techniques based on attention mechanisms have excelled in capturing complex interactions between agents. For example, [81] introduces a social-temporal attention module to capture temporal dynamics and interpersonal interactions simultaneously.

Formally, let the future trajectory be denoted as $y \in \mathbb{R}^{T \times d}$, where T represents the number of future time steps and d the dimensionality of

the input (e.g., $d = 2$ for pedestrian trajectories in a bird's-eye view). The predicted trajectory $\hat{y}$ is generated by a learnable model usually conditioned on two inputs: *i)* the observed trajectory $x \in \mathbb{R}^{T_p \times d}$, representing the agent's past coordinates over T_p time steps, and *ii)* the neighboring trajectories $\mathcal{X} = x_1, x_2, \ldots, x_N$, which include all other agents within the scene, without imposing a distance threshold. Usually, the model is trained to minimize the discrepancy between the predicted trajectory and the ground-truth future path, as measured by a suitable loss function $\mathcal{L} := \mathcal{L}(\hat{y}, y)$.

4.1.1 Trajectory Generation: a Probabilistic Perspective

Given the inherent uncertainty and multimodal nature of future motion, simplistic predictive approaches are often insufficient to capture the complexity and variability of possible outcomes. Modern *probabilistic* approaches address this by modeling the *distribution* of potential outcomes. For instance, Social-GAN (S-GAN) [46] uses a conditional Generative Adversarial Network (GAN) [42] to produce socially plausible trajectories, and SoPhie [98] enhances GANs with social and visual interaction modules. Similarly, Variational Autoencoders (VAEs) [58] and Conditional Variational Autoencoders (C-VAEs) [106] have been applied for multimodal trajectory prediction, as seen in works like [127, 54, 111, 99, 128]. For example, Trajectron++ [99] combines VAEs with graph-structured recurrent neural networks to model agent interactions, while PECNet [76] integrates VAEs with goal conditioning for endpoint-driven forecasting.

GAN and VAE-based approaches often face challenges such as mode collapse, necessitating specialized loss functions to encourage diverse predictions [46]. For instance, Gupta et al. [46] introduced the variety loss, defined as $\mathcal{L}_{\text{variety}} = \min_k |y - \hat{y}^{(k)}|_2$, which aims to promote the generation of K diverse trajectory predictions by minimizing the error between the ground truth and the closest prediction among multiple samples. However, while this loss function enhances prediction diversity, it inherently distorts the representation of the data's true Probability Density Function (PDF). To address this limitation, Thiede and Brahma [113], for example, proposed corrective measures to adjust the learned distribution, improving its approximation of the true PDF.

A breakthrough in trajectory prediction comes from adopting denoising diffusion models [50], as introduced by [45]. These models iteratively refine predictions by progressively denoising potential trajectories, offering a novel way to handle uncertainty and multimodality in forecasting tasks.

4.1.2 Trajectory Forecasting Benchmarks

To evaluate the performance of trajectory prediction models, three established datasets are utilized in this chapter: the Stanford Drone Dataset

(SDD) [94], the NBA SportVU Dataset (NBA) [68], and the NFL Football Dataset (NFL) [1]. These datasets are widely acknowledged in the trajectory prediction literature and encompass different scenarios, such as pedestrian movement, basketball games, and American football plays. A detailed description of each dataset is provided below:

- **Stanford Drone Dataset (SDD)** [94] gathers trajectories of pedestrians within the Stanford University campus in a bird's eye view. Given 8 time steps ($\approx$ 3.2 seconds), methods must predict the following 12 frames (4.8 seconds). In this thesis, the established train-test split [76] is used;

- **NBA SportVU Dataset (NBA)** collected by the NBA's SportVU automatic tracking system, this dataset [68] provides the trajectories of 10 players and the ball in real basketball games. Given 10 previous time-steps ($\approx$ 2.0 seconds), the models predict the subsequent 20 steps (4.0 seconds);

- **NFL Football Dataset (NFL)** [1] records the movements of every player throughout each play of the 2017 season. The goal is to predict the trajectories of the 22 players (11 per team) and the ball for the ensuing 3.2 seconds (16 steps), given the preceding 1.6 seconds (8 steps).

4.1.3 Evaluation Metrics

The evaluation of trajectory prediction models is based on two widely and commonly used metrics [86, 3]:

- **Average Displacement Error (ADE)**: ADE computes the average Euclidean distance between the predicted trajectory and the ground-truth trajectory across all time steps. It is defined as:

$$\text{ADE} = \frac{1}{T}\sum_{t=1}^{T} \|\hat{y}_t - y_t\|, \tag{4.1}$$

  where T is the total number of time steps, $\hat{y}_t$ is the predicted position at time t, and y_t is the ground-truth position at time t;

- **Final Displacement Error (FDE)**: FDE measures the Euclidean distance between the predicted and ground-truth positions at the final time step. It is given by:

$$\text{FDE} = \|\hat{y}_T - y_T\|, \tag{4.2}$$

  where T is the final time step;

- **Best-of-K Protocol**: To evaluate stochastic models, the best-of-K protocol [54, 76] is employed. Under this protocol, $K = 20$ [124, 123] trajectories are generated for each scenario, and the trajectory with the smallest error is selected for evaluation. The corresponding metrics are denoted as:

$$\text{ADE}_K = \frac{1}{T} \sum_{t=1}^{T} \min_{k=1,\dots,K} \|\hat{y}_{t,k} - y_t\|, \tag{4.3}$$

$$\text{FDE}_K = \min_{k=1,\dots,K} \|\hat{y}_{T,k} - y_T\|, \tag{4.4}$$

  where $\hat{y}_{t,k}$ represents the predicted position at time t for the k-th trajectory.

For sports datasets, these metrics are further evaluated at varying time intervals to provide a more granular performance assessment.

4.1.4 State of the Art

The list below outlines the most recent and significant approaches to trajectory generation, which will be used as benchmarks for the experiments detailed in the upcoming chapters. The methods are categorized based on the underlying model architecture and the key components they leverage to improve prediction accuracy.

- **Social-GAN (S-GAN)** [46] utilizes a Generative Adversarial Network (GAN) framework to predict socially acceptable trajectories for multiple agents. It incorporates a novel pooling mechanism that aggregates information across individuals to model social interactions, enabling the generation of diverse and realistic human motion behaviors;

- **Trajectron++** [99] employs a graph-structured recurrent neural network combined with variational autoencoders (VAEs) to forecast trajectories. It integrates dynamic and heterogeneous data, ensuring that the predicted paths are dynamically feasible and account for the interactions between different types of agents;

- **Predicted Endpoint Conditioned Network (PECNet)** [76] enhances a VAE framework with goal-oriented reasoning by first predicting potential endpoints of trajectories and then generating paths conditioned on these endpoints. This approach allows for accurate long-term trajectory predictions by focusing on likely destinations;

- **Latent Belief Energy-Based Model (LB-EBM)** [85] focuses on predicting long-range trajectories by introducing a belief vector that

encapsulates the energy distribution within the environment. This method models the underlying energy landscape to generate trajectories that are both feasible and contextually appropriate;

- **GroupNet** [123] introduces a multiscale hypergraph neural network to capture both pair-wise and group-wise interactions at various scales. By modeling complex relational reasoning, it effectively predicts trajectories in scenarios involving multiple interacting agents;

- **Memo-Net** [124] emulates the concept of retrospective memory from neuropsychology by maintaining a memory bank of past instances. It retrieves similar instances to inform intention prediction, allowing for more accurate forecasting of agent behaviors based on historical patterns;

- **Motion Indeterminacy Diffusion (MID)** [45] employs a diffusion model to incrementally minimize uncertainty in possible future trajectories. By iteratively refining its predictions, it effectively captures the stochastic nature of human motion, leading to more accurate and dependable trajectory forecasts.

4.2 Motivation

In generative modeling, adopting a variational approach often encounters a critical challenge known as *posterior collapse*: a phenomenon where the latent variables collapse to the prior, rendering them uninformative. Consequently, the decoder disregards these variables, leading to suboptimal generative capabilities (Sec. 2.2). In trajectory generation, this issue manifests as predictions concentrated on a single, often trivial trajectory, with reduced uncertainty and diversity. A similar challenge, termed *mode collapse*, has been observed in adversarial networks. Efforts to address this problem typically involve complex learning objectives aimed at promoting diversity [46, 98] or the use of multiple generator networks [29], often at the cost of increased computational and architectural complexity.

In image generation, **Vector Quantized Variational Autoencoders (VQ-VAEs) [116]** have shown promise in mitigating posterior collapse. Unlike traditional variational models, VQ-VAEs replace the hand-crafted Gaussian prior with a learnable categorical prior, creating a discrete latent space. This space is represented by a fixed-size dictionary, or **codebook**, whose entries are learnable latent codes (see Sec. 2.2). This structured latent space enhances flexibility, making VQ-VAEs particularly promising for trajectory forecasting, where modeling diverse and consistent trajectories is paramount.

The primary contribution of this chapter is to extend the capabilities of the VQ-VAE codebook. While the original formulation employs a single,

static codebook shared across all examples, we propose a **context-aware instance-based codebook**, where the codebook dynamically adapts to the specific *context* of each example. We refer as *context* to the set of historical information related to each agent, namely the past steps of its trajectory as well as its interactions with nearby agents. In this way, we aim to encourage even more flexibility during the discretization process, as distinct motion patterns can be discretized with varying granularity.

Moreover, we envision the customization of the codebook as an **adaptation** of the shared original VQ-VAE codebook. By doing so, our goal is to strike a balance between per-instance customization and the emergence of cross-instance concepts that are relevant across multiple examples. In practice, we draw inspiration from recent advances in Parameter Efficient Fine Tuning and represent the dynamic adjustments to the codebook as **low-rank updates** of its values (see Fig. 4.1). We show that such a modeling constraint improves the representation capabilities of the learned latent space, thereby encoding additional information and facilitating the reconstruction task.

The traditional subsequent stage in VQ-VAEs involves fitting the distribution on the discrete latent codes. In this respect, we make use of a vector-quantized diffusion model [44] to learn the implicit prior, departing from existing approaches [116, 38] that rely on autoregressive priors, which are more susceptible to issues related to error accumulation and unidirectional bias. We further introduce a novel sampling strategy based on **k-means clustering** to enhance the diversity and quality of the generated trajectories.

4.3 Low-rank Adaptation for VQ-VAE

We herein present our approach to pedestrian trajectory generation, which we name **Low-rank Adaptation for VQ-VAE (LRVQ)**, depicted in Fig. 4.1. Briefly, we exploit VQ-VAEs to encode the future trajectory y of a given agent. On top of that, the following main novelties are introduced:

- we extend VQ-VAE to predict a trajectory coherent with the observed historical trend. To do so, we feed **additional contextual** information to the VQ-VAE, conditioning both the prior and the posterior distributions. The contextual information consists of the past observed trajectory x, and a summary of the interactions between the agent and its neighbors. The structure of the resulting quantization model is presented in Sec. 4.3.1;

- to encourage further **flexibility**, the codebook itself is conditioned on the additional contextual information (see Sec. 4.3.2). As discussed later, the context is introduced by devising a **low-rank** adjustment to the codebook;

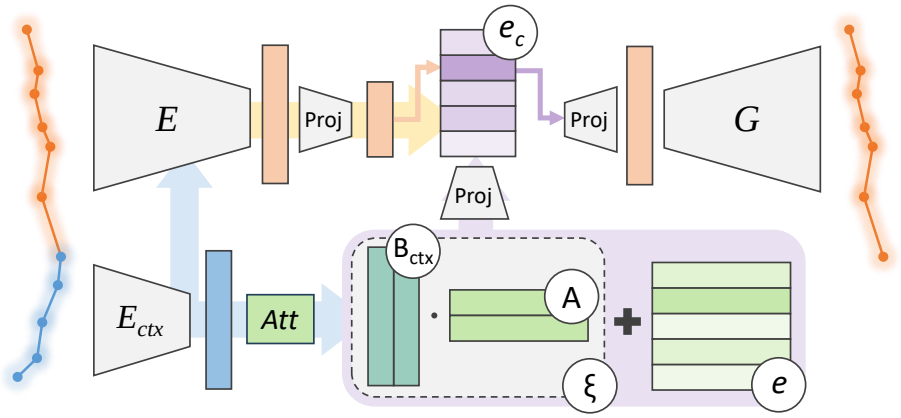


Fig. 4.1: Overview of our approach to trajectory prediction, based on Vector Quantization and Low-Rank adaptation of the codebook (highlighted in the purple box).

- to avoid the error accumulation and the *unidirectional bias* problem, typical of auto-regressive methods [44], we make use of a **discrete diffusion model** for the generation of the sequence of indices (see Sec. 4.3.3). We also introduce a **new sampling technique**, based on the k-means clustering algorithm, to produce better and more consistent generations (see Sec. 4.3.4).

4.3.1 Trajectory Forecasting with VQ-VAEs

Formally, our VQ-VAE can be summarized as:

$$h_{\text{ctx}} = \text{E}_{\text{ctx}}([x, \mathcal{X}]) \qquad \text{(context encoding)} \qquad (4.5a)$$

$$z^q = \text{E}(y, [\mathcal{Y}, h_{\text{ctx}}]) \qquad \text{(encoding)} \qquad (4.5b)$$

$$\hat{y} = \text{G}(z^q, \mathcal{Z}^q), \qquad \text{(decoding)} \qquad (4.5c)$$

where $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}^q$ represent respectively the past, the future, and the latent quantized representation of the nearby agents' trajectories (see Sec. 4.1). The modules $\text{E}_{\text{ctx}}(\cdot)$, $\text{E}(\cdot)$, $\text{G}(\cdot)$ are three neural networks, each of which exploits social-temporal transformer [81] to account for social-temporal relations.

In particular, a contextual encoder $\text{E}_{\text{ctx}}(\cdot)$ computes hidden features $h_{\text{ctx}} \in \mathbb{R}^{T_p \times D}$ that summarize both the past trend $x \in \mathbb{R}^{T_p \times 2}$ of the trajectory and spatial interactions (Eq. 4.5a). The function $\text{E}(\cdot)$ plays the role of the VQ-VAE encoder, transforming the future trajectory y into a discrete representation $z^q \in \mathbb{R}^{T \times D}$ (see Eq. 4.5b). To condition the model on historical information, the encoder is fed also with the hidden contextual

information h_{ctx}; in detail, a tailored cross-attention layer is devised to mix future and past information. Finally, in step (4.5c) we achieve the estimated future trajectory $\hat{y} \in \mathbb{R}^{T\times 2}$ through the decoder $\mathrm{G}(\cdot)$.

As well as traditional VQ-VAEs, we employ Mean Squared Error (MSE) as our reconstruction term between the ground truth and predicted trajectory.

4.3.2 Instance-based Codebook

The codebook plays a crucial role in VQ-VAEs and can cause instabilities during optimization. For instance, the uneven utilization of the vectors of the codebook is a factor that may lead to inefficiencies in representation learning. This imbalance often results in certain elements of the codebook being underutilized, while others never match with real-valued embeddings. To mitigate these issues, the authors of [126] resort to reducing the latent-space dimensionality, showing that it leads to a condensed but richer codebook. In practice, before quantization, each vector z is projected from $\mathbb{R}^D$ to a lower-dimension space $D_r \ll D$. In the following, we will refer to this strategy as **static codebook**, to distinguish it from our proposal that instead leverages dynamic cues.

Our idea is to modify the content of the codebook, such that it reflects the motion observed in the past trajectory. The intuition is that different motion styles (*e.g.*, straight *vs.* curvilinear) could prefer distinct latent codes and discretization strategies. On this basis, we exploit again the contextual features h_{ctx} to generate an **instance-based codebook** $\xi = f_\xi(\cdot, h_{\text{ctx}})$, computed through a tailored learnable module f_ξ. The latter shares the same design of the above-described encoding networks and hence builds upon social-temporal transformers [81]. Afterwards, we combine static and instance-based codebooks by means of summation, thus obtaining a **conditioned** codebook e_c:

$$e_c = \texttt{l2\_norm}(e) + \lambda_\xi \texttt{l2\_norm}(\xi) \tag{4.6}$$

where `l2_norm` indicates the row-wise l2-normalization $v/\|v\|_2$ and λ_ξ is an hyperparameter that weighs the sum. We leverage normalizing layers to ensure that the two components contribute almost equally to the final embedding table.

Moreover, the way we define the codebook draws inspiration from the successes of low-rank adaptation [52] for fine-tuning Large Language Models (LLMs). Namely, we opt for a *low-rank characterization* of f_ξ, which means that the instance-driven modifications to the static codebook lie on a lower-dimensional manifold of the parameter space. We hence define the instance-based codebook ξ as a matrix product of two low-rank matrices B_{ctx} and A, as follow:

$$\begin{aligned} B_{\text{ctx}} &= f_\xi(B, h_{\text{ctx}}) \quad \text{where } B, B_{\text{ctx}} \in \mathbb{R}^{D\times r} \\ \xi &= B_{\text{ctx}} A \quad \text{where } A \in \mathbb{R}^{r\times C}. \end{aligned} \tag{4.7}$$

Considering B as a set of learnable tokens, f_ξ adopts cross attention between the conditioning information h_{ctx} and B to create an instance-based B_{ctx}.

4.3.3 Diffusion-based Categorical Prior

As previously mentioned, the second main stage regards the training of the parametric categorical prior $p_{\theta_p}(c|x, \mathcal{X})$ (note that the p_{θ_p} is also conditioned on historical information), where $c = \{c_1, c_2, \ldots, c_T\}$. Notably, the learned prior serves to forecast the future trajectory y at inference time, when the posterior distribution of y is not available. Sec. 4.3.4 provides a detailed description of the sampling procedure, while the rest of this section describes the architectural and training aspects of the categorical prior.

We borrow the design of the categorical prior from the framework of Denoising Denoising Diffusion Probabilistic Modelss (DDPMs). In particular, we employ vector-quantized diffusion models [44], as they naturally handle discrete distributions. Notably, the application of DDPMs allows one to learn the categorical prior without the need for autoregressive modeling, as commonly employed in many existing approaches [115, 38]. In the context of trajectory prediction, we view the adoption of a non-autoregressive model as an additional strength. On the one hand, auto-regressive methods can leverage the inherent inductive bias of time-series data, where consecutive time steps relate to each other. However, this often results in error accumulation issues and in the so-called *unidirectional bias* [44], which blurs contextual information that flows in a direction not coherent with the chosen auto-regressive order. In the task under consideration, this means that auto-regressive approaches may struggle to leverage cues emerging in later moments of the trajectory, as *the goal* or the long-range intention of the agent. These crucial aspects of trajectory prediction [76] could be better addressed by the approach proposed in this work, which is **order-free** and capable of capturing multiple plausible trends.

Formally, we define q^{diff} as the diffusion process that injects incremental noise to the token sequence c for Ψ diffusion steps. Instead, p_θ^{diff} is the denoising process that gradually reduces the noise of the noised sequence. The parameters θ of the denoising module are trained with the variational lower bound [105]:

$$\mathcal{L}_{\text{vlb}} = \mathcal{L}^0 + \mathcal{L}^1 + \cdots + \mathcal{L}^{\Psi-1} + \mathcal{L}^{\Psi}, \tag{4.8a}$$

$$\mathcal{L}^{\psi} = D_{KL}(q^{diff}(c^{\psi}|c^{\psi-1}) \parallel p_\theta^{diff}(c^{\psi}|c^{\psi+1}, \widehat{\mathcal{C}}^{\psi}, x, \mathcal{X})), \tag{4.8b}$$

$$\mathcal{L}^{c^0} = -\log p_\theta^{diff}(c^0|c^{\psi}, \widehat{\mathcal{C}}^{\psi}, x, \mathcal{X}), \tag{4.8c}$$

where we use x, $\mathcal{X}$ and $\widehat{\mathcal{C}}^{\psi}$ – the token sequence of neighboring agents at diffusion step ψ – as conditioning information during denoising. (4.8c) is

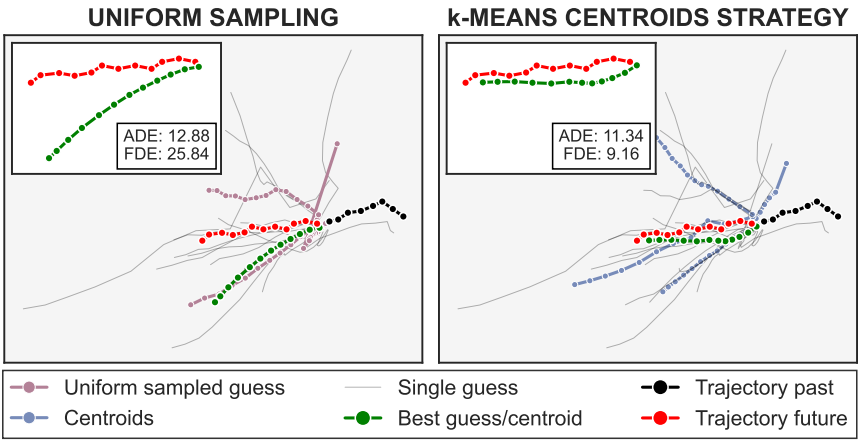


Fig. 4.2: Comparison between the $K = 5$ samples obtained from a uniform sampling strategy (on the left) and the ones given as output from the proposed k-means centroids sampling strategy (on the right), starting from the same $N = 20$ initial *guesses*.

an auxiliary objective encouraging the prediction of a noiseless token c_0. The loss function:

$$\mathcal{L} \leftarrow \begin{cases} \mathcal{L}^0, & \text{if } \psi = 1 \\ \mathcal{L}^{\psi-1} + \lambda \mathcal{L}^{c^0} & \text{otherwise.} \end{cases} \tag{4.9}$$

We refer to [44] for more exhaustive details on the diffusion steps and the prior.

Generation. At inference time, the past and social information is encoded using $\mathrm{E}_{\mathrm{ctx}}$ and then passed to the diffusion process p_θ^{diff}. The latter, after Ψ denoising steps, provides a (denoised) sequence of T indices $\hat{c} \in \mathbb{R}^T$. These indices represent the encoding of the future unobserved trajectory; therefore, we used them to select the proper elements of the codebook e_c, thus allowing us to create a quantized sequence representation z^q. Then z^q undergoes decoding through the VQ-VAE decoder G, which finally yields the generation of trajectories $\hat{y}$.

4.3.4 Enforcing Effective Multi-modal Forecasting

The sampling approach described above represents the common way to draw new samples from the learned prior of a VQ-VAE. However, we build upon it to create a stronger and richer selection strategy that furthers the multi-modal capabilities of DDPMs. The standard evaluation process involves sampling K distinct trajectories from the model and

assessing the top-performing one (as described in Sec. 4.1.3). Therefore, each methodology must find the right balance between accuracy in its prediction and potential for exploration. The proposed procedure goes in this direction: we generate numerous *raw* future paths, called *guesses*, and then condense them into the most representative ones. In formal terms, we sample N guesses and then perform the k-means clustering algorithm, with a number of clusters equal to $K < N$ (in our experiments, we set $N = 200$ and $K = 20$). We view the resulting *centroids* as the principal modes of the predictive distribution learned by the DDPM and thus use them for prediction in place of the original samples. This strategy guarantees a twofold advantage compared to naive prediction: firstly, out-of-distribution samples typically form independent clusters, thus enhancing exploration; secondly, the use of centroids reduces the quantization noise, as in-distribution samples are grouped into large clusters and averaged element-wise (see Fig. 4.2).

4.4 Experiments

We evaluate our proposed approach on three widely-used trajectory prediction benchmarks, as detailed in Sec. 4.1.2: SDD, NBA and NFL. Performance is assessed using the standard metrics described in Sec. 4.1.3, specifically ADE_K and FDE_K, computed over $K = 20$ generated samples. For comparative analysis, we benchmark against state-of-the-art methods outlined in Sec. 4.1.4.

Implementation Details. We set the number of codewords C to 16 for all datasets, while we take the best rank r for each dataset (*e.g.*, 8 for SDD and NBA, 4 for NFL). For the first stage, we use AdamW [71] as optimizer with $\text{lr} = 5 \times 10^{-4}$, $\beta_1 = 0.5$ and $\beta_2 = 0.9$. We train on SDD for 7000 epochs with batch size equal to 256. For NBA and NFL, we instead optimize for 700 epochs (the batch size equals 64). We use a cosine schedule for λ_ξ from an initial value of 0 to a final value of 1. In this way, we can introduce the instance-level codebook gradually during training.

For the second stage, we re-use the same optimizer/batch-size setup, while training for 3000 epochs for SDD, 1000 epochs for NBA, and 700 for NFL. As an augmentation technique, we rotate the trajectories by a random angle, ranging between 0 and θ_{max}. We set θ_{max} to 180° for the first stage, while we find it beneficial to adopt a lower value (5°) for the second stage.

Comparison with SOTA Methods. We report the comparison in Tab. 4.1, Tab. 4.2, and Tab. 4.3. To sum up, our LRVQ demonstrates superior performance across all the considered benchmarks.

On the SDD dataset (Tab. 4.1), we attain superior ADE results, matching closely MemoNet in FDE. While PECNet and GroupNet, among C-VAE

Tab. 4.1: SDD results (ADE_{20}/FDE_{20}). * represents the reproduced results from open source. Best results in **bold**, second-best underlined.

Time	S-GAN	Trajectron++	PECNet	MemoNet	GroupNet	MID*	**LRVQ**
4.8*s*	27.23/41.44	19.30/32.70	9.96/15.88	8.56/**12.66**	9.31/16.11	9.73/15.32	**7.86**/12.68

Tab. 4.2: NBA results (ADE_{20}/FDE_{20}). Best results in **bold**, second-best underlined.

Time	S-GAN	PECNet	Trajectron++	MemoNet	GroupNet	MID	**LRVQ**
1.0*s*	0.41/0.62	0.40/0.71	0.30/0.38	0.38/0.56	0.26/0.34	0.28/0.37	**0.19**/**0.29**
2.0*s*	0.81/1.32	0.83/1.61	0.59/0.82	0.71/1.14	0.49/0.70	0.51/0.72	**0.41**/**0.63**
3.0*s*	1.19/1.94	1.27/2.44	0.85/1.24	1.00/1.57	0.73/1.02	0.71/0.98	**0.64**/**0.96**
4.0*s*	1.59/2.41	1.69/2.95	1.15/1.57	1.25/1.47	0.96/1.30	0.96/**1.27**	**0.89**/**1.27**

Tab. 4.3: NFL results (ADE_{20}/FDE_{20}). Best results in **bold**, second-best underlined.

Time	S-GAN	PECNet	Trajectron++	LB-EBM	GroupNet	MID	**LRVQ**
1.0*s*	0.37/0.68	0.52/0.97	0.41/0.65	0.75/1.05	0.32/0.57	0.30/0.58	**0.23**/**0.35**
2.0*s*	0.83/1.53	1.19/2.47	0.93/1.65	1.26/2.28	0.73/1.39	0.71/1.31	**0.53**/**0.92**
3.2*s*	1.44/2.51	1.99/3.84	1.54/2.58	1.90/3.25	1.21/2.15	1.14/1.92	**0.98**/**1.68**

methods, demonstrate noteworthy performance compared to the older S-GAN and Trajectron++, they struggle in FDE, especially when compared to MemoNet. This could be ascribed to the effective sampling strategy of MemoNet, which integrates a tailored clustering phase to generate multiple overall intentions.

Additionally, our approach showcases robust performance across all examined partial timestamps for both the NBA (Tab. 4.2) and NFL datasets (Tab. 4.3). The two most competing methods are GroupNet – based on the C-VAE framework – and more importantly MID, which akin to our approach utilizes a diffusion process. However, we highlight an important distinction with MID, which we consider as a motivation for our improvements: while MID adopts diffusion modeling directly in output space, we instead apply it to the discrete variables extracted by the VQ-VAE encoder. We believe that our latent-based formulation further promotes the emergence of multi-modal generative capabilities.

Tab. 4.4: Impact of distinct VQ-VAE codebooks on performance (ADE_{20}/FDE_{20}).

Dataset	Static	Full-Rank	Low-Rank
SDD	8.29/13.44	8.07/12.89	**7.86**/**12.68**
NBA	0.895/1.279	0.894/1.275	**0.893**/**1.267**
NFL	0.993/1.702	0.993/1.702	**0.982**/**1.679**

4.5 Analysis

In this section, we briefly present some additional experiments aimed at showing the effectiveness of the proposed approach. We first investigate the impact of the instance-based codebook, comparing it with alternative strategies. Then, we analyze the influence of the rank r on the behavior of the model. Finally, we provide a qualitative comparison between the proposed *low-rank* codebook and alternative strategies.

4.5.1 On the Impact of the Instance-based Codebook

Comparison with alternative strategies. To assess the merits of our *low-rank* instance-based codebook, we herein empirically compare it with two alternative strategies. On the one hand, we devise a comparison with a *static* codebook ($\rightarrow$ standard VQ-VAEs, lacking instance-level conditioning). Secondly, we contrast it with a *full-rank* codebook (which includes instance-level conditioning but lacks low-rank design constraints). To be more precise, the *full-rank* codebook is a baseline approach herein provided, which computes the values of the codebook through a learnable module fed with historical information as input. Unlike the proposed *low-rank* counterpart, the *full-rank* codebook does not adapt a shared static codebook but directly outputs its values. Through such a comparison, we can evaluate the efficacy of constraining the updates to the dictionary within a low-dimensional manifold.

Tab. 4.4 presents the related results: as can be observed, the *low-rank* model outperforms both the *static* and *full-rank* variants. In particular, the improvements are remarkable for SDD and NFL and more modest for NBA. Moreover, the presence of instance-level conditioning, common to *full-* and *low-* approaches, proves particularly beneficial for the SDD dataset, as demonstrated by the gap w.r.t. the static codebook (similar evidence emerges for the NBA dataset).

Impact of the rank dimension. In the second place, we aim to investigate the impact of the rank r, which controls the dimension of the matrix B_{ctx}

Tab. 4.5: Impact of varying the rank of B on the behavior of the model. Optimal performance (ADE_{20}) is achieved by identifying a sweet spot characterized by a low reconstruction error (ADE_{rec}) and a high accuracy in code prediction (Acc).

Dataset	Rank	ADE_{rec} ↓	Acc(%) ↑	ADE_{20} ↓
SDD	4	3.41	26.38	**7.96**
	16	2.97	22.20	8.06
NBA	4	0.207	15.92	0.898
	16	0.164	13.27	**0.892**
NFL	4	0.227	15.30	**0.982**
	16	0.177	11.95	0.996

(*i.e.*, the degree of instance-level cues introduced into the codebook). In particular, we want to measure how the rank r affects: *i)* the reconstruction capabilities of the VQ-VAE decoder (learned during the first stage); *ii)* the generative capabilities of the diffusion model (learned during the second stage). For point *i)*, we exploit the Average Displacement Error (ADE_{rec}) to assess the reconstruction performance. Instead, to characterize the generative capabilities, we resort to the mean accuracy achieved by the diffusion model in predicting codebook indexes, as well as the already mentioned ADE_{20}.

Tab. 4.5 presents the results for different ranks r. We observe that a higher reconstruction capability during the initial training stage is associated with increased difficulty in the diffusion task, resulting in lower accuracy. This indicates a correlation between the two phases: achieving optimal results in the first phase does not necessarily yield the best final generation metrics, as it complicates the joint task of trajectory generation (*i.e.*, sampling from the prior and reconstructing through the decoder). Tab. 4.5 demonstrates that the most favorable final metrics are achieved by striking a balance between low reconstruction error and good diffusion accuracy.

4.5.2 On the Impact of the *k-means Centroids* Sub-sampling Strategy

To evaluate the effectiveness of the *k-means centroids* sub-sampling strategy, we conduct experiments across varying numbers of guesses and compare the resulting metrics with those obtained using a uniform sub-sampling baseline (Tab. 4.6). Notably, the *k-means centroids* approach consistently outperforms the uniform baseline, with its advantage be-

Tab. 4.6: Comparison between *k-means centroids* and uniform sub-sampling strategies on SDD. Varying the number of guesses (N), we report ADE_{20} and FDE_{20} on the obtained 20 generations.

N	Uniform	**k-means centroids**
Naive (20)	9.40/15.66	9.40/15.66
50	9.35/15.53	8.59/14.07
100	9.40/15.97	8.23/13.28
150	9.42/15.90	8.07/13.16
200	9.31/15.73	7.86/12.68

coming more pronounced as N increases. This superiority stems from the dual benefits outlined in Sec. 4.3.4: first, out-of-distribution samples naturally form independent clusters, which enhances exploration, and second, in-distribution samples aggregate into larger clusters, reducing quantization noise through element-wise averaging.

4.5.3 Qualitative Results

Figure 4.3 provides a qualitative comparison on 20 generations (with sub-sampling) produced by a VQ-VAE trained with a *static* codebook, a *dynamic* codebook, and the *low-rank* conditioned codebook (see Sec. 4.5.1). Each row illustrates a different scene from the SDD dataset, showcasing different agent behaviors: in the first one, the agent remains stationary, while in the others, it either turns left or proceeds straight ahead. Compared to the other two methods, low-rank conditioning appears to be more accurate, particularly in complex scenarios where the agent stays still or changes its direction of movement.

4.6 Conclusions

In this work, we present a stochastic approach to trajectory prediction that leverages discretization techniques to address the challenges of sampling fidelity, diversity, and multimodality. Building upon Vector Quantization, our method introduces a dynamic, instance-based codebook, enabling the incorporation of contextual information such as past trajectories and interactions. By employing a low-rank update mechanism, we enhance the flexibility of the discretization process, enabling the codebook to adapt effectively to diverse motion patterns.

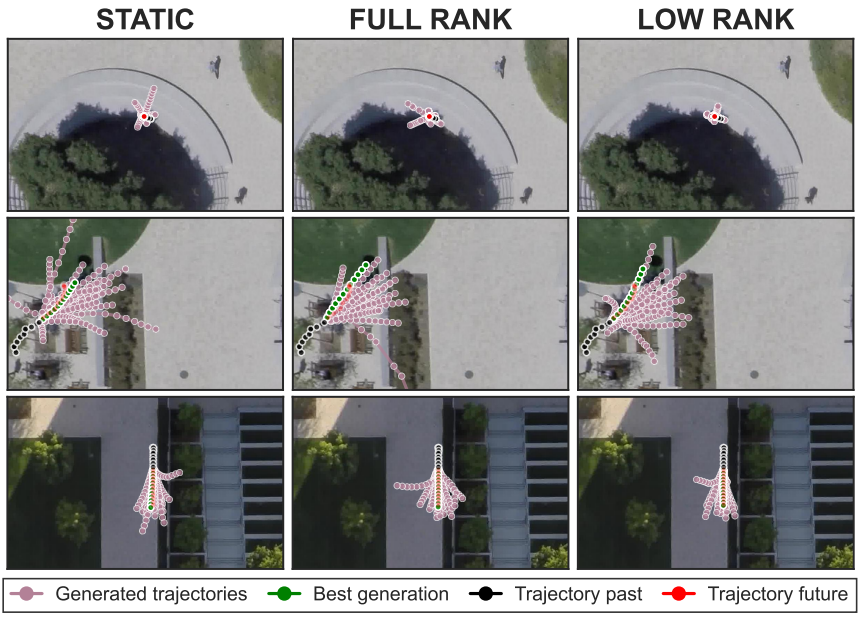


Fig. 4.3: Qualitative comparison for three SDD scenes (one for each row of the figure) between the trajectories obtained from a VQ-VAE with a static codebook, a full rank codebook the proposed *low-rank* codebook (from left to right).

Our empirical evaluation across three well-established benchmarks demonstrates the effectiveness of our approach, achieving state-of-the-art results. The dynamic discretization of the latent space not only preserves fidelity but also promotes diversity in the generated trajectories, making our method particularly suited for applications requiring reliable multimodal predictions.

However, our approach has certain limitations that must be acknowledged. First, the two-step training procedure inherent to VQ-VAE introduces additional complexity and increases the total training time. While this staged training is crucial for achieving the desired representation quality, it may pose challenges for scenarios demanding rapid prototyping or large-scale deployment. Second, the inference time scales with the number N of initial guesses, as detailed in Sec. 4.3.4. Striking a balance between computational efficiency and the accuracy of the ensemble of K final generations requires careful tuning of N, which may complicate its practical adoption in time-sensitive applications.

Despite these limitations, the integration of discretization techniques, particularly the use of a dynamically adaptive codebook, provides a robust

solution to the challenges of trajectory prediction. We believe this work opens avenues for further research into combining advanced discretization strategies with scalable inference methods to address these challenges and extend the applicability of our approach to even more complex domains (*e.g.*, image generation).

Part III

Conclusion

Chapter 5

Conclusion

This thesis explored the transformative potential of data discretization and discrete representation techniques across various machine learning domains, emphasizing their applications in continual learning and generative modeling. By leveraging the discrete paradigm, we demonstrated advancements in interpretability, robustness, and multimodal data processing that extend the capabilities of traditional continuous representation methods.

In Chapter 2, we established a comprehensive technical foundation, highlighting the role of i) discretization in spectral geometry for analyzing continuous structures through discrete graph representations, leveraging spectral clustering techniques to uncover intrinsic data structures, and applying spectral methods for manifold learning, and ii) discrete representation learning in advancing machine learning paradigms. Specifically, we explored Vector Quantization Variational Autoencoders (VQ-VAEs) as a transformative framework, demonstrating their ability to mitigate posterior collapse, enhance interpretability, and adapt to diverse data patterns through discrete latent representations. These representations enable more robust encoding of input data, support efficient clustering of latent features, and facilitate tasks such as anomaly detection, generative modeling, and downstream analysis, making VQ-VAEs highly versatile across a range of applications.

Building on this foundation, Chapter 3 introduced Continual Spectral Regulariser for Incremental Learning (CaSpeR-IL), a novel approach to regularizing Rehearsal-Based CL Methods (RBMs) by applying spectral geometry principles. By leveraging graph-based spectral properties, CaSpeR-IL effectively disentangles latent representations, promoting class separation and interpretability in supervised and low-supervised continual learning scenarios. Our findings revealed that carefully applied geometric constraints can improve model accuracy and accelerate convergence, even in demanding learning scenarios with limited annotations.

Finally, in Chapter 4, we presented Low-rank Adaptation for VQ-VAE (LRVQ), a stochastic trajectory prediction framework grounded in dynamic vector quantization techniques. This method addresses the inherent challenges of multimodality and diversity in generative tasks, introducing an adaptive, low-rank codebook to refine the representation and prediction of motion patterns. Empirical evaluations across benchmarks validated the efficacy of our approach, achieving state-of-the-art performance and showcasing the broader applicability of advanced discretization strategies.

The contributions of this thesis underscore the versatility and promise of discrete representation learning. While the proposed methods reveal new possibilities for data modeling and analysis, they also highlight practical challenges, such as increased training complexity and computational demands, particularly in large-scale or time-sensitive applications. Addressing these limitations remains a fertile area for future research, with opportunities to explore hybrid paradigms that integrate discrete and continuous frameworks more seamlessly.

Chapter 6

Acknowledgements

I would like to express my deepest gratitude to my advisor, Prof. Dr. Simone Calderara, for his unwavering support, guidance, and mentorship throughout my doctoral studies. I am especially grateful to him for introducing me to the world of machine learning research, despite my industrial background and limited experience in academia. His encouragement and belief in my potential have been instrumental in shaping my journey.

I would like to thank my company, *Ammagamma*, for providing me with the opportunity to pursue a Ph.D. while working full-time. I am grateful for their flexibility and understanding, which allowed me to balance my professional responsibilities with my academic pursuits. I would also like to thank my colleagues for their support and encouragement, especially during challenging times.

I would like to extend my heartfelt thanks to my co-authors, Emanuele, Matteo, Pietro, and Angelo, for their invaluable contributions to our collaborative research projects. A special mention goes to Angelo for his continuous support, countless hours spent discussing and brainstorming ideas, and his exceptional patience and willingness to assist me whenever I needed help. I owe a great deal to him for his friendship and guidance.

I am deeply grateful to my family, especially my mother and father, for their unwavering support and encouragement throughout this journey. I owe a special thanks to my brother, Simone, for inspiring me to pursue my passion for research and for always believing in my abilities, which has been a constant source of motivation.

Finally, I want to express my deepest gratitude to my partner, Martina, for her love, patience, and understanding. Balancing work and personal life was not always easy, but her constant support and encouragement helped me overcome the most challenging moments. Her presence has been my anchor throughout this journey.

Appendices

Appendix A

List of Publications

The following list of publications includes all conference papers, journal articles and recent pre-prints published during my Ph.D.; the contents and experimental results published in these papers have been included in the previous chapters.

[A] Emanuele Frascaroli, Riccardo Benaglia, Matteo Boschini, Luca Moschella, Cosimo Fiorini, Emanuele Rodolà, and Simone Calderara. CaSpeR: Latent Spectral Regularization for Continual Learning. In *Pattern Recognition Letters*. 2024.

[B] Riccardo Benaglia, Angelo Porrello, Pietro Buzzega, Simone Calderara, and Rita Cucchiara. Trajectory Forecasting through Low-Rank Adaptation of Discrete Latent Codes. In *International Conference on Pattern Recognition*. 2024.

Appendix B

Activities carried out during Ph.D.

Thesis supervision

- May 22, 2024. Alessandro Crescenzi. Computer Engineering Master's Degree, UNIMORE.

Participation to national and international projects

- "LEGO.AI: LEarning the Geometry of knOwledge in AI systems" – Italian Ministerial grant PRIN 2020 n. 2020TA3K9N.
- "ECOSISTER" – PNRR project (ECS 00000033 CUP E93C22001100001).
- "ELIAS: European Lighthouse of AI for Sustainability" – European Horizon No. 101120237.

Reviewing

- ACM Transactions on Multimedia Computing Communications and Applications (TOMM)

Conferences and schools attended

- Causal Inference Summer School 2022, July 11-15, 2022, University of Trento, Italy.
- Ellis Summer School, September 2023, University of Modena and Reggio Emilia, Italy.
- ICPR 2024: 27th Conference on Pattern Recognition December 01-05, 2024, Kolkata, India.

Bibliography

[1] A-vhadgar. Big data bowl. `https://github.com/a-vhadgar/Big-Data-Bowl`. Accessed: 2017.

[2] Hongjoon Ahn, Jihwan Kwak, S. Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. SS-IL: Separated Softmax for Incremental Learning. In *IEEE International Conference on Computer Vision*, 2021.

[3] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016.

[4] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient Based Sample Selection for Online Continual Learning. In *Advances in Neural Information Processing Systems*, 2019.

[5] Gianluca Antonini, Michel Bierlaire, and Mats Weber. Discrete choice models of pedestrian walking behavior. *Transp. Res. B Methodol*, 40, 2006.

[6] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *International Conference on Learning Representations Workshop*, 2018.

[7] Arjun Ashok, KJ Joseph, and Vineeth N Balasubramanian. Class-incremental learning with cross-space clustering and controlled transfer. In *Proceedings of the European Conference on Computer Vision*, 2022.

[8] Stefan Becker, Ronny Hug, Wolfgang Hubner, and Michael Arens. Red: A simple but effective baseline predictor for the trajnet benchmark. In *European Conference on Computer Vision Workshops*, 2018.

[9] Lorenzo Bonicelli, Matteo Boschini, Emanuele Frascaroli, Angelo Porrello, Matteo Pennisi, Giovanni Bellitto, Simone Palazzo, Concetto Spampinato, and Simone Calderara. On the effectiveness

of equivariant regularization for robust online continual learning. *arXiv preprint arXiv:2305.03648*, 2023.

[10] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[11] Matteo Boschini, Pietro Buzzega, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Continual semi-supervised learning through contrastive interpolation consistency. *Pattern Recognition Letters*, 2022.

[12] Deutsche Bundespost. 200th day of birth of joseph von fraunhofer (1787−1826). `https://commons.wikimedia.org/wiki/File:DBP_1987_1313_Joseph_von_Fraunhofer,_Sonnenspektrum.jpg`. First day of issue: 12 February 1987, Design: Kößlinger.

[13] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *Advances in Neural Information Processing Systems*, 2020.

[14] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New Insights on Reducing Abrupt Representation Change in Online Continual Learning. In *International Conference on Learning Representations Workshop*, 2022.

[15] Ruben Cartuyvels, Graham Spinks, and Marie-Francine Moens. Discrete and continuous representations and processing in deep learning: Looking forward. *AI Open*, 2:143–159, 2021.

[16] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[17] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *IEEE International Conference on Computer Vision*, 2021.

[18] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[19] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning:

Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision*, 2018.

[20] Arslan Chaudhry, Albert Gordo, Puneet Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[21] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. In *International Conference on Learning Representations Workshop*, 2019.

[22] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. In *International Conference on Machine Learning Workshop*, 2019.

[23] Jeff Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. In *Problems in analysis*. Princeton University Press, 1969.

[24] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

[25] David KY Chiu, Benny Cheung, and Andrew KC Wong. Information synthesis based on hierarchical maximum entropy discretization. *Journal of Experimental & Theoretical Artificial Intelligence*, 2(2):117–129, 1990.

[26] Luca Cosmo, Mikhail Panine, Arianna Rampini, Maks Ovsjanikov, Michael M. Bronstein, and Emanuele Rodolà. Isospectralization, or How to Hear Shape, Style, and Correspondence. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[27] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[28] Riccardo Del Chiaro, Bartłomiej Twardowski, Andrew Bagdanov, and Joost Van de Weijer. Ratt: Recurrent attention to transient tasks for continual image captioning. In *Advances in Neural Information Processing Systems*, 2020.

[29] Patrick Dendorfer, Sven Elflein, and Laura Leal-Taixé. Mg-gan: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction. In *IEEE International Conference on Computer Vision*, 2021.

[30] Patrick Dendorfer, Vladimir Yugay, Aljosa Osep, and Laura Leal-Taixé. Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking? *Advances in Neural Information Processing Systems*, 35:15657–15671, 2022.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2009.

[32] Remi Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv:1611.06430*, 2016.

[33] Mohammad Mahdi Derakhshani, Xiantong Zhen, Ling Shao, and Cees Snoek. Kernel continual learning. In *International Conference on Machine Learning*, 2021.

[34] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations Workshop*, 2021.

[35] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the European Conference on Computer Vision*, 2020.

[36] Xuefeng Du, Yiyou Sun, and Yixuan Li. When and how does in-distribution label help out-of-distribution detection? 2024.

[37] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *Proceedings of the European Conference on Computer Vision*, 2020.

[38] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[39] Sebastian Farquhar and Yarin Gal. Towards Robust Evaluations of Continual Learning. In *International Conference on Machine Learning Workshop*, 2018.

[40] Usama M Fayyad and Keki B Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *International Joint Conference on Artificial Intelligence*, volume 93, pages 1022–1029. Citeseer, 1993.

[41] Enrico Fini, Victor G Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[42] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2014.

[43] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Trans. Inf. Theory*, 44, 1998.

[44] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[45] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[46] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[47] Jeff Z. HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. In *Advances in Neural Information Processing Systems*, 2021.

[48] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[49] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[50] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020.

[51] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[52] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations Workshop*, 2021.

[53] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *IEEE International Conference on Computer Vision*, 2019.

[54] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *IEEE International Conference on Computer Vision*, 2019.

[55] Yu Jin, Andreas Loukas, and Joseph JaJa. Graph Coarsening with Preserved Spectral Properties. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[56] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[57] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*, 2020.

[58] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations Workshop*, 2014.

[59] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017.

[60] Alexander Kolesnikov, André Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. Uvim: A unified modeling approach for vision with learned guiding codes. *Advances in Neural Information Processing Systems*, 2022.

[61] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[62] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*, 2018.

[63] Carlos Lassance, Vincent Gripon, and Antonio Ortega. Representing deep neural networks latent space geometries with graphs. *MDPI Algorithms*, 2021.

[64] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[65] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022.

[66] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway Spectral Partitioning and Higher-Order Cheeger Inequalities. *Journal of the ACM*, 2014.

[67] Yuanman Li, Rongqin Liang, Wei Wei, Wei Wang, Jiantao Zhou, and Xia Li. Temporal pyramid network with spatial-temporal attention for pedestrian trajectory prediction. 2021.

[68] linouk23. Nba player movements. `https://github.com/linouk23/NBA-Player-Movements`. Accessed: 2016.

[69] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[70] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, 2017.

[71] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations Workshop*, 2019.

[72] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations Workshop*, 2022.

[73] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, pages 3589–3599, 2021.

[74] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[75] Gianluca Mancusi, Aniello Panariello, Angelo Porrello, Matteo Fabbri, Simone Calderara, and Rita Cucchiara. Trackflow: Multi-object tracking with normalizing flows. In *IEEE International Conference on Computer Vision*, 2023.

[76] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *Proceedings of the European Conference on Computer Vision*, 2020.

[77] Riccardo Marin, Arianna Rampini, Umberto Castellani, Emanuele Rodolà, Maks Ovsjanikov, and Simone Melzi. Instant recovery of shape from spectrum via latent space connections. In Vitomir Struc and Francisco Gómez Fernández, editors, *International Conference on 3D Vision*, 2020.

[78] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*, 1989.

[79] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799. PMLR, 2014.

[80] Andriy Mnih and Danilo Rezende. Variational inference for monte carlo objectives. In *International Conference on Machine Learning*, pages 2188–2196. PMLR, 2016.

[81] Alessio Monti, Angelo Porrello, Simone Calderara, Pasquale Coscia, Lamberto Ballan, and Rita Cucchiara. How many observations are enough? knowledge distillation for trajectory forecasting. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[82] Luca Moschella, Simone Melzi, Luca Cosmo, Filippo Maggioli, Or Litany, Maks Ovsjanikov, Leonidas J. Guibas, and Emanuele Rodolà. Learning Spectral Unions of Partial Deformable 3D Shapes. *Computer Graphics Forum*, 2022.

[83] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[84] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)*, 2012.

[85] Bo Pang, Tianyang Zhao, Xu Xie, and Ying Nian Wu. Trajectory prediction with latent belief energy-based model. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[86] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *IEEE International Conference on Computer Vision*, 2009.

[87] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020.

[88] Federico Pernici, Matteo Bruni, Claudio Baecchi, Francesco Turchini, and Alberto Del Bimbo. Class-incremental learning with pre-allocated fixed classifiers. In *International Conference on Pattern Recognition*, 2021.

[89] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. GDumb: A simple approach that questions our progress in continual learning. In *Proceedings of the European Conference on Computer Vision*, 2020.

[90] Arianna Rampini, Franco Pestarini, Luca Cosmo, Simone Melzi, and Emanuele Rodolà. Universal Spectral Adversarial Attacks for Deformable Shapes. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021.

[91] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological Review*, 1990.

[92] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in Neural Information Processing Systems*, 2019.

[93] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[94] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proceedings of the European Conference on Computer Vision*, 2016.

[95] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 1995.

[96] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer Graphics Forum*, 2017.

[97] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. 39, 2020.

[98] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[99] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Proceedings of the European Conference on Computer Vision*, 2020.

[100] Rico Sennrich. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[101] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27:379–423, 1948.

[102] Hang Shao, Abhishek Kumar, and P. Thomas Fletcher. The Riemannian Geometry of Deep Generative Models. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[103] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[104] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, 1989.

[105] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

[106] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[107] Hugo Steinhaus et al. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci*, 1(804):801, 1956.

[108] Chen Sun, Per Karlsson, Jiajun Wu, Joshua B Tenenbaum, and Kevin Murphy. Stochastic prediction of multi-agent interactions from partial observations. *International Conference on Learning Representations Workshop*, 2019.

[109] Jingchen Sun, Jiming Chen, Tao Chen, Jiayuan Fan, and Shibo He. Pidnet: An efficient network for dynamic pedestrian intrusion detection. 2020.

[110] Yuhta Takida, Takashi Shibuya, WeiHsiang Liao, Chieh-Hsin Lai, Junki Ohmura, Toshimitsu Uesaka, Naoki Murata, Shusuke Takahashi, Toshiyuki Kumakura, and Yuki Mitsufuji. Sq-vae: Variational bayes on discrete representation with self-annealed stochastic quantization. *International Conference on Machine Learning*, 2022.

[111] Charlie Tang and Russ R Salakhutdinov. Multiple futures prediction. *Advances in Neural Information Processing Systems*, 2019.

[112] Zhicong Tang, Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Improved vector quantized diffusion models. *arXiv preprint*, 2022.

[113] Luca Anthony Thiede and Pratik Prabhanjan Brahma. Analyzing the variety loss in the context of probabilistic trajectory prediction. In *IEEE International Conference on Computer Vision*, 2019.

[114] Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 2022.

[115] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 2016.

[116] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[117] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[118] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

[119] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 1985.

[120] Han Wang and Yixuan Li. Bridging ood detection and generalization: A graph-theoretic view. 2024.

[121] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[122] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[123] Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[124] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: Retrospective-memory-based trajectory prediction. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022.

[125] Guanglei Yang, Enrico Fini, Dan Xu, Paolo Rota, Mingli Ding, Moin Nabi, Xavier Alameda-Pineda, and Elisa Ricci. Uncertainty-aware contrastive distillation for incremental semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[126] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *International Conference on Learning Representations Workshop*, 2022.

[127] Ye Yuan and Kris Kitani. Diverse trajectory forecasting with determinantal point processes. *International Conference on Learning Representations Workshop*, 2020.

[128] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *IEEE International Conference on Computer Vision*, 2021.

[129] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 2017.

[130] Mengyao Zhai, Lei Chen, Jiawei He, Megha Nawhal, Frederick Tung, and Greg Mori. Piggyback GAN: Efficient Lifelong Learning for Image Conditioned Generation. In *Proceedings of the European Conference on Computer Vision*, 2020.

[131] Jiahui Zhang, Fangneng Zhan, Christian Theobalt, and Shijian Lu. Regularized vector quantization for tokenized image synthesis. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2023.

[132] Hao Zhu and Piotr Koniusz. Simple spectral graph convolution. In *International Conference on Learning Representations Workshop*, 2021.