Hyperbolic Residual Quantization: Discrete Representations for Data with Latent Hierarchies

Piotr Piękos^{1,2*}, Subhradeep Kayal^{1}, Alexandros Karatzoglou^{1}

^{1}Amazon

^{2}KAUST, AI Initiative, Thuwal, Saudi Arabia

piotr.piekos@kaust.edu.sa

Abstract

Hierarchical data arise in countless domains, from biological taxonomies and organizational charts to legal codes and knowledge graphs. Residual Quantization (RQ) is widely used to generate discrete, multitoken representations for such data by iteratively quantizing residuals in a multilevel codebook. However, its reliance on Euclidean geometry can introduce fundamental mismatches that hinder modeling of hierarchical branching, necessary for faithful representation of hierarchical data. In this work, we propose Hyperbolic Residual Quantization (HRQ), which embeds data natively in a hyperbolic manifold and performs residual quantization using hyperbolic operations and distance metrics. By adapting the embedding network, residual computation, and distance metric to hyperbolic geometry, HRQ imparts an inductive bias that aligns naturally with hierarchical branching. We claim that HRQ in comparison to RQ can generate more useful for downstream tasks discrete hierarchical representations for data with latent hierarchies. We evaluate HRQ on two tasks: supervised hierarchy modeling using WordNet hypernym trees, where the model is supervised to learn the latent hierarchy - and hierarchy discovery, where, while latent hierarchy exists in the data, the model is not directly trained or evaluated on a task related to the hierarchy. Across both scenarios, HRQ hierarchical tokens yield better performance on downstream tasks compared to Euclidean RQ with gains of up to 20% for the hierarchy modeling task. Our results demonstrate that integrating hyperbolic geometry into discrete representation learning substantially enhances the ability to capture latent hierarchies.

1 Introduction

Hierarchical structures appear throughout human knowledge and information organization, serving as essential frameworks for understanding complex relationships between entities. These structures can be found in biological classifications of living organisms [34], business organizational structures [13], and computer file systems [36]. Studies show that when children learn, they organize their knowledge in hierarchies [26]. This pattern extends to numerous other domains as well: from taxonomic categorization in libraries and archives to the nested organization of legal codes and regulations. Government systems typically follow hierarchical arrangements, with federal, state,

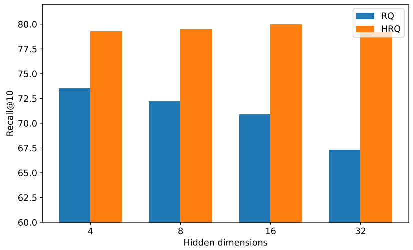

Figure 1: Recall@10 of the hypernym generation based on tokens generated by HRQ vs tokens generated by RQ. HRQ consistently outperforms RQ.

[*] The majority of this work was performed during an internship with Amazon Science.

Presented at the NeurIPS 2025 Workshop on Non-Euclidean Foundation Models and Geometric Learning.

and local levels each containing their own internal hierarchies. Similarly, academic disciplines are organized into fields, subfields, and specialized areas. The presence of hierarchical structures across such diverse domains reflects their importance in how humans conceptualize and organize information.

In the current state of machine learning modeling, most often continuous vectors are used to represent entities [5, 37, 7]. However, it can sometimes be beneficial to use discrete representations rather than continuous vector embeddings. Discrete tokens function effectively as labels because in the discrete domain, generation is equivalent to prediction. This equivalence allows models to avoid complicated generation methods like GANs [19] or diffusion models [25], and instead rely on more straightforward prediction tasks. Discrete representations also tend to be more interpretable as each token can correspond to a specific concept or attribute in the hierarchy [44].

Residual Quantization Variational Autoencoders (RQ-VAE) [31, 58] leverage these benefits by creating semantic hierarchical discrete representations through a multilevel quantization process [51]. At each step of residual quantization, the model encodes increasingly fine-grained details, with earlier levels capturing broader structural elements and later levels representing more specific attributes. The result is a list of tokens that together create an identifier of the entity. We will refer to this hierarchical discrete representation as *Multitoken(MT)*. By learning discrete tokens at multiple levels of abstraction, RQ-VAE provides a framework for modeling hierarchical relations directly from dense embedding.

However, RQ-VAE operates within Euclidean space, which imposes fundamental limitations on its ability to capture hierarchical relationships. Euclidean geometry struggles to efficiently represent tree-like structures [22], as the volume of space grows polynomially with distance from the origin, while the number of nodes in a hierarchy typically grows exponentially with depth. This geometric mismatch means that Euclidean-based models like RQ-VAE inevitably lose important hierarchical information during encoding.

In contrast, hyperbolic space [22], a Riemannian manifold with constant negative curvature, has been shown to model hierarchies remarkably well [40, 41]. The hyperbolic space can approximately isometrically embed any tree already in two dimensions [22], whereas the same cannot be said for the Euclidean space of any dimension. The volume in the hyperbolic space grows exponentially with distance from the origin, aligning well with the growth of number of nodes in hierarchy.

The ability to encode trees by hyperbolic geometry has inspired numerous advances in machine learning. Hyperbolic neural networks have also been extensively leveraged in continuous embedding models to exploit latent hierarchies in a variety of domains. Poincaré embeddings [40], learn continuous hierarchies by mapping symbolic data into an n-dimensional Poincaré ball. The authors showed that these embeddings outperform the Euclidean ones on tree-structured data in terms of both representation capacity and generalization ability. Hyperbolic embeddings found use in data domains rich in latent hierarchies, like knowledge-graph representation[3, 12, 32] and recommender systems [49, 10, 39], and other [54, 18].

Despite these advances in continuous embedding models, the application of hyperbolic geometry to discrete representation learning has remained mostly underexplored. HyperVQ [20] proposes to perform vector quantization in a hyperbolic space by phrasing it as a hyperbolic multinomial logistic regression. In this paper, we use hyperbolic distance to find the nearest codebook vector and focus on the hyperbolic version of Residual Quantization. We introduce Hyperbolic Residual Quantization (HRQ), which performs residual quantization (RQ) in a hyperbolic space with an adapted process of residual quantization to accommodate the hyperbolic structure. We claim that for **data with latent hierarchies** residual quantization benefits from hierarchical inductive bias induced by hyperbolic space. We implement this approach through several key adaptations: first, we employ hyperbolic neural networks for the embedding process, ensuring that data representations reside natively in hyperbolic space. Second, we utilize hyperbolic operations to calculate the residuals between quantization levels, preserving the geometric properties of the space throughout the quantization process. Finally, we incorporate hyperbolic distance metrics in the clustering algorithm, allowing the model to properly capture the hierarchical relationships between data points. These modifications enable HRQ-VAE to make use of the natural advantages of hyperbolic geometry to represent hierarchical structures while maintaining the benefits of discrete token-based representations.

We evaluate the quality of the multitokens created by HRQ in two scenarios. First, we test its ability to model hierarchies with supervision on the hierarchy. (H)RQ creates multitokens of nouns [38] based on their hypernymy relation. Then, we test which representation is more useful in generating the hypernym for a given noun. We show that multitokens learned with Hyperbolic Residual Quantization significantly outperform tokens learned with Residual Quantization. Furthermore, we test the model's ability to create meaningful hierarchies without direct supervision on the hierarchy. Specifically, we evaluate it in a scenario where hierarchy exists, but the model is not supervised on modeling the hierarchy and is used for a task not directly related to the hierarchy. We show that the multitokens generated by HRQ outperform the multitokens generated by RQ in this scenario as well.

The paper is organized as follows. In Section 2.1 we introduce necessary concepts from the theory of hyperbolic spaces for our method and describe the RQ-VAE algorithm. In Section 3 we describe HRQ-VAE. In Section 4 we demonstrate our experimental results. In section 5. Finally, in Section 6 we summarize our findings and propose future directions.

2 Background

2.1 Hyperbolic Space

Hyperbolic geometry operates on manifolds with constant negative Gaussian curvature. A fundamental characteristic of hyperbolic geometry is its exponential spatial expansion relative to the distance from any reference point, creating abundant capacity to represent branching structures. This property enables hyperbolic spaces to accommodate the embedding of complex hierarchical relationships with minimal distortion. Research has demonstrated that arbitrary tree structures can be embedded within a hyperbolic space while approximately preserving their metric properties [22, 23]. Because of these results, hyperbolic space can be conceptualized as "a continuous version of a tree," making it exceptionally valuable for computational representations of hierarchical data structures, complex networks with inherent branching patterns, and systems characterized by nested relationships.

In this work, we use the Poincaré ball model, which is the most widely used representation of the hyperbolic space in the context of neural networks. The definition of the Poincaré ball we use follows Ganea et al. [16].

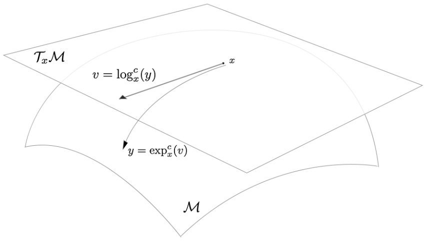

Figure 2: Visualization of the tangent space and related operations. Exponential map exp_x^c maps from the tangent space attached at x to the manifold and logarithmic map log_x^c maps from the manifold to the tangent space attached at point x.

The Poincaré Ball Model. The n-dimensional Poincaré Ball $\mathbb{P}_c^n$ with curvature c is a set $\{x \in \mathbb{R}^n : c||x||^2 < 1\}$ with Riemannian metric $g_x^{\mathbb{P}} = \lambda_x^2 g^E$, where g^E is the Euclidean metric tensor and $\lambda_x := \frac{2}{1-c||x||^2}$. The gyrovector spaces [50] allow one to define the operations corresponding to the standard operations in the euclidean vector spaces. In the Poincaré ball model $\mathbb{P}_c^n$ the **Möbius addition** is a hyperbolic analogue of a standard addition operation, defined as

$$x \oplus_c y := \frac{(1 + 2c\langle x, y\rangle + c||y||^2)x + (1 - c||x||^2)y}{1 + 2c\langle x, y\rangle + c^2||x||^2||y||^2} \quad \Big| \quad x \ominus_c y := x \oplus_c (-y)$$

Hyperbolic Distance. Distance in the Poincaré ball model of the hyperbolic space is defined as

$$d^{\mathbb{P}_c}(u, v) = \text{arcosh}(1 + 2\frac{c||u - v||^2}{(1 - c||u||^2)(1 - c||v||^2)}) \tag{1}$$

As points get farther from the center, the distance between them grows exponentially, creating increasingly more space near the boundaries. Conversely, there is limited space near the center, naturally constraining which points can occupy these central positions - a property that aligns with hierarchical structures where few elements serve as high-level abstractions. The metric treats distance differently when moving toward/away from the center versus moving side-to-side, which helps capture both how deep items are in the hierarchy and how they branch apart. Items that belong to the same branch end up close to each other but at different depths, while the exponential growth of distances ensures effective separation between different branches. This makes it easy to preserve both the local structure (items close to each other in the hierarchy) and the overall organization (how different branches relate to each other).

The Tangent Space. The tangent space $\mathcal{T}_x\mathcal{M}$ of the manifold $\mathcal{M}$ at point x is an euclidean space attached to the manifold at point x that intuitively contains all possible velocities the vector attached to x can have.

In order to translate between manifold and tangent space, two special maps are used. The exponential map projects vectors from the tangent space $\mathcal{T}_x\mathcal{M}$ to the manifold $\mathcal{M}$. In contrast, the logarithmic map is used to project from the manifold $\mathcal{M}$ to the tangent space $\mathcal{T}_x\mathcal{M}$. For the Poincaré ball model, the exponential and logarithmic maps are equal to

$$\exp_x^c(v) = x \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_x\|v\|}{2}\right)\frac{v}{\sqrt{c}\|v\|}\right)$$
$$\log_x^c(y) = \frac{2}{\sqrt{c}\lambda_x}\tanh^{-1}(\sqrt{c}\|-x \oplus_c y\|)\frac{-x \oplus_c y}{\|-x \oplus_c y\|}$$

Similarly to the addition, scalar multiplication has its own hyperbolic version. These operations suffice to derive linear layers. Furthermore, with exponential and logarithmic maps, it is possible to add nonlinearities by translating back and forth from the manifold to the tangent space. Here, we defined only necessary the concepts that will be explicitly used in the HRQ-VAE algorithm, and omitted others that are necessary to derive hyperbolic layers (like scalar multiplication). We refer interested readers to Ganea et al. [16] or Cannon et al. [8]

3 Method

Multitoken(MT) is a list of discrete tokens that *together* identify an entity in the dataset D. While the typical flat discrete representation is a single number from 0 to $|D|-1$, the multitoken of length k, is a list of k tokens $[t_0, ..., t_{k-1}]$, such that jointly they identify a corresponding entity. If multitokens are structured in a semantic way, they can offer representational benefits over flat tokens. Specfically, tokens can be shared across different multitokens, leading to information sharing and a more efficient and robust representation than flat tokens, where each entity is treated independently. For example, a tiger might be identified by a multitoken [12,24] and a lion might be identified by a token [12,364]. In this case, the first token 12 is shared between the two entities and leads to a shared part of the representation. The difficulty lies in creating good, structurally semantic multitokens.

3.1 Hyperbolic Residual Quantization.

Hyperbolic Residual Quantization (HRQ) is a method for hierarchical multitoken representation that performs residual quantization directly in hyperbolic space. The method is inspired by classical residual quantization (RQ), which approximates vectors by iteratively quantizing their residuals with respect to multiple codebooks. HRQ generalizes this process to hyperbolic geometry, ensuring that the hierarchical structure induced by quantization is better aligned with latent hierarchies in the data.

Let $C = [C_0, \dots, C_{k-1}]$ be a sequence of codebooks, where each C_i contains s vectors in $\mathbb{P}_c^h$, the h-dimensional Poincaré ball of curvature c. For a vector $x_s^{\mathbb{P}_c} \in \mathbb{P}_c^h$, HRQ produces a sequence of tokens $[t^0, t^1, \dots, t^{k-1}]$ and corresponding codebook embeddings $[e^0, e^1, \dots, e^{k-1}]$ as follows. The initial residual is set to $r^0 = x_s^{\mathbb{P}_c}$. At each step i, we quantize the current residual using hyperbolic distance:

$$e^i, t^i = q_{C_i}^{\mathbb{P}_c}(r^i), \quad \text{where } e^i \in C_i,\ t^i \in \{0, \dots, s-1\}.$$

The residual is then updated via Möbius subtraction, $r^{i+1} = r^i \ominus_c e^i$, and the process repeats until k tokens are obtained. The multitoken $[t^0, t^1, \dots, t^{k-1}]$ uniquely identifies the representation of $x_s^{\mathbb{P}_c}$. The reconstruction is given by the Möbius sum of selected embeddings denoted as $y_s^{\mathbb{P}_c} = \bigoplus_{i=0}^{k-1} e^i$.

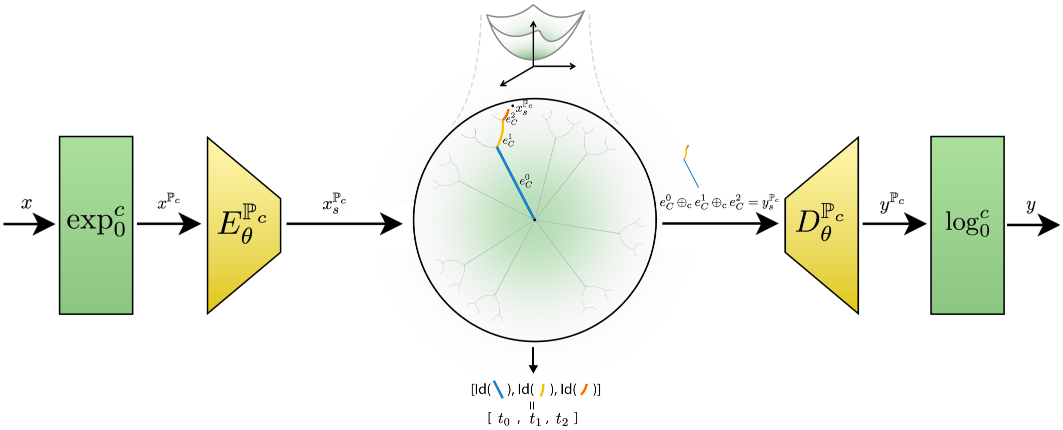

Figure 3: HRQ-VAE visualized. In the image HRQ-VAE quantizes given vector x into a multitoken $[t_0, t_1, t_2]$ and its corresponding embeddings e_C^0, e_C^1, e_C^2. Green blocks represent mapping to and from hyperbolic space. Yellow blocks represent hyperbolic autoencoder. The detailed part in the middle is responsible for hyperbolic residual quantization. The space expands exponentially the further we go away from the center. In fact, The circle's border is at infinite distance from point 0. As a consequence, most of the points must be distant from the center and only a small number of points can be at a privileged position close to the center. This leads to natural occurence of hierarchies. Light gray branches represent the possible HRQ-VAE and

We denote the quantization process by $HRQ_C(x_s^{\mathbb{P}_c}) = \left([t^0, \ldots, t^{k-1}], \, y_s^{\mathbb{P}_c}\right)$. In cases where the final residual matches the last codebook, i.e. $r^{k-1} \in C_{k-1}$, the reconstruction is exact, $y_s^{\mathbb{P}_c} = x_s^{\mathbb{P}_c}$. Otherwise, the approximation error $d_{\mathbb{P}_c}(x_s^{\mathbb{P}_c}, y_s^{\mathbb{P}_c})$ is minimized during training. We optimize the codebooks with the loss

$$L_{HRQ}(x_s^{\mathbb{P}_c}) = \sum_{i=1}^{k} \left(\|sg[r^i] - e^i\|^2 + \alpha \|r^i - sg[e^i]\|^2 \right),$$

where $sg[\cdot]$ denotes the stop-gradient operator and α controls whether residuals are pulled toward the codebook vectors or vice versa. This objective ensures stable training and prevents codebook collapse.To allow backpropagation through the quantization step, the derivative with respect to $x_s^{\mathbb{P}_c}$ is modeled using the straight-through estimator, $\frac{dy_s^{\mathbb{P}_c}}{dx_s^{\mathbb{P}_c}} \approx I$. To solve conflicts between items in the representations generated by RQ, it adds an additional token that extends multitoken to uniquely identify the item. In practice, this is rarely necessary to uniquely identify an item as the multitokens from RQ most often suffice for the identification.

By embedding the residual quantization mechanism in hyperbolic space, HRQ directly exploits the curvature of the geometry to encode latent hierarchies. The resulting multitokens are more structured, semantically meaningful, and efficient than those produced in Euclidean space.

3.2 Hyperbolic Residual Quantization VAE (HRQ-VAE)

We introduce **Hyperbolic Residual Quantization VAE (HRQ-VAE)**, a generative model that integrates Hyperbolic Residual Quantization into an autoencoder framework. The method is inspired by RQ-VAE, which applies residual quantization in Euclidean space, but adapted to hyperbolic space, ensuring that the learned multitoken representations better align with latent hierarchies.

Formally, let $E_\theta : \mathbb{R}^d \to \mathbb{P}_c^{h_s}$ and $D_\theta^{\mathbb{P}_c} : \mathbb{P}_c^{h_s} \to \mathbb{P}_c^h$ denote the encoder and decoder networks, parameterized as hyperbolic neural networks [16]. Since the input $x \in \mathbb{R}^d$ lies in Euclidean space, we map it to the hyperbolic manifold via the exponential map:

$$z^{\mathbb{P}_c} = E_\theta(\exp_0^c(x)).$$

HRQ quantizes the latent embedding: $([t^0, \ldots, t^{k-1}], \, y_s^{\mathbb{P}_c}) = HRQ_C(z^{\mathbb{P}_c})$,. The decoder reconstructs the embedding, which is mapped back to Euclidean space with the logarithmic map:

$$\hat{x} = \log_0^c \left(D_\theta^{\mathbb{P}_c}(y_s^{\mathbb{P}_c}) \right).$$

The model is trained with two objectives. The reconstruction loss $L_R(x) = \|x - \hat{x}\|^2$ encourages faithful reconstruction, while the quantization loss $L_{HRQ}(x_s^{\mathbb{P}_c})$ updates the codebooks and controls the interaction between residuals and code vectors. The total loss is $L(x) = L_R(x) + L_{HRQ}(x_s^{\mathbb{P}_c})$.

Optimization is performed with Riemannian SGD [4], ensuring that both codebook vectors and network parameters remain consistent within hyperbolic space. We denote the full process as

$$HRQ\text{-}VAE(x) = \big([t^0, \dots, t^{k-1}], \hat{x}\big).$$

By embedding HRQ within a hyperbolic autoencoder, HRQ-VAE learns multitoken representations that are discrete and natively hierarchical due to hyperbolic structure. The visualization of HRQ-VAE is shown in the Figure 3. The pseudocode for HRQ-VAE is in the Algorithm 1.

$\lvert C_i \rvert$	k	Token type	Hidden dimensions			
			4	8	16	32
64	3	RQ	71.2%	73.7%	69.8%	67.1%
		HRQ	**79.0%**(+10.9%)	**79.6%**(+8.0%)	**79.1%**(+13.3%)	**78.3%**(+16.7%)
	4	RQ	71.2%	70.9%	70.3%	64.6%
		HRQ	**78.8%**(+10.7%)	**79.2%**(+11.8%)	**79.2%**(+12.6%)	**78.9%**(+22.1%)
128	3	RQ	71.3%	72.5%	72.4%	66.3%
		HRQ	**79.5%**(+11.4%)	**79.5%**(+9.7%)	**79.5%**(+9.8%)	**79.1%**(+19.3%)
	4	RQ	72.2%	72.7%	70.9%	64.6%
		HRQ	**79.1%**(+9.6%)	**79.4%**(+9.2%)	**79.6%**(+12.3%)	**78.7%**(+21.8%)
256	3	RQ	72.4%	73.2%	71.2%	66.2%
		HRQ	**78.9%**(+9.0%)	**80.0%**(+9.3%)	**80.3%**(+12.9%)	**79.9%**(+20.7%)
	4	RQ	73.5%	72.2%	70.9%	67.3%
		HRQ	**79.3%**(+7.9%)	**79.5%**(+10.1%)	**80.0%**(+12.9%)	**79.2%**(+17.7%)

Table 1: Top 10 Recall of hypernymy prediction models trained on multitokens generated by RQ and HRQ. Despite operating on the same model and differing only in the structure of the multitokens, model that operated on HRQ multitokens produced significantly higher recall than model operating on RQ multitokens. The value $(+x.x\%)$ represents a percentage gain of HRQ w.r.t. RQ: (HRQ-RQ)/RQ. These results demonstrate that HRQ multitokens capture significantly more semantic information.

4 Experiments

In this section, we empirically evaluate the quality of multitokens produced by RQ and HRQ. We evaluate the quality of tokens in a two-step pipeline. First, we learn the tokens for all entities. Then, we fix the tokens and investigate how well they perform in a downstream task.

We focus on data with latent hierarchies and our claim is that hyperbolic residual quantization produces better multitokens for data that contain latent hierarchies. Therefore, all our experiments are characterized by the clear existence of hierarchies in datasets. We evaluate HRQ in two distinct scenarios. First, we look at Hierarchy Modeling(Section 4.1), in which the multitokens are explicitly trained on the hierarchy and the downstream task is related to the hierarchy multitokens are modeling. The second scenario, which we call Hierarchy Discovery(Section 4.2), operates a dataset which contains latent hierarchies, but the model that learns multitokens is not supervised on these hierarchies. Furthermore, the downstream task is not directly related to the latent hierarchy as well.

Additionally, in Appendix D we inspect the structure of the space hypothesize on what causes benefits of HRQ multitokens. The implementation details for all methods are in Appendix C.

4.1 Hierarchy Modeling

In this section, we investigate how effectively HRQ tokens capture hierarchical relationships compared to RQ. To do that, (H)RQ creates tokens by learning directly to predict hierarchical relation. Our experimental setup is similar to the main experiments from Nickel and Kiela [40] that is adapted to discrete setting to compare the quality of discrete multitokens. Specficially, we use the transitive closure of the WordNet [38] noun taxonomy. The WordNet taoxnomy contains 82,115 nouns and 743,241 hypernymy relations. A hypernymy is a semantic "is-a" link where a general word(the hypernym) covers a group of more specific words (its hyponyms). We first learn the multitokens by simultaneously training an embedding and learning (H)RQ.

After we learn and create multitokens for all nouns, we fix multitokens, and we train a sequence-to-sequence transfomer model that translates noun to its hypernym, both represented with their corresponding multitokens. We evaluate the model by measuring recall@10 in the test dataset, which was not visible neither for the multitoken creation nor for the training of the sequence-to-sequence model. The test dataset is a randomly selected 15% of all hypernymy relations. We learn the multitoken of the noun by embedding it in a continuous space, then contrastively pushing away nouns that are not in the hypernymy relation and pulling closer nouns that are. At the same time, the embedding is being quantized into multitokens by RQ or HRQ. Both embedding and codebook vectors are trained joinlty at the same time.

Formally, let N be the set of nouns, and $H = \{(u, v) : u \in N, v \in N : u \text{ is a hypernym of } v\}$ be the set defining the hypernymy relation. Let E_θ be the h-dimensional embedding network, that embeds either in Euclidean or hyperbolic space depending on the model. Let $H'(u) = \{v : (u, v) \notin H\} \cup \{u\}$. We also have a (H)RQ algorithm with a codebook of length k, each codebook having s vectors. Then the total loss for $(u, v) \in H$ is given by:

$$L(u, v) = \log \frac{e^{-d(E_\theta(u), E_\theta(v))}}{\sum_{v' \in H'(u)} e^{-d(E_\theta(u), E_\theta(v'))}} + L_{RQ}(E_\theta(u)) + L_{RQ}(E_\theta(v))$$

In practice, we limit $H'(u)$ to 50 sample nouns from N that are not hypernyms of u and v.

$L(u, v)$ is minimized for θ, C with d being either the euclidean distance for RQ or hyperbolic distance for HRQ. As a result, it produces multitokens for all nouns $T(u) = [t_0, ..., t_{k-1}]$. In the next step, a transformer sequence-to-sequence model is trained to predict a hypernym for a given noun, both represented as their learned multitokens. The idea is that multitokens that better capture the structure of the space will serve as a more useful representation for the hypernymy generation.

To evaluate the representation quality of multitokens generated by RQ and HRQ, we investigate different combinations of parameters. We investigate the results for token lengths $k \in \{3, 4\}$. We vary the size of codebooks $s \in \{64, 128, 256\}$ and the dimensions of dense embeddings $h \in \{4, 8, 16, 32\}$. We focus on small dimensionalities of the dense embedding because usually before the residual quantization occurs, the embedding is mapped to a low-dimensional space.

The results for $k = 4$ and $|C_i| = 256$ are shown in the Figure 1. The complete results are shown in Table 1. Although the final sequence-to-sequence models differ only in the representations of the nouns and otherwise have the same architecture, the tokens generated by HRQ sometimes lead to an improvement of up to 20% over the tokens generated by RQ. It clearly demonstrates the quality difference in favor of HRQ. The significant improvement is consistent across all dimensions tested. This demonstrates that HRQ is able to create significantly more semantic multitokens than the RQ, when it is trained to predict hierarchical relations.

4.2 Hierarchy Discovery

In real-world applications, the data often contains inherent hierarchical structures that are not explicitly labeled or available during model training. Although approaches directly supervised on the hierarchy can effectively learn to mimic known hierarchies, discovering latent hierarchical relationships without direct supervision presents a more challenging task. We call it the "Hierarchy Discovery", where the (H)RQ model creates hierarchical structures based solely on patterns present in the embeddings.

In this setting, we evaluate whether the hierarchical inductive bias of HRQ-VAE leads to multitokens that capture more semantic information compared to the standard RQ-VAE, when neither model has access to hierarchical supervision during training. Both approaches must rely entirely on patterns within the embeddings themselves, but in HRQ-VAE there is an additional inductive bias towards the formation of hierarchical structures. Our evaluation focuses on downstream task performance as the primary measure of representation quality, reflecting the practical perspective that better representations should yield improved results on real-world problems.

We use the Amazon Reviews 2014 [35] dataset, which contains a product catalog with detailed descriptions. We will generate multitokens of the products and then use them in a recommender system. We first generate dense embeddings with the MPNet [48] from product descriptions, which serve as input for both our RQ-VAE and HRQ-VAE models.

RQ-VAE and HRQ-VAE are trained to produce multitokens without explicit hierarchical supervision. The quality of these discrete representations is subsequently measured by using them to train a sequence-to-sequence recommender system, where performance differences directly reflect the semantic richness captured by each quantization approach. The recommender system is a transformer encoder-decoder that predicts the next bought item based on all the previous history. Following the protocol of Rajput et al. [44] we limit the user histories to those that have at least five items and truncate the histories to 20 items.

In order to test the model beyond the Amazon Reviews 2014 dataset, we also include the evaluation on the MovieLens 10M [24] dataset. Note that both product and movies can be structured in latent taxonomical hierarchies, making them suitable for our case. As the MovieLens dataset does not contain the movie description, we first generate the descriptions with the LLM Claude [1]. The prompt used to generate the description is included in the Appendix B.

Dataset	Metric	Random	RQ-VAE	HRQ-VAE
AR Beauty	NDCG@5	1.66%±0.07	2.29%±0.03	**2.41%**±0.04 (+5.2%)
	Recall@5	2.06%±0.09	3.68%±0.03	**3.74%**±0.04 (+1.6%)
	NDCG@10	2.35%±0.09	2.83%±0.05	**2.89%**±0.05(+2.1%)
	Recall@10	3.87%±0.17	4.83%±0.06	**5.01%**±0.06(+3.7%)
AR TaG	NDCG@5	1.51%±0.07	1.91%±0.02	**1.94%**±0.02(+1.6%)
	Recall@5	1.97%±0.09	2.82%±0.03	**2.93%**±0.03(+3.9%)
	NDCG@10	1.94%±0.09	2.45%±0.03	**2.47%**±0.03(+0.8%)
	Recall@10	2.76%±0.16	4.22%±0.08	**4.53%**±0.09(+7.3%)
AR SaO	NDCG@5	0.95%±0.07	**1.03%**±0.02	**1.03%**±0.02 (+0.0%)
	Recall@5	1.34%±0.08	1.58%±0.02	**1.62%**±0.02(+2.5%)
	NDCG@10	1.29%±0.09	**1.50%**±0.03	1.48%±0.02(−1.4%)
	Recall@10	2.41%±0.14	2.78%±0.04	**2.85%**±0.04(+4.0%)
MovieLens	NDCG@5	11.42%±0.32	11.45%±0.20	**11.76%**±0.24(+2.7%)
	Recall@5	17.43%±0.54	17.62%±0.21	**17.90%**± 0.25(+1.6%)
	NDCG@10	13.21%±0.58	13.89%±0.28	**14.27%**± 0.40(+2.7%)
	Recall@10	23.52%±0.73	25.11%±0.37	**25.49%**± 0.33(+1.5%)

Table 2: Results of recommender systems for different multitokens (Random, RQ-VAE and HRQ-VAE) across four datasets. The table reports average metric over 8 runs. The observed standard deviation is written on the right of the results. For the HRQ-VAE, percentage improvement over RQ-VAE is in the parantheses.

Apart from the RQ-VAE and HRQ-VAE we also include a baseline that consists of randomly sampled tokens with additional token that distinguishes conflicts, similarly to (H)RQ. The main results are shown in Table 2. The multitokens generated by HRQ-VAE consistently outperform RQ-VAE and the random baseline. The reported results are on the test set with each model type selected with the highest performance on the validation set.

5 Related Work

Quantized representations. Quantized discrete representations are an alternative to dense embeddings, which recently gained popularity. The aim of a quantized representation is to create coarse information representations that focus on qualitative properties [21]. Van Den Oord et al. [52] proposes VQ-VAE that learns the vector codebook simultaneously together with the embeddings.

This was further enhanced by RQ-VAE [31, 58] that calculates the sequence of discrete tokens by iteratively quantizing the residuals. Discrete representations are beneficial to use as labels, as they avoid issues of high-dimensional continuous generation. VQ-GANs [15, 57] utilize vector quantization for adversarial [19, 47] image generation. Zeghidour et al. [58], Yang et al. [55] uses VQ-VAE for audio generation. RQ-VAE has been introduced both in audio [58] and image processing [31].

Hyperbolic Neural Networks. Neural networks operating in hyperbolic space have been demonstrated to perform well in tasks and modalities with hierarchical structures. Sala et al. [46], Nickel and Kiela [40], Ganea et al. [17] demonstrate benefits of hyperbolic embeddings for data with latent hierarchies. Ganea et al. [16] derives multi-layer fully connected hyperbolic neural network. The benefits of utilizing hyperbolic neural networks can be observed in multiple areas containing hierarchies. Ma et al. [33], Yang et al. [56] model the taxonomy of objects in hyperbolic space. [2, 28] applies hyperbolic nets to computer vision. Hyperbolic neural networks have shown their benefits in reinforcement learning [9] due to the hierarchical nature of the unrolling episodes. Chamberlain et al. [10], Chen et al. [14], Sun et al. [49] applies hyperbolic networks to recommender systems with two-fold motivation: 1) The bipartite graph nature of the interactions between users and items, which has been shown to correspond to a complex network [30], and 2) Taxonomical nature of the items. Hyper-VQ [20] proposes vector quantization in hyperbolic space. It presents the quantization problem as a hyperbolic multinomial regression and is orthogonal to our contributions for HRQ-VAE. Both can be combined together and we consider that a promising future work.

Recommender Systems. Traditional recommender systems represent items with ID-based tokens, though with the advent of LLM usage in recommender systems, content-based tokenazation methods have been proposed. RQ-VAE has been used in recommender systems [44] to tokenize item representation and train a transformer-based recommender algorithm [27]. Another sequential generative recommendation model Petrov and Macdonald [42] also applies a quantization scheme based on collaborative filtering and matrix factorization computed embedding.

6 Conclusions and future work

The results shown in this work indicate that HRQ-VAE creates hierarchical representations more robust than RQ-VAE, when latent hierarchies appear in the dataset. We show that even if the model is not directly supervised on the latent hierarchy, the multitoken generated by HRQ-VAE might still be more robust than multitoken generated by RQ-VAE. Due to ubiquity of latent hierarchies in practical dataset, we see potential for number of applications of HRQ-VAE.

Furthermore, improving the performance of discrete hierarchical tokens leads to more interpretable models, as the hierarchical tokens can be related to the data taxonomies. This direction of research might lead to models whose discrete representations remain robust under domain shifts and noisy inputs, leading to societal benefits such as enhanced transparency in AI-driven decision-making, and greater public trust through auditability of the deployed systems.

In this work, we limited the scope of investigation to datasets that exhibit clear latent hierarchies. However, RQ-VAE has shown impressive results in several domains that do not follow this assumption, such as image and audio processing. HRQ-VAE, after appropriate adaptation, can potentially be applied to these domains as well. Each modality presents its own unique challenges related to the scale of experiments, hyperbolic adaptations, and the analysis of performance-contributing factors. Due to these complexities, we considered these additional modalities outside the scope of the current paper. However, exploring the application of HRQ-VAE to these diverse domains remains an exciting direction for future work.

Acknowledgements

We would like to thank the Amazon DiscoTec Science Team, in particular Fernando Morales, Eva Mohedano Robles, David Buchaca Prats and Miriam Bellver Bueno, for the insightful discussions during the early stages of this work. We also thank Firas Laakom for his valuable comments on draft versions of the manuscript.

References

[1] Anthropic. Claude 3.5 sonnet. `https://anthropic.com/claude`, 2024.

[2] Mina Ghadimi Atigh, Julian Schoep, Erman Acar, Nanne Van Noord, and Pascal Mettes. Hyperbolic image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4453–4462, 2022.

[3] Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. Multi-relational poincaré graph embeddings. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4465–4475, 2019. URL `https://proceedings.neurips.cc/paper/2019/hash/f8b932c70d0b2e6bf071729a4fa68dfc-Abstract.html`.

[4] Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760*, 2018.

[5] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[6] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[7] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795, 2013. URL `https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html`.

[8] James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31(59-115):2, 1997.

[9] Edoardo Cetin, Benjamin Chamberlain, Michael Bronstein, and Jonathan J Hunt. Hyperbolic deep reinforcement learning. *arXiv preprint arXiv:2210.01542*, 2022.

[10] Benjamin Paul Chamberlain, Stephen R Hardwick, David R Wardrope, Fabon Dzogang, Fabio Daolio, and Saúl Vargas. Scalable hyperbolic recommender systems. *arXiv preprint arXiv:1902.08648*, 2019.

[11] Ines Chami, Albert Gu, Vaggos Chatziafratis, and Christopher Ré. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. *Advances in Neural Information Processing Systems*, 33:15065–15076, 2020.

[12] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-dimensional hyperbolic knowledge graph embeddings. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6901–6914. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.617. URL `https://doi.org/10.18653/v1/2020.acl-main.617`.

[13] Alfred D Chandler Jr. *Strategy and structure: Chapters in the history of the American industrial enterprise*, volume 461. MIT press, 1969.

[14] Yankai Chen, Menglin Yang, Yingxue Zhang, Mengchen Zhao, Ziqiao Meng, Jianye Hao, and Irwin King. Modeling scale-free graphs with hyperbolic geometry for knowledge-aware recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 94–102, 2022.

[15] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[16] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

[17] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *International conference on machine learning*, pages 1646–1655. PMLR, 2018.

[18] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pages 1646–1655. PMLR, 2018. URL `http://proceedings.mlr.press/v80/ganea18a.html`.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[20] Nabarun Goswami, Yusuke Mukuta, and Tatsuya Harada. Hypervq: Mlr-based vector quantization in hyperbolic space. *arXiv preprint arXiv:2403.13015*, 2024.

[21] Robert Gray. Vector quantization. *IEEE Assp Magazine*, 1(2):4–29, 1984.

[22] M Gromov. Hyperbolic groups. *Essays in Group Theory*, pages 75–263, 1987.

[23] Matthias Hamann. On the tree-likeness of hyperbolic spaces. In *Mathematical proceedings of the cambridge philosophical society*, volume 164, pages 345–361. Cambridge University Press, 2018.

[24] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[26] Barbel Inhelder and Jean Piaget. *The early growth of logic in the child: Classification and seriation*. Routledge, 2013.

[27] Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. pages 197–206, 2018. doi: 10.1109%2fICDM.2018.00035.

[28] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6418–6428, 2020.

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[30] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 82(3):036106, 2010.

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

[32] Qiuyu Liang, Weihua Wang, Feilong Bao, and Guanglai Gao. Fully hyperbolic rotation for knowledge graph embedding. In Ulle Endriss, Francisco S. Melo, Kerstin Bach, Alberto José Bugarín Diz, Jose Maria Alonso-Moral, Senén Barro, and Fredrik Heintz, editors, *ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024)*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, pages 1615–1622. IOS Press, 2024. doi: 10.3233/FAIA240668. URL `https://doi.org/10.3233/FAIA240668`.

[33] Mingyu Derek Ma, Muhao Chen, Te-Lin Wu, and Nanyun Peng. Hyperexpan: Taxonomy expansion with hyperbolic representation learning. *arXiv preprint arXiv:2109.10500*, 2021.

[34] Ernst Mayr. The role of systematics in biology: The study of all aspects of the diversity of life is one of the most important concerns in biology. *Science*, 159(3815):595–599, 1968.

[35] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52, 2015.

[36] Marshall K McKusick, William N Joy, Samuel J Leffler, and Robert S Fabry. A fast file system for unix. *ACM Transactions on Computer Systems (TOCS)*, 2(3):181–197, 1984.

[37] Tomas Mikolov. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[38] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38 (11):39–41, 1995. doi: 10.1145/219717.219748.

[39] Leyla Mirvakhabova, Evgeny Frolov, Valentin Khrulkov, Ivan V. Oseledets, and Alexander Tuzhilin. Performance of hyperbolic geometry models on top-n recommendation tasks. In Rodrygo L. T. Santos, Leandro Balby Marinho, Elizabeth M. Daly, Li Chen, Kim Falk, Noam Koenigstein, and Edleno Silva de Moura, editors, *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, pages 527–532. ACM, 2020. doi: 10.1145/3383313.3412219. URL `https://doi.org/10.1145/3383313.3412219`.

[40] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

[41] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International conference on machine learning*, pages 3779–3788. PMLR, 2018.

[42] Aleksandr V. Petrov and Craig Macdonald. Generative sequential recommendation with gptrec, 2023. URL `https://arxiv.org/abs/2306.11114`.

[43] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[44] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=BJ0fQUU32w`.

[45] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[46] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR, 2018.

[47] Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. 1991.

[48] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867, 2020.

[49] Jianing Sun, Zhaoyue Cheng, Saba Zuberi, Felipe Pérez, and Maksims Volkovs. Hgcf: Hyperbolic graph convolution networks for collaborative filtering. In *Proceedings of the Web Conference 2021*, pages 593–601, 2021.

[50] Abraham Albert Ungar. *Analytic hyperbolic geometry and Albert Einstein's special theory of relativity*. World Scientific, 2008.

[51] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6306–6315, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html`.

[52] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[53] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[54] Benjamin Wilson. Learning phylogenetic trees as hyperbolic point configurations. *CoRR*, abs/2104.11430, 2021. URL `https://arxiv.org/abs/2104.11430`.

[55] Dongchao Yang, Songxiang Liu, Rongjie Huang, Jinchuan Tian, Chao Weng, and Yuexian Zou. Hifi-codec: Group-residual vector quantization for high fidelity audio codec. *arXiv preprint arXiv:2305.02765*, 2023.

[56] Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. Hypformer: Exploring efficient hyperbolic transformer fully in hyperbolic space. *arXiv preprint arXiv:2407.01290*, 2024.

[57] 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. *arXiv preprint arXiv:2110.04627*, 2021.

[58] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

[59] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902, 2020.

A HRQ-VAE

Algorithm 1 HRQ-VAE

```
**Input:** $x \in \mathbb{R}^d$
$x^{\mathbb{P}_c} \leftarrow \exp_0^c(x)$
$x_s^{\mathbb{P}_c} \leftarrow E_\theta^{\mathbb{P}_c}(x^{\mathbb{P}_c})$
$y_s^{\mathbb{P}_c} \leftarrow 0, r_C^0 = x_s^{\mathbb{P}_c}$
**for** $i \in \{0, .., k-1\}$ **do**
    $e_C^i, t_i = q_C(r_C^i)$
    add $t_i$ to the return sequence
    $r_C^{i+1} \leftarrow r_C^i \ominus_c e_c^{i-1}$
    $y_s^{\mathbb{P}_c} \leftarrow y_s^{\mathbb{P}_c} \oplus_c e_C^i$
**end for**
$y^{\mathbb{P}_c} \leftarrow D_\theta^{\mathbb{P}_c}(y_s^{\mathbb{P}_c})$
$y \leftarrow \log_0^c(y^{\mathbb{P}_c})$
$l_{\text{rec}} = ||x - y||^2$
$l_{\text{cmt}} = \sum_{i=0}^{k-1}(||sg[r_C^i] - e_C^i||^2 + \alpha||r_C^i - sg[e_C^i]||^2)$
$l \leftarrow l_{\text{rec}} + l_{\text{cmt}}$
$\nabla\theta \leftarrow \frac{dl}{d\theta}$ ; $\nabla C \leftarrow \frac{dl}{dC}$
**return** $t_0, t_1, ..., t_{k-1}, \nabla\theta, \nabla C$
```

Algorithm 2: HRQ-VAE

B Datasets details

B.1 Hierarchy Modeling

WordNet is a large, manually curated lexical database of English that groups words into synonym sets (synsets) and interlinks these synsets via semantic relations such as hypernymy and hyponymy, enabling rich hierarchical modeling of concepts [38]. Each synset contains a gloss (brief definition) and example usages, and synsets are organized into noun, verb, adjective, and adverb hierarchies [**?**]. For our hierarchy modeling, we focus exclusively on the noun subnetwork, where the "is-a" (hypernym) relation defines a directed acyclic graph representing a noun hierarchy.
The noun subnetwork consists of $82,115$ nouns and $743,241$ hypernymy relations. We split it into the train set and test set by randomly choosing 85% of the hypernymy relations to be selected for the the train set. The Embedding, RQ and the sequence-to-sequence models are all trained on the train set. We use the remaining 15% as the test set on which we report the performance.

B.2 Hierarchy Discovery

We used four datasets to evaluate the HRQ-VAE performance in the Hierarchy Discovery section. Three data sets are the categories 'Beauty', 'Sports and Outdoors' and 'Toys and Games' from the Amazon Reviews 2014 suite [35]. We also evaluate HRQ-VAE on the MovieLens10M dataset [24]. The (H)RQ-VAE uses dense embeddings of the items to learn the corresponding hierarchical tokens. In order to create dense embeddings of the items, we use a pretrained, fixed language model embedding [48], which embeds the description of the item. The descriptions of the items are included in the Amazon Reviews 2014 datasets. For MovieLens, we first create the description from the movie title with the help of a Claude 3.5 Sonnet [1] language model.

Dataset	Users	Items
AR Beauty	22,363	12,101
AR Toys and Games	35,598	18,357
AR Sports and Outdoors	19,412	11,924
MovieLens10M	71,567	10,681

Table 3: Quantitative statistics of datasets used in Hierarchy Discovery experiments.

In all experiments, we focus on predicting the next item the user interacted with (whether watched a movie or bought a product) and disregard the scores. This is a standard practice in the area of recommender systems [44, 27, 59].
In order to use MovieLens, we first create the descriptions with Claude 3.5 Sonnet [1]. We use the following prompt to generate the movie description:

```
You are an expert in movie descriptions.  Your task is to
generate movie description that:
- contains a maximum of 100 words
- captures the general theme of the movie and interesting
specifics of the story
- can be used adequately in a search engine to search for a
movie Your task is to generate a movie description for the
following movie title.  Return the movie description and do
not return anything else.
```

From the description, we generate a dense embedding in the same way as for AR datasets. In all datasets, we cut the histories shorter than 5 elements and limit the length of user histories to 20.

Test/train split. Following the standard evaluation [44] method, we divide user histories into the test, validation, and training part with a leave-one-out strategy. If the user history is a sequence of items $[i_1, ..., i_T]$, with T elements. The training set consists of history limited to $T-2$ tokens. The validation set is a prediction of i_{T-1} based on $[i_1, ..., i_{T-2}]$ and the test set is a prediction of i_T based on $[i_1, ..., i_{T-1}]$. The last and second-to-last items are taken from all users for the validation and test split, regardless of the length trajectory. Note that i_T in the notation above represents an item, not a token. Hence, for a multitoken scenario of tokens trained with (H)RQ-VAE, a single item i_T will be represented by a multitoken of length k and all k atomic tokens will be selected for the test/validation set.

C Implementation details

C.1 Hierarchy Modeling

(H)RQ. To create multitokens of nouns we learn at the same time the embedding of the nouns and the codebook that quantizes the tokens.
We investigate the results for token lengths $k \in \{3, 4\}$. We vary the size of the codebooks $s \in \{64, 128, 256\}$ and the dimensions of dense embeddings $h \in \{4, 8, 16, 32\}$. Other parameters follow Nickel and Kiela [40]. We use Stochastic Gradient Descent [45] or Riemannian Stochastic Gradient Descent [6] for the optimization of encoders and RQ/HRQ codebook respectively. We use the learning rate 1.0. We train both models for 1500 epochs, out of which first 20 epochs are warm-up epochs with learning rate equal to 0.01.

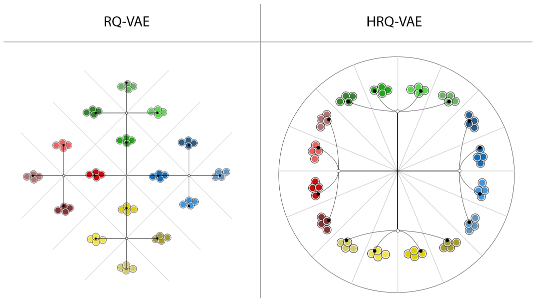

Figure 4: The embedding space structure induced by RQ-VAE and HRQ-VAE, respectively for a hierarchical tokens of length 2. The data is represented by coloured dots. Hue of the dot represents first hierarchical token. The shade represents second token. For the RQ-VAE the result is a typical effect of hierarchical clustering. HRQ-VAE due to exponential growth of the space has inductive bias to putting leaf nodes away on a similar distance away from the center.

Downstream Model. The sequence-to-sequence model is trained to generate hypernyms of a noun, both represented as multitokens. Hence, both the input and the output of the model are a list of k tokens from 0 to s. The transformer model has 4 layers for both the encoder and the decoder. The hidden dimension is equal to 256 with the feedforward dimension equal to 1024 and 8 attention heads. The embeddings of the encoder and decoder are tied. It is trained for 100 epochs with Adam [29] optimizer with a learning rate equal to 0.001.

C.2 Hierarchy Discovery

(H)RQ-VAE. The initial dense embedding of the text is calculated with 768 dimensional MPNET [48]. From the dense embedding, we train the (H)RQ-VAE and assign the new hierarchical token produced by the model to each item. The encoder in (H)RQ-VAE has 3 intermediate layers of size 512, 256, 128 with (H)ReLU activation and the output layer of size 32. The decoder has symmetric architecture to the encoder. The codebook has length 256 for each token and is not shared across tokens. We use batch size of 128, and train the (H)RQ-VAE for 5000 epochs with learning rates $[10^{-3}, 10^{-4}, 10^{-5}]$. We choose the learning rate that performed the best on the validation split of the downstream task and report the corresponding test result.

Downstream Model. We train the recommender system to evaluate the quality of the discrete representations produced by (H)RQ-VAE. User history is a sequence of items the user interacted with: either a movie they watched, or an item they bought. At each step, we predict the next item the user will interact with; specifically, we generate $k \in \{5, 10\}$ ranked guesses. To evaluate the quality of the set of guesses, we use two most popular recommender system metrics: Recall@K and NDCG@K. In our case of multitokens, each item is represented by a multitoken. Each user history has concanted multitokens of all items given a user bought(or a movie watched), and each specific recommendation is considered good if the entire multitoken corresponds to the true item the user interacted with. We split all the datasets into train,validation, and test set in the same way. We limit the histories to 20 interactions and filter the histories with less than 5 interactions. Furthermore, we select the last interaction as a test set, the second to last as a validation set, and everything else as a training set. We train a sequence-to-sequence transformer model [53] with T5 [43] architecture. For each datapoint, an output sequence is the hierarchical representation of the next item, whereas the input is all their previous history. The model has a token embedding size of 384, 6 attention heads with 64 dimension each. and 1024 dimension of the feedforward net.

Our setup for hierarchy discovery follows the parameters of Rajput et al. [44]. However, the results differ significantly on the AR dataset. The fact that they differ consistently across all tokens and also across random baselines suggests that the cause of the inconsistency must lie in the final recommender system. However, after a detailed inspection and testing of different libraries, we were unable to reproduce the original results. However, please note that, contrary to Rajput et al. [44] our claim is not about creating the best recommender system, but about comparing HRQ to RQ, and if the shift in the performance is caused by the downstream model - it is not important for our claim, as we use recommender system only as a downstream task to evaluate the quality of HRQ multitokens in comparison to RQ multitokens. All experiments were ran on a device equiped in a single 16GB Nvidia-V100 card.

	RQ-VAE	**HRQ-VAE**
Variable	$\|x_s\|_2$	$\|\log_0^c(x_s^{\mathbb{P}_c})\|_2$
EV	0.7213	0.3251
Std. dev	0.2696	0.0664
CV	0.3738	0.2042

Table 4: Analysis of the norms for RQ-VAE and HRQ-VAE. We compare the euclidean norm of the low dimensional vector x_s to the euclidean norm of hyperbolic $x_s^{\mathbb{P}_c}$ after mapping to the tangent space with logarithmic map. For the comparison we use the Coefficient of Variation defined as $CV(X) = \frac{\sigma(X)}{\mu(X)}$. It is used to compare the variability of a random variables with different orders of magnitude. RQ-VAE has almost twice the CV of HRQ-VAE which supports our claim about the structure of their corresponding spanning trees.

D Structure of The Space

Suppose we have a set of points S and we want to find a point that minimizes average distance to all points from S. In Euclidean space this point will be the center of mass of points, a simple average of all points from S. However, in hyperbolic space, the point that minimizes the average hyperbolic distance (Eq. 1) will be a continuous analogue to the nearest common ancestor node of all the nodes.

This leads to a vastly different structures when these spaces are clustered hierarchically and, as a consequence, to a vastly different structures for spanning trees of the residual quantization. In the Euclidean space the points corresponding to the leafs will be splattered around the space with the qunatization tree cutting into the centers of respective subclusters. Meanwhile, in the hyperbolic case, the leafs will be mostly spread around with the cluster "centers" being closer to 0 than the cluster points. This behavior has been observed in the hyperbolic clustering [11]. We visualize the structural differences in Fig. 4.

This structure is beneficial for learning hierarchical relations for several reasons. Because the space is split radially most of the time the regions can have their own infinite part of the space, whereas in Euclidean division some regions are crammed close to center. As a consequence, the edges between regions are sharper than in the hyperbolic space, which might lead to poorer generalization. Finally, the structure imposed by the hierarchical euclidean quantization leads to strong utilization of the vector norms to select the cluster. On the other hand, hyperbolic quantization that leads to leafs being set the most outward in the spanning tree leaves the norm for the optimizer to choose, which can be an important benefit for gradient-based learning.

We argue that these structural difference of the hierarchical space of HRQ-VAE in comparison to space of RQ-VAE leads to the superior performance of HRQ-VAE in downstream tasks.

To quantitatively support this argument we inspect the norms of low-dimensional encoded representations x_s and $x_s^{\mathbb{P}_c}$. Specifically, we argue that the norms will vary less in the hyperbolic space. To make a fair comparison we compare Euclidean norms, so the hyperbolic $x_s^{\mathbb{P}_c}$ vector is first transformed to the tangent space with logarithmic map. Moreover, as the models differ in the average norm we look at the Coefficient of Variation as a measure of interest. The coefficient of variation is defined for positive variables as $CV(X) = \frac{\sigma(X)}{\mu(X)}$. The results are shown in Table 4 and confirm that the norms vary significantly more for the vectors to be quantized in the euclidean space.

E Limitations

Although HRQ and HRQ-VAE demonstrate better performance in the discussed tasks, they come with some limitations. The biggest limitation is the strong assumptions about the type of data. Currently, we limit the claim to the situation where the dataset has latent hierarchies, and at the same time, we are interested in discrete representations. This is a very specific situation. Extending the evaluation to domains of general application in which RQ-VAE succeeded, such as image or audio, would greatly increase the influence. However, the current version does not investigate performance in this direction.

F Reproducibility Statement

To ensure reproducibility, we report standard deviations in Table 2. In addition, we provide detailed descriptions of our models and implementation choices, including the exact prompt used to generate the movie title descriptions for the Movielens dataset, in the Appendix.

G LLM Usage

In preparing this manuscript, large language models (LLMs) were employed solely as writing assistants to polish the text. Their role was limited to improving readability, grammar, and style, without contributing to the development of ideas, the design of experiments, the analysis of results, or the generation of original content. All scientific contributions, including conceptualization, methodology, and interpretation, are entirely the authors' own.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

   Answer: [Yes]

   Justification: Both sections support the claim that for data with latent hierarchies representations generated by HRQ are better for downstreamt asks than discrete representations generated by RQ.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The papers list the assumption about the data it operates on multiple times in the paper, abstract and even in the title. Furthermore, Appendix E is dedicated for limitations.

   Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: There are no theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation details are discussed in the Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All the datasets that we use are public. However, the code used to generate the results is proprietary at this point.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Training setup is described in the Appendix B and the hyperparameters are described in the Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The table with results of the hierarchy discovery(Table 2) section includes statistical significance in the form of standard deviation of the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resource requirements are listed in the Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: While the work does not directly have a strong societal impact, we briefly discuss it in the Section 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release any new data or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites and lists used datasets in the Appendix B.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release any new data or models.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method does not use LLM in any way. However, for the MovieLens dataset in Section 4.2 we used LLM to generate movie descriptions. We cited the appropriate LLM and detailed the process of description generation in the Appendix B.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.