Breaking the Hourglass Phenomenon of Residual Quantization: Enhancing the Upper Bound of Generative Retrieval

Zhirui Kuai[1,†] and **Zuxu Chen**[2,†], and **Huimu Wang**[3,*] **and Mingming Li**[3,*]
Dadong Miao[3] and **Binbin Wang**[3] and **Xusong Chen**[3] and **Li Kuang**[1] and **Yuxing Han**[2,*]
Jiaxing Wang[3] and **Guoyu Tang**[3] and **Lin Liu**[3] and **Songlin Wang**[3] and **Jingwei Zhuo**[3]
[1]Central South University, School of Computer Science and Engineering, China
[2]Shenzhen International Graduate School, Tsinghua University, China
[3]JD.com, Beijing, China
kuaizhirui@csu.edu.cn and chen-zx22@mails.tsinghua.edu.cn and yuxinghan@sz.tsinghua.edu.cn
{wanghuimu1,limingming65,zhuojingwei}@jd.com

Abstract

Generative retrieval (GR) has emerged as a transformative paradigm in search and recommender systems, leveraging numeric-based identifier representations to enhance efficiency and generalization. Notably, methods like TIGER employing Residual Quantization-based Semantic Identifiers (RQ-SID), have shown significant promise in e-commerce scenarios by effectively managing item IDs. However, a critical issue termed the "**Hourglass**" phenomenon, occurs in RQ-SID, where intermediate codebook tokens become overly concentrated, hindering the full utilization of generative retrieval methods. This paper analyses and addresses this problem by identifying path sparsity and long-tailed distribution as the primary causes. Through comprehensive experiments and detailed ablation studies, we analyze the impact of these factors on codebook utilization and data distribution. Our findings reveal that the "Hourglass" phenomenon substantially impacts the performance of RQ-SID in generative retrieval. We propose effective solutions to mitigate this issue, thereby significantly enhancing the effectiveness of generative retrieval in real-world E-commerce applications.

1 Introduction

In recent years, GR has surfaced as a groundbreaking retrieval paradigm, marking significant advancements in search and recommendation environments including recommender systems (Rajput et al., 2024; Tan et al., 2024; Wang et al., 2024), search question answering (Liu et al., 2023; Qin et al., 2023), and E-commerce retrieval (Tay et al., 2022; Wang et al., 2022; Li et al., 2024). In this paradigm, target items are initially represented as identifiers (e.g., numbers, subwords, n-grams, token IDs, URLs, semantic codes). Subsequently, leveraging input information such as queries and user details, large models are employed to output the final items in an end-to-end manner. This approach not only enhances retrieval efficiency but also improves the model's generalization capability.

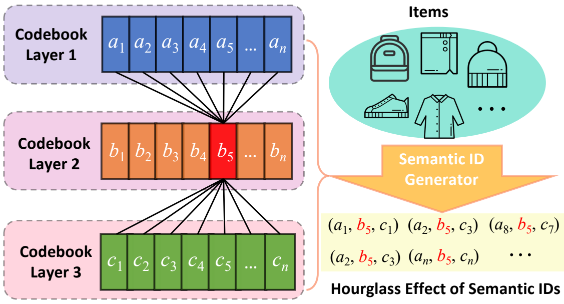

Codebook Layer 1
a_1 a_2 a_3 a_4 a_5 ... a_n
Codebook Layer 2
b_1 b_2 b_3 b_4 b_5 ... b_n
Codebook Layer 3
c_1 c_2 c_3 c_4 c_5 ... c_n
Items
Semantic ID Generator
(a_1, b_5, c_1) (a_2, b_5, c_3) (a_8, b_5, c_7)
(a_2, b_5, c_3) (a_n, b_5, c_n) ···
Hourglass Effect of Semantic IDs

Figure 1: The Hourglass Phenomenon of Semantic IDs

In generative retrieval, numeric-based identifier representation methods are widely adopted in the industry due to their simplicity, efficiency, and strong generalization, especially in long behavior sequence recommendations. These methods significantly reduce sequence lengths and accelerate the inference process. Notable methods include DSI (Tay et al., 2022), NCI (Wang et al., 2022), TIGER (Rajput et al., 2024), GDR (Yuan et al., 2024), and GenRet (Sun et al., 2024a). Among these, the TIGER method generates Semantic Identifiers (SID) through Residual Quantization (RQ) (Lee et al., 2022; Zeghidour et al., 2021), effectively capturing both semantic information and hierarchical structures. This approach is particularly advantageous in item-dominated e-commerce scenarios, where it accurately reflects the complex hierarchical relationships and semantic features inherent in e-commerce data, thereby significantly enhancing recommendation performance.

It is important to highlight that the performance upper bound of RQ-based methods critically depends on the generation of SID. However, we have identified a significant "hourglass" phenomenon in

[*]Corresponding Author. †Equal Contribution.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 677–685
November 12-16, 2024 ©2024 Association for Computational Linguistics

SID produced via RQ, as illustrated in Figure 1. Specifically, the codebook tokens in the intermediate layers are excessively concentrated, leading to a one-to-many and many-to-one mapping structure. This concentration results in path sparsity, where the matching paths for the item constitute a minimal fraction of the total path space and a long-tail distribution of intermediate layer tokens with a majority of SID concentrated in a few head tokens. This hourglass effect is particularly exacerbated in datasets with long-tail characteristics, which substantially constrains the representational capacity of GR methods. The underlying cause of this issue stems from the intrinsic nature of progressively quantizing high-dimensional vector residuals.

Furthermore, we analyzed the process of generating SID from residuals, demonstrating that sparsity and long-tail distributions are inevitable. To assess the general impact of SID on downstream GR tasks, we trained models of different scales (such as 0.8B, 7B) and types (Qwen1.5 (Bai et al., 2020), Baichuan2 (Yang et al., 2023), LLaMA2 (Touvron et al., 2023)) based on RQ-SID. Through a series of experiments, including altering the distribution of Semantic IDs by interacting with the first and second layers and swapping tokens between the first and second layers, we not only confirmed the existence of the Hourglass effect but also detailed its specific impact on model performance. This analysis provides a robust foundation for future model optimization.

To alleviate the hourglass effect, we propose two straightforward yet effective methods: the heuristic approach and the adaptive variable-length token strategy. The heuristic method involves directly removing the second layer, while curtailing the long-tail impact, it may lead to insufficient spatial capacity. The second method implements an adaptive token distribution adjustment to remove the top tokens from the second layer, thereby transforming the semantic ID into a variable-length structure. This strategy ensures that the overall distribution remains consistent while effectively mitigating the hourglass effect by selectively token removal. Extensive experimental results reveal that although both methods are straightforward, they successfully alleviate the impact of the hourglass effect to varying extents. Notably, the adaptive variable-length token strategy method emerges as the most effective.

The contributions of this paper can be summarized as follows:

- To our knowledge, this is the first study to systematically investigate the deficiencies of residual quantization-based semantic identifiers in generative retrieval, specifically identifying the "hourglass" phenomenon where intermediate layer codebook tokens are overly concentrated.

- We conduct thorough experiments and ablation studies that reveal path sparsity and long-tail distributions as the primary causes of the "hourglass" effect, limiting the representation and performance capabilities of generative models.

- We propose and validate a novel method to alleviate the **"hourglass"** effect, which significantly enhances model performance by improving codebook utilization and addressing token long-tail distributions.

2 Related Works

Recent advancements in generative retrieval have significantly influenced various domains, such as recommendation systems, search question answering, and E-commerce retrieval. This paradigm shift, as evidenced by works like (Tay et al., 2022; Wang et al., 2022, 2024; Li et al., 2024), involves representing target items using identifiers such as numbers, sub-words, and semantic codes.

Within the industry, numeric-based identifier representation methods are prevalent due to their simplicity and efficiency. These methods, including DSI (Tay et al., 2022), NCI (Wang et al., 2022), TIGER (Rajput et al., 2024), GDR (Yuan et al., 2024), and GenRet (Sun et al., 2024b), are particularly effective in long behavior sequence recommendations. They reduce sequence lengths and accelerate inference processes. Notably, the TIGER method employs RQ(Lee et al., 2022; Zeghidour et al., 2021) to generate SID, capturing semantic information and hierarchical structures. This is especially beneficial in item-dominated e-commerce contexts, where complex hierarchical relationships and semantic features are crucial for enhancing recommendation performance. However, the performance upper limit of RQ-based methods largely depends on the generation of SID, which is also the central focus of analysis and discussion in this paper.

3 Preliminary

3.1 Residual Quantization

Residual-quantized is a multi-level vector quantizer that applies quantization on residuals to generate a tuple of codewords (i.e., Semantic IDs). Residual-quantized variational AutoEncoder (RQ-VAE) (Rajput et al., 2024; Lee et al., 2022; Zeghidour et al., 2021) is jointly trained by updating the quantization codebook and the encoder-decoder reconstruction parameters.

Support that there is a vector $\mathbf{x} \in \mathcal{R}^D$, we aim to quantize it using L codebooks (L layer) of M elements each, where codebook could be denoted as $\mathbf{C} \in \mathcal{R}^{L \times M \times D}$, D is the dimension of vector. When $l = 1$, the initial residual is simply defined as $\mathbf{r}_1 = \mathbf{x}$. Then, $\mathbf{r}_l$ is quantized by mapping it to the nearest embedding from that layer's codebook $\mathbf{C}_l \in \mathcal{R}^{M \times D}$. The index of the closest embedding at this layer could be computed as follows:

$$c_l = \arg\min_{m \in M} \parallel \mathbf{r}_l - \mathbf{C}_{l,m} \parallel_2^2 \tag{1}$$

where c_l represents the l-th codeword(semantic ID). Note that, at the l-th layer (l > 1), the residual is:

$$\mathbf{r}_l = \mathbf{r}_{l-1} - \mathbf{C}_{l,c_{l-1}} \tag{2}$$

The above process is repeated recursively L times to get a tuple of L codewords that represent the Semantic ID for the given $\mathbf{x}$, denoted as $(c_1, c_2, \ldots, c_L)$.

To reconstruct the raw vector, we sum the corresponding codebook elements as:

$$\hat{\mathbf{x}} = \sum_{l=0}^{L} \mathbf{C}_{l,c_l} \tag{3}$$

This method could approximate the raw vector from a coarse-to-fine granularity by the norm of residuals decreasing, i.e., $\|\hat{\mathbf{x}} - \mathbf{x}\|^2 < \epsilon, \epsilon \ll 0.001$.

3.2 Generative Retrieval

Generative retrieval (Wang et al., 2022; Tay et al., 2022; Tang et al., 2023; Bevilacqua et al., 2022; Zhou et al., 2023), has been proposed in the recommendation field, search field and question-answer field. These models advocate generating identifiers of target passages/items directly through the autoregressive language models.

In personalized search scenarios, a core task is to provide the most relevant candidates that the user is likely to purchase based on their given query and historical interaction behaviors. In this paper, we re-frame this task as a Next Token Prediction (NTP) problem utilizing LLM and Semantic ID. Specifically, given user u, query q, and the user's historical item sequence, we first convert the sequence into a Semantic ID sequence, denoted as $Seq :=$ $\left\{\underbrace{(c_{1,1}, \cdot, c_{1,M})}_{item_1}; \underbrace{(c_{2,1}, \cdot, c_{2,M})}_{item_2}; \ldots; \underbrace{(c_{t,1}, \cdot, c_{t,M})}_{item_t}\right\}$ where $(c_{i,1}, \;\cdot,\; c_{i,M})$ denotes the M-length Semantic ID for $item_i$. The LLM is then trained to predict the Semantic ID of $item_{t+1}$, represented as $(c_{t+1,1}, \cdot, c_{t+1,M})$. The generation objective could be formulated as,

$$\mathcal{L}_{sft} = -\sum_{i}^{M} \log p_\theta(i|q, u, Seq, I_{<i}) \tag{4}$$

where $I_{<i} = \{c_{t+1,1}, \cdots, c_{t+1,i}\}$, p_θ is the supervised fine-tuning (SFT) model.

4 Problem of GR based on RQ

4.1 Hourglass Phenomenon

To generate the semantic IDs used RQ, we first leverage the query-item data from billions of search logs within the company to train dual-tower models such as DSSM and BERT (Li et al., 2020; Fan et al., 2019; Karpukhin et al., 2020; Li et al., 2023a; Qiu et al., 2022). Subsequently, we obtain the embeddings for hundreds of millions of items using the item tower. Finally, we employ RQ to generate semantic IDs for all items.

Upon the successful generation of semantic IDs, we proceed to aggregate and compute the three-layer distribution maps for all items. As illustrated in Figure 2, it is evident that the second layer of the Semantic ID architecture is concentrated with a substantial number of routing nodes. The overall distribution of the three-layer code exhibits an hourglass phenomenon.

To investigate the generalizability of this phenomenon, we conducted multiple visualization experiments under various parameter combinations, e.g., code table size and number of layers. As shown in Figure 6 in the appendix, the results indicate that the hourglass effect is highly pronounced, and the path distribution among the tokens across the three layers of the code table is relatively sparse.

Additionally, based on the aforementioned experiments, we conducted statistical analyses of the

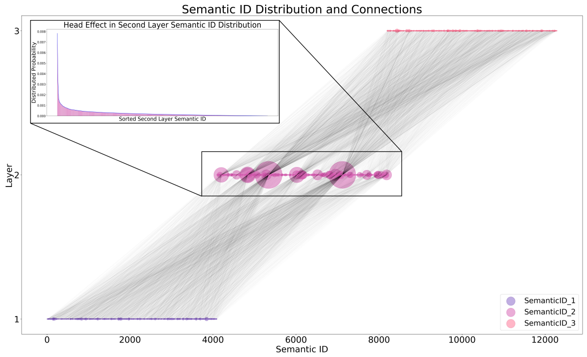


Figure 2: Distribution and Connections of Semantic IDs

token distribution in the second layer using three metrics: entropy (Shannon, 1948), Gini coefficient (Yitzhaki, 1979), and standard deviation (Pal et al., 2019), as shown in the Figure 3. The results indicate that the token distribution in the second layer exhibits low entropy, high Gini coefficient, and large standard deviation, suggesting that the distribution is highly skewed and exhibits a long-tail effect.

Overall, this hourglass phenomenon is statistically evidenced in the code table by path sparsity and a long-tail distribution of tokens. 1) Path sparsity, resulting from the Semantic ID structure, leads to low code table utilization. 2) The long-tail distribution indicates that in the intermediate layer, a predominant number of routes converge on a single token.

4.2 Analysis of Residual Quantization

To explore the causes of the hourglass phenomenon, we will conduct an in-depth analysis and discussion based on the operating mechanism of the RQ. Without loss of generality, we consider two distributions of raw embedding: un-uniform and uniform, denoted as $\mathbf{X} = \{\mathbf{x} | \mathbf{x} \in \mathbf{X}\} \in \mathcal{R}^{N \times M}$, N is the size of the dataset. Now, we use the RQ to produce the semantic ID for $\mathbf{X}$.

In the first layer, all candidate's points are divided into M different cluster buckets. Each cluster bucket contains n_m data points and has a radius of e_m. For the uniform distribution, $n_m = N/M$, and $e_1 = e_2 = \ldots = e_m$. Therefore, the in-degree of all tokens in this layer are equal.

In the second layer, all input embedding is $\mathbf{X}^{'}$, the residual of the first layer. Due to the difference in the magnitude of residual values, the input distribution in this layer is non-uniform. There are a large number of points with smaller magnitudes (points near the cluster centers in each bucket from the previous layer), which is equal to $n_m * M * \rho = N * \rho$, ρ is the ratio. At the same time, there are small points with larger magnitudes, which are considered as outliers. To reduce the clustering loss, the clustering process in this layer focuses on these outliers. As a result, the points with smaller magnitudes will occupy fewer cluster centers, while the outliers will either occupy individual cluster centers or multiple cluster centers. Therefore, this layer's semantic IDs will form large routing nodes, exhibiting a long-tail phenomenon, which is also demonstrated in the second layer of Figure 4.

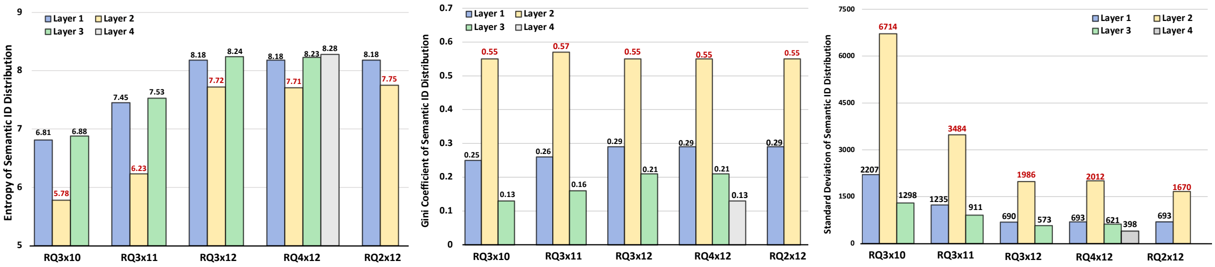
In the third layer, all input point magnitudes become consistent again and relatively uniform. Therefore, the code distribution in this layer is similar to the first layer, with a uniform distribution. As a result, it can be directly observed that the large routing nodes from the second layer diverge into multiple smaller nodes in the third layer, creating a one-to-many situation, as shown in the third layer of Figure 4. At the same time, if the residuals in the second layer tend towards zero, there will still be some clustering in the third layer. However, since all magnitudes are very small at this point, the impact of the clustering effect is limited.

As we continue to iterate through the layers, this phenomenon of non-uniform distribution and long-tail clustering followed by uniform distribution will alternate. However, as the number of layers increases, the residuals become smaller (refer to layer 4 of Figure 4), and the clustering effect weakens, so it can be ignored. Ultimately, this leads to the formation of an hourglass-like structure, where the input data is first compressed into a smaller number of clusters, then expands back out into a larger number of clusters, and finally converges to a uniform distribution. Upon the completion of SID construction, the influence of the RQ quantization method, coupled with the dominance of head tokens in the intermediate layer, naturally leads to the sparsity of paths.

Similarly, for the un-uniform distribution, such as long-tail distribution, the residual distribution becomes even more uneven, resulting in a more severe phenomenon.

4.3 Impact on the GR

In the above section, we have discussed the long-tail distribution in the second layer of Semantic ID, indicating a one-to-many and many-to-one structure. We argue that this phenomenon significantly



(a) Entropy

(b) Gini Coefficient

(c) Standard Deviation

Figure 3: Illustrating the Hourglass Phenomenon in Semantic IDs with Different Statistical Metrics

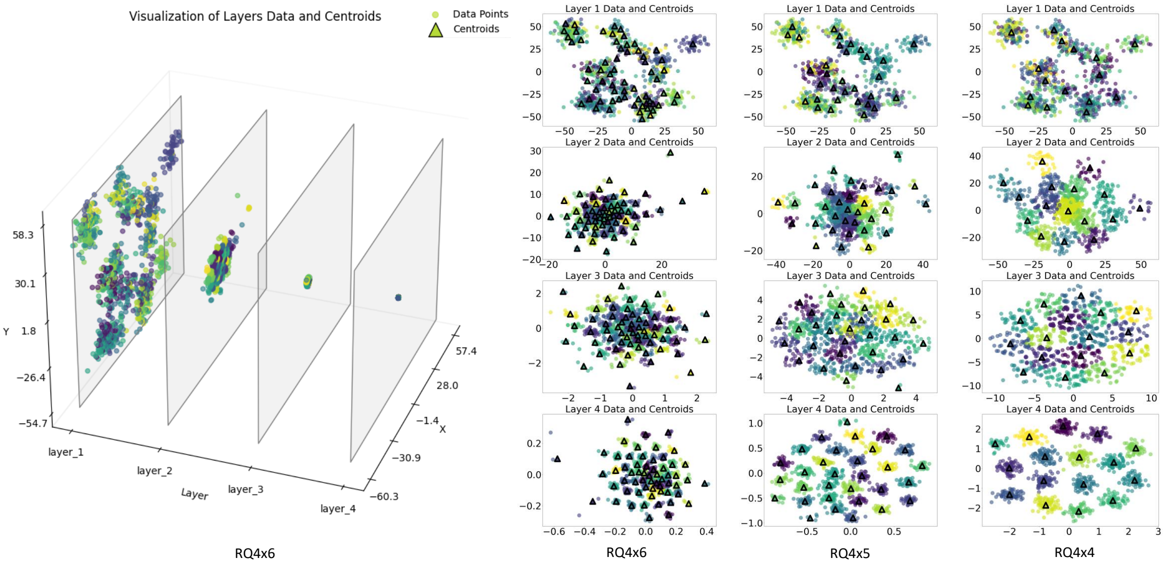


Figure 4: Hierarchical Residual Reduction and Dimensional Analysis Across Layers

impacts the generation of downstream tasks, especially for generative retrieval task.

To measure this impact, we conducted various experiments. First, we altered the distribution of Semantic ID by interacting with the first and second layers. On this basis, we only predicted the tokens of the second and third layers while keeping the tokens of the first layer fixed.

During the evaluation process, we divide the test set into two groups according to the distribution of second-layer tokens: the head token test set and the tail token test set. As shown in Table 1, the performance of the head token test set significantly improved, whereas the performance of the tail token test set was notably poorer. This performance disparity can be attributed to the previously analyzed path sparsity and long-tail distribution of tokens, leading to biased results. This phenomenon has been observed across models of different scales (LLaMA2, Baichuan2, and Qwen1.5) and different parameters of RQ, highlighting the widespread impact of long-tail token distribution and path sparsity on model performance.

To further investigate the impact of the hourglass phenomenon on model performance, we conduct three critical experiments: 1) give the first token directly as input, 2) exchange the tokens of the first and second layers, and 3) give the first token of the swapped sequence as input.

Swapping only the first and second layers results in a significant long-tail distribution in the first layer, and the issue of the long-tail distribution remains unresolved. As shown in Table 1, the changes in metrics are minimal. However, if we swap the layers and provide the 1st token, the task shifts to predicting the 2nd and 3rd layer. This simplifies the task since the true first-layer is given, mitigating the long-tail distribution's impact and significantly improving performance. Conversely, if we don't swap the layers and still provide the first token, the second-layer SID maintains its long-tail distribution. These results shown in Table 1 are higher than the baseline but worse than when the first token is given after swapping.

Table 1: The performance of generative retrieval on E-commerce datasets with RQ3x12, i.e., $L = 3, M = 2^{12}$. The head/tail token denotes the head/tail semantic ID in the second layer, respectively.

Method	Recall@1	Recall@3	Recall@5	Recall@10	Recall@30	Recall@50
LLaMA2-0.8B*	0.2480	0.4080	0.4990	0.590	0.7080	0.7480
Head Token	0.3617	0.5745	0.6894	0.7745	0.8894	0.9191
Tail Token	0.2131	0.3569	0.4405	0.5333	0.6523	0.6954
Qwen1.5-7B	0.2770	0.4720	0.5700	0.6600	0.7700	0.7930
Head Token	0.3450	0.5970	0.7040	0.8020	0.8960	0.9120
Tail Token	0.2470	0.4160	0.5100	0.5950	0.7190	0.7470
Baichuan2-7B	0.2730	0.4900	0.5900	0.6760	0.7670	0.8040
Head Token	0.3440	0.6000	0.7200	0.8140	0.9020	0. 9210
Tail Token	0.2480	0.4360	0.5250	0.6110	0.7180	0.7540
Given Layer 1*	0.340	0.497	0.567	0.632	0.722	0.756
Exchange Layer 1&2*	0.2390	0.4190	0.5100	0.6070	0.7150	0.7540
+ Given Layer 1*	**0.6600**	**0.8240**	**0.8650**	**0.8910**	**0.9160**	**0.9190**

* These experiments are based on the LLaMA2-0.8B model, which adopts the LLaMA2 structure and SFT on Chinese corpora.

These approaches aim to mitigate the effects of the long-tail distribution, and results verify a significant improvement. This finding indicates that the hourglass phenomenon has a substantial negative impact on model performance. Through the above experiments, we not only confirmed the existence of the hourglass effect but also elucidated its specific impact on model performance, thereby providing a robust basis for future optimization.

5 Methods and Experiments

To alleviate the hourglass effect, we propose two simple yet effective methods.

5.1 Heuristic Method

One heuristic approach is to directly remove the second layer, eliminating the impact of the long tail. However, it can lead to insufficient spatial capacity, i.e., $M^L \rightarrow M^{L-1}$. Note that, here needs first to generate an L-layer SID and then remove the second layer, which differs from directly generating a two-layer SID, where large routing nodes may still exist.

5.2 Variable Length of SID

Another simple method is to adaptively remove the top tokens of the second layer, making the semantic ID a variable-length structure. Here, a top@K strategy is used, with p as a threshold. This approach ensures that the distribution remains unchanged while reducing the impact of the hourglass effect selectively. What's more, the spatial capacity is sufficient, i.e., $M^L \rightarrow M^L + K(M^{L-2} - M^{L-1})$. Note that the choice of top-k depends on the actual data distribution, so ablation testing is necessary. In summary, while this method is simple and efficient, it is not optimal and can only alleviate, but not completely resolve, the hourglass phenomenon.

5.3 Experiments

To further validate the effectiveness of the method, experiments are conducted on the LLaMA model and on a real large-scale e-commerce platform. We randomly selected hundreds of millions of training samples from nearly sixty days of data, with a user base reaching tens of millions and a product catalog of two hundred million items (Li et al., 2023b; Wang et al., 2023). The average length of user behavior sequences is 100.

Results indicate that by applying the adaptive token removal strategy, the performance of the model is improved while maintaining a similar computational cost compared to the base model, and several objective optimizations, such as Focal Loss (Lin et al., 2017) and Mile Loss (Su et al., 2024).

Specifically, experimental results showed that the model with top@400 token removal outperforms the baseline model in terms of most evaluation metrics. This suggests that the method effectively reduces the impact of the long-tail effect. As the number of tokens removed increases, the performance improvement of the model encoun-

Table 2: The performance of generative retrieval on E-commerce based on RQ3x12.

Method	Recall@1	Recall@3	Recall@5	Recall@10	Recall@30	Recall@50
LLaMA2-0.8B	0.2480	0.4080	0.4990	0.590	0.7080	0.7480
Focal Loss (Lin et al., 2017)	0.2310	0.4270	0.5050	0.6110	0.7300	0.7640
Mile Loss (Su et al., 2024)	0.2590	0.4380	0.5110	0.6090	0.7250	0.7600
Remove 2-th layer	0.3090	0.4310	0.4970	0.5640	0.6580	0.7020
Remove 2-th layer top@20	0.2500	0.4270	0.5130	0.6120	0.7250	0.7580
Remove 2-th layer top@200	0.3190	0.4740	0.5600	0.6550	0.7450	0.7760
Remove 2-th layer top@400	**0.3340**	0.5070	**0.5950**	**0.6800**	**0.7760**	0.7990
Remove 2-th layer top@600	0.3320	**0.5080**	0.5850	0.6720	0.7700	**0.8010**

ters a bottleneck. Especially when all tokens are removed, this limitation is particularly pronounced, which is presumed to be due to the absence of long-tail tokens, resulting in a loss of recall. At the same time, removing the second layer directly will cause one SID to correspond to multiple items.

This fine-grained analysis provides strong evidence for the effectiveness of the proposed method, which selectively removes less important tokens while retaining the most informative ones, leading to improved model performance even when a substantial amount of data is removed.

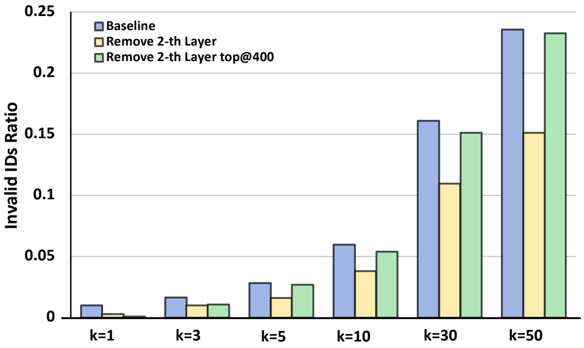


Figure 5: Invalid IDs Ratio when generating Semantic IDs using Beam Search for various values of k

5.4 Valid Ratio

During the autoregressive decoding process, as the model decodes the next token of the target SID, it may predict invalid SIDs, SIDs that are not in the SID's vocabulary, or do not correspond to any item in the full dataset. Therefore, we have calculated the proportion of invalid SIDs on the LLaMA2-0.8B model with RQ3x12. As shown in Figure 5, we can see the base model, the invalid ratio of the proposed method is lower than the base model, indicating that the higher-quality generation items with a lower ratio of hallucination. Furthermore, when the number of recalls is less than 10, the invalid ratio is below 5%. Thus, the effectiveness of generation is to meet practical needs. In other situations, where a higher number of recalls is required (k=50), the invalid ratio is higher. Across various sizes of base models and different RQ parameter settings, the results tend to converge on the same conclusion. Therefore, it is necessary to employ the retrieval augmented generation (RAG) (Lewis et al., 2020; Ding et al., 2024) for processing during the inference process, such as prefix-tree (Beurer-Kellner et al., 2024), and FM_Index (Herruzo et al., 2021).

6 Conclusion

This study systematically explores the limitations of RQ-SID in GR, particularly identifying the "hourglass" phenomenon in the intermediate layer where codebook tokens are overly concentrated, leading to path sparsity and long-tail distribution. Through extensive experiments and ablation studies, we have demonstrated the existence of this phenomenon and conducted an in-depth analysis attributing its root cause to the characteristics of residuals. To alleviate this issue, we propose two methods: a heuristic approach that removes the second layer and a variable-length token strategy that adaptively adjusts token distribution. Experimental results show both methods effectively mitigate the bottleneck effect, with the adaptive token distribution adjustment yielding the best results. While this method is simple and efficient, it is not optimal and can only alleviate, but not completely resolve, the hourglass phenomenon. To the best of our knowledge, this is the first systematic exploration of the deficiencies of RQ-SID in GR, providing a solid foundation for future model optimizations and significantly enhancing model performance by improving codebook utilization.

Acknowledgments

This work has been supported by the National Key R&D Program of China under grant No. 2022YFF0902500, the National Natural Science Foundation of China under grant No.62472447, the Science and Technology Innovation Program of Hunan Province under grant No.2023RC1023, the Hunan Provincial Natural Science Foundation of China under grant No.2024JK2006, and Shenzhen Startup Funding No. QD2023014C.

References

Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. Sparterm: Learning term-based sparse representation for fast text retrieval. *arXiv preprint arXiv:2010.00768*.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2024. Guiding llms the right way: Fast, non-invasive constrained generation. *arXiv preprint arXiv:2403.06988*.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. *Advances in Neural Information Processing Systems*, 35:31668–31683.

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. 2024. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. *arXiv preprint arXiv:2402.10612*.

Miao Fan, Jiacheng Guo, Shuai Zhu, Shuo Miao, Mingming Sun, and Ping Li. 2019. Mobius: towards the next generation of query-ad matching in baidu's sponsored search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2509–2517.

Jose M Herruzo, Ivan Fernandez, Sonia González-Navarro, and Oscar Plata. 2021. Enabling fast and energy-efficient fm-index exact matching using processing-near-memory. *The Journal of Supercomputing*, 77(9):10226–10251.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Mingming Li, Huimu Wang, Zuxu Chen, Guangtao Nie, Yiming Qiu, Binbin Wang, Guoyu Tang, Lin Liu, and Jingwei Zhuo. 2024. Generative retrieval with preference optimization for e-commerce search. *arXiv preprint arXiv:2407.19829*.

Mingming Li, Chunyuan Yuan, Binbin Wang, Jingwei Zhuo, Songlin Wang, Lin Liu, and Sulong Xu. 2023a. Learning query-aware embedding index for improving e-commerce dense retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 3265–3269. ACM.

Mingming Li, Chunyuan Yuan, Huimu Wang, Peng Wang, Jingwei Zhuo, Binbin Wang, Lin Liu, and Sulong Xu. 2023b. Adaptive hyper-parameter learning for deep semantic retrieval. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 775–782.

Mingming Li, Shuai Zhang, Fuqing Zhu, Wanhui Qian, Liangjun Zang, Jizhong Han, and Songlin Hu. 2020. Symmetric metric learning with adaptive margin for recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 4634–4641.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. Webglm: Towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4549–4560.

Manoranjan Pal, Premananda Bharati, Manoranjan Pal, and Premananda Bharati. 2019. Introduction to correlation and linear regression analysis. *Applications of regression techniques*, pages 1–18.

Yujia Qin, Zihan Cai, Dian Jin, Lan Yan, Shihao Liang, Kunlun Zhu, Yankai Lin, Xu Han, Ning Ding, Huadong Wang, et al. 2023. Webcpm: Interactive web search for chinese long-form question answering. *arXiv preprint arXiv:2305.06849*.

Yiming Qiu, Chenyu Zhao, Han Zhang, Jingwei Zhuo, Tianhao Li, Xiaowei Zhang, Songlin Wang, Sulong Xu, Bo Long, and Wen-Yun Yang. 2022. Pre-training tasks for user intent detection and embedding retrieval in e-commerce search. In *Proceedings of the*

31st ACM International Conference on Information & Knowledge Management, pages 4424–4428.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2024. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Zhenpeng Su, Zijia Lin, Baixue Baixue, Hui Chen, Songlin Hu, Wei Zhou, Guiguang Ding, and W Xing. 2024. Mile loss: a new loss for mitigating the bias of learning difficulties in generative language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 250–262.

Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten Rijke, and Zhaochun Ren. 2024a. Learning to tokenize for generative retrieval. *Advances in Neural Information Processing Systems*, 36.

Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten Rijke, and Zhaochun Ren. 2024b. Learning to tokenize for generative retrieval. *Advances in Neural Information Processing Systems*, 36.

Juntao Tan, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Zelong Li, and Yongfeng Zhang. 2024. Towards llm-recsys alignment with textual id learning. *arXiv preprint arXiv:2403.19021*.

Yubao Tang, Ruqing Zhang, Jiafeng Guo, Jiangui Chen, Zuowei Zhu, Shuaiqiang Wang, Dawei Yin, and Xueqi Cheng. 2023. Semantic-enhanced differentiable search index inspired by learning strategies. *arXiv preprint arXiv:2305.15115*.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Binbin Wang, Mingming Li, Zhixiong Zeng, Jingwei Zhuo, Songlin Wang, Sulong Xu, Bo Long, and Weipeng Yan. 2023. Learning multi-stage multi-grained semantic embeddings for e-commerce search. In *Companion Proceedings of the ACM Web Conference 2023*, pages 411–415.

Yidan Wang, Zhaochun Ren, Weiwei Sun, Jiyuan Yang, Zhixiang Liang, Xin Chen, Ruobing Xie, Su Yan, Xu Zhang, Pengjie Ren, et al. 2024. Enhanced generative recommendation via content and collaboration integration. *arXiv preprint arXiv:2403.18480*.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35:25600–25614.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

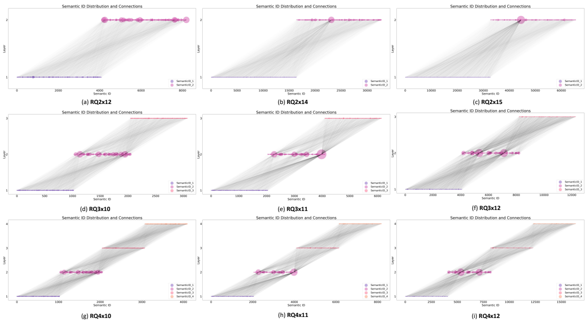
Shlomo Yitzhaki. 1979. Relative deprivation and the gini coefficient. *The quarterly journal of economics*, 93(2):321–324.

Peiwen Yuan, Xinglin Wang, Shaoxiong Feng, Boyuan Pan, Yiwei Li, Heda Wang, Xupeng Miao, and Kan Li. 2024. Generative dense retrieval: Memory can be a burden. *arXiv preprint arXiv:2401.10487*.

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

Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2023. Enhancing generative retrieval with reinforcement learning from relevance feedback. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

A Distribution and Connections of Different RQ-Semantic IDs



(a) **RQ2x12** (b) **RQ2x14** (c) **RQ2x15**

(d) **RQ3x10** (e) **RQ3x11** (f) **RQ3x12**

(g) **RQ4x10** (h) **RQ4x11** (i) **RQ4x12**

Figure 6: Distribution and Connections of Different RQ-Semantic IDs