The Interpretability of Codebooks in Model-Based Reinforcement Learning is Limited

Kenneth Eaton[1], Jonathan Balloch[2], Julia Kim[2], Mark Riedl[2]
[1]Georgia Tech Research Institute, [2]Georgia Institute of Technology

Abstract

Interpretability of deep reinforcement learning systems could assist operators with understanding how they interact with their environment. Vector quantization methods—also called *codebook* methods—discretize a neural network's latent space that is often suggested to yield emergent interpretability. We investigate whether vector quantization in fact provides interpretability in model-based reinforcement learning. Our experiments, conducted in the reinforcement learning environment Crafter, show that the codes of vector quantization models are inconsistent, have no guarantee of uniqueness, and have a limited impact on concept disentanglement, all of which are necessary traits for interpretability. We share insights on why vector quantization may be fundamentally insufficient for model interpretability.

1 Introduction

Although deep neural networks have advanced the performance of reinforcement learning (RL) agents, these "black box" models give little insight as to the agent's decision making and learned transition models. Interpretability of RL agents is important in high-stakes domains that demand human trust, such as autonomous vehicles and infrastructure. Interpretability is also crucial to analyze behavior and develop adaptations when trained agents make mistakes or experience novel changes to their environment.

There has been significant recent interest in vector quantization (VQ) in neural networks, largely driven by the work of Van Den Oord et al. (2017); the authors show that VQ latent space discretization in generative models helps regularize the latent space, thereby improving performance. Although not claimed by Van Den Oord et al. (2017), several subsequent works suggest that the latent disentanglement caused by VQ enables greater semantic interpretability of neural models (Hsu et al., 2017; Tamkin et al., 2023; Aloufi et al., 2020).

We examine whether VQ within model-based reinforcement learning (MBRL) captures semantic information about entities in the environment. Specifically, we use the IRIS (Micheli et al., 2022) MBRL agent, which quantizes the latent vector encoding of the world state and uses a transformer to model transition dynamics. Grad-CAM (Selvaraju et al., 2017) is used to perform qualitative and quantitative analysis to determine whether codes are consistent and represent semantically groundable entities, such as objects, across diverse inputs.

We find VQ lacks the necessary constraints to enforce that disentangled codes correspond to semantic entities; codes are not consistent enough to be grounded to a semantic concept. Based on our findings, we disprove the intuition that VQ alone is sufficient for interpretability and hypothesize that latent semantic alignment is needed alongside discretization to make latent spaces generally interpretable.

2 Background and Related Work

In MBRL, the agent learns a model to approximate the transition dynamics of the environment (Sutton & Barto, 1998). An agent can benefit from having a good predictive model of the future

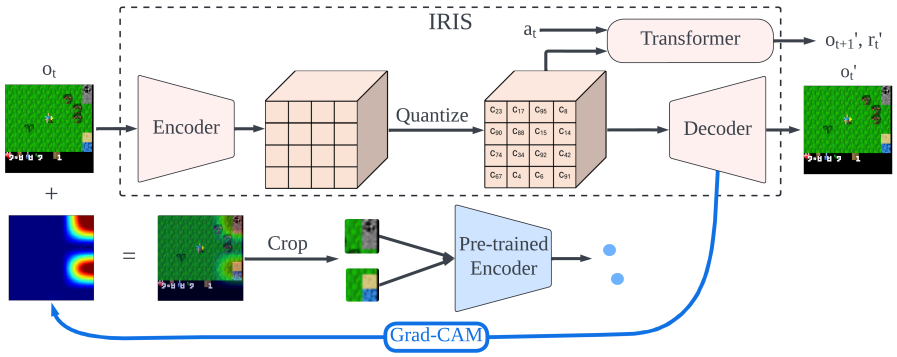


Figure 1: IRIS transition model architecture and our evaluation process. Heatmaps from Grad-CAM crop the input image to focus regions, which are then embedded by a pre-trained encoder.

state (Werbos, 1987), commonly dubbed a *World Model* (Ha & Schmidhuber, 2018; Hafner et al., 2020). IRIS (Micheli et al., 2022) is a MBRL agent that uses VQ to replace the latent state vectors of an encoded observation with the most similar codes, which are input to a decoder and a transformer learning the World Model. Since the codebook sits between the encoder and decoder, individual codes have the potential to represent grounded entities (Khazatsky et al., 2021).

VQ is a data compression technique that models a latent distribution as a discrete set of vectors or "codes." Formally, a VQ function Q can be defined as: $Q : R^d \rightarrow C$, where the "codebook" $C = \{c_i | i \in I, c_i \in R^d\}$ is composed of k vectors referred to as *codes*. For a given input x, Q assigns a code c_i that minimizes the MSE between the vectors. In deep neural networks, VQ is most often used in the latent space of generative models, replacing the encoder output with learned codes (Van Den Oord et al., 2017; Esser et al., 2021; Ramesh et al., 2021). VQ is also used in the latent space of RL models to constrain and regularize them, which improves accuracy, robustness to noise, and sample efficiency (Zhang et al., 2022; Robine et al., 2020).

Several works have examined and suggested the emergent interpretability of VQ methods. Chen et al. (2022); Zhang et al. (2022) demonstrate that the entire latent embedding created from a combination of codes is capable of capturing consistent states. There are also works that suggest the interpretability of some specific codes using qualitative assessments such as latent traversals (Aloufi et al., 2020; Shu et al., 2020; Desticourt et al., 2022; Zou et al., 2023; Tamkin et al., 2023; Wallingford et al., 2023; Shao et al., 2023; Hsu et al., 2024). Despite evidence from works such as Locatello et al. (2019), the suggestion of VQ interpretability comes from an unsubstantiated connection between latent *disentanglement* and latent semantics (Guo et al., 2020; Khazatsky et al., 2021) or interpretability (Hsu et al., 2017; Liu et al., 2022; Zou et al., 2022; Klein et al., 2022). These methods rarely quantify the interpretability provided by VQ codes, and we found no prior work that studies VQ interpretability in MBRL.

This work is similarly motivated to the parallel research area of *mechanistic interpretability* Nanda (2022); Elhage et al. (2022) (MI), which studies interpreting neural network behavior based on an interpretation of the network's internal structures Olah et al. (2020). This stands in contrast to "black-box" Molnar (2022) interpretability approaches, such as saliency maps of inputs from outputs Adebayo et al. (2018); Jain & Wallace (2019); Wiegreffe & Pinter (2019), which attempt to explain relationships between the inputs and outputs without considering the network's internal mechanisms. Interpreting neural networks using codebooks falls in between these methods (in what is sometimes called "white-box" Molnar (2022); Räuker et al. (2023) interpretability), where (as in MI) the internal structures of the neural networks are constrained or investigated to establish what activations of the network are representing, while using tools like saliency maps to convey feature importance. By taking a white-box approach to interpreting codes while using MI concepts such as superposition Elhage et al. (2022), our analysis balances the theoretical robustness of MI with the ability to interpret almost any network without the need to understand all its internal structures.

Table 1: Example heatmaps for codes.

Code 304								
Code 434								
Code 381								

3 Experiments in VQ Interpretability

Experimental Approach. In our work, we use the state-of-the-art MBRL model IRIS (Micheli et al., 2022) featuring a codebook with 512 codes. We make no modifications to the IRIS architecture. To evaluate interpretability, we apply Grad-CAM to the code inputs of the decoder. Grad-CAM scores importance of input image pixels based on the activations and gradients of a selected weight vector or layer. We apply this process individually for each code in the model by masking out the other codes' values. The end result is a heatmap for each code that overlays the input image with the regions that were most influential in that code being selected.

To measure the quantitative properties of the learned codes, we evaluate the code consistency over agent observations and episodes by looking at the areas of high Grad-CAM activation for the same code. We split images into connected components of high activation and filter out those with an area smaller than a threshold. The remaining connected components are used to crop the original images to the bounding box region they encapsulate. We use a frozen pre-trained model to compare the cropped inputs and gauge their semantic similarity.

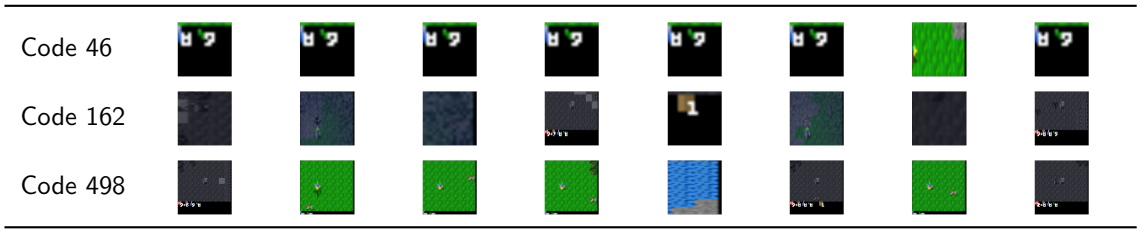
We conducted our experiments in Crafter (Hafner, 2021), an open-world survival game designed for RL with similar dynamics to Minecraft. The model and corresponding policy was first trained to convergence. Then we collected a dataset of 127,434 $(o_t, s_t, a_t, s_{t+1}, o_{t+1})$ transitions of the agent acting in the environment over 714 episodes, where o is the input observation, a is the selected action, and s are the latent states. We applied our interpretation approach to investigate different codes correspondance to the inputs on newly collected runs, and saved the Grad-CAM activation heatmaps for every code the model selected in each observation.

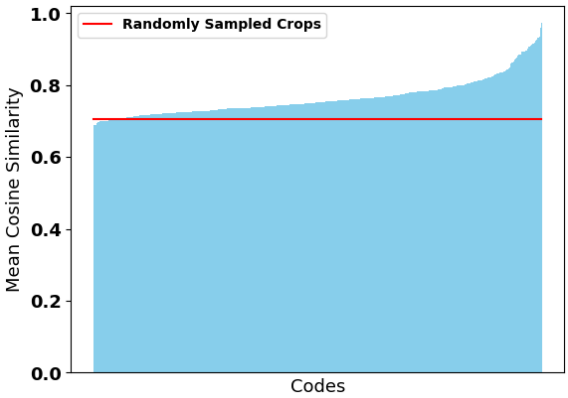
We found that 90% of the heatmaps contained all zero values. Since all the values being zero is not useful for our interpretability evaluation, those samples were dropped. 477 of the 512 codes had non-zero heatmaps, equating to 205,231 heatmaps for the final dataset.

Results. Table 1 displays eight heatmaps overlaid on their observation image, randomly sampled for each code, from the most frequently occurring code in our dataset, the tenth most frequently occurring, and the median occurring. We chose these codes based on their frequency to avoid biasing the results and capture diverse regions of the frequency distribution, as discussed further in Appendix A. In general, the heatmaps from a code focus on the same regions of the image, especially for the more frequently occurring codes. To examine the content the codes focus on, in Table 2 we show the crops resulting from the heatmaps. To display results for more codes while still focusing on diverse frequencies, we show crops for the second most frequent code, eleventh most frequent, and the next most frequent code after the median code used earlier. In the crops, code 46 is consistently focused on numbers indicating agent inventory, while the other two have several similar crops but no clear, singular focus. Therefore, we conclude codes did not consistently associate with objects or concepts. Additional examples of crops are presented in Appendix B.

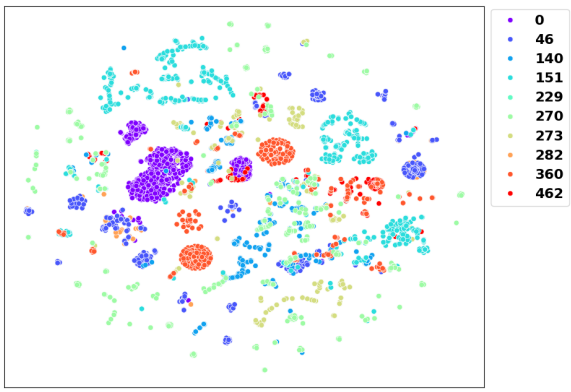
To quantify code interpretability, we compared the embeddings for each crop from the last hidden layer of a pre-trained, frozen ResNet50 (He et al., 2016) model trained on ImageNet-1k (Russakovsky et al., 2015). Given these crop embeddings, we calculated the mean embedding of each code and

Table 2: Example crops from codes.





(a) Mean cosine similarity of cropped images



(b) t-SNE of codes with high cosine similarity

Figure 2: The data from our GradCAM experiments show that VQ codes are inconsistent quantitatively (low mean cosine similarity) and qualitatively (in the lack of t-SNE separation).

measured code consistency as the average cosine similarity between the mean code embedding and all embeddings for that code.

In Figure 2a we show the mean cosine similarity results. The similarity rises gradually with a spike for the best codes. As a baseline, we calculate the mean cosine similarity of randomly sampled crops, averaged over ten trials, shown as the red line in the plot. The minimal difference between the random crops and the majority of individual codes suggests at best there are a handful of consistent codes. In Figure 2b, this is examined further with a t-SNE plot of the embeddings for the ten codes with the highest mean cosine similarity and at least 500 embeddings, to ensure similar order of magnitude of samples. We observe only a handful of these codes with strong clustering. In addition, some codes have multiple clusters, which makes a code's semantic meaning less clear.

4 Discussion

As reported above, the model learned a limited number of codes that capture entities such as grass, water, and resources in the inventory. In Appendix C, we examine co-occurring codes to find additional interpretable examples occurring in superposition Elhage et al. (2022), however, they occur very sparingly. Outside of these instances, the codes lacked a consistent semantic theme. Thus, VQ alone cannot provide interpretability of transition models in MBRL. One of the reasons we believe this happened is that VQ offers insufficient constraints to enforce semantic disentanglement of codes. Specifically, the model is always trained to reconstruct the entire image using a combination of codes, and due to the depth of the CNN in our model, the latent vectors have all seen the entire image. This creates a condition where there is no need for the model to cleanly isolate entities in the image to a single code. Thus, indicating there are characteristics of discretization that matter to ensure semantic associations, a deeper study of which is left for future work.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Ranya Aloufi, Hamed Haddadi, and David Boyle. Privacy-preserving voice analysis via disentangled representations. In *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pp. 1–14, 2020.

Yu-Jie Chen, Shin-I Cheng, Wei-Chen Chiu, Hung-Yu Tseng, and Hsin-Ying Lee. Vector quantized image-to-image translation. In *European Conference on Computer Vision*, pp. 440–456. Springer, 2022.

Etienne Desticourt, Véronique Letort, and Florence D'Alché-Buc. Interpretable generative modeling using a hierarchical topological vae. In *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 1415–1421, 2022. doi: 10.1109/CSCI58124.2022.00253.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

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*, pp. 12873–12883, 2021.

Daya Guo, Duyu Tang, Nan Duan, Jian Yin, Daxin Jiang, and Ming Zhou. Evidence-aware inferential text generation with vector quantised variational AutoEncoder. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6118–6129, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.544. URL https://aclanthology.org/2020.acl-main.544.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*, 2021.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kyle Hsu, William Dorrell, James Whittington, Jiajun Wu, and Chelsea Finn. Disentanglement via latent quantization. *Advances in Neural Information Processing Systems*, 36, 2024.

Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/0a0a0c8aaa00ade50f74a3f0ca981ed7-Paper.pdf.

Sarthak Jain and Byron C Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, 2019.

Alexander Khazatsky, Ashvin Nair, Daniel Jing, and Sergey Levine. What can i do here? learning new skills by imagining visual affordances. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14291–14297. IEEE, 2021.

Lukas Klein, João B. S. Carvalho, Mennatallah El-Assady, Paolo Penna, Joachim M. Buhmann, and Paul F Jaeger. Improving explainability of disentangled representations using multipath-attribution mappings. In *Medical Imaging with Deep Learning*, 2022. URL https://openreview.net/forum?id=3uQ2Z0MhnoE.

Xiao Liu, Pedro Sanchez, Spyridon Thermos, Alison Q. O'Neil, and Sotirios A. Tsaftaris. Learning disentangled representations in the imaging domain. *Medical Image Analysis*, 80:102516, 2022. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2022.102516. URL https://www.sciencedirect.com/science/article/pii/S1361841522001633.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4114–4124. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/locatello19a.html.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Christoph Molnar. *Interpretable Machine Learning*. Christoph Molnar, 2 edition, 2022. URL https://christophm.github.io/interpretable-ml-book.

Neel Nanda. A comprehensive mechanistic interpretability explainer & glossary, Dec 2022. URL https://neelnanda.io/glossary.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SATML)*, pp. 464–483. IEEE, 2023.

Jan Robine, Tobias Uelwer, and Stefan Harmeling. Discrete latent space world models for reinforcement learning, 2020.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Nan Shao, Zefan Cai, Hanwei xu, Chonghua Liao, Yanan Zheng, and Zhilin Yang. Compositional task representations for large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6axIMJA7ME3.

Lei Shu, Alexandros Papangelis, Yi-Chia Wang, Gokhan Tur, Hu Xu, Zhaleh Feizollahi, Bing Liu, and Piero Molino. Controllable text generation with focused variation. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3805–3817, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.339. URL https://aclanthology.org/2020.findings-emnlp.339.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.

Alex Tamkin, Mohammad Taufeeque, and Noah D Goodman. Codebook features: Sparse and discrete interpretability for neural networks. *arXiv preprint arXiv:2310.17230*, 2023.

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

Matthew Wallingford, Aditya Kusupati, Alex Fang, Vivek Ramanujan, Aniruddha Kembhavi, Roozbeh Mottaghi, and Ali Farhadi. Neural radiance field codebooks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=mX56bKDybu5.

P. J. Werbos. Learning how the world works: Specifications for predictive networks in robots and brains. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, N.Y.*, 1987.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 11–20, 2019.

Liang Zhang, Justin Lieffers, and Adarsh Pyarelal. Deep reinforcement learning with vector quantized encoding. *arXiv preprint arXiv:2211.06733*, 2022.

Kaifeng Zou, Sylvain Faisan, Fabrice Heitz, and Sébastien Valette. Joint disentanglement of labels and their features with vae. In *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 1341–1345, 2022. doi: 10.1109/ICIP46576.2022.9898046.

Kaifeng Zou, Sylvain Faisan, Fabrice Heitz, and Sébastien Valette. Disentangling high-level factors and their features with conditional vector quantized vaes. *Pattern Recognition Letters*, 172:172–180, 2023.

A Code Frequency

In order to decide which codes would be the most beneficial to examine for consistency, we organized them by how frequently they occur in the dataset. As seen in Figure 3, the model is not overly reliant on any codes because the most frequent one occurs in barely over 1%. However, there is a significant dropoff between the top ten codes and the rest, which is the reason we presented heatmaps and crops with three distinct frequencies. The top two codes occur significantly more than the tenth and eleventh codes. The median codes then occur at a much lower rate. We chose to focus on the higher frequency end of the distribution because we believe even if the low frequency codes are consistent, they will not occur enough to provide useful interpretability, since in many observations, none of those codes will appear.

B Consistent and Inconsistent Code Crops

In Table 3, we show additional cropped images randomly sampled from six codes. The top three shown are examples of the more consistent codes we observed. The first one focuses on stone, while the second captures grass, and the third focuses on water. Throughout our entire dataset of 477

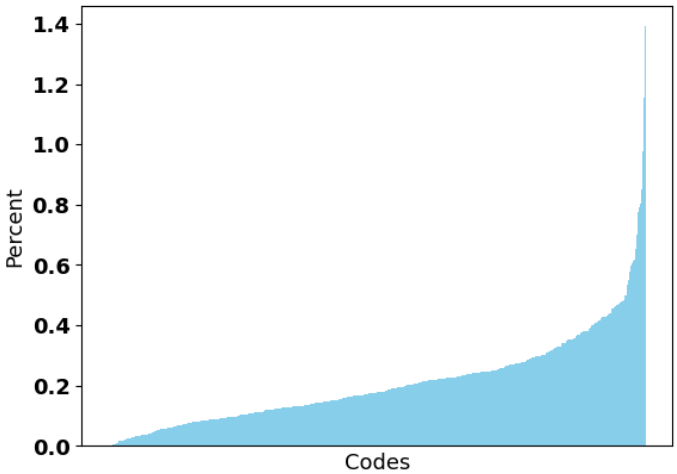


Figure 3: Percentage of each code's occurrence in the dataset.

Table 3: Additional examples of consistent (top three) and inconsistent (bottom three) codes.

Code 0								
Code 360								
Code 192								
Code 15								
Code 329								
Code 438								

codes, only a handful are close to as consistent as these. Even the ones that are have inconsistent instances, like the fourth and fifth crops for code 192, that limit how confidently they can be interpreted. Below them, we show examples for three inconsistent codes. The majority of the codes look closer to these when randomly sampling their cropped images. Across all three of the codes, the crops are widely varied in what they contain, providing no value for interpretability.

C Code Co-occurrence

One of the potential reasons for the lack of interpretability of codes is that the model learned a combination of codes to capture entities. In order to investigate this, we found the codes that most frequently appear together and calculated their co-occurrence rate. We define co-occurrence rate as the number of appearances together divided by the average number of times the two codes are used. This is related to the mechanistic interpretability concept of superposition Elhage et al. (2022), where given a latent feature space represented by a set of d-dimensional vectors v, optimization tries to use these vectors to represent more features than they have the dimensional capacity. This leads to phenomena such as representing concepts with groups of features, or individual features alternately representing more than one concept depending on the input, both of which we observe in our work Elhage et al. (2022); Nanda (2022). Figure 4 shows a heatmap of the rates for the ten

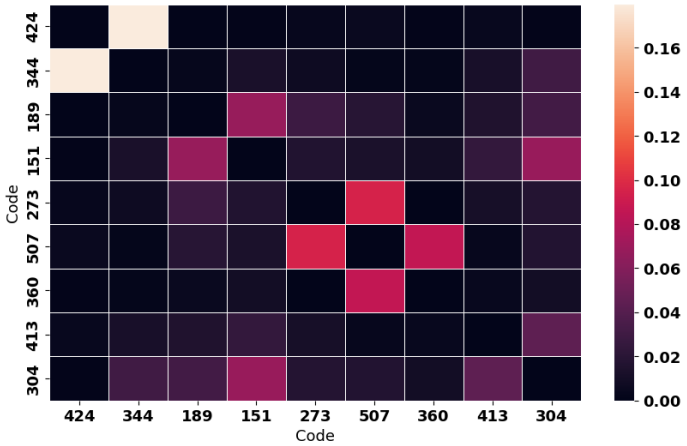


Figure 4: Co-occurrence rate for the ten most frequently co-occurring pairs of codes

Table 4: Example crops from co-occurring codes, randomly sampled from observations where they co-occur.

Code 424								
Code 344								
Code 273								
Code 507								

most frequently co-occurring pairs. Note the co-occurrence of codes to themselves has been set to zero to better distinguish the rates between different codes. The highest rate observed was between codes 344 and 424 at around 18%.

In Table 4, we visualize the crops from the observations where codes co-occur for the two highest co-occurring pairs, codes 344 and 424 and codes 273 and 507. The crops reveal that each code is extremely consistent in what it captures in these instances. Although this is a promising indication of interpretability, we believe it is still greatly limited by two factors. The first applies only to the pair of codes 344 and 424, which is that all their instances of co-occurrence came from a single episode. The codes selected at a given time step exhibit significant co-occurance with the codes from time steps shortly before and after them, since the observations are very similar. Therefore, the high consistency of the codes is less trustworthy because it is an expected behavior within an episode and not demonstrated across episodes. The second factor limiting the impact of co-occuring codes is that, while these instances can indeed provide interpretable observations, they account for a very small portion of the entire observations collected. Specifically, the two pairs, codes 344 and 424 and codes 273 and 507, each co-occur in roughly 0.1% of the observations and the co-occurence rates for all codes suggest very few other pairs will provide percentages near that high. Based on the target application of the RL system, the percent of interpretable samples needed will vary. Given an example target of 10%, our results suggest this may not be possible to reach by combining all the interpretable codes and co-occurring pairs without extensive analysis of all the codes and code combinations. We argue if arduous analysis is required to reach a sufficient amount of interpretability, vector quantization by itself is not truly providing interpretability to the meaningful extent suggested by prior literature.