Level Blending with Evolutionary Algorithm in Latent Space of VQ-VAE

Bahar Boroomand*, **Mohammadreza Hami***, **Shayan Karimi***, **Parham Mohammad Panahi***

University of Alberta
{bborooma, hami, skarimi3, parham1}@ualberta.ca

1 Introduction

Procedural Content Generation (PCG) by definition means automated novel content generation using algorithms for video games, and it has different approaches such as search-based, constructive generation,solver-based, etc (Shaker, Togelius, and Nelson 2016). As the importance of content generation increased, Summerville et al. explored generation using machine learning models trained on existing game data called Procedural Content Generation via Machine Learning (PCGML) (Summerville et al. 2017).

Using PCG can boost our creativity, which is one of its benefits. Even people who consider themselves "creative" have a tendency to copy other people and themselves. However, algorithmic techniques may produce content that is vastly different from what a human would produce by providing a valid but unexpected answer to a particular content generation issue (Shaker, Togelius, and Nelson 2016). PCGML has the same benefit, but it uses machine learning techniques to extract knowledge as mentioned earlier. One approach to achieve this creativity is to generate levels with properties that are a mixture of levels of two games called level blending. In this article, we propose and investigate PCGML methods to blend levels from two different games to generate co-creative levels.

Recently Sarkar et al. proposed Variational Autoencoders (VAE) for blending levels of *Super Mario Bros.* (SMB) and *Kid Icarus* (KI). They showed that by capturing the latent space across both games, a more holistic blending of these level properties is enabled by VAE (Sarkar, Yang, and Cooper 2020). In this work, we want to confirm the effectiveness of discrete latent space of Vector Quantised Variational AutoEncoder (Oord, Vinyals, and Kavukcuoglu 2017), instead of VAE for the same purpose. Moreover, we use evolutionary search in this latent space to increase the controlled creativity of the maps and manage the proportions of elements of *Super Mario Bros.* and *Kid Icarus* presented within the generated maps.

In this work, (1) an exploratory study about the effectiveness of VQ-VAEs and their discrete latent space is conducted. Moreover, (2) an evolutionary search algorithm (to be specific Genetic Algorithm) in the latent space of VQ-VAE is used to help us inject controlled creativity into the blended maps using mutation and create new maps at a desirable level of blending by using a fitness function over tile distributions. We also, (3) present that changing the latent vector elements of one SMB map with elements of KI leads to interpolation between maps. Finally, (4) the combination of VQ-VAE with an evolutionary search algorithm (to be specific Genetic Algorithm) is compared with GA and VQ-VAE itself using two evaluation metrics, Expressive Range Analysis (Summerville 2018), and Quantify Span of Generation (QSG). The latter is a metric we developed to measure how well a model can generate outputs with properties in the middle of input maps' expressive ranges.

2 Related Work

Game-level blending is one of the topics that has been investigated in the computational creativity area. There are different computational creativity techniques. One of them is conceptual blending which was used by Fauconnier et al. for the purpose of level blending (Fauconnier and Turner 1998). Also, Guzdial et al. proposed a process called conceptual expansion to generate new games by recombining existing ones (Guzdial and Riedl 2018a). They also proposed a concept blending approach for blending different Mario-level generation models (Guzdial and Riedl 2016) and explored different blending strategies for the level generation (Guzdial and Riedl 2018b). In this article, we want to use this idea of level blending and computational creativity for blending SMB levels with KI levels but with the use of machine learning techniques.

One approach to achieving blended levels is using deep learning methods. Sarkar et al. proposed the idea of using LSTMs to blend levels from SMB and KI together (Sarkar and Cooper 2018). Variational Autoencoders (Kingma and Welling 2013) are one of the recent methods which have been investigated in this research area as well. Sarkar et al. utilized the VAE idea and Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) (Hansen, Müller, and Koumoutsakos 2003) for generating blended maps using the same datasets (Sarkar, Yang, and Cooper 2020). In this work, they used tile-based properties (like difficulty and density) as their evaluation metrics. They suggested that the VAE's latent space captures the nature of the training data better than the GAN's latent space which was proposed by Volz et al. for the same purpose of blending levels (Volz et al. 2018). Similar to these works we want to blend SMB

*These authors contributed equally.

and KI levels via machine learning, however, we use the discrete latent space of VQ-VAE and combine this latent space with a Genetic Algorithm to control the distribution of blended tiles using fitness. As our evaluation metric we also use tile-based properties. However, besides the tile-based properties evaluation metrics which are used in previous works, we proposed the Quantify Span of Generation as another evaluation metric.

Sarkar et al. generated blended game maps based on 5 different games in their paper (Sarkar and Cooper 2021). They used MAP-Elites (Mouret and Clune 2015) as one of the famous quality diversity algorithms for evolutionary search in VAE's latent space to generate a diverse set of playable levels. They used QD-score and Coverage values as their evaluation metric. The idea of using MAP-Elites as an evolutionary search in the latent space can be one future work to be examined in the latent space of VQ-VAE.

3 Methodology

The goal of our work is to check the effectiveness of VQ-VAE in order to generate blended maps of SMB and KI games, as well as use the GA algorithm in the latent space of VQ-VAQ and compare them. Also, there exists an additional idea in which we choose one of the parents for the crossover function of GA from a specific set and compare the results. Hence, in order to confirm the effectiveness of each one of the ideas, the methodology of this work is divided into four phases. In the first two phases, we implement two blending approaches GA and VQ-VAE. Then, in each one of the following steps, we combine one approach with the other and compare their outputs.

VQ-VAE

There exist two key differences between VQ-VAE and VAE. First, the output of the VQ-VAE's encoder is discrete in contrast with VAE which is continuous; and second, the latent space is learned during the training process but in VAEs it is static (Oord, Vinyals, and Kavukcuoglu 2017). In other words, each element of vectors in its latent space exists in the codebook of the VQ-VAE, and the number of elements in that codebook is finite.

Dataset The dataset for this work includes chunks of maps of Super Mario Bros and Kid Icarus Games which are taken from the Video Game Level Corpus VGLC[1](Summerville et al. 2016).

To create the dataset, we first collected text representations of both game maps. Then, we created square chunks from these representations by moving a 16x16 window on the KI maps. Since the number of rows in the SMB maps is 14, we created 14x16 chunks from this game maps and used the padding technique to add two rows of empty tiles to the top of each chunk to make it 16x16.

Next, we created a unified one-hot representation of both of the games and transferred the text representations of the created chunks into their one-hot representations. As the final step, we combined 2268 chunks of SMB and 1153 KI chunks and shuffled them.

Training Our VQ-VAE encoder had 3 convolutional layers separated by max-pooling layers. Our vector quantizer had a codebook of size 64 and a latent size of 16. The decoder had 3 transposed convolutional layers separated by up-sampling layers. We used the Adam optimizer and the learning rate of 0.001. This model was trained for 50 epochs with a batch size of 32.

Generation We used the elements in the codebook to generate random vectors in the latent space. Then, using the decoder of the VQ-VAE we extracted the map representation of those vectors. We noticed that the generated outputs are blended levels of the two SMB and KI game maps since they had tiles of both of the games. This hypothesis is also evaluated in the evaluation section.

Secondly, to confirm the ability of interpolation in the latent space, we selected two random maps, one from each game maps' original dataset, and passed them through the encoder to access their latent vectors. Starting from one of the vectors, at each step, we selected one element V_i of the first vector that differed from the same index in the second vector W_i and we changed V_i to W_i.

Genetic Algorithm

As the second step, a Genetic Algorithm was implemented which used an initial population of SMB and KI chunks mentioned in the prior section and generated new maps. Being blended in this algorithm is defined using the distribution of the tiles in the maps. Based on the attributes given for each tile[1], we divided them into two separate groups named passable and solid. To find a map's fitness, we calculated the distributions of its passable and solid tiles and compared them with the same distributions for the original chunks of maps. A map is a blending of the two games if its distributions lie between the average of all chunk distributions of the SMB and KI datasets. We also defined a blending point metric to measure how close an expected blended map distribution should be to the average of distributions for the SMB dataset. Inspired by membership functions in fuzzy mathematics (Zadeh, Klir, and Yuan 1996), we created a similar function to determine the fitness score based on the distance of generated maps' tiles distributions and the blending point.

In addition to the fitness function, crossover and mutation functions play significant roles in generating the blended maps in the second phase. The crossover was defined as selecting two individuals randomly from the population as the parents, choosing one column from each, and swapping them to create two new maps which are called children. The mutation was defined as selecting one of the existing tiles in the map and changing it to one another possible random tile.

VQ-VAE + Genetic Algorithm

As the next step, we used Genetic Algorithm to search in the latent space of VQ-VAE for blended maps. The implementation of VQ-VAE is the same as the first phase of this section. As the initial population for GA, we used the latent vectors of the original chunks. We redefined mutation over one latent vector V as randomly selecting one element V_i and changing it to one other random element from the

[1]https://github.com/TheVGLC/TheVGLC

Figure 1: Five samples of generated maps using Genetic Algorithm

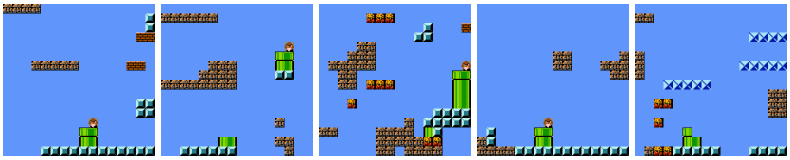
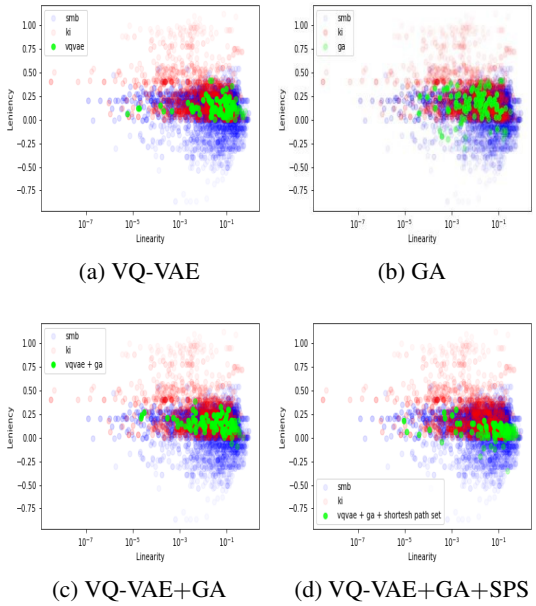


Figure 2: Five samples of generated maps using VQ-VAE

codebook. Also, the crossover was redefined as swapping a random number of elements from two randomly selected vectors as the parents. To calculate the fitness of each individual, we decoded the latent vector using the VQ-VAE's decoder and used the fitness function described in the GA section. In what follows, we note two key values of this work that occurs due to this combination.

First, with the help of the mutation function, injecting more creativity into the maps would be possible. Second, in contrast with the first phase, in this one we have a fitness function that not only controls the creativity injected by mutation by keeping the ones that are close to what we expect the outputs to look like but also we have the blending point and it would help to create new maps at a desirable level of blending. In other words, we can specify a numeric measure of which game should our outputs look closer to.



(a) VQ-VAE (b) GA

(c) VQ-VAE+GA (d) VQ-VAE+GA+SPS

Figure 3: Expressive range analysis for each generator

VQ-VAE + Genetic Algorithm + Shortest Path Set

The final phase of our methodology is about implementing the idea of the Shortest Path Set (SPS) and checking whether it affects generation of better outputs. As mentioned in the VQ-VAE part, one of the key contributions of this work is to confirm the ability to interpolate between two latent vectors in the discrete latent space of VQ-VAE. We used this ability to create the SPS.

Consider V and W as two latent vectors of random maps where the prior is from the SMB dataset and the posterior is from the KI dataset. The length of the shortest path interpolation between these two would be equal to the number of elements V_i in V such that V_i is not equal to W_i. Hence, we can define the SPS as the vectors generated in this interpolation.

In each generation in our Genetic Algorithm, we used the crossover function to mate two individuals as parents and create their children. The SPS was created using two random maps, one from the SMB dataset and one from the KI dataset. Then, we selected one of the parents from SPS and the other one randomly from the population as before. Since one of the parents was chosen from the interpolation of two real maps, we hypothesized that this idea will help to generate better results. The prior statement is evaluated in the next section.

4 Evaluation

To evaluate the quality of each generator, we employed expressive range analysis (Summerville 2018) to plot distributions of generated maps based on two metrics: linearity, and leniency. We made comparisons with the distributions of original maps on these two metrics.

Linearity is defined as the coefficient of determination for linear regression when applied to the generated map segment. Leniency is a measure of difficulty. More specifically, it is the number of enemies in each map plus the number of gaps minus the number of rewards. We scaled each of these three metrics to range $[0, 1]$ before computing leniency.

We also defined a metric to quantify the spanning power of our generative models named Quantify Span of Generation (QSG). Using this metric, one can see how close a map is to the centers of both clusters in this expressive space. In following equations, the QSG is provided where i is each

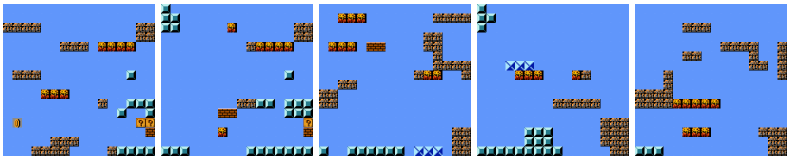


Figure 4: Five samples of generated maps using VQ-VAE + GA

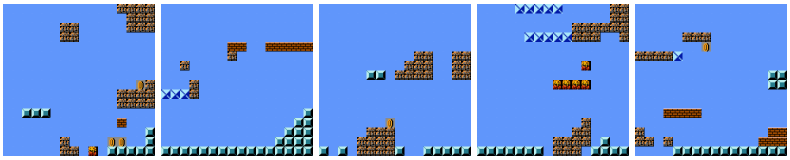


Figure 5: Five samples of generated maps using VQ-VAE + GA + Shortest Path Set

one of the models' output maps and r_i is the blended score of that map.

$$QSG(model) = \sum_i r_i \tag{1}$$

$$r_i = \frac{||C_a - C_b||}{||x_i - C_a|| + ||x_i - C_b||} \tag{2}$$

In equation 2, C_a and C_b are centers of the two clusters for maps of SMB and KI. The best blending model is expected to generate maps with blending score 1 that lie exactly on the line starting from C_a to C_b. The larger the sum of its distances from C_a and C_b gets, the lower its blended score becomes.

5 Results and Discussion

Figure 3 depicts expressive range analysis plots for each model. Blue dots represent the original SMB chunks and red dots are the original KI chunks. The green dots are 100 generated level chunks. The distributions of the two games overlap in the middle and separate out along leniency. One noteworthy observation is the overlap of the generated levels with both games. Generated levels lie in the middle of two games, and the highest concentration is around the centers of the two clusters. This observation implies that these maps are indeed blended. Another observation is that the model which outputs best cover the span between the two distributions is the VQ-VAE+GA model.

Table 1 summarizes the Quantify Span of Generation metric for each model which here is sum of blended scores for 100 generated maps. Since the range of blended score for each map is $[0, 1]$, the maximum score for each model in the following table would be 100. The VQ-VAE+GA achieved the highest QSG among all of the models which is also aligned with the results of the expressive analysis range evaluations.

In Figures 1,2,4 and 5 the results of each of the models are shown. As can be seen in Figure 1, the Genetic Algorithm results have more unexpected and infrequent elements like the coins and the question boxes. However, the tiles in Figure 1 are separated from each other so the maps are neither realistic nor holistic.

In Figure 2, the results of VQ-VAE are shown. The VQ-VAE trained on levels of SMB and KI seems to generate

Model	QSG
VQ-VAE	76.87
GA	74.13
VQ-VAE+GA	**81.26**
VQ-VAE+GA+SPS	62.32

Table 1: Sum of blending scores for 100 generated maps

holistic levels in our opinion. The levels also have interesting elements such as pipes and enemies. This shows that VQ-VAE is suitable for generating blended levels using latent spaces of multiple games which was one of the objectives we achieved.

Figure 4, is the combination of VQ-VAE and Genetic Algorithm, which is explained in the methodology section. The maps generated in this part are also holistic but with the use of our GA fitness function, we can optimize the desired distribution of tiles from each of the games as well. In this Figure, the generated maps are two times closer to SMB than KI due to the blending point we defined in our fitness function. Moreover, one feature of SMB maps is horizontality (the player moves from left to right), while for KI is verticality. In the fourth Figure, it can be seen that maps are both horizontal and vertical, meaning that they can be played in both directions.

The fourth phase of our methodology section's results can be observed in Figure 5. We expected to see a big difference in these maps compared to the ones in phase three, though, there is not. Indeed, in our evaluation results, it is shown that the phase three score is better as we mentioned earlier. However, some specific connected objects of the original maps can be seen in Figure 5 which we hypothesize is the result of using our Shortest Path Set nodes as one of the parents in GA. Evaluating this hypothesis can be explored in the future works.

The last hypothesis is to interpolate in the latent space of VQ-VAE as mentioned in the methodology section. Figure 6 includes maps captured from the interpolation starting from the reconstruction of one KI map and moving toward SMB. Therefore, maps in between have the properties of both games.

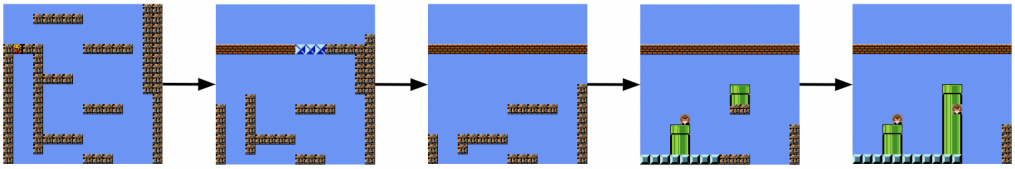


Figure 6: Interpolation in the discrete latent space of VQ-VAE

6 Conclusion and Future Work

In this paper, we confirmed the effectiveness of discrete latent space VQ-VAE and the ability of interpolation in that space. Moreover, we proposed the idea of using the Genetic Algorithm in the latent space of VQ-VAE for generating new blended maps. To ensure the correctness of this idea, we implemented four phases such that each one of the modules, was applied to the previous one. Using expressive range analysis and QSG, we evaluated all the phases. Considering the evaluation results, using GA in the latent space of VQ-VAE helped to create better-blended maps regarding the metrics we used.

We intend to improve our approach by investigating playability in blended maps. One idea can be adding the paths of the player in the original maps of SMB and KI (Sarkar et al. 2020), and encoding the maps with these paths and generating the blended outputs. The playability of generated paths in the output maps should be examined. Another future work could be using MAP-Elites (Mouret and Clune 2015) in the latent space of VQ-VAE in order to increase diversity.

References

Fauconnier, G.; and Turner, M. 1998. Conceptual integration networks. *Cognitive science*, 22(2): 133–187.

Guzdial, M.; and Riedl, M. 2016. Learning to blend computer game levels. *arXiv preprint arXiv:1603.02738*.

Guzdial, M.; and Riedl, M. 2018a. Automated game design via conceptual expansion. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 14, 31–37.

Guzdial, M. J.; and Riedl, M. O. 2018b. Combinatorial creativity for procedural content generation via machine learning. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.

Hansen, N.; Müller, S. D.; and Koumoutsakos, P. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, 11(1): 1–18.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Mouret, J.-B.; and Clune, J. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Oord, A. v. d.; Vinyals, O.; and Kavukcuoglu, K. 2017. Neural Discrete Representation Learning.

Sarkar, A.; and Cooper, S. 2018. Blending Levels from Different Games using LSTMs. In *AIIDE Workshops*.

Sarkar, A.; and Cooper, S. 2021. Generating and blending game levels via quality-diversity in the latent space of a variational autoencoder. In *The 16th International Conference on the Foundations of Digital Games (FDG) 2021*, 1–11.

Sarkar, A.; Summerville, A.; Snodgrass, S.; Bentley, G.; and Osborn, J. 2020. Exploring level blending across platformers via paths and affordances. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, 280–286.

Sarkar, A.; Yang, Z.; and Cooper, S. 2020. Controllable Level Blending between Games using Variational Autoencoders.

Shaker, N.; Togelius, J.; and Nelson, M. J. 2016. *Procedural Content Generation in Games*. Computational Synthesis and Creative Systems. Springer. ISBN 978-3-319-42714-0.

Summerville, A. 2018. Expanding Expressive Range: Evaluation Methodologies for Procedural Content Generation. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 14(1): 116–122.

Summerville, A.; Snodgrass, S.; Guzdial, M.; Holmgård, C.; Hoover, A. K.; Isaksen, A.; Nealen, A.; and Togelius, J. 2017. Procedural Content Generation via Machine Learning (PCGML).

Summerville, A. J.; Snodgrass, S.; Mateas, M.; and Ontanón, S. 2016. The vglc: The video game level corpus. *arXiv preprint arXiv:1606.07487*.

Volz, V.; Schrum, J.; Liu, J.; Lucas, S. M.; Smith, A.; and Risi, S. 2018. Evolving mario levels in the latent space of a deep convolutional generative adversarial network. In *Proceedings of the genetic and evolutionary computation conference*, 221–228.

Zadeh, L. A.; Klir, G. J.; and Yuan, B. 1996. *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers*, volume 6. World Scientific.