Evaluating Quality Diversity Success by Transferring MAP Elites Archives

Maxx Batterton

Introduction

Quality Diversity (QD) algorithms seek to find a diverse collection of distinct high quality solutions to a problem.

QD algorithms find different solutions to the same problem, diversifying and finding many unique good solutions.

MAP-Elites

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites [1]), is a QD algorithm that maps solutions to an archive divided up by features of a solution.

- Each bin in the archive represents solutions with a particular set of features.
- Each solution has features corresponding to exactly one bin.
- The quality of the solution is evaluated. There are three cases:
  - fills an empty bin,
  - improves an older, worse solution
  - is discarded for being worse

Binning Schemes

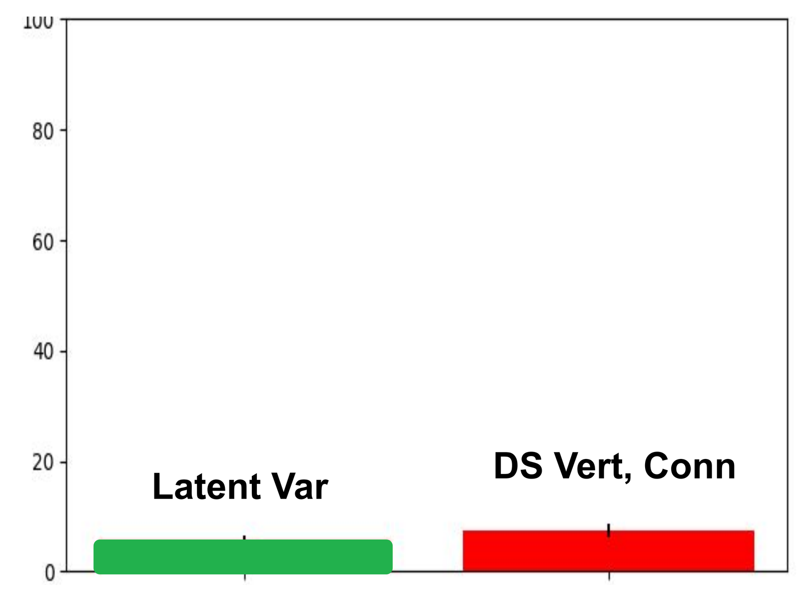
Binning schemes consider the features of a solution and output the bin that the solution corresponds to.

Comparing Binning Schemes

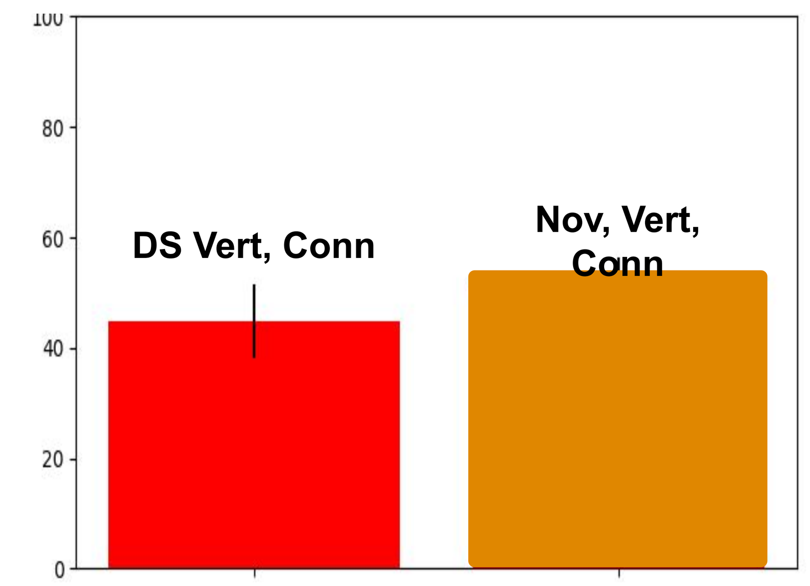
To evaluate a binning scheme, every elite solution from an archive is re-evaluated with respect to a different binning scheme, and survivors and fill percentage are recorded. If many bins were filled and the percentage is high, the target scheme is successfully capturing the diversity in the source scheme.

MEGAMAN (% Surviving)

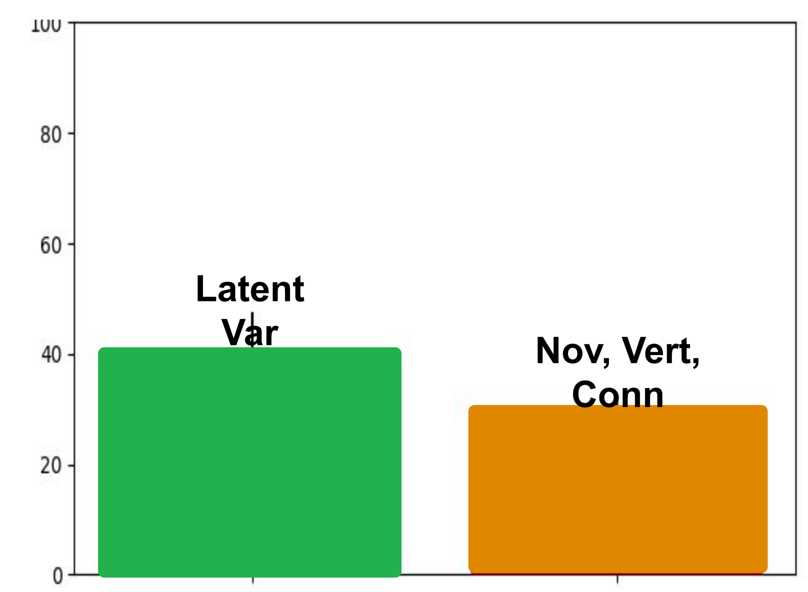
From Novelty, Verticality, Connectivity

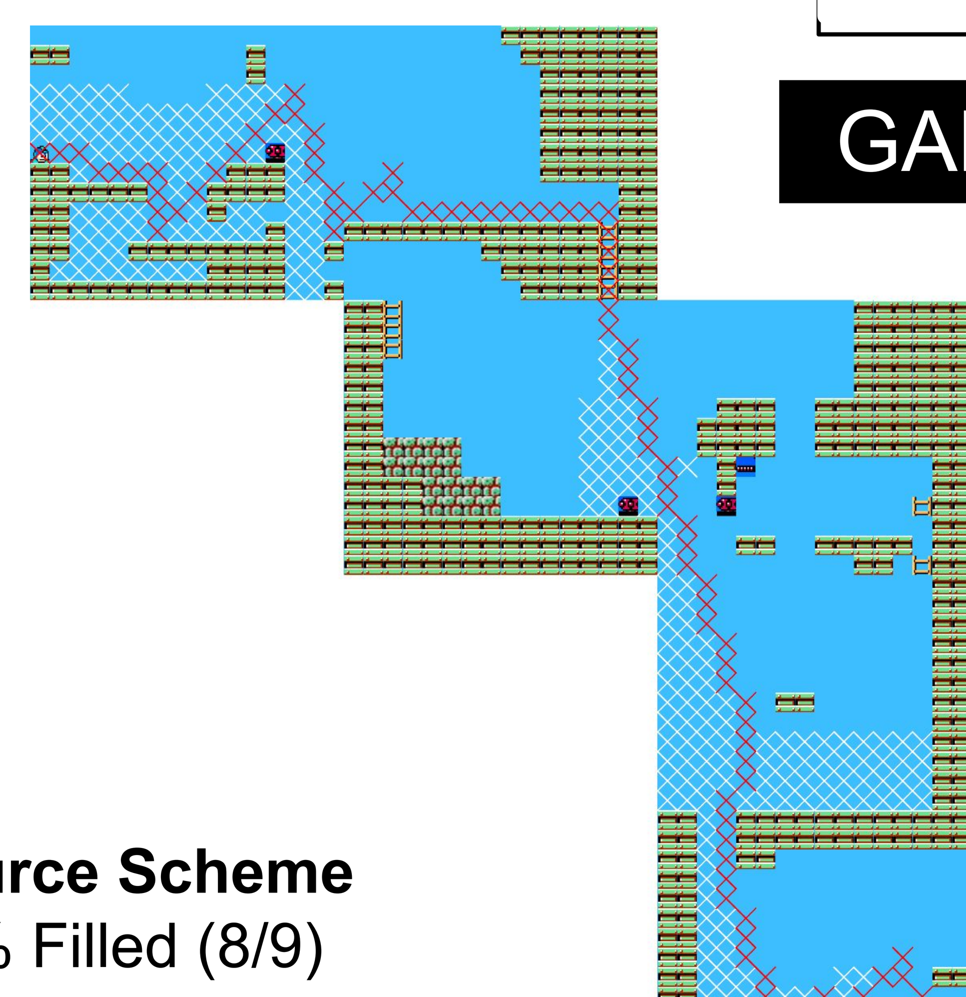
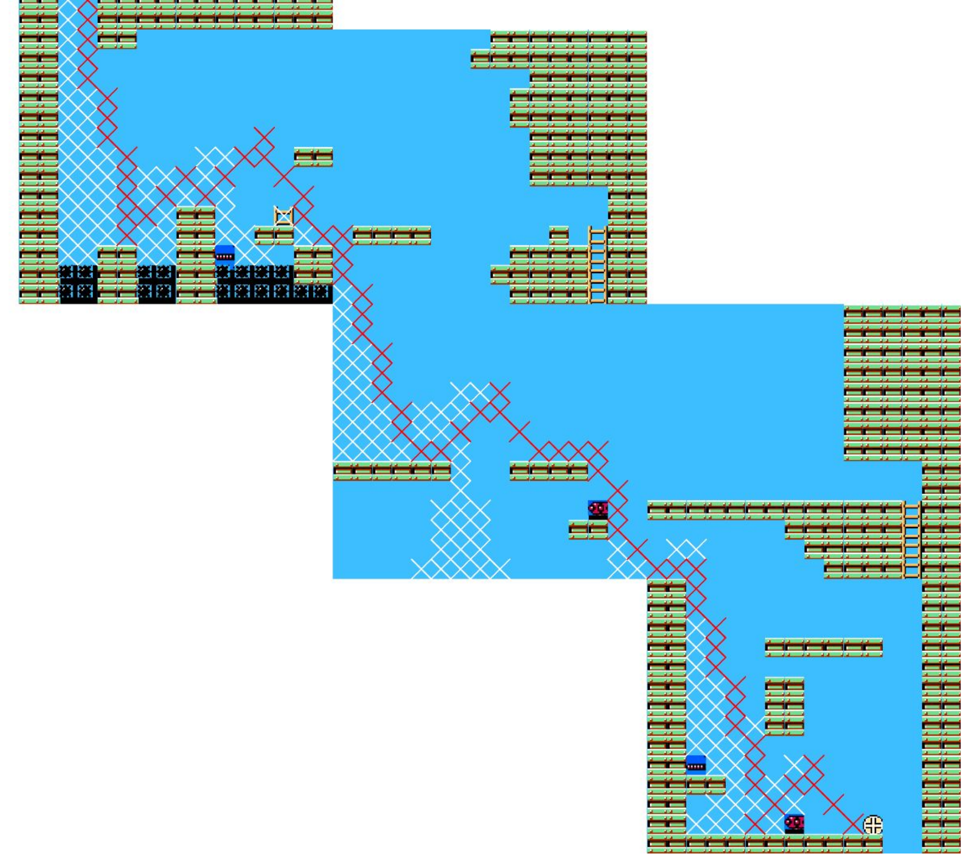


From Latent Variable Half Sums



From Distinct Segments, Verticality, Connectivity

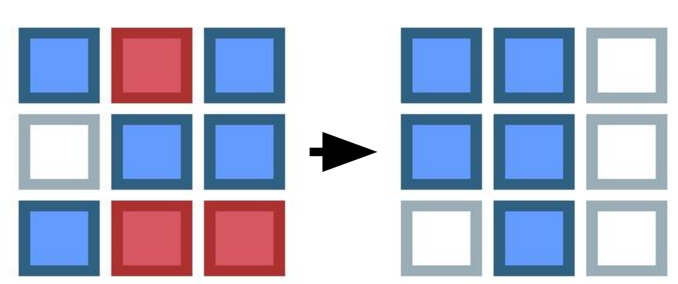


GAN Level Generation

Using existing Generative Adversarial Networks, or GANs, created by past SCOPE students [2,3], we use the GANs to map input vectors to game levels for evaluation.

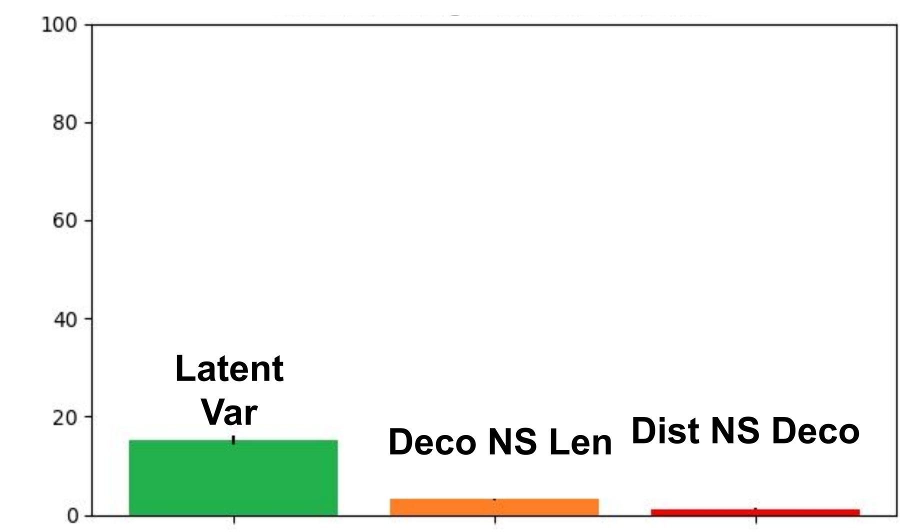
Source Scheme
88% Filled (8/9)



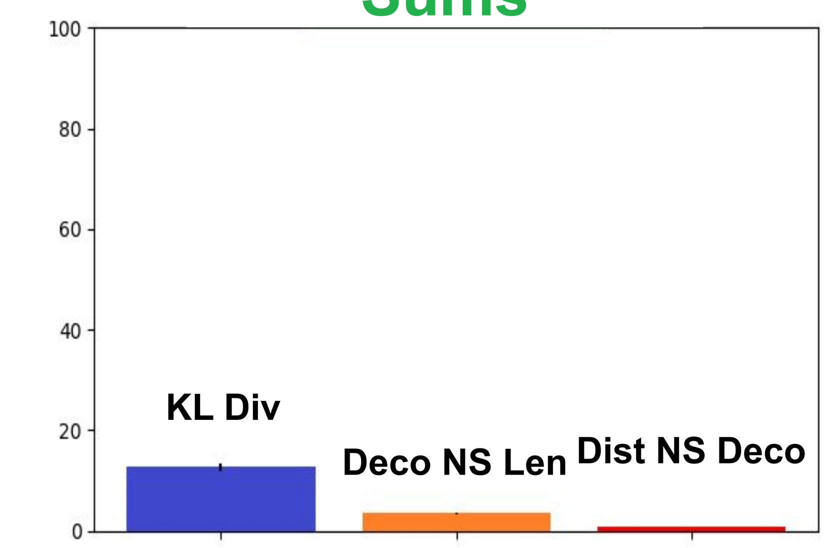
Target Scheme
55% Filled (5/9)
62.5% Surviving (5/8)

MARIO (% Surviving)

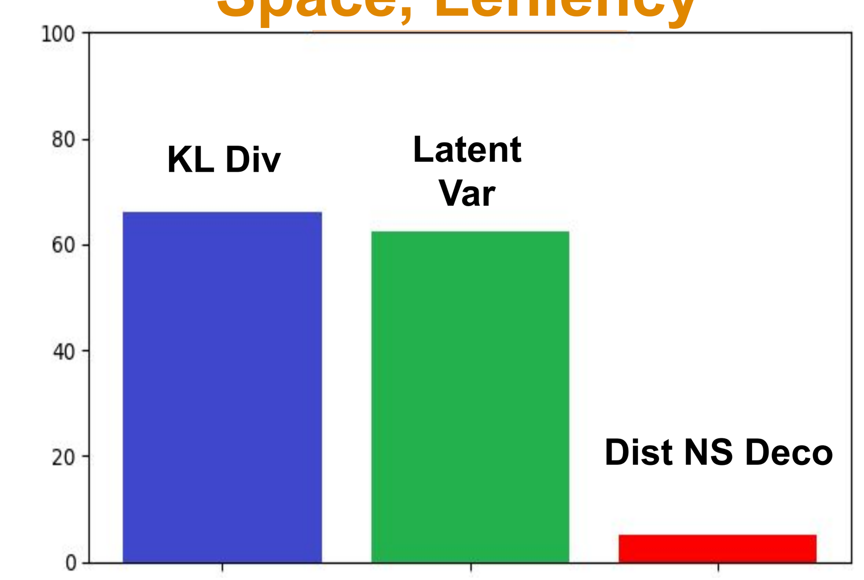
From KL Divergence



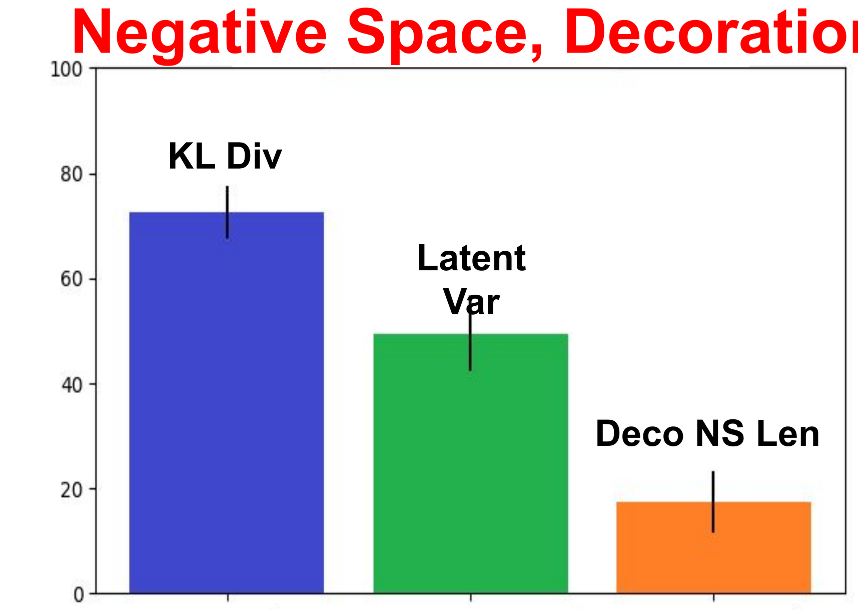
From Latent Variable Half Sums



From Decoration, Negative Space, Leniency



From Distinct Segments, Negative Space, Decoration



Results, Conclusions, and Observations

- In Mario, the general schemes (**KL Div** and **Latent Variable**) were not able to effectively transfer, but in MegaMan, **Latent Variable** nearly filled 50% of the other archives.
- The schemes are not all equal sizes. When the sizes differ, there are either less bins than possible to fully fill the target archive, or too many bins that it becomes easy to fill. If our archives were all the same size, comparison would be easier.
- In MegaMan, our "**Novelty, Verticality, Connectivity**" and "**Distinct Segments, Verticality, Connectivity**" schemes both share Verticality and Connectivity, yet the scheme with novelty did terrible compared to the one with distinct segments

References

[1] A. Cully, J. Clune, D. Tarapore, and JB. Mouret. Robots that can adapt like animals. Nature. 2015; 521(7553):503-7.

[2] J. Schrum, J. Gutierrez, V. Volz, J. Liu, S. Lucas, and S. Risi. Interactive evolution and exploration within latent level-design space of Generative Adversarial Networks. Genetic and Evolutionary Computation Conference. 2020. ACM.

[3] B. Capps and J. Schrum. Using multiple Generative Adversarial Networks to build better-connected levels for Mega Man. Genetic and Evolutionary Computation Conference. 2021. ACM