DSA-ME: DEEP SURROGATE ASSISTED MAP-ELITES

Yulun Zhang[1], **Matthew C. Fontaine**[1], **Amy K. Hoover**[2], **Stefanos Nikolaidis**[1]
[1] Department of Computer Science, University of Southern California
{yulunzha, mfontain, nikolaid}@usc.edu
[2] Ying Wu College of Computing, New Jersey Institute of Technology
{ahoover}@njit.edu

ABSTRACT

We study the problem of efficiently generating high-quality and diverse content in games. Previous work on automated deckbuilding in Hearthstone shows that the quality diversity algorithm MAP-Elites can generate a collection of high-performing decks with diverse strategic gameplay. However, MAP-Elites requires a large number of expensive evaluations to discover a diverse collection of decks. We propose assisting MAP-Elites with a deep surrogate model trained online to predict game outcomes with respect to candidate decks. MAP-Elites discovers a diverse dataset to improve the surrogate model accuracy, while the surrogate model helps guide MAP-Elites towards promising new content. In a Hearthstone deckbuilding case study, we show that our approach improves the sample efficiency of MAP-Elites and outperforms a model trained offline with random decks, as well as a linear surrogate model baseline, setting a new state-of-the-art for quality diversity approaches in automated Hearthstone deckbuilding. We include the source code for all the experiments at: `https://github.com/icaros-usc/EvoStone2`.

1 INTRODUCTION

In this work, we study the problem of *efficiently* generating high-quality and diverse content in games. We believe automated deckbuilding in collectable card games (i.e., Magic the Gathering (Mag, 1993), Legends of Runeterra (Leg, 2020), or Hearthstone (hea, 2014)) forms an ideal case study for this problem. In collectable card games, designers want to automatically generate high performing decks with diverse strategies for both player experience (Heijne & Bakkes, 2017) and playtesting (García-Sánchez et al., 2018). Example strategies include *aggro* decks that focus on reducing the opponent hero's health to zero as quick as possible, and *control* decks that only deal damage to the opponent once the board is under control (Fontaine et al., 2019).

In the Hearthstone deckbuilding problem, generating high-quality and diverse decks is challenging. To evaluate a candidate deck requires at least 200 games even against restricted opponents (Hoover et al., 2019; Bhatt et al., 2018), because of the large number of initial game states and stochastic player actions. In other words, the outcome of a single Hearthstone game has high variance even with deterministic strategies.

To generate high-quality decks with diverse strategic gameplay, we build upon previous work (Fontaine et al., 2019) that formulates the problem of automated deckbuilding as a quality diversity (QD) problem and generates collections of decks via the QD algorithm MAP-Elites. A QD formulation requires an objective function to maximize and measure functions to span. In the context of Hearthstone, we specify the objective as winning games and the measure functions as the average game length and average cards in hand. MAP-Elites will then generate a diverse collection of decks that vary according to these measures. For example, the algorithm will produce decks that end games quickly as well as decks that extend the game as long as possible. It will also attempt to find the best-performing decks for each measure output combination, e.g., for different game lengths.

However, searching for game content with MAP-Elites requires a large number of expensive evaluations, where each evaluation requires many runs in a game engine. We propose assisting the

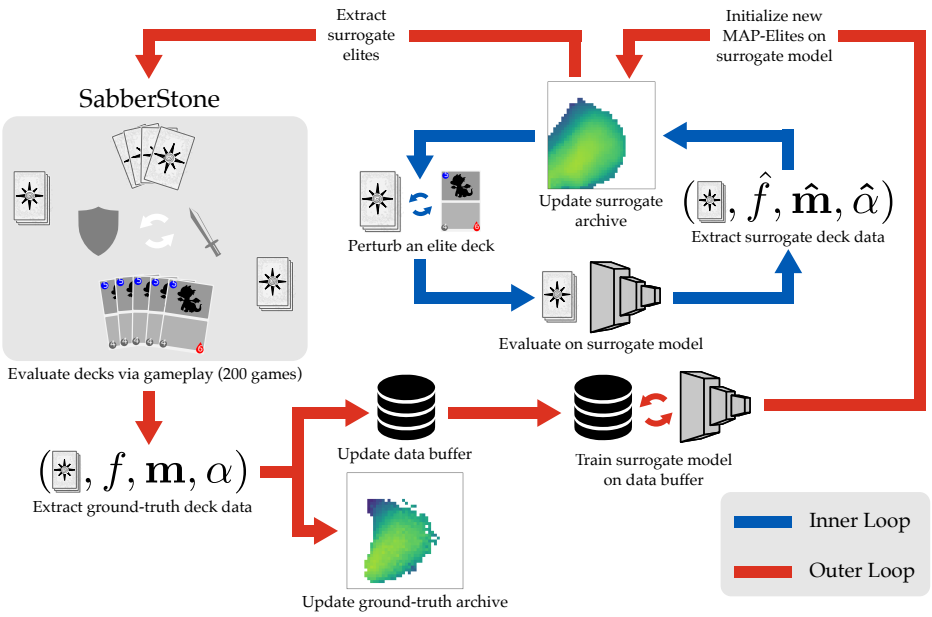

Figure 1: An overview of the Deep Surrogate Assisted MAP-Elites (DSA-ME) algorithm applied to Hearthstone Deckbuilding: Our approach consists of an inner loop (in blue) and an outer loop (in red). In the inner loop, we exploit a deep surrogate model of the SabberStone simulator by searching for a diverse collection of high performing decks with the MAP-Elites algorithm. Each iteration of the inner loop is a full run of the MAP-Elites algorithm on an input surrogate model. The elites of the MAP-Elites archive form candidate decks to be evaluated on the Hearthstone simulator *SabberStone* in the outer loop. We perform an expensive evaluation of each deck by playing 200 games and extract ground-truth data about performance and playstyle. These candidate decks are both added to a data buffer as labeled data to train the surrogate model and to a ground-truth archive. After training the surrogate model on this new data, we initialize a new MAP-Elites to run in the inner loop on the updated surrogate model, completing one iteration of our algorithm. The result of DSA-ME is the ground-truth archive after several iterations of the outer loop.

search with trained surrogate models that predict game outcomes for specific content. Previous work (Karavolos et al., 2021) from procedural content generation showed how to train surrogate models offline from labeled gameplay data in a supervised fashion. Any search algorithm can then exploit the surrogate model directly on the game itself. However, a supervised surrogate model approach requires a large diverse labeled dataset and is vulnerable to the search discovering adversarial examples (Trabucco et al., 2021), low-quality solutions that are predicted by the surrogate model to have high-quality.

As an alternative to offline surrogate models, we can acquire online labeled data by evaluation of game content with simulated agents. The Surrogate-Assisted Illumination (SAIL) (Gaier et al., 2018) algorithm trains on online surrogate model to improve the sample efficiency of MAP-Elites on generative design tasks. SAIL combines a surrogate model, modeled as a Gaussian process, that predicts the utility of solutions with an acquisition function that predicts where to sample. However, Gaussian processes can only be applied on low dimensional continuous data, whereas procedurally generated data is often high dimensional and discrete. For example, a Hearthstone cardset used to create decks may contain several hundred cards.

We propose training a deep neural network as a surrogate model that allows us to scale to high-dimensional search spaces. The deep neural network predicts both the objective and measures, thus we can run many iterations of MAP-Elites on a given surrogate model. Our method yields a two-fold benefit: *by exploiting the surrogate model with MAP-Elites, we obtain diverse data to train a better surrogate model in future iterations and a better surrogate model helps guide MAP-Elites towards high-quality and diverse content.*

Overall, we introduce a novel model-based QD algorithm that maintains an online deep surrogate model predicting the objective and measures of generated game content. Our experiments in automated Hearthstone deckbuilding show that our approach improves the sample efficiency of MAP-Elites and outperforms a model trained offline, as well as an alternative linear model, setting a new state-of-the-art for quality diversity approaches to automated Hearthstone deckbuilding.

2 BACKGROUND

Hearthstone and Automated Deckbuilding. Hearthstone (hea, 2014) is a two-player turn-based virtual collectable card game published by Blizzard in 2014. Players need to form their own decks by choosing cards from a provided cardset. Each deck contains exactly 30 cards and allows just two copies of each card with the exception of *legendary* cards, which are restricted to one copy per deck. The two players' objective of the game is to reduce the opponent's health point to zero by playing cards on a virtual board.

Due to the large cardinality of the Hearthstone deckspace, automated deckbuilding is considered as one of the two main AI challenges of Hearthstone (Hoover et al., 2019). At the time of this writing, Blizzard has published over 2500 collectible cards, which form a large search space of possible decks that players can design. We focus on the basic and classic card sets of the game, which are the first two sets of cards published by Blizzard in Hearthstone. Each of them contains 171 playable cards and each card can be added to a deck at most twice, which would create approximately 1.42×10^{35} possible decks (Fontaine et al., 2019). Although we only focus on these two initial sets, the size of the search space defined by them is large enough for us to demonstrate the effectiveness of our method.

Both evolutionary and non-evolutionary approaches have been applied to the automated deckbuilding problem. Evolutionary approaches include genetic algorithms (García-Sánchez et al., 2016; 2018) and evolutionary strategies (Bhatt et al., 2018) that find a single high-performing deck. Later work (Fontaine et al., 2019) applied the MAP-Elites (Mouret & Clune, 2015) algorithm to generate a collection of high-performing decks with diverse strategic gameplay. Non-evolutionary approaches include a utility based system that automatically constructs decks based on predefined metrics (Stiegler et al., 2016), and a deckbuilding recommendation system (Chen et al., 2018).

Quality Diversity Algorithms. Quality diversity (QD) optimization (Pugh et al., 2016; Chatzilygeroudis et al., 2021) is a class of stochastic optimization algorithms. We define the QD problem over a search space with an objective function to maximize – known as a fitness function in evolutionary computation – and measure functions to span. Quality diversity algorithms have been proposed based on genetic algorithms (Lehman & Stanley, 2011a; Cully et al., 2015; Lehman & Stanley, 2011b), evolution strategies (Fontaine et al., 2020; Colas et al., 2020; Conti et al., 2017; Nordmoen et al., 2018), Bayesian optimization (Kent & Branke, 2020), and gradient ascent (Fontaine & Nikolaidis, 2021b). The MAP-Elites (Cully et al., 2015; Mouret & Clune, 2015) QD algorithm pre-tessellates the space of measure function outputs. MAP-Elites then searches for a solution for each cell in the tessellation that maximizes the given objective. MAP-Elites returns an *archive* of solutions; each cell of the archive contains at most one solution, i.e., an *elite*, which is the highest performing solution in that cell. QD algorithms have been applied to different domains such as procedural content generation (Gravina et al., 2019), robotics (Cully et al., 2015; Mouret & Clune, 2015), aerodynamic shape design (Gaier et al., 2018), and scenario generation in human-robot interaction (Fontaine & Nikolaidis, 2021a; Fontaine et al., 2021).

Model-based Quality Diversity Optimization. Several works in quality diversity optimization have combined insights from model-based optimization (MBO) (Bartz-Beielstein, 2016) or model-based reinforcement learning (MBRL) (Moerland et al., 2020). These approaches broadly fit under the category of model-based quality diversity (MBQD). In MBO there are two main categories of methods: *surrogate* methods and *generative* methods.

Surrogate model methods learn a predictive model of the optimization objective and then exploit that model through an acquisition function or direct optimization. In MBQD, the Surrogate-Assisted Illumination (SAIL) (Gaier et al., 2018) algorithm and the Bayesian Optimization of Elites (BOP-Elites) algorithm (Kent & Branke, 2020) both can be considered surrogate model methods as they learn a Gaussian process that is exploited via an acquisition function. While SAIL learns a surrogate model only of the objective, Surrogate-Assisted Phenotypic Niching (SPHEN) (Hagg et al., 2020) uses

Algorithm 1: DSA-ME

Input: ***evaluate***: function that returns objective f, measures $\mathbf{m}$, and ancillary data α
random_solution: function that generates a random solution x
perturb_elite: function that generates a new solution x given an input archive
N: Number of evaluations on *evaluate* function
n: Number of inner loop iterations
G: Initial population size
Output: Ground-truth Archive M

Function `InnerLoop` *(G, M_s, sm)*:
  for $j \leftarrow 0$ **to** n **do**
    if $j < G$ **then**
      $x \leftarrow$ *random_solution()*
    else
      $x \leftarrow$ *perturb_elite(M_s)*
    end
    $\hat{f}, \hat{\mathbf{m}}, \hat{\alpha} \leftarrow sm.predict(x)$
    Add (x, $\hat{f}$, $\hat{\mathbf{m}}$, $\hat{\alpha}$) to M_s
  end

Initialize a ground-truth archive M, an surrogate model sm, a data buffer D
$i \leftarrow 0$
while $i < N$ **do**
  if $i < G$ **then**
    $x \leftarrow$ *random_solution()*
    $f, \mathbf{m}, \alpha \leftarrow$ *evaluate(x)*
    Add (x, f, $\mathbf{m}$, α) to D
    Add (x, f, $\mathbf{m}$, α) to M
    $i \leftarrow i + 1$
  else
    $sm.train(D)$
    Initialize a surrogate archive M_s
    `InnerLoop` (G, M_s, sm)
    for $x \in M_s$ **do**
      $f, \mathbf{m}, \alpha \leftarrow$ *evaluate(x)*
      Add (x, f, $\mathbf{m}$, α) to D
      Add (x, f, $\mathbf{m}$, α) to M
      $i \leftarrow i + 1$
    end
  end
end

Gaussian processes to predict both the objective and the measures. Previous work (Cazenille et al., 2019) has shown that MAP-Elites with surrogate models of micro-robots physics and chemical reaction dynamics resulted in performance comparable to a complete physical model. Our proposed DSA-ME algorithm falls under this category of MBQD.

Generative model methods learn a generative model of solutions as a learned representation of the search space. By creating a generative model of a design space, an optimization method can more efficiently search for complex solutions indirectly. From an evolutionary computation perspective, these methods can also be viewed as a type of indirect encoding of their search space (D'Ambrosio et al., 2014). In model-based quality diversity, the Data-Driven Encoding MAP-Elites (DDE-Elites) algorithm (Gaier et al., 2020) and the Policy Manifold Search (PoMS) algorithm (Rakicevic et al., 2021) both construct an online learned representation of their search space from elites in the MAP-Elites archive.

Several works in MBQD have drawn inspiration from MBRL. Previous work (Keller et al., 2020) proposes a model-based method for the Novelty Search via Local Competition (NSLC) (Lehman & Stanley, 2011b) algorithm that predicts both the agent reward and a skill descriptor of a reinforcement learning-based robotics controller. Another approach (Lim et al., 2021) trained a model to predict the environment dynamics of a robot. The dynamics model can then be leveraged to learn robotic skills for downstream tasks. Given that the model predicts the dynamics, rather than reward or skill descriptors, new models can be trained to solve different downstream tasks by defining a reward function for the dynamics model. These approaches specialize for skill learning in reinforcement learning domains. Our approach differs as it can be applied to any optimization problem that can be modeled as a quality diversity problem, assuming that a suitable surrogate model exists for the search space.

3 Approach: Deep Surrogate Assisted MAP-Elites

We introduce the Deep Surrogate Assisted MAP-Elites (DSA-ME) algorithm, a model-based quality diversity algorithm derived from MAP-Elites. Fig 1 gives an overview of the DSA-ME framework and Algorithm 1 provides a pseudocode implementation of DSA-ME.

3.1 DEEP SURROGATE MODEL

In our Hearthstone example domain, a deep surrogate model predicts gameplay outcomes for an input Hearthstone deck. The model acts as an efficient surrogate for the game engine, predicting the objective f and measures $\mathbf{m}$ needed to run MAP-Elites. By evaluating game content on the surrogate model instead of the game engine, we aim to improve the efficiency of generating high quality and diverse decks. Additionally, we explore a surrogate model that predicts ancillary data α extracted from gameplay, hypothesizing that this helps form better internal representations and thus generalizes better to new decks.

3.2 INNER AND OUTER LOOPS

The DSA-ME algorithm consists of an inner and outer loop visualized in Fig 1. The inner loop exploits the surrogate model to discover a diverse archive of high-quality solutions, *according to the surrogate model*. The outer loop evaluates candidate archives produced by the inner loop on the Hearthstone game engine as a way to (1) generate labeled data to improve the surrogate model and (2) update a running ground-truth archive of high-quality decks with diverse strategic gameplay.

We initialize the outer loop in the same manner as MAP-Elites, by generating G random decks and evaluating them on the Hearthstone game engine. These initial decks as well as their gameplay data form the initial training data D for the surrogate model and the initial solutions for the outer archive M.

After generating the initial G decks and training the initial surrogate model (line 21), we initialize a new archive M_s (line 22) containing decks with gameplay outcomes predicted by the surrogate model sm. The algorithm then starts running the inner loop (lines 1-10). As in MAP-Elites, the inner loop generates the first G decks by randomly sampling decks (line 4) and later decks by perturbing existing elites (line 6). Within the inner loop, we evaluate new candidate decks directly on the surrogate model (line 8). After evaluation, each candidate deck x and surrogate predictions $\hat{f}$, $\hat{\mathbf{m}}$, and $\hat{\alpha}$ get added to the surrogate archive M_s (line 9).

After the inner-loop runs n iterations, we return to the outer loop. We extract all elites from the surrogate archive M_s and evaluate them on the Hearthstone to obtain ground-truth predictions f, $\mathbf{m}$, and α (lines 24-29). These decks and their ground-truth data get added to both the training data buffer for the surrogate model (line 26) and the outer loop's ground truth archive (line 27). The algorithm results in a ground-truth archive M with decks that have been evaluated on the Hearthstone game engine.

3.3 ON THE BENEFITS OF ONLINE TRAINING

In this section, we provide our intuition on why we expect online training of the surrogate model to improve search performance over surrogate models trained from offline data.

The initial surrogate model will struggle to predict gameplay outcomes as the model is trained only on a few randomly chosen initial decks. When we run MAP-Elites to exploit the surrogate model in the inner loop, we will discover areas of the search space where the surrogate model incorrectly predicts the objective or measure functions of a given deck. Moreover, the data extracted from the surrogate model is diverse according to the surrogate model. By training the surrogate model on these adversarial, with respect to the objective, yet diverse solutions, the model will update its predictions on what constitutes a good Hearthstone deck.

As the outer loop progresses, the surrogate model will become more accurate. When the inner loop starts generating more accurate solutions, the surrogate archive will provide better estimates for high quality decks that vary according to the measure functions. Gradually, the role of the inner loop MAP-Elites shifts from generating adversarial decks that exploit the surrogate model to generating high quality decks that populate the ground-truth archive.

We contrast the online approach with surrogate models trained only with offline data. In the offline case, MAP-Elites will exploit the surrogate model for decks that the surrogate model predicts will perform well, but that perform poorly when evaluated on Hearthstone. Without including these adversarial decks in the training set, the model cannot correct its erroneous predictions.

4 EXPERIMENTS

Our experiments show that combining the QD algorithm MAP-Elites with a *deep surrogate* model trained *online* improves the quality and diversity of automatically generated Hearthstone decks.

4.1 EXPERIMENT SETUP

Search Space. We search for decks for the *Rogue* class in the basic and classic card sets. We are interested in *Rogue* because decks in this class require long-term planning due to the class's combo mechanic; some cards played in the early states of the game might be useful in the later states when played with other cards (Fontaine et al., 2020). Each deck contains 30 cards, selected from 180 candidate cards. Each card can be selected at most twice.

To input a deck into the surrogate model, we encode the deck as an integer vector that represents the frequency of each card chosen. We call this encoding method *Bag-of-Cards* and provide detailed information in Appendix A.1

SabberStone Simulator. To simulate games of Hearthstone, we run our experiments on SabberStone (darkfriend77, 2021), a community developed simulator of Hearthstone games.

Objective and Measures. We adopt the same objective and measures proposed by previous work (Fontaine et al., 2020) for both strategy training and deck search. To evaluate a deck or strategy, we play 200 games of Hearthstone distributed evenly among our opponent suite. To provide a smooth approximation of winning percentage, we specify an objective of the average health difference between the two player at the end of each game. For measures, we calculate the average number of cards in hand at the start of each turn and the average number of turns across the 200 games. The average number of turns measure defines the spectrum between *aggro*, which ends the game quickly, and *control*, which tries to prolong the game, playstyles.

Player Strategy. We use CMA-ME (Fontaine et al., 2020) to train a neural network to play a human-designed *Miracle Rogue* deck (Stonekeep, 2021) with the objective and measures defined in section 4.1. We provide more details on the player strategy in Appendix A.2.

Opponents. We selected six high-performing decks discovered by previous work (Fontaine et al., 2019) in the *Hunter*, *Paladin*, and *Warlock* classes as our opponents. For each class, we include both an *aggro* and *control* opponent. We choose these opponents to provide our deck search with high quality and strategically diverse opponents.

Surrogate Models and Baselines. We use a multilayer perceptron (MLP) as a deep surrogate model. To show the benefit of deep surrogate models, we include a linear surrogate model baseline. We refer to these models as Linear Surrogate Assisted MAP-Elites (LSA-ME) and Deep Surrogate Assisted MAP-Elites (DSA-ME), respectively. To demonstrate the importance of training the surrogate model online, we add a baseline that exploits a fixed MLP trained in a supervised fashion. We collect the training data for the offline MLP by sampling 10000 randomly generated decks and evaluating them on SabberStone. The fixed model is not updated online after training. We denote this baseline as Offline DSA-ME. Finally, we include MAP-Elites without a surrogate model as a model-free baseline. We run 5 trials for each algorithm. We describe the hyperparameters in Appendix A.3.

Perturbation Operator. All implemented QD algorithms require a perturbation operator. To perturb elites, we replace k cards from a parent elite with k cards chosen uniformly from the cardset. We vary k geometrically, similarly to previous work (Fontaine et al., 2019).

Ancillary Data. Drawing upon insights from self-supervised learning (Ren & Lee, 2018), we explore whether adding ancillary data to the prediction of the surrogate models helps form better internal representations and thus generalize better to new decks. We provide a full list of ancillary data in Appendix A.4. We denote surrogate model algorithms trained with ancillary data as DSA-ME (ad) and Offline DSA-ME (ad).

Table 1: Hearthstone numerical results

Algorithm	Max Avg Health Diff	Max Win Percentage	Cells Filled Percentage	QD-Score
MAP-Elites	15.72 ± 0.62	90.6 ± 0.91 %	14.36 ± 0.32 %	136.89 ± 1.95
Offline DSA-ME	18.69 ± 0.17	97.1 ± 0.19 %	21.64 ± 0.12 %	194.94 ± 1.65
LSA-ME	20.07 ± 0.18	96.2 ± 0.12 %	27.89 ± 0.64 %	276.23 ± 6.10
DSA-ME	$\mathbf{22.30 \pm 0.16}$	$\mathbf{98.4 \pm 0.10}$ **%**	$\mathbf{31.86 \pm 0.71}$ **%**	$\mathbf{338.52 \pm 5.86}$
Offline DSA-ME (ad)	20.23 ± 0.20	97.7 ± 0.30 %	15.65 ± 0.13 %	168.67 ± 0.99
DSA-ME (ad)	21.15 ± 0.35	97.4 ± 0.43 %	30.78 ± 1.35 %	312.66 ± 11.72

4.2 Experiment Design

Independent Variables. We follow a between-groups design with two independent variables: (1) The type of surrogate model: DSA-ME, LSA-ME, Offline DSA-ME, and MAP-Elites without any surrogate model. (2) The presence of ancillary data: none or ancillary data (ad) if the model predicts the additional metrics described in section 4.1.

Dependent Variables. The dependent variables measure the quality and diversity of the decks found by the algorithms. Specifically, the dependent variables are the maximum average health difference, the maximum win percentage, the percentage of cells filled in the ground-truth archive, and the QD-score (Pugh et al., 2015). The maximum average health difference and maximum win percentage measure the capability of the algorithm to search for globally optimal solutions. The percentage of cells filled in the ground-truth archive measures the diversity of the solutions. The QD-score measures the quality and diversity of the solutions altogether and is defined as the sum of the objective values of all solutions in the ground-truth archive.

Hypotheses.
H1: *The DSA-ME, LSA-ME, and Offline DSA-ME algorithms will have better performance than MAP-Elites*. We hypothesize that the surrogate models will result in MAP-Elites finding more diverse and high-performing solutions than without the models.

H2: *DSA-ME will have better performance than Offline DSA-ME and LSA-ME*. We choose as baseline the linear model to demonstrate the benefit of a deep model. We also compare against Offline DSA-ME to demonstrate the benefit of training the model online with data generated from MAP-Elites.

H3: *Adding the ancillary data to DSA-ME and Offline DSA-ME will improve performance*. We expect the ancillary data to give an additional training signal to the surrogate models so that the models can learn a better internal representation and more accurately predict the objective and measures. We only include ancillary data in DSA-ME and Offline DSA-ME, since linear models do not have internal representations.

5 Results

Table 1 shows the average maximum objective value, average maximum win percentage, average percentage of cells filled, and average QD-score over 5 trials.

To test hypotheses 1 and 2, we performed a one-way ANOVA test for each of the metrics. There were statistically significant differences in all four metrics: QD-score ($F(3, 16) = 403.49, p < 0.001$), maximum win percentage ($F(3, 16) = 52.81, p < 0.001$), maximum average health difference ($F(3, 16) = 64.65, p < 0.001$), and percentage of cells filled ($F(3, 16) = 228.30, p < 0.001$). Post-hoc pairwise comparisons with Bonferroni corrections showed that all three variations of DSA-ME performed significantly better than MAP-Elites in all four metrics. This supports **H1**.

In addition, DSA-ME outperforms LSA-ME and Offline DSA-ME in QD-score, percentage of cells filled, and maximum average health difference. There is no statistically significant difference in the maximum win percentage among the three algorithms. We attribute this to the fact that the maximum win percentage is high for all algorithms. Overall, we observe that DSA-ME finds more diverse and high-performing solutions, supporting **H2**.

To test the third hypothesis, we first performed a two-way ANOVA test for each metric, with the type of the surrogate model (DSA-ME, Offline DSA-ME) and the presence of ancillary data as independent variables. There was no statistically significant interaction between the surrogate model

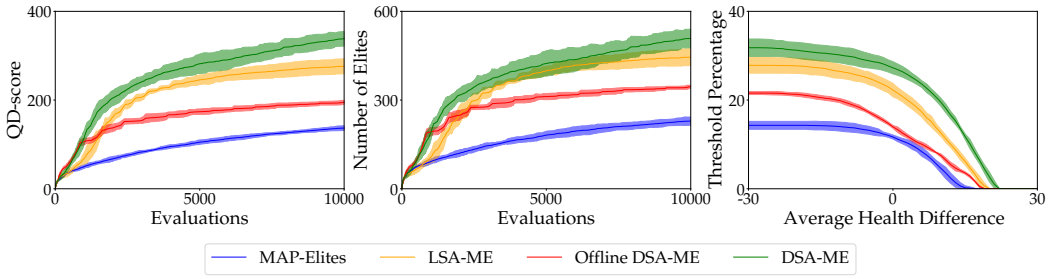

Figure 2: Quantitative analysis of the elites in the archive with the surrogate model predicting objective and measures (without ancillary data). In the rightmost plots, the y-axis shows the percentage of filled cells in the archive that had average health difference larger than the corresponding value in the x-axis. Larger area under each curve indicates better performance

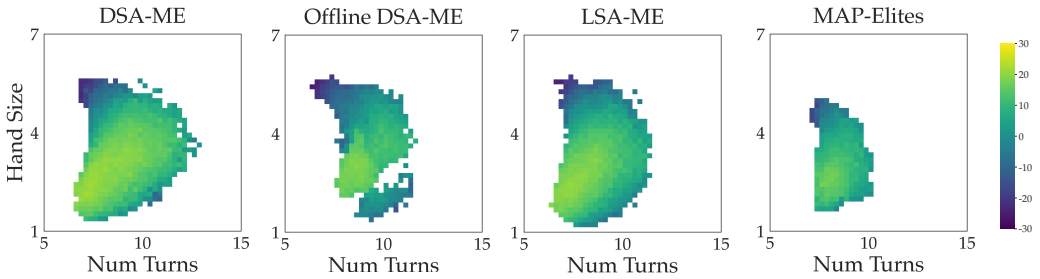

Figure 3: Example archives from single runs of vanilla MAP-Elites and variants of DSA-ME. The color indicates the objective value (average health difference) of each solution. Qualitatively DSA-ME finds more diverse and high performing solutions than MAP-Elites.

type and the presence of ancillary data for QD-score. Main effect analysis showed that surrogate models with ancillary data had significantly smaller QD-score than the models without the ancillary data ($p = 0.001$).

We found a statistically significant interaction between the two independent variables for maximum win percentage ($F(1, 16) = 8.00, p = 0.012$), maximum average health difference ($F(1, 16) = 33.277, p < 0.001$), and percentage of cells filled ($F(1, 16) = 10.148, p = 0.006$). Simple main effect analysis showed that ancillary data helped only in one case: maximum average health difference was significantly higher for Offline DSA-ME (ad) compared to Offline DSA-ME ($p < 0.001$). For DSA-ME (ad), maximum win percentage and maximum average health difference are both significantly worse than DSA-ME ($p = 0.024, p = 0.003$) and we found no significance in percentage of cells filled. For Offline DSA-ME (ad), percentage of cells filled is significantly worse than Offline DSA-ME ($p < 0.001$) and we found no significance for maximum win percentage. These results show that adding ancillary data did not improve performance and they do not support **H3**. We discuss this finding in section 7.

Figure 2 shows the QD-score and the percentage of cells filled. We note that the QD-score metric combines both the number of cells filled in the archive (diversity) and the objective value of each occupant (quality). To disambiguate the two, we also show in Fig. 2 the Complementary Cumulative Distribution Function (CCDF) plot of the solutions in the archive. The CCDF plot shows the percentage of cells (y-axis) that have objective value greater than the threshold specified in the x-axis. Fig. 3 shows example heatmaps of the ground-truth archives of the four algorithms (without the ancillary data).

The heatmaps and plots support the results from the statistical analysis, showing that DSA-ME finds more diverse and higher quality solutions than the baselines. They also show that Offline DSA-ME, although using a deep surrogate model, performs worse than LSA-ME in both QD-score and number of elites. In addition, as shown in the CCDF plots, DSA-ME consistently finds more solutions in the archive over all average health differences.

6 On the Performance of DSA-ME

We perform a post-hoc experimental analysis to shed more light on what makes DSA-ME achieve high performance. We are interested in the quality of the surrogate model and the labelled elite data as well as the effect of resetting MAP-Elites in the inner loop. We describe the experiments and results in Appendix B. The results suggests that (1) although the surrogate model does not always predict the exact cell, the generated candidate solutions are diverse enough to illuminate the measure space; (2) DSA-ME generates a diverse, high-quality dataset of decks and training on the dataset enables the surrogate model to make accurate predictions; (3) resetting MAP-Elites in the inner loop enables DSA-ME to generate more diverse candidate solutions.

7 Discussion

Our work presents a method to efficiently generate diverse and high-quality game content for collectable card games. We introduce the Deep Surrogate Assisted MAP-Elites (DSA-ME) algorithm, a novel model-based quality diversity algorithm.

In our experiments, we observe that DSA-ME outperforms surrogate models trained on offline, randomly generated data and a model-free MAP-Elites approach, setting a new state-of-the-art for QD approaches in automated Hearthstone deckbuilding. By dynamically forming a surrogate model of Hearthstone gameplay, DSA-ME explores more complex decks, whose performance is hypothesized by surrogate model predictions. This reduces the number of simulated games required to find high-performing decks with differing strategic gameplay. We anticipate that our results will benefit game designers looking to balance collectable card games through automated deckbuilding (de Mesentier Silva et al., 2019).

While our DSA-ME approach improves upon existing methods for automated deckbuilding, we found that adding ancillary data to the surrogate model predictions did not improve performance. This result conflicts with our intuitions from recent advancements in self-supervised learning (Ren & Lee, 2018); we hypothesized that ancillary data would help the deep surrogate model learn a better internal representation of the Hearthstone cardset.

A potential reason for this result is the simplicity of the Hearthstone deck representation. We encode decks as a Bag-of-Cards input feature vector, which marks if a card is included in the deck. This encoding does not capture a card's mana cost, attack, health, or card text that would be available to a human player when building a deck. We conjecture that encoding these card attributes could help the surrogate model form a better assessment of individual cards and generalize better to new cards.

An alternative reason could be the low fidelity of our surrogate model. A low fidelity model would predict only game outcomes, while a high fidelity model would replicate the entire game engine. Low-fidelity models allow for game content to be evaluated efficiently as a single forward pass in a feed-forward neural network. However, the low fidelity model may not be able to represent the level of context necessary to take advantage of ancillary data.

An alternative approach would be to use a high-fidelity model that predicts the game dynamics directly, similarly to model-based reinforcement learning (Lim et al., 2021). A high-fidelity neural simulation of Hearthstone games could help improve the prediction accuracy of the surrogate model. However, such a model could be less efficient than the Hearthstone simulator itself. Future work will explore this trade-off across the fidelity spectrum of model-based quality diversity for procedural content generation.

Acknowledgments

The authors thank Adam Gaier for his feedback in early versions of this work.

References

Magic the gathering, 1993. URL https://magic.wizards.com/en.

Hearthstone, 2014. URL https://playhearthstone.com/en-us.

Legends of runeterra, 2020. URL https://playruneterra.com/en-us/.

Thomas Bartz-Beielstein. A survey of model-based methods for global optimization. pp. 1–, 01 2016.

Aditya Bhatt, Scott Lee, Fernando de Mesentier Silva, Connor W Watson, Julian Togelius, and Amy K Hoover. Exploring the hearthstone deck space. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, pp. 1–10, 2018.

Leo Cazenille, Nicolas Bredeche, and Nathanael Aubert-Kato. Exploring self-assembling behaviors in a swarm of bio-micro-robots using surrogate-assisted map-elites. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 238–246. IEEE, 2019.

Konstantinos Chatzilygeroudis, Antoine Cully, Vassilis Vassiliades, and Jean-Baptiste Mouret. Quality-diversity optimization: a novel branch of stochastic optimization. In *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*, pp. 109–135. Springer, 2021.

Zhengxing Chen, Chris Amato, Truong-Huy Nguyen, Seth Cooper, Yizhou Sun, and Magy Seif El-Nasr. Q-deckrec: A fast deck recommendation system for collectible card games, 2018.

Cédric Colas, Vashisht Madhavan, Joost Huizinga, and Jeff Clune. Scaling map-elites to deep neuroevolution. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pp. 67–75, 2020.

Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *arXiv preprint arXiv:1712.06560*, 2017.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, May 2015. ISSN 1476-4687. doi: 10.1038/nature14422. URL http://dx.doi.org/10.1038/nature14422.

rnilva darkfriend77. Sabberstone. https://github.com/icaros-usc/SabberStone, 2021.

Fernando de Mesentier Silva, Rodrigo Canaan, Scott Lee, Matthew C Fontaine, Julian Togelius, and Amy K Hoover. Evolving the hearthstone meta. In *2019 IEEE Conference on Games (CoG)*, pp. 1–8. IEEE, 2019.

David B D'Ambrosio, Jason Gauci, and Kenneth O Stanley. Hyperneat: The first five years. *Growing adaptive machines*, pp. 159–185, 2014.

Matthew Fontaine and Stefanos Nikolaidis. A quality diversity approach to automatically generating human-robot interaction scenarios in shared autonomy. *Robotics: Science and Systems*, 2021a.

Matthew C. Fontaine and Stefanos Nikolaidis. Differentiable quality diversity, 2021b.

Matthew C Fontaine, Scott Lee, Lisa B Soros, Fernando de Mesentier Silva, Julian Togelius, and Amy K Hoover. Mapping hearthstone deck spaces through MAP-elites with sliding boundaries. In *Proceedings of The Genetic and Evolutionary Computation Conference*, pp. 161–169, 2019.

Matthew C. Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K. Hoover. Covariance matrix adaptation for the rapid illumination of behavior space. *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, Jun 2020. doi: 10.1145/3377930.3390232. URL http://dx.doi.org/10.1145/3377930.3390232.

Matthew C. Fontaine, Ya-Chuan Hsu, Yulun Zhang, Bryon Tjanaka, and Stefanos Nikolaidis. On the importance of environments in human-robot coordination. *Robotics: Science and Systems*, 2021.

Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. Data-efficient design exploration through surrogate-assisted illumination. *Evolutionary computation*, 26(3):381–410, 2018.

Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. Discovering representations for black-box optimization. *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, Jun 2020. doi: 10.1145/3377930.3390221. URL http://dx.doi.org/10.1145/3377930.3390221.

Pablo García-Sánchez, Alberto Tonda, Giovanni Squillero, Antonio Mora, and Juan J Merelo. Evolutionary deckbuilding in hearthstone. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8. IEEE, 2016.

Pablo García-Sánchez, Alberto Tonda, Antonio M Mora, Giovanni Squillero, and Juan Julián Merelo. Automated playtesting in collectible card games using evolutionary algorithms: A case study in hearthstone. *Knowledge-Based Systems*, 153:133–146, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Daniele Gravina, Ahmed Khalifa, Antonios Liapis, Julian Togelius, and Georgios N Yannakakis. Procedural content generation through quality diversity. In *2019 IEEE Conference on Games (CoG)*, pp. 1–8. IEEE, 2019.

Alexander Hagg, Dominik Wilde, Alexander Asteroth, and Thomas Bäck. Designing air flow with surrogate-assisted phenotypic niching. In *International Conference on Parallel Problem Solving from Nature*, pp. 140–153. Springer, 2020.

Nikolaus Hansen. The cma evolution strategy: A tutorial, 2016.

Norbert Heijne and Sander Bakkes. Procedural zelda: A pcg environment for player experience research. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450353199. doi: 10.1145/3102071.3102091. URL https://doi.org/10.1145/3102071.3102091.

Amy K. Hoover, Julian Togelius, Scott Lee, and Fernando de Mesentier Silva. The many ai challenges of hearthstone, 2019.

Daniel Karavolos, Antonios Liapis, and Georgios N. Yannakakis. A multifaceted surrogate model for search-based procedural content generation. *IEEE Transactions on Games*, 13:11–22, 2021.

Leon Keller, Daniel Tanneberg, Svenja Stark, and Jan Peters. Model-based quality-diversity search for efficient robot learning. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9675–9680, 2020.

Paul Kent and Juergen Branke. Bop-elites, a bayesian optimisation algorithm for quality-diversity search. *arXiv preprint arXiv:2005.04320*, 2020.

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

Joel Lehman and Risto Miikkulainen. Enhancing divergent search through extinction events. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 951–958, 2015.

Joel Lehman and Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011a.

Joel Lehman and Kenneth O Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pp. 211–218, 2011b.

Bryan Lim, Luca Grillotti, Lorenzo Bernasconi, and Antoine Cully. Dynamics-aware quality-diversity for efficient learning of skill repertoires. *ArXiv*, abs/2109.08522, 2021.

Thomas M. Moerland, Joost Broekens, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *ArXiv*, abs/2006.16712, 2020.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites, 2015.

Jørgen Nordmoen, Eivind Samuelsen, Kai Olav Ellefsen, and Kyrre Glette. Dynamic mutation in map-elites for robotic repertoire generation. In *Artificial Life Conference Proceedings*, pp. 598–605. MIT Press, 2018.

Justin Pugh, Lisa Soros, and Kenneth Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3, 07 2016. doi: 10.3389/frobt.2016.00040.

Justin K. Pugh, L. B. Soros, Paul A. Szerlip, and Kenneth O. Stanley. Confronting the challenge of quality diversity. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pp. 967–974, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334723. doi: 10.1145/2739480.2754664. URL https://doi.org/10.1145/2739480.2754664.

Nemanja Rakicevic, Antoine Cully, and Petar Kormushev. Policy manifold search: Exploring the manifold hypothesis for diversity-based neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '21, pp. 901–909, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383509. doi: 10.1145/3449639.3459320.

Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 762–771, 2018.

Andreas Stiegler, Claudius Messerschmidt, Johannes Maucher, and Keshav Dahal. Hearthstone deck-construction with a utility system. In *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, pp. 21–28. IEEE, 2016.

Stonekeep. Classic miracle rogue, 2021. URL https://www.hearthstonetopdecks.com/decks/classic-miracle-rogue-2/.

Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *International Conference on Machine Learning*, pp. 10358–10368. PMLR, 2021.

Frank Veenstra, Pablo González de Prado Salas, Kasper Stoy, Josh Bongard, and Sebastian Risi. Death and progress: How evolvability is influenced by intrinsic mortality. *Artificial life*, 26(1): 90–111, 2020.

Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.

A IMPLEMENTATION DETAILS

A.1 BAG-OF-CARDS ENCODING

we encode the deck as an integer vector that represents the frequency of each card chosen. Formally, each input vector $x \in \mathbb{R}^c$ represents a deck where c is the cardinality of the cardset. Each element x_i represents the number of copies of card i included in the deck. The range of x_i is bounded to $\{0, 1, 2\}$ because each card can be added to a deck at most twice in Hearthstone. We call this encoding method *Bag-of-Cards* as it is similar to the Bag-of-Words (Zhang et al., 2010) embedding used in natural language processing and information retrieval problems.

A.2 PLAYER STRATEGY

SabberStone includes a library containing simple AI agents that play the game via a tree search. At each turn, the tree search explores possible moves and discovers the best action sequence that maximizes the heuristic game state score at the end of the turn. Included heuristics emulate *aggro* strategies, which end the game quickly, and *control* strategies, which try to prolong the game, for arbitrary decks. The library allows for custom scoring functions that result in more advanced strategies (Fontaine et al., 2020).

In our preliminary experiments, we found that the heuristic strategies included in SabberStone struggle to play decks of the *Rogue* class. To create a stronger agent, we train a custom neural network heuristic function with CMA-ME following previous work (Fontaine et al., 2020). The neural network maps 15 game related features defined by SabberStone (darkfriend77, 2021) to a single value representing the preference for an end-of-turn state. The neural network is a multilayer perceptron with layer sizes $[15, 5, 4, 1]$ and 109 parameters.

We train the neural network to play a human designed *Miracle Rogue* deck (Stonekeep, 2021) as the player's deck with the objective and measures defined in section **??**. The *Classic Miracle Rogue* deck requires the player to have long-term planning to play well, thus we expect a high-performing strategy using this deck to generalize to other advanced *Rogue* decks as well.

While CMA-ES (Hansen, 2016) is usually preferred while optimizing a single objective black-box function, previous work (Fontaine et al., 2020) showed that CMA-ME is able to find a globally better neural network strategy than CMA-ES because CMA-ME finds more diverse solutions and is less likely to converge to a local minimum. Therefore, the neural network is trained using CMA-ME in 50000 evaluations. The highest performing solution was chosen as the final strategy.

A.3 HYPERPARAMETERS

The MLP of the surrogate model has three hidden layers with sizes $[128, 32, 16]$ with an Exponential Linear Unit (ELU) activation functions. All surrogate models are trained using the mean squared error loss function and Adam optimizer (Kingma & Ba, 2014) with learning rate $\eta = 0.01$ and batch size 64. In each outer loop, all models are trained for 20 epochs.

We set the number of inner loop iterations $n = 10^6$ with a batch size of 10. We set the number of evaluations on the SabberStone game simulator $N = 10,000$, each with 200 simulated Hearthstone games. The initial population size G for both inner and outer loop is 100. We run all algorithms for 5 trials on a high-performing cluster, utilizing 200 Xeon CPUs. Each trial lasted 10 to 12 hours.

A.4 ANCILLARY DATA

As ancillary data, we include the win percentage, the total damage done, the number of cards drawn, the total mana spent, the total mana wasted, the sum of mana cost for all cards in the deck, the variance of mana costs in the deck, the number of minion cards in the deck, and the number of spell cards in the deck. We chose data that relates to either deck composition or how the player strategy utilized the given deck.

B ON THE PERFORMANCE OF DSA-ME

We perform a post-hoc experimental analysis to shed more light on what makes DSA-ME achieve high performance.

B.1 ON THE QUALITY OF THE SURROGATE MODEL

First, we analyze the quality of the surrogate model to determine if high quality predictions of the objective and measure functions result in the improved performance. Consider instead an alternative hypothesis that the surrogate model *drifts* in prediction error and the fact the surrogate model changes throughout DSA-ME is responsible for good performance rather than prediction accuracy.

First, we confirm that the final trained surrogate model from DSA-ME achieves high accuracy by evaluating it on a different test data created from a separate run of DSA-ME. The average mean squared error over 5 runs is 0.44 for the predicted number of turns whose range is [5, 15], 0.07 for the hand size with range [1, 7], and 14.70 for the average health difference with range [-30, 30]. We observe that the model predicts the measure function outputs with high accuracy, while it has higher error for the objective.

While the above measure function predictions are accurate, they are not accurate enough to ensure the candidate solutions from the surrogate archive will remain in the same MAP-Elites bin when evaluated on the game engine. To evaluate this, we run MAP-Elites on the final DSA-ME surrogate model for 5 trials. We then take elites from the surrogate archive, evaluate each deck on the game engine, and populate a new archive with the ground truth objective and measures (see Fig. 5). We observe that 44% of the elites from the surrogate archive are retained in the new archive. While only 3% of decks are placed in the same cell as the surrogate archive, the average Manhattan distance between the cell in the surrogate and ground-truth archives is 2.45 cells. This result shows that although the surrogate model does not frequently predict the exact cell, candidate solutions remain in the same region of the archive.

To show that the quality predictions of the surrogate model are the reason for the high performance of DSA-ME, we add as baseline Offline DSA-ME (Surrogate Model), which uses a "frozen" surrogate model that is fully trained from a previous run of DSA-ME. Fig. 4 shows that Offline DSA-ME (Surrogate Model) outperforms DSA-ME. This indicates that using a fully-trained model for evaluation results in better performance than training the model online. We note that *this holds only when the surrogate model has been trained with data generated from a complete DSA-ME run*; if it is trained with randomly generated data, Offline DSA-ME performs significantly worse, as shown in section 5.

B.2 ON THE QUALITY OF THE LABELED ELITE DATA

We note the similarity of training the surrogate archive to the online training of the discriminator network in generative adversarial networks (Goodfellow et al., 2014). We test whether we can only train the surrogate model incrementally in a closed-loop manner, or we can train it from scratch using the entire dataset generated from a DSA-ME run. We compare the final surrogate model of a DSA-ME run, Offline DSA-ME (Surrogate Model), with a surrogate model trained from scratch on the data of the DSA-ME run, Offline DSA-ME (Elite Data). In both conditions we train the surrogate models and then run DSA-ME with the frozen models. Fig. 4 shows that there is no significant difference in performance between the two models, indicating that we can use the training data from a DSA-ME run to train an accurate surrogate model from scratch.

Overall, results in sections B.1 and B.2 show that *running MAP-Elites in the inner-loop of Algorithm 1 results in a diverse, high-quality dataset of decks for the surrogate model. Training on the dataset enables the surrogate model to make accurate predictions*.

B.3 ON THE IMPORTANCE OF RESETTING MAP-ELITES

We visualize the archive generated by running MAP-Elites using the surrogate model predictions (Fig. 5-left). We then evaluate each elite on the Sabberstone game engine and use the Sabberstone

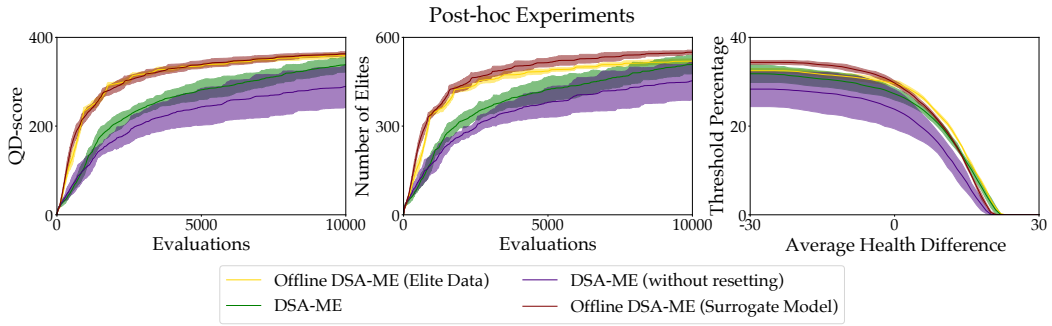

Figure 4: Post-hoc experiments to evaluate the quality of the surrogate model and the effect of resetting MAP-Elites in the inner loop of DSA-ME.

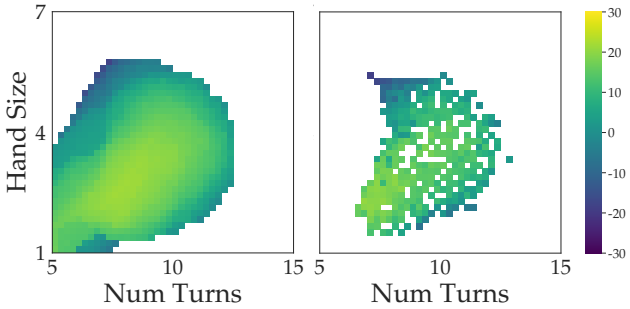

Figure 5: (Left) Archive from running MAP-Elites on the surrogate model. (Right) Archive by evaluating the elites on the left with the game engine.

evaluations to place the elites on a new archive (Fig. 5-right). We observe that the new archive covers the space reasonably well but has holes in different areas.

Thus, we hypothesize that resetting MAP-Elites in the inner-loop (line 22) with random initialization produces archives with different missing cells and that the union of these archives covers the space well. We test this hypothesis by running DSA-ME and keeping the same archive in the inner-loop, instead of initializing a new archive (lines 22-23 in Algorithm 1). We call this baseline DSA-ME (without resetting). Fig. 4 shows that this baseline has lower QD-score and coverage. We find this result consistent with previous work (Lehman & Miikkulainen, 2015; Veenstra et al., 2020) on the benefits of mortality on evolvability in divergent search algorithms.