Learning With Generalised Card Representations for “Magic: The Gathering”

Timo Bertram*[†]
*Dept. of Computer Science
Johannes-Kepler Universität
Linz, Austria
tbertram@faw.jku.at

Johannes Fürnkranz*[†]
[†]LIT Artificial Intelligence Lab
Johannes-Kepler Universität
Linz, Austria
juffi@faw.jku.at

Martin Müller
Dept. of Computing Science and Amii
University of Alberta
Edmonton, Canada
mmueller@ualberta.ca

***Abstract*—A defining feature of collectable card games is the deck-building process prior to actual gameplay, in which players form their deck out of large collections of cards or small, fixed sets. Learning to build decks is difficult for players and models alike due to the large card variety and highly complex semantics, as well as requiring meaningful card and deck representations when aiming to utilise AI. In addition, regular releases of new card sets lead to unforeseeable fluctuations in the available card pool, thus affecting possible deck configurations and requiring continuous updates. Previous GameAI approaches to building decks have often been limited to fixed sets of possible cards, which greatly limits their utility in practice. In this work, we explore possible card representations that generalise to unseen cards, thus greatly extending the real-world utility of AI-based deck-building for the game “Magic: The Gathering”. We study such representations based on numerical, nominal, and text-based features of cards, card images, and meta information about card usage from third-party services. Our results show that while the particular choice of generalised input representation has little effect on learning to predict human draft picks among known cards, the performance on new, unseen cards can be greatly improved. Our generalised model is able to predict 55% of human choices on completely unseen cards, thus showing a deep understanding of card quality and strategy.**

***Index Terms*—Magic: The Gathering, card games, neural networks, Siamese networks, comparison training, game AI**

I. Introduction

Modern board and card games are popular domains for AI research due to their inherent structure and wide popularity. While super-human has been achieved in some games such as Go, chess, DOTA and poker [1], [6], [11], many others still provide numerous obstacles. Providing additional difficulty, commercial games, and especially collectable card games such as *Hearthstone* and *Magic: The Gathering (MTG)*, are constantly evolving due to card changes and the release of novel content. This implies that any model for such games must be able to continuously adapt to new releases to maintain usefulness. In this work, we focus on one crucial aspect of the popular card game *Magic: The Gathering*; deck-building. Before card play begins, players build their deck of cards by selecting from a large pool of available choices. This initial stage of the game is crucial for success, as without strong and general deck-building strategies, strong performance is not achievable. We focus on learning a generalised model for drafting, a sub-area of deck-building, in *Magic: The Gathering* that is readily expandable to new content and can accurately predict human decisions. To achieve this, we incorporate the **contextual preference ranking** (CPR) framework of Bertram et al. [2] and adapt it in several ways:

- The original work on CPR is limited to a single, fixed set of cards, using one-hot encoding to represent inputs to a Siamese neural network. We explore different generalised card representations for use in CPR, thus extending the method to arbitrary and potentially previously unseen cards.
- We train a large model for drafting in *Magic: The Gathering* on a large, heterogeneous dataset of 100 million examples. The model obtains general knowledge of card semantics, showing usefulness for drafting decks with new cards.
- We explore fine-tuning the model to quickly adapt to newly released card sets, achieving rapid performance improvements.

II. Related Work

We build on the **contextual preference ranking** (CPR) approach [2] by making it more general and widely usable, while also improving upon its overall accuracy. CPR functions by constructing preferences between choices within an explicitly stated context in which the decision was made. Data on such decisions-in-context can be coded as triplets which are used to train **Siamese neural networks** (SNNs) [5]. As one such case, card drafting decisions (see Section III) in *Magic: The Gathering* can be regarded as preferences among the different available cards in the context of all previously chosen cards. In the initial CPR research [2], this approach was shown to outperform classical fully-connected neural networks [15].

Other research on drafting, or deckbuilding in general, is largely based on reinforcement learning or genetic algorithms [4]. For the game of *Legends of Code and Magic*, a simple card game developed for testing AI agents, earlier work [8], [13] separates the drafting and playing phases and learns to draft while approximating the strength of the resulting decks through heuristic-based gameplay. Such approaches depend

979-8-3503-5067-8/24/$31.00 ©2024 IEEE

strongly on the chosen gameplay strategies and therefore limit the depth of strategy. Later, end-to-end approaches for drafting and playing were developed [16], which led to drafting strategies decoupled from heuristics. However, in the LOCM test-bed, card complexity is low and gameplay is simple, which limits applicability to real games. Xiao et al. [17] used a similar algorithm, an end-to-end reinforcement learning agent combining card selection and gameplay, for the highly successful commercial game of *Hearthstone*. This approach yields impressive results and appears to be a promising direction when fast-forward game simulations are possible. However, in order to allow exploration of the enormous game space, the card pool and selection options were strongly restricted, thus slightly limiting overall usefulness. So far, it appears that end-to-end reinforcement learning of drafting and gameplay is not yet practical for unrestricted game environments, especially when regarding ever-changing domains. Our work combats this problem from a different angle: instead of attempting to learn extremely complex relations with no prior knowledge, we rely on human behaviour and model decisions seen in historic game data in a generalised manner.

Overall, human players of *Magic: The Gathering* have shown significant interest in using drafting agents as tools to aid their decision-making. Popular tools for this task[1] attempt to solve the problem though hand-constructed rules, and thus rely on humans in the design, and prevent the usage AI models. One of these [15] employs a classical fully-connected neural network trained with supervised learning, but received worse predictive accuracy than CPR [2]. Peer-reviewed research results on the other tools are not available at this time. Finally, general work on card games used different card representations [9], [19], opting to represent features or images of cards for classification and regression, showing promising directions.

III. Drafting in collectable card games

Most collectable card games (CCG) provide two overarching modes of gameplay: *Constructed* and *Limited*. This includes commercial games such as *Hearthstone*, *Magic: The Gathering*, *Lorcana* or *Flesh and Blood*, as well as simple AI counterparts such as *Legends of Code and Magic*. To simplify the following explanation we focus on *Magic: The Gathering (MTG)*, but most rules apply to other games with minimal adaptions.

In **constructed** modes, each player independently builds a deck of at least 60 cards in advance of playing the game. Players possess mostly free reign over which cards to include in their deck, leading to an enormous space of potential configurations. Theoretically, the possible deck space is infinitely large as no upper bound on legal card quantity exists. In practice, the vast majority of players select the smallest legal number of 60 cards to maximise the consistency of their decks by minimising the variance of drawn cards. At the time of writing, the number of unique cards to choose from ranges from 3,037 cards in the smallest version of the game to 26,524 in the largest format. Based on this, and disregarding some specifics of deck-building, the number of unique decks players could potentially build lies between $\binom{3037}{60} > 10^{126}$ and $\binom{26524}{60} > 10^{183}$. While an enormous space, the majority of such deck compositions are completely useless in practice and high-level human competition mostly converges to fewer than 100 deck archetypes with slight individual adaptions. How one could train models that emulate such an enormous reduction in deck space remains an open research question.

Limited games, or in this context **drafts**, incorporate deck-building into the actual game. Before play commences, players take turns to build their decks sequentially. Players start the draft with a choice of 15 initial cards, passing unpicked cards to the other players, thus decreasing the number of options, until all initial cards are selected. This process repeats thrice with 15 cards per player, thus leading to a total of 45 cards and 42 choices with 15 to 2 options for every participant. Therefore, within one draft, one player's decisions result in $(15!)^3 > 10^{36}$ different possible final deck configurations. While still large, the sequential nature of decisions leads to smaller, more manageable and largely independent tasks. In this work, we focus on such drafting environments, viewing each decision as independent and disregarding the effect of card choices on other players. As such, we solely focus on each individual choice in the context of the player's previous decisions, enabling the use of CPR.

IV. Adapting Contextual Preference Ranking

A. Contextual Preference Ranking

Contextual preference ranking (CPR) is a framework for learning to predict the better of two choices $\mathbf{c}_j$ and $\mathbf{c}_k$ in the context of a set of previously made decisions $\mathcal{C}$, formally denoted as

$$(\mathbf{c}_j \succ \mathbf{c}_k \mid \mathcal{C})\,. \tag{1}$$

In our setting, the context $\mathcal{C}$ represents the (incomplete) deck of cards a player is holding, $\mathbf{c}_j$ and $\mathbf{c}_k$ are two possible additions to it, while the preference encodes which one of the two is a better fit to $\mathcal{C}$.

Bertram et al. [2] tackled this problem by training a Siamese neural network (SNN) from a database of human card selection decisions. The key idea is to train a single neural network $N(.)$ that maps card sets and individual cards into a uniform embedding space by minimising the triplet loss

$$L_{\text{triplet}}(\mathbf{a},\mathbf{p},\mathbf{n}) = \max\left(d(\mathbf{a},\mathbf{p}) - d(\mathbf{a},\mathbf{n}) + m, 0\right) \tag{2}$$

over three weight-sharing copies of the network, where the anchor $\mathbf{a} = N(\mathcal{C})$ represents the embedding of the context $\mathcal{C}$, $\mathbf{p} = N(\mathbf{c}_j)$ the embedding of the preferred choice $\mathbf{c}_j$, and $\mathbf{n} = N(\mathbf{c}_k)$ the other choice. The parameter m represents a desired margin (typically set to $m = 1$), and d is a distance metric (typically the Euclidean distance).

[1] https://draftsim.com/arenatutor/, https://mtga.untapped.gg/companion, and https://mtgaassistant.net/

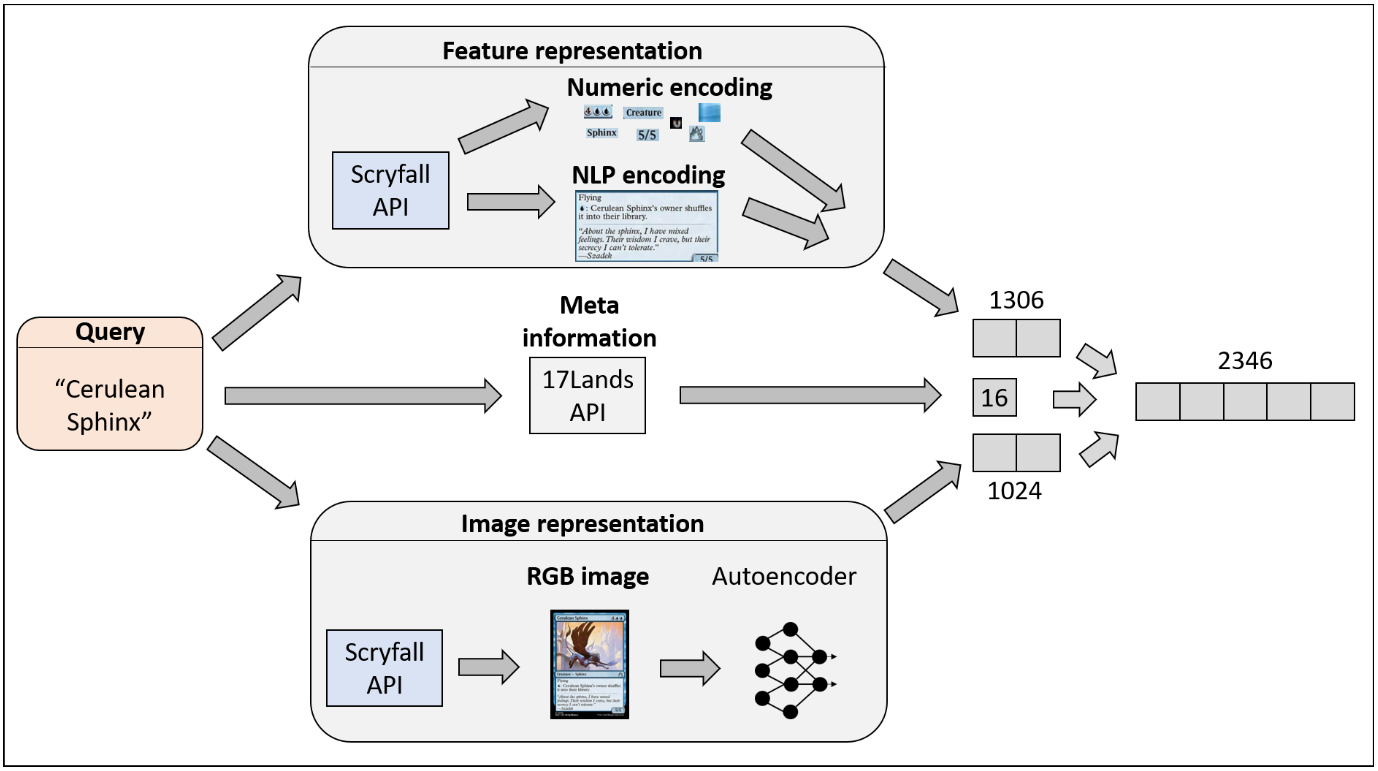


Fig. 1. High-level overview of generating card representations according to Section IV-B. The representation-variant shown here is *Features* + *Meta* + *Image*, simpler representations only use parts of the pipeline.

In testing, the model embeds the context $\mathcal{C}$ and all possible choices $\mathbf{c}_i$, and ranks choices by their embedded distance to $\mathcal{C}$. A card

$$\mathbf{c}^* = \arg\min_{\mathbf{c}_i} d(N(\mathbf{c}_i), N(\mathcal{C})) \tag{3}$$

with minimal distance is chosen as the best addition to the current deck $\mathcal{C}$.

We follow this framework but with a generalised implementation. In the former work, the presence of 265 unique cards was represented as one-hot encodings used as inputs to the SNNs. This input representation can not be adapted for training or prediction on unseen cards and separate models would need to be trained for each new set of cards. The majority of our work focuses on this problem of training on arbitrary sets and generalising to unseen inputs.

B. Input representation

In *MTG*, cards are released in so-called *expansions* or *sets*. Every such set features 200–300 unique cards. The majority of these cards are completely new and often feature new rules and mechanics. In order to generalise CPR for this type of application, we study different representations of cards and explore their advantages and disadvantages.

1) Random vectors: Representing cards by randomly generated vectors of arbitrary size enables an infinite space of possible inputs. However, this representation prevents generalising knowledge to unseen cards and learning semantics and is thus only useful for in-sample cards. Still, we experiment with random vectors as card representations to provide insight into the importance of representation choice for seen sets. In these experiments, each unique card is encoded as a vector of size 1024.

2) Hand-coded features: To generalise across cards and sets, we can encode cards by their unique*features* which capture the rules and semantics associated with them. *Magic* card features include numerical values, categorical features of large categories, and a card text of great importance 2. To a lesser extent, they contain an RBG image, which will be regarded in the following section. For the *Features* representation, we encode a card as a vector of all numerical and categorical features and append a text-embedding of the full card text generated with a sentence transformer [10].

3) Image Representation: When simply encoding a card by its pixel-wise RGB values, no handcrafted features are required. However, this tasks the model to learn the complex semantics of a card purely from visual cognition. Text recognition from images is challenging [14] and likely leads to a loss of information compared to an explicit encoding. In addition, full-scale RGB images of cards are large ($3 \times 936 \times 672$), leading to an explosion in input size. To combat this, we train a basic convolutional autoencoder [7] to construct latent representations of card images which can be used as compressed representations of inputs to the SNN. In our experiments, we found strong influence, both visually (see Figure 3) and numerically (see Table VI-A), of the bottleneck

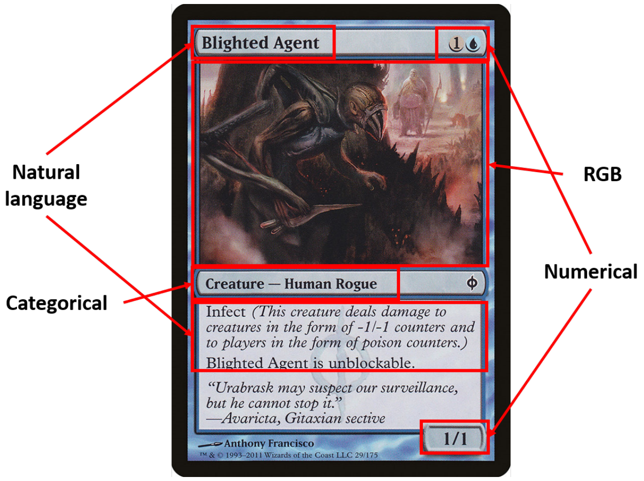

Natural language
Categorical
RGB
Numerical
Blighted Agent
Creature — Human Rogue
Infect (This creature deals damage to creatures in the form of -1/-1 counters and to players in the form of poison counters.)
Blighted Agent is unblockable.
"Urabrask may suspect our surveillance, but he cannot stop it."
—Avaricta, Gitaxian sective
1/1
Anthony Francisco

Fig. 2. Anatomy of a *Magic: The Gathering* card. Cards consist of a number of different features of varying importance and representation.

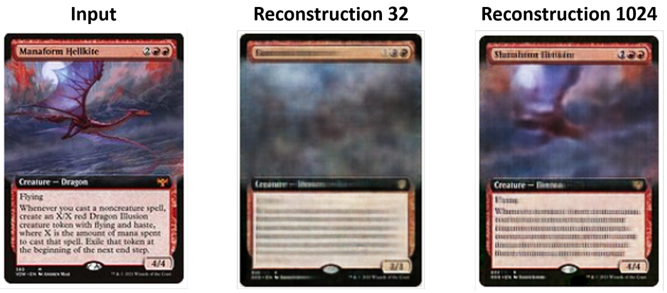


Fig. 3. Inputs and reconstructions of trained autoencoders. Input and outputs are of shape 3 × 936 × 672, the latent spaces are 32-dimensional and 1024-dimensional respectively. The latent card representations are used as compressed input to the SNN (see Figure VI-A)

dimensionality on the received reconstructions. We present the results with latent space dimensionalities of 32 and 1024, finally choosing 1024 dimensions for later experiments to minimise information loss.

4) Statistical Meta Information on Cards: Meta information about card use by human players, such as pick-rate and player win-rates when using the card, is publicly available [2]. Such usage information provides direct insight into human decisions for the cards involved. This is an especially useful feature when little useful context is available, e.g. when comparing similar cards, or at stages of deck-building when few cards have been selected. We provide an experiment where we train a model solely on the meta-information (Table VI-A), but generally, such statistics should be viewed as additional information rather than a stand-alone encoding. Noteworthy, card statistics are only available when a set has seen sufficient server play and is thus impossible to use for entirely new sets. In our experiments, we encode statistics as a scalar vector of size 16.

C. Encoding networks

Regardless of the specific encoding used, adapting CPR from one-hot to generalised encodings provides the additional challenge of deriving a representation for card decks from the representation of single cards [3]. With one-hot encodings of cards, a deck can simply be represented as the sum of card encodings without loss of information. However, with feature-based card encodings, a sum-based encoding of a deck makes it impossible to reconstruct individual cards. To prevent this, we represent a deck of cards as a two-dimensional array of size 45 × n, where 45 is the maximum number of cards in the deck and n is the representation size of individual cards. Slots for not-yet-picked cards are set to all zero. When decks and cards are used as inputs to the SNN, this first requires translating them to a common representation (see Figure 1), which is achieved by adding two separate encoding networks that output 512-dimensional representations, of cards and decks respectively, to the training pipeline. These encoding networks are trained end-to-end with the SNN.

V. Experimental Setup

A. Data

With the outlined adaptions to CPR, we evaluate their influence on the accuracy of the resulting model. In addition to the changes outlined in Section IV, the experiments in this section feature a larger variety of card sets from different sources, collectively encoding approximately 100 million total card picks. Table I shows all used datasets along with their respective release dates and training set sizes. We used all released datasets up to and including *LTR*. Data of M19 is obtained from *DraftSim*[3], all other datasets are provided by *17Lands*[4]. Each dataset was split into 80% training and 20% test set. Note, however, that these numbers show the individual card picks. In our preference-based formulation, each pick results in 1–14 preferences, thus the total number of triples is even larger.

TABLE I
All card sets used in this research with their respective release data and training set size.

Set	Release date	Training size
LTR	23-06-23	684,724
MOM	21-04-23	5,085,312
SIR	21-03-23	2,422,668
ONE	10-02-23	5,260,169
BRO	18-11-22	4,153,162
DMU	09-09-22	7,887,976
HBG	07-07-22	1,680,866
SNC	29-04-22	5,753,840
NEO	18-02-22	5,122,921
VOW	19-21-21	4,012,657
MID	24-09-21	3,363,477
AFR	23-07-21	959,794
STX	23-04-21	3,809,102
M19	07-07-18	29,094,192

[2] https://www.17lands.com/card_data

[3] https://draftsim.com/draft-data/

[4] https://www.17lands.com/public_datasets

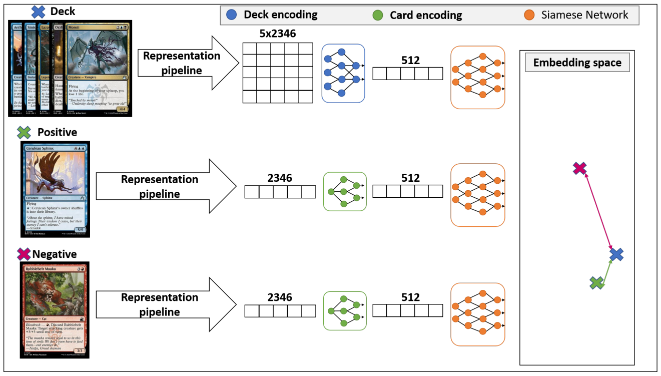


Fig. 4. General overview of training. Refer to Figure 1 for the explanation of the representation pipeline. Training loss is computed based on Equation (2).

B. Evaluation Measure

The accuracy of the model is defined as the *top-1 accuracy* when ranking all possible picks. To generate this, we embed the current deck and all possible options, rank all options by their distance to the deck in the embedding space, and choose the card with the minimal distance as shown in (3). The selected card is compared to the human choice captured in the dataset and the percentage of matching choices is reported. Note that, as this aims to predict human choices, it is generally not possible to achieve perfect accuracy, as human decisions are not necessarily correct or consistent.

C. Siamese Neural Network configuration

As explained in Section IV-C, using CPR with generalised card representation requires separate encoding networks for cards and decks. The card encoding network uses 4 fully-connected layers of size 1024 with dropout, normalisations and ELU activations. The deck encoding network consists of 6 convolutional layers with 1 to 16 filters, normalisations, maxpooling and ELU activations with a single fully-connected output layer. Both encoding networks feed into the same Siamese neural network, which uses 5 fully-connected layers of size 512, normalisations, dropout and ELU activations. The final output layer is of size 512 and uses the tangens hyperbolicus activation, thus creating a 512-dimensional embedding space of range $[-1, 1]$. For a general overview, refer to Figure 4.

VI. Results

Experiments in this section are split: Section VI-A investigates the different input encodings (see Section IV-B) and Section VI-B evaluates fine-tuning a pre-trained model on previously unseen cards, i.e. simulating the release of a new card set.

A. Representation

To investigate the influence of card representation (see Section IV-B) on the received models, we conduct a series of experiments with SNNs that only differ in the shape of the input signals. To reduce the computational effort of this experiment, models are trained solely on the *NEO* set. The accuracy of the received models is reported on held-out samples of the *NEO* set and on the average accuracy across all unseen sets. Section VI-B will report on experiments with the whole dataset to maximise predictive performance and simulate real-world use.

TABLE II
Top-1 testing accuracy of models with differently encoded inputs. All models are trained solely on *NEO* and tested on a held-out *NEO* test-set and unseen cards. Accuracy is averaged across all unseen sets.

Model	Input size	*NEO* test	unseen sets
One-hot	302	67.80%	NaN
Random	1024	67.87%	23.79%
Image *32*	32	65.93%	28.69%
Image *1024*	1024	68.09%	31.10%
Meta	16	64.73%	42.14%
Features	1306	67.76%	33.57%
Features + Meta	1322	68.07%	34.74%
Features + Image *1024*	2330	67.81%	35.59%
Features + Meta + Image *1024*	2346	68.00%	42.87%

In Table VI-A we see the test accuracies obtained on *NEO* and the unseen sets. Interestingly, the top-1 accuracy on the seen card set is consistent across most models apart from the two with small input spaces and thus limited information. However, accuracy on unseen sets, i.e. the ability of the model to learn general characteristics of cards, differs drastically. The *Random* model is still able to accurately model decisions in *NEO*, thus showing that with sufficient data on a set, generalised knowledge is not required. However, this model provides no insight into unseen cards where we receive the accuracy of random predictions[5].

All other representations are able to improve upon random predictions on unseen cards. Regarding the autoencoded image representations, we receive worse accuracy with the small latent space, likely due to the generally too-small input space. The more information-rich compression reaches high accuracy on the *NEO* test-set, as well as achieving some generalisation to the unseen sets. Surprisingly, hand-engineered features only receive slightly better general results, thus leading us to speculate that with limited data, models do not generate rich semantics of cards but rather use shallow features like their colour, which are easily recognisable from the image representations.

The *Meta* model using only 16 features based on meta attributes achieves among the best accuracy on unseen sets, underlining that when limited information is available, simply using card statistics leads to reasonable results. However, combining features and meta-information is not able to utilise this, barely improving upon the *Features* model. Similarly, combining *Features* and *Image 1024* only leads to slightly better performance on unseen cards. The reason for this is not

[5] Decisions follow a uniform random distribution with 1 to 15 choices, thus random predictions would yield an accuracy of $1/15 \cdot \sum_{i=1}^{15} 1/i \approx 0.22$. The observed accuracy for the *random* representation is slightly higher because some datasets are missing a small number of picks.

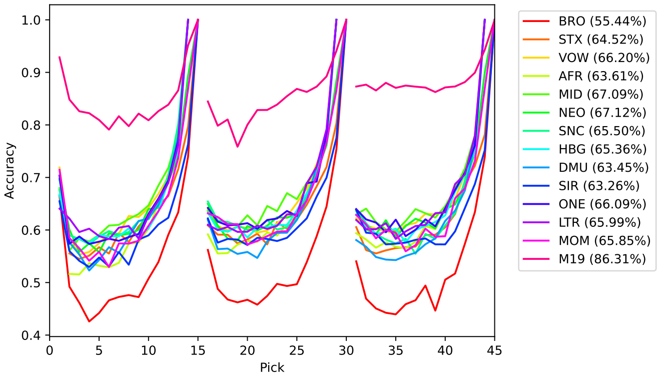


Fig. 5. Accuracy of *Features+Meta+Image* model trained on all sets apart from *BRO*, split by set and pick. Numbers in brackets show the average accuracy per set. Graphs are separated by the three packs.

clear to us and requires further investigation. The overall best-performing model is *Features + Meta + Image*, being able to predict almost 43% of decisions on completely unseen cards while retaining accuracy on *NEO*.

From this experiment, the initial card representation appears largely irrelevant when CPR is used to construct an embedding space for known cards. Even random vectors as card representations and one-hot encodings yield similar results as encoding a card's semantics. However, input representation clearly matters when aiming to generalise to new, previously unseen cards. Additionally, meta-information about a card's usage, if available, is highly relevant for unknown cards and already leads to reasonable results.

B. Transfer

Following these experiments, the *Features+Meta+Image* model with the highest combined accuracy is chosen for continued exploration. Our goal in this section is to model a real-world setting where a model is trained on all available card sets, and then tested on or adapted to a newly issued set of cards. To test this ability for transferring obtained knowledge to new card sets, we use all but one of the datasets listed in Table I for training. On the one hand, this can be expected to improve the overall performance of the model by increasing training set size and card heterogeneity, and on the other hand, it is also meant to simulate the intended application scenario, where we can assume to train on all available card sets and test on a newly issued set. To simulate a new card set being released, we hold out a single set of cards (*BRO*), reporting accuracy on held-out test sets by averaging across all seen sets and explicitly reporting the test set performance on *NEO* and *BRO*.

As expected, by simply increasing the sample quantity (75 million instead of 5 million) and unique card variety (2,990 cards instead of 302), the general accuracy of the model increases consierably (see Table VI-B). Although performance on the seen sets is comparable to the previous results, the model *Pre-training*, which is the same model as *Features+Meta+Image 1024* but trained on more data, achieves 55.44% accuracy on the unknown *BRO* set. Splitting the received accuracy over all sets and picks (Figure 5), we see similar patterns on all sets, regardless of whether the model was trained on it or not. The accuracy of picks at the start and end of each pack is higher, while picks in the middle appear more difficult. Compared to previous work with CPR [2], the accuracy on *M19* is improved by more than 2 percentage points.

Building on this, we explore fine-tuning or transfer learning [18], with the model on two individual sets, *NEO* and *BRO*. When fine-tuning, the large corpus of training data is replaced with a single set of cards, thus aiming to use previously obtained general knowledge for a smaller task. For our experiment, a model is initialised with the parameters obtained from pre-training on the large corpus and trained exclusively on *NEO* and *BRO* without further adjustments to the training process, omitting freezing layers or other attempts to knowledge retainment.

The two fine-tuning settings serve different purposes. When fine-tuning on *NEO*, no new data or cards are seen. Rather, the training data is reduced to decrease the task complexity, thus gaining the opportunity to improve the accuracy. Using *BRO* to fine-tune replaces previous data with unseen cards, thus investigating whether the model can quickly adjust to new information, simulating the release of a new set of cards in real applications.

TABLE III
TOP-1 TESTING ACCURACY OF MODELS TRAINED ON A LARGER CORPUS OF DATA.

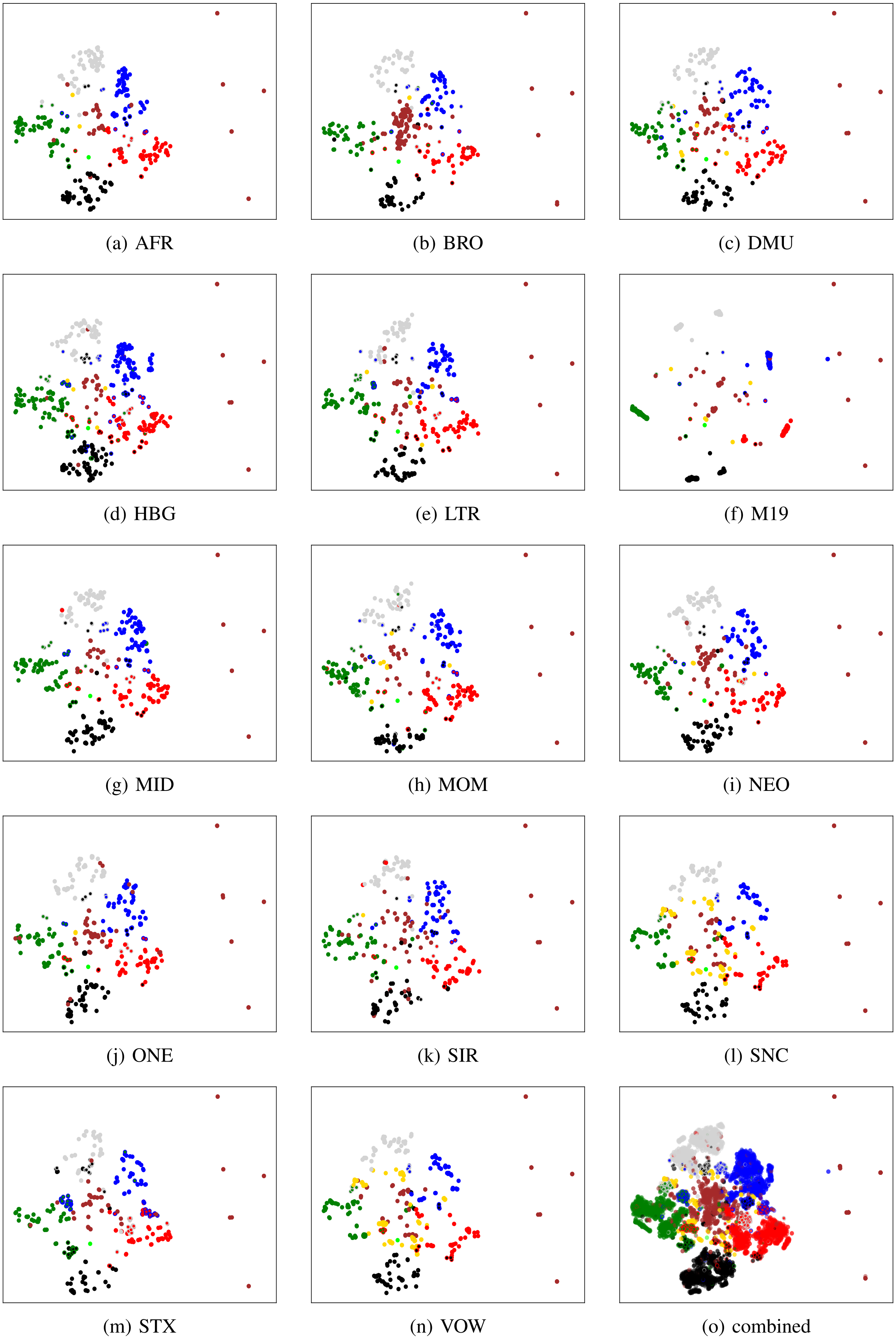
	Test data		
Training data	Seen sets hold-out	*NEO* test-set	*BRO*
Pre-training	66.49%	67.20%	55.44%
+ tuning on *NEO*	58.20%	67.97%	52.25%
+ tuning on *BRO*	57.97%	58.26%	62.27%

We find that fine-tuning on *NEO* barely improves accuracy on the set and does not achieve better overall results than random initialisation. This again shows that when regarding a known, fixed set of cards, generalised knowledge of card semantics seems to not affect accuracy. Additionally, when fine-tuning on one set, accuracy on the others diminishes. While potentially preventable by freezing layers or using small amounts of pre-training data, we omit such experiments.

Fine-tuning on *BRO* naturally leads to improved accuracy as the model was not trained with these cards previously. While we do not find higher peak accuracy compared to random initialisation, using the pre-trained model leads to much quicker adaption to new data (see Figure 7) and is thus useful for fast-paced environments.

VII. VISUALISATION

Although this work mainly aims to explore card representation and general accuracy of results when using CPR to learn a universal drafting model, CPR provides the inherent advantage of an intuitively interpretative embedding space because all



(a) AFR

(b) BRO

(c) DMU

(d) HBG

(e) LTR

(f) M19

(g) MID

(h) MOM

(i) NEO

(j) ONE

(k) SIR

(l) SNC

(m) STX

(n) VOW

(o) combined

Fig. 6. 2-Dimensional Representation of embedding space per set. This space is created with the *Pre-training* model (Table VI-B), thus cards visualised in (b) are unseen. However, a similar clustering structure emerges, clearly showing a generalised understanding of card semantics. The lime-green point denotes the embedding of an empty deck, thus each card's distance to it serves as an approximation of its absolute strength.

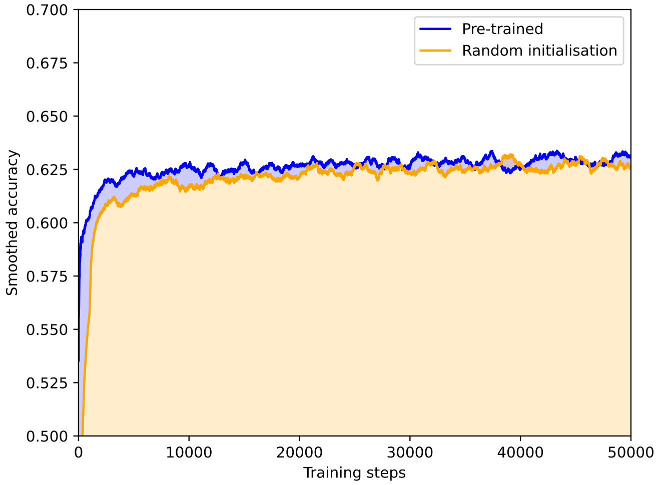


Fig. 7. Accuracy curve of pre-trained model and randomly initialised model on newly seen dataset *BRO*. The pre-trained model does not achieve higher peak accuracy but meaningfully improves training speed.

model predictions are directly based on minimising distances. As the embedding space is high-dimensional, we use TSNE [12] to compute a 2-dimensional approximation (see Figure 6) and separately plot each set of cards. Each card is plotted with its respective colour in the game (see Figure 2), one of the most defining features. For this plot, the *Pre-training* model from Section VI-B is used but all strong models provide similar visualisations.

Highly intuitive clusters emerge in this embedding space. All colours are grouped together and multiple-coloured cards, visible as either two-coloured points with different borders or as golden points for more than two colours, are placed between their respective colours. All plots feature 5 points at large distance from the main plot, which are the 5 cards included in every set that provide no value to gameplay and should thus never be picked. Interestingly, the *BRO* set, which consists of previously unseen cards, exhibits similar structures as the sets which were used for training the model, indicating a clear semantic understanding of the model.

VIII. Conclusion

In this work, we generalised the contextual preference ranking framework [2] by extending it to generalised card representations. The key idea is to replace the previously utilised one-hot encodings with alternative representations, consisting of card features, text-embeddings and auto-encoded image compressions of *Magic: The Gathering* cards. Following CPR, the card representations are mapped into a high-dimensional embedding space using Siamese neural networks where distances model the relationship between cards and decks. We find that the input representation is largely irrelevant when modelling singular, fixed sets. In such cases, where data for all cards is available, all models with reasonable inputs perform similarly well. However, large discrepancies in accuracy emerge when aiming to understand card semantics, e.g. when predicting decisions with previously unseen cards, a common occurrence when new cards are released. Here, we find that hand-engineered features of cards, combined with meta-statistics and auto-encoded images lead to the highest generalisation capabilities and provide quick fine-tuning opportunities when new data is available.

References

[1] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[2] T. Bertram, J. Fürnkranz, and M. Müller, "Predicting human card selection in *Magic: The Gathering* with contextual preference ranking," in *2021 IEEE Conference on Games (CoG)*. IEEE, 2021, pp. 1–8.

[3] ——, "Weighting information sets with Siamese neural networks in Reconnaissance Blind Chess," in *2023 IEEE Conference on Games (CoG)*. IEEE, 2023, pp. 1–8.

[4] S. J. Bjørke and K. A. Fludal, "Deckbuilding in *Magic: The Gathering* using a genetic algorithm," Master's thesis, NTNU, 2017.

[5] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "Siamese" time delay neural network," *Advances in Neural Information Processing Systems*, vol. 6, 1993.

[6] N. Brown and T. Sandholm, "Superhuman AI for multiplayer poker," *Science*, vol. 365, no. 6456, pp. 885–890, 2019.

[7] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *21st International Conference on Artificial Neural Networks (ICANN), Part I*. Espoo, Finland: Springer, 2011, pp. 52–59.

[8] R. Miernik and J. Kowalski, "Evolving evaluation functions for collectible card game AI," in *14th International Conference on Agents and Artificial Intelligence (ICAART), Volume 3*, 2022, pp. 253–260.

[9] M. Pawlicki, J. Polin, and J. Zhang, "Prediction of price increase for *Magic: The Gathering* cards," Class report, Stanford University, 2014. [Online]. Available: http://cs229.stanford.edu/proj2014/Matt\%20Pawlicki,\%20Joe\%20Polin,\%20Jesse

[10] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China, 2019, pp. 3980–3990.

[11] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[12] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[13] R. Vieira, A. R. Tavares, and L. Chaimowicz, "Drafting in collectible card games via reinforcement learning," in *2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. IEEE, 2020, pp. 54–61.

[14] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 3304–3308.

[15] H. N. Ward, B. Mills, D. J. Brooks, D. Troha, and A. S. Khakhalin, "Ai solutions for drafting in *Magic: The Gathering*," in *2021 IEEE Conference on Games (CoG)*. IEEE, 2021, pp. 1–8.

[16] W. Xi, Y. Zhang, C. Xiao, X. Huang, S. Deng, H. Liang, J. Chen, and P. Sun, "Mastering strategy card game (Legends of Code and Magic) via end-to-end policy and optimistic smooth fictitious play," *arXiv preprint arXiv:2303.04096*, 2023.

[17] C. Xiao, Y. Zhang, X. Huang, Q. Huang, J. Chen *et al.*, "Mastering strategy card game (Hearthstone) with improved techniques," in *2023 IEEE Conference on Games (CoG)*. IEEE, 2023, pp. 1–8.

[18] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[19] F. Zilio, M. Prates, and L. Lamb, “Neural networks models for analyzing *Magic: The Gathering* cards,” in *25th International Conference on Neural Information Processing (ICONIP), Part II*. Siem Reap, Cambodia: Springer, 2018, pp. 227–239.