The Workshops of the Thirty-Second
AAAI Conference on Artificial Intelligence

Combinatorial Creativity for Procedural Content Generation via Machine Learning

Matthew Guzdial, Mark O. Riedl
School of Interactive Computing
Georgia Institute of Technology
mguzdial3@gatech.edu, riedl@cc.gatech.edu

Abstract

In this paper we propose the application of techniques from the field of creativity research to machine learned models within the domain of games. This application allows for the creation of new, distinct models without additional training data. The techniques in question are combinatorial creativity techniques, defined as techniques that combine two sets of input to create novel output sets. We present a survey of prior work in this area and a case study applying some of these techniques to pre-trained machine learned models of game level design.

Introduction

Procedural content generation (PCG) represents a set of varied approaches in which a designer encodes some design knowledge in an algorithm, which then generates novel video game content (Hendrikx et al. 2013). This set of techniques has some drawbacks, in particular that it typically requires high quality design knowledge to function. Procedural content generation via machine learning (PCGML) makes use of knowledge extraction from games via machine learning as an alternative to hand-coded design knowledge. However, PCGML shares the limitations of machine learning approaches. Namely a dependence on the type and quantity of training data available. Most games have a limited amount of content of a limited number of types, meaning that knowledge extracted from a game is limited in terms of scale and descriptive power. Alternatively, one could consider training on content from multiple games, but individual games tend to vary too much to apply machine learning techniques (Summerville et al. 2017). In addition, extracting knowledge from games is not a fully automated process, requiring that a human designer implement a game-specific knowledge scraping tool, seek out some general representation (Guzdial and Riedl 2016a), or make use of a pre-authored corpus, which only shifts the burden to an earlier designer (Summerville et al. 2016). We propose a solution to both of these problems, recombining learned models to create novel models to maximize the generative space while minimizing the required training data and human effort.

Computational creativity is a field that represents the intersection of artificial intelligence and creativity (Colton and Wiggins 2012). We focus on a set of techniques within this field referred to as combinatorial creativity techniques. These techniques recombine existing knowledge to create new usable knowledge, relying on structure from the original knowledge representations in order to create new knowledge representations with similar value (Boden 2004). For example, consider combining the concepts of "man" and "wolf" to create the concept "werewolf", one can use similarities between the original concepts to ensure a consistent, novel concept. One could consider applying these techniques to recombine knowledge extracted from games, but these techniques historically have required hand-authored knowledge representations, which differ from typically messy machine learned models. Alternatively one could directly recombine knowledge extracted from a game (levels, rules, etc.) rather than a model trained on that content. However, machine learned models typically capture a higher abstraction of structure, which benefits combinatorial creativity techniques in terms of the size of output content and likelihood of valid structure.

In this paper we propose applying combinatorial creativity techniques to machine learned models of knowledge extracted from games. We survey related work in this area, including prior applications of combinatorial creativity techniques to games. We present formalizations of three historical combinatorial creative techniques and one novel technique. In a case study on preexisting machine learned level design models extracted from games we demonstrate the trade-offs of the various approaches in terms of novelty and value of their output. We end with a discussion of remaining work.

Related Work

There have been many approaches to combinatorial creativity over the years, which we briefly summarize. Case-based reasoning (CBR) represents a general AI problem solving approach that relies on the storage, retrieval, and adaption of existing solutions (De Mantaras et al. 2005). The adaption function has lead to a large class of combinatorial creativity approaches, which we can place into the categories of substitutional adaption and structural adaption (Wilke and Bergmann 1998; Fox and Clarke 2009). These techniques tend to be domain-dependent, for example for the problem of text generation or tool creation (Hervás and Gervás

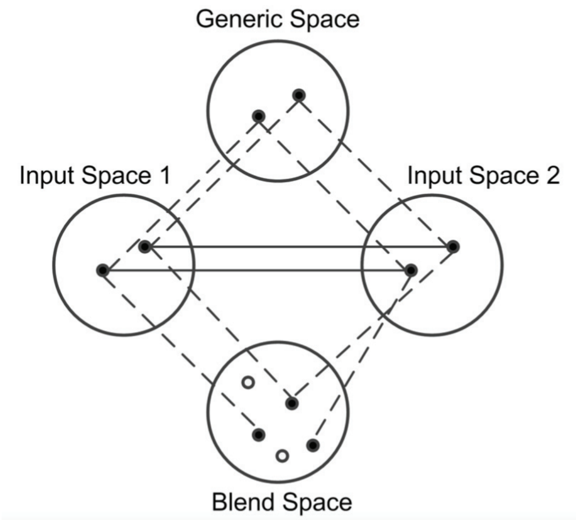

Figure 1: Representation of the canonical four space concept blending approach from (1998).

2006; Sizov, Öztürk, and Aamodt 2015). Murdock and Goel (2001) combine reinforcement learning with case-based reasoning, which aligns with our work to combine computational creativity and machine learning research. However, the technique does not look to automatically derive cases or concept spaces and then combine these structures.

Beyond CBR there exists a wide range of combinatorial creativity techniques. The area of belief revision, modeling how beliefs change, includes a function to merge prior existing beliefs with new beliefs(Konieczny, Lang, and Marquis 2004; Steels and De Beule 2006; Cojan and Lieber 2008; 2009; Konieczny and Pérez 2011). Belief merging has been applied to CBR as an adaption function (Cojan and Lieber 2012), further aligning it as a combinatorial creativity technique. However, due to the nature of its work it focuses on a single final merged belief state, and we are interested in larger output sets. The mathematical notion of convolution has been applied to blend weights between two neural nets in work that parallels our desire to combine computational creativity and ML, but without promising results (Thagard and Stewart 2011).

We identify three specific combinatorial creativity techniques for deeper investigation, due to the fact that they are well-formed in domain-independent terms, with distinct and large spaces of output combinations: concept blending, amalgams, and compositional adaption.

Concept Blending

Fauconnier and Turner (1998) formalized the "four space" theory of concept blending. In this theory they described four spaces that make up a blend as seen in Figure 1: two *input spaces* represent the unblended elements, input space points are projected into a common *generic space* to identify equivalence, and these equivalent points are projected into a *blend space*. In the blend space, novel structure and patterns arise from the projection of equivalent points. Fauconnier and Turner (1998; 2002) argued this was a ubiquitous process, occurring in discourse, problem solving, and general meaning making.

Concept blending typically requires a large amount of human authoring for individual concept spaces. Recently O'Donoghue et al. (2015) have looked into deriving this knowledge automatically from text corpora, producing graphical representations of nodes and their verb connections. Our own work runs parallel to O'Donoghue et al., but in the domain of two dimensional video games levels and without the dependency rules that exist in the english language. There has been work in blending individual tagged exemplars together based on surface level features of components (Alhashim et al. 2014).

Fauconnier and Turner originally developed a set of heuristics for domain-independent measures of quality for blends. As an alternative more recent work has looked to the introduction of goals for blends (Li et al. 2012).

Amalgamation

Ontañón designed amalgams as a formal unification function between multiple cases (Ontañón and Plaza 2010). Similar to concept blending, amalgamation requires a knowledge base that specifies when two components of a case share a general form, for example "French" and "German" can both share the more general form "nationality". Unlike concept blending, this shared generalization does not lead to a merging of components, but requires that only one of the two can be present in a final amalgam. For example, a "red French car" and a "old German car" could lead to an "old red French car" or an "old red German car".

Amalgams have been utilized as the adaption function in CBR systems (Manzano, Ontanón, and Plaza 2011), combined with concept blending for product development (Besold and Plaza 2015), and adapted to an asymmetrical form for story generation (Ontanón, Zhu, and Plaza 2012). Amalgamation represents a strong general method for combinatorial creativity. However it suffers from the same fallbacks of other adaption methods in terms of a traditional reliance on authored knowledge bases and domain-specific generalization.

Compositional Adaption

Compositional adaption arose as a CBR adaption approach (Holland 1989; Fox and Clarke 2009), but has found significant applications in adaptive software (McKinley et al. 2004; Eisenbach, Sadler, and Wong 2007). The intuition behind compositional adaption is that individual component cases can be broken apart and reused based on their connections. In adaptive software this takes the shape of sets of functions with given inputs and outputs that can be strung together to achieve various effects, which makes compositional adaption similar to planning when it includes a goal state or output. However, it can also be applied in a goal-less way to generate sequences of components.

Compositional adaption has been applied to recipe generation (Müller and Bergmann 2014; Badie and Mahmoudi 2017), intelligent tutoring systems (Reyhani, Badie, and Kharrat 2003), and to traditional CBR approaches

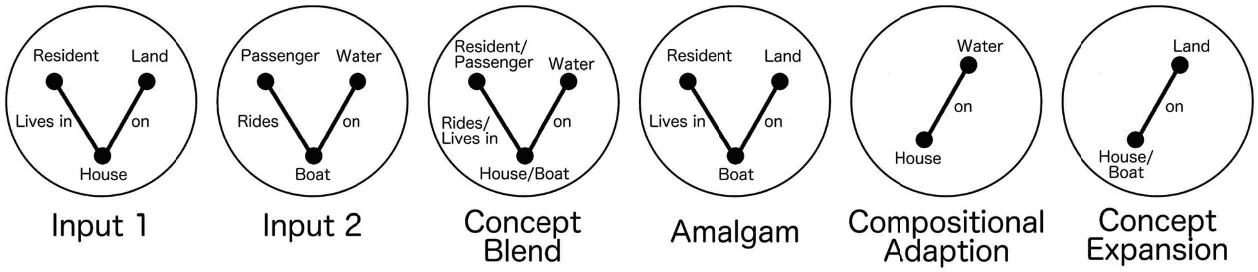

Figure 2: Example of the four combinatorial creativity techniques. Two input spaces on left with example output from the four techniques on the right.

(Chedrawy and Abidi 2006). Unlike amalgamation and concept blending, compositional adaption does not require an explicit knowledge base by default. However, it is common to make use of the knowledge base to generalize components and their relationships in order to expand the set of possible recombinations. As with the prior combinatorial creativity approaches, this knowledge base is almost always human-authored.

Computational Creativity and Games

Combinatorial creativity and computational creativity research in general are rarely applied to video games. Prior work has looked into knowledge intensive concept blending systems to create new elements of video games such as sound effects and 3D models (Ribeiro et al. 2003; Martins et al. 2004). The Game-O-Matic system made use of concept mapping to match verbs onto game mechanics to create arcade-style games based on human-authored mapping knowledge (Treanor et al. 2012). Gow and Corneli (2015) proposed a system to generate small games via amalgamation. Permar and Magerko (2013) presented a system to produce novel interactive narrative scripts via concept blending, using analogical processing. The work presented in this paper focuses on a two-dimensional platformer game, a very different domain. Most related in terms of computational creativity, Cook et al.'s ANGELINA system produces entire video games with a computational creativity bent, though it does not make use of machine learning or combinatorial creativity (Cook, Colton, and Gow 2017).

Snodgrass and Ontañón leveraged transfer learning to train a machine learning approach on levels from two separate platform games, meant to address a lack of training data, but with the effect of combining level structure in the generated levels (Snodgrass and Ontañón 2016).

Approaches

In this section we present brief descriptions of our implementations of three historical and one novel combinatorial creativity techniques. We chose the three historical techniques, concept blending, amalgamation, and compositional adaption, due to their relative domain independence. The last technique, conceptual expansion, can be understood as a hybrid technique combining elements of concept blending and compositional adaption. For a grounded example of these approaches in terms of their output see Figure 2.

For this implementation we make use of a graph-based representation, which assumes individual nodes with edges connecting nodes with some relationship. At one level of abstraction all of these approaches follow the same process. First, take in two graphs as input, as seen on the left side of Figure 2, Second, construct a mapping between elements of the two graphs. This mapping typically requires some outside knowledge, either in the form of hierarchical information (e.g. houses and boats can be said to have the same parent object), distance functions, or an authored mapping. Alternatively, mapping can make use of graph structure, looking for similarities in node and edge relationships, irrespective of the semantic meaning of the node and edge values. Finally, once a mapping is reached each approach makes use of a unique algorithm to construct final combinations. Notably there are typically many possible final combinations for each approach, depending on the two input spaces and constructed mapping. In addition, this is a traditional usage example for illustration purposes, while we propose utilizing machine learned models as input.

We now list the four techniques in terms of the algorithmic approach each uses to construct combinations in the third step of the process outlined above. We present each technique in terms of its algorithmic complexity. For the purposes of the example we assume the following mappings of the two inputs presented in Figure 2: *house* and *boat*, *live in* and *ride*, *passenger* and *resident*, and *land* and *water* as taken from (Goguen 2006).

- **Amalgamation**: Amalgamation, the process that produces amalgams, considers mapped elements to be incapable of co-existing in a final amalgam. In the example of Figure 2 house and boat could not both be present in a final amalgam. Instead amalgamation first chooses one of each of the mapped elements, then adds all of these chosen elements and as many of the non-mapped elements as possible based on node-edge relationships to the final output graph. We can understand amalgamation as most like the every day usage of the term combination.
- **Conceptual Blending**: Concept blending produces blends. As described in the related work section, historically concept blending relies on four distinct spaces: the

two input spaces, a generic space, and the actual blend space. The abstract mapping stage we discussed above can be understood as the mapping into a generic space. From this generic space mapping the blending algorithm derives a final mapping to apply in the blend space, which can vary from an empty mapping to the full generic space mapping. For example in Figure 2 the displayed output is from a blend space mapping equivalent to the generic space mapping except for the mapping of land and water as discussed in (Goguen 2006).

All individual mapped elements are combined into a single new blended element, such as *house/boat* and *rides/lives in*. The final blend is composed of all of these blended elements and as many of the non-blended elements as possible given open node-edge relationships. The goal is to maximize the shared information between the blend and the two input spaces. The choice of what blend space mappings to apply represents the majority of conceptual blending's expressive power.

- **Compositional Adaption**: Compositional adaption produces compositions. In compositional adaption the nodes and edges from the original input spaces are broken apart into individual pieces. These pieces can then be strung back together to create compositions based on the given mapping. For example the edge *on* from Input 2 in Figure 2 connects to nodes *boat* and *water*. But because of the mapping of *house* and *boat*, the compositional adaption can attach the node *house* in the place of *boat*, leading to the composition "house on water". The process can be understood as randomly choosing nodes and edges to add to an initially empty composition based on currently open slots. This process can stop whenever there are no unconnected edges, meaning we can end up with smaller graphs than the input spaces, which accounts for most of compositional adaptions expressive power. In many cases the output of compositional adaption can be understood as a superset of the output of amalgamation.
- **Conceptual Expansion**: Conceptual expansion is a process that produces combinations referred to as expansions. Expansion in this case refers to a more general form of blending between mapped elements, in which we define N variables [0,1] representing the amount to which each of the N mapped elements are expressed in the expanded element. For example if we imagine two mapped nodes, one with a value 5.0 and one with a value -5.0, we can imagine a number of expanded nodes with all possible values between 5.0 and -5.0. If the node values were categorical values of A and B we could expand final nodes of either A, B, or AB. The process begins by randomly generating either a node or edge, and then expanding appropriate nodes or edges based on unconnected relationships.

  We can understand conceptual expansion in the case with all categorical values as a hybridization of conceptual blending and compositional adaption. In particular conceptual expansion creates the individual blended elements based on the given mappings as in conceptual blending, then constructs expansions piece-by-piece according to these mappings as in compositional adaption. We can understand conceptual expansion's output as representing the superset of output of both of these techniques, which further includes output neither technique could produce individually as seen in Figure 2.

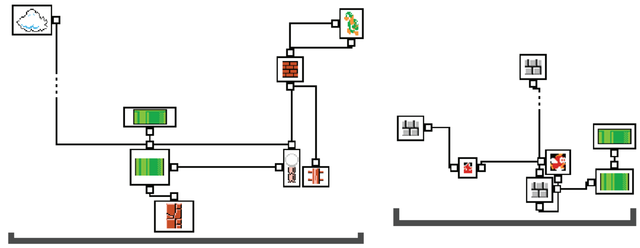

Figure 3: Example of the machine-learned level graphs for overworld (left) and castle (right) type levels, reproduced with permission from (Guzdial and Riedl 2016b).

Case Study: Combining Machine-learned Game Level Models

In this paper we propose applying combinatorial creativity techniques to machine learned models in order to generate new models without additional training data. As an illustration of the potential for this, and to give a deeper understanding of these techniques we ran a case study applying combinatorial creativity techniques to machine-learned models of Super Mario Bros. levels. In particular we drew on the models applied to a prior approach at marrying machine learning and creativity techniques, Guzdial and Riedl's (2016b) probabilistic graphical models. We lack the space to fully describe these models, but they can be understood as graphical models that probabilistically specify the relative positions of level components. The graphical nature of these models makes for a simple adaption of the techniques discussed in this paper. Notably we remove the weights from the edges of these models as all of the historic combinatorial creativity techniques were not designed to handle numeric variables.

We drew on five of the Guzdial and Riedl (2016b) learned graphs for this case study representing each class of Super Mario Bros. level: overworld, underground, athletic (sometimes called treetop levels), castle, and underwater. We visualize the overworld and castle graph in Figure 3. We pulled from the simplest machine-learned graphs of each type, the graphs having an average size of six nodes and seven edges. For mapping, we made use of hierarchical information as our mapping strategy with hierarchical information adapted from the classes used by Summerville and Mateas (2016) (e.g. "enemy" as a parent of all game enemies, "solid" as a parent of all unbreakable level components, etc.). We made use of every pair of input graphs, making for a total of ten pairs, and generated all possible unique output combinations for each approach. In the case of blended elements we combined the sets of shapes and relationships contained in the original graphical model's nodes and recalculated the probability distributions based on these sets. For further details please see (Guzdial and Riedl 2016b).

To classify the output we measured a normalized edit dis-

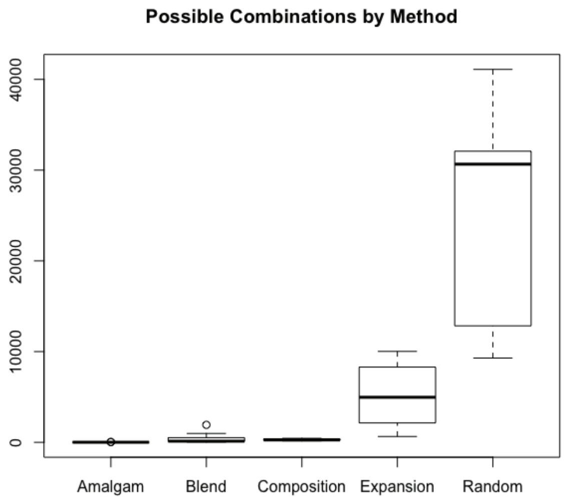

Figure 4: Output counts for each of the approaches for all the combination tasks.

tance from the two input graphs as graph novelty. We note that we are measuring novelty from both inputs, we will never see a novelty value of 0 as some amount of the output will be shared by the input. Given that these are generative level design models we also measured the playability value of each output graph by generating a hundred levels with each output and determining the percentage that could be completed with an A* agent.

We created a random baseline for comparison purposes by creating a random graph and assigning random values from the input elements and all possible blends of the input elements to the edges and nodes of each graph. The output of this approach can be understood as a superset of all possible output of the other approaches, but without any of the benefits that might come from leveraging the existing structure of the input graphs. Notably we limited the size of the possible graphs to the maximum size of the two input graphs.

Case Study Results

We present the total number of unique outputs given the input of all ten combinations by each approach in Figure 4. The random baseline produces by far the most unique output graphs, followed by conceptual expansion, and then the remaining three approaches. While it is difficult to tell on the same graph we note that compositional adaption and conceptual blending both produced significantly more output than amalgamation based on the pairwise Wilcoxon Mann-Whitney U test ($p < 0.001$). However conceptual blending and compositional adaption did not produce significantly different numbers of unique output graphs.

Given the massive disparity in the size of the output, we chose to break the output into four categories based on the metrics of novelty and playability, defined according to low vs. high novelty (with a cut-off of 0.5) and low vs. high playability (with a cut-off of 0.5 or 50% playable levels). We present the average percentage of unique output in each of these categories by approach in Table 1 (with playability value represented as P and novelty represented as N) and the average values of unique output per combination in Table 2. Following the practice of expressive range (Smith and Whitehead 2010), we present scatterplots of our generators output in Figure 5 of playability (x-axis) and novelty (y-axis). These values reflect our intuitions of the approaches. For example: output amalgams have generally high value but low novelty, blends have consistently high value, and compositions have a bias towards high value and high novelty. This represents a clear trade-off in terms of approach, and can potentially inform when each should be used. Expansions are more even across all four categories, with a bias towards higher novelty. In addition conceptual expansion produces nearly ten times more high quality output, producing 15,290 highly playable and novel expansions in comparison to 1,800 compositions, 1,820 blends, and only 10 amalgams in this category. The random output has a strong bias towards non-playable, highly novel output, which matches our expectation given it doesn't make use of any of the structural information from the inputs.

Table 1: Output summary of percentages of the different approaches.

Approach	Low P/Low N	Low P/High N	High P/Low N	High P/High N
Amalgams	3.57%	4.23%	**86.3%**	5.88%
Blends	6.74%	12.6%	36.7%	**43.9%**
Compositions	3.33%	25%	9.17%	**62.5%**
Expansions	10.5%	**46.8%**	14.5%	30.7%
Random	2.5%	**73.5%**	1.07%	23.6%

Table 2: Output summary of the average whole values of the different approaches.

Approach	Low P/Low N	Low P/High N	High P/Low N	High P/High N
Amalgams	1	1	**11**	1
Blends	28	52	152	**182**
Compositions	10	72	26	**180**
Expansions	523	**2,331**	722	1,529
Random	621	**18,268**	266	5,866

Case Study Illustrative Output

In order to give a more in depth understanding as to the performance of these different approaches we present an illustrative example of a high novelty, high playability output for each of the combinatorial creativity techniques for the overworld castle blend. We also include a single generated output level sections for each output graph, though we note each graph is capable of generating a variety of level structure. As one can see in Figure 6 each graph is distinct. The amalgam and blend are the most complex, due to the requirement that they make use of the maximum amount of elements possible. Of the remaining three, the composition and expansion differ in that the expansion has a blended element with the two "enemies" combined. Lastly, while the random output is technically novel and playable it involves merging clouds with the ground and having floating fences, which leads to generated levels that differ significantly from the standard Super Mario Bros. levels.

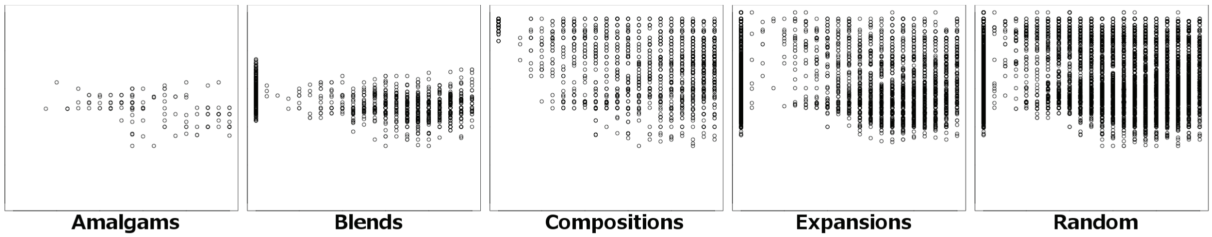

Figure 5: Scatterplots of the output values for each approach, the x-axis is playability [0,1] and the y-axis is novelty [0,1].

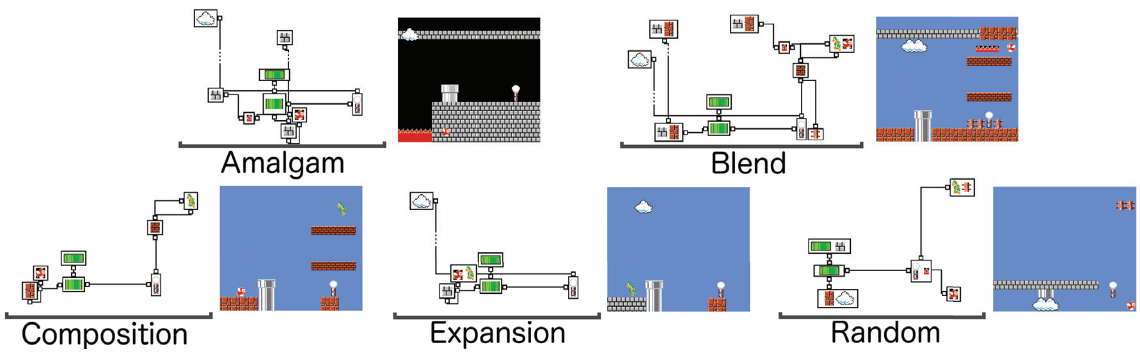

Figure 6: Illustrative output of a high novelty, high playability output graph for each technique for the overworld-castle combination, with an associated generated output level chunk.

Discussion

The initial results from our case study identify cases in which each presented combinatorial creative technique could be applied. Amalgamation presents a safe option when one lacks the ability to differentiate between valuable and non-valuable output, but only produces a small number of additional unique models. Concept blending and compositional adaption appear similar in the table representations, but significantly different output spaces when visualized. Conceptual blending produces output similar to amalgams, though an order of magnitude more results. Compositional adaption produces roughly equivalent output to concept blending, but a much larger variety in terms of model novelty. We note that there are domains that are more or less suited to the notion of blended elements. For example, when the input variables have numeric values (e.g. does 3 and 4 make 7, 34, 43, or something entirely different). Lastly, conceptual expansion produced by far the most unique output graphs, and could therefore prove helpful in cases where one can differentiate between low and high quality content and desires an order of magnitude increase in the number of output models.

We note there are limitations with this case study, particularly with our choice of playability as a metric. Even if only twenty percent of the output from a level design model is playable, if one can differentiate between playable and unplayable content then the unplayable output can be avoided. Further, playability does not account for many important aspects of a level, such as style and subjective player experience. One can see this most clearly in the results of the random baseline, which produced many levels that were technically playable but from the authors' perspective in no way resembled Super Mario Bros. levels. We note this case is partially but not completely captured by novelty, as identical elements arranged in entirely new ways would still represent a low novelty score. Despite this we find that playability is still a helpful comparative metric at least for this initial case study.

We do not anticipate combinatorial creativity techniques to function in all circumstances. As noted above they require inputs, some ability to map between the inputs, and some way to evaluate output combinations. These limitations offer some ability to influence the performance of the presented methods, for example different mappings will lead to very different output.

Conclusions and Future Work

In this paper we propose the application of combinatorial creativity techniques to machine learned models derived from knowledge extracted from games. In particular we propose applying combinatorial creativity techniques to maximize the expressive power of PCGML given minimal training data. Towards this end we present a brief survey and case study applying combinatorial creativity techniques to machine learned models of level design. We note a single case study is inconclusive, but we present some evidence that these techniques can be used to create orders of magnitude more novel, high quality level design models.

In the future we look to apply combinatorial creativity techniques to other types of machine learned models trained on knowledge extracted from games. In particular we're interested in non-level game content such as game rules and aesthetic content. Outside of PCGML we intend to investigate the ability for combinatorial creativity to aide in automated game playing, adapting prior game playing agent knowledge to novel environments.

Acknowledgments

We gratefully acknowledge the NSF for supporting this research under NSF award 1525967.

References

Alhashim, I.; Li, H.; Xu, K.; Cao, J.; Ma, R.; and Zhang, H. 2014. Topology-varying 3d shape creation via structural blending. *ACM Transactions on Graphics (TOG)* 33(4):158.

Badie, K., and Mahmoudi, M. T. 2017. Compositional adaptation in case-based reasoning based on the semantic relations between the components in the cases. In *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, 449–454. IEEE.

Besold, T. R., and Plaza, E. 2015. Generalize and blend: Concept blending based on generalization, analogy, and amalgams. In *ICCC*, 150–157.

Boden, M. A. 2004. *The creative mind: Myths and mechanisms*. Psychology Press.

Chedrawy, Z., and Abidi, S. R. 2006. Case based reasoning for information personalization: using a context-sensitive compositional case adaptation approach. In *Engineering of Intelligent Systems, 2006 IEEE International Conference on*, 1–6. IEEE.

Cojan, J., and Lieber, J. 2008. Conservative adaptation in metric spaces. In *European Conference on Case-Based Reasoning*, 135–149. Springer.

Cojan, J., and Lieber, J. 2009. Belief merging-based case combination. In *ICCBR*, 105–119. Springer.

Cojan, J., and Lieber, J. 2012. Belief revision-based case-based reasoning. In *Proceedings of the ECAI-2012 Workshop SAMAI: Similarity and Analogy-based Methods in AI*, 33–39.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proceedings of the 20th European conference on artificial intelligence*, 21–26. IOS Press.

Cook, M.; Colton, S.; and Gow, J. 2017. The angelina videogame design system, part i. *IEEE Transactions on Computational Intelligence and AI in Games*.

De Mantaras, R. L.; McSherry, D.; Bridge, D.; Leake, D.; Smyth, B.; Craw, S.; Faltings, B.; Maher, M. L.; T COX, M.; Forbus, K.; et al. 2005. Retrieval, reuse, revision and retention in case-based reasoning. *The Knowledge Engineering Review* 20(3):215–240.

Eisenbach, S.; Sadler, C.; and Wong, D. 2007. Component adaptation in contemporary execution environments. In *Distributed Applications and Interoperable Systems*, 90–103. Springer.

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

Fauconnier, G., and Turner, M. 2002. *The way we think: Conceptual blending and the mind's hidden complexities*. Basic Books.

Fox, J., and Clarke, S. 2009. Exploring approaches to dynamic adaptation. In *Proceedings of the 3rd International DiscCoTec Workshop on Middleware-Application Interaction*, 19–24. ACM.

Goguen, J. 2006. Mathematical models of cognitive space and time. In *Reasoning and Cognition: Proc. of the Interdisciplinary Conference on Reasoning and Cognition*, 125–128.

Gow, J., and Corneli, J. 2015. Towards generating novel games using conceptual blending. In *Proceedings of the 11th AIIDE Conference*.

Guzdial, M., and Riedl, M. 2016a. Game level generation from gameplay videos. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Guzdial, M., and Riedl, M. 2016b. Learning to blend computer game levels. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*.

Hendrikx, M.; Meijer, S.; Velden, J. V. D.; and Iosup, A. 2013. Procedural content generation for games: A survey. *ACM Trans. Graph.* 9(1):1:1–1:22.

Hervás, R., and Gervás, P. 2006. Case-based reasoning for knowledge-intensive template selection during text generation. In *European Conference on Case-Based Reasoning*, 151–165. Springer.

Holland, J. H. 1989. *Induction: Processes of inference, learning, and discovery*. Mit Press.

Konieczny, S., and Pérez, R. P. 2011. Logic based merging. *Journal of Philosophical Logic* 40(2):239–270.

Konieczny, S.; Lang, J.; and Marquis, P. 2004. Da2 merging operators. *Artificial Intelligence* 157(1-2):49–79.

Li, B.; Zook, A.; Davis, N.; and Riedl, M. O. 2012. Goal-driven conceptual blending: A computational approach for creativity. In *Proceedings of the 2012 International Conference on Computational Creativity, Dublin, Ireland*, 3–16.

Manzano, S.; Ontanón, S.; and Plaza, E. 2011. Amalgam-based reuse for multiagent case-based reasoning. In *International Conference on Case-Based Reasoning*, 122–136. Springer.

Martins, J.; Pereira, F.; Miranda, E.; and Cardoso, A. 2004. Enhancing sound design with conceptual blending of sound descriptors. In *Proceedings of the 1st joint workshop on computational creativity*.

McKinley, P. K.; Sadjadi, S. M.; Kasten, E. P.; and Cheng, B. H. 2004. A taxonomy of compositional adaptation. *Rapport Technique numéroMSU-CSE-04-17*.

Müller, G., and Bergmann, R. 2014. Compositional adaptation of cooking recipes using workflow streams. In *Computer cooking contest, workshop proceedings ICCBR*.

Murdock, J., and Goel, A. 2001. Meta-case-based reasoning: Using functional models to adapt case-based agents. *Case-based reasoning research and development* 407–421.

O'Donoghue, D. P.; Abgaz, Y.; Hurley, D.; and Ronzano, F. 2015. Stimulating and simulating creativity with dr inventor. In *Proceedings of the 6th ICCC*.

Ontañón, S., and Plaza, E. 2010. Amalgams: A formal approach for combining multiple case solutions. In *Case-Based Reasoning. Research and Development*. Springer. 257–271.

Ontanón, S.; Zhu, J.; and Plaza, E. 2012. Case-based story generation through story amalgamation. In *Proceedings of the ICCBR 2012 Workshops*, 223–232.

Permar, J., and Magerko, B. 2013. A conceptual blending approach to the generation of cognitive scripts for interactive narrative. In *Proceedings of the 9th AIIDE Conference*.

Reyhani, N.; Badie, K.; and Kharrat, M. 2003. A new approach to compositional adaptation based on optimizing the global distance function and its application in an intelligent tutoring system. In *Information Reuse and Integration, 2003. IRI 2003. IEEE International Conference on*, 285–290. IEEE.

Ribeiro, P.; Pereira, F. C.; Marques, B.; Leitao, B.; and Cardoso, A. 2003. A model for creativity in creature generation. In *Proceedings of the 4th GAME-ON Conference*, 175.

Sizov, G.; Öztürk, P.; and Aamodt, A. 2015. Evidence-driven retrieval in textual cbr: bridging the gap between retrieval and reuse. In *International Conference on Case-Based Reasoning*, 351–365. Springer.

Smith, G., and Whitehead, J. 2010. Analyzing the expressive range of a level generator. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, 4. ACM.

Snodgrass, S., and Ontañón, S. 2016. An approach to domain transfer in procedural content generation of two-dimensional videogame levels. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Steels, L., and De Beule, J. 2006. Unify and merge in fluid construction grammar. *EELC* 4211:197–223.

Summerville, A., and Mateas, M. 2016. Super mario as a string: Platformer level generation via lstms. In *Proceedings of the First International Conference of DiGRA and FDG*.

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

Summerville, A.; Snodgrass, S.; Guzdial, M.; Holmgård, C.; Hoover, A. K.; Isaksen, A.; Nealen, A.; and Togelius, J. 2017. Procedural content generation via machine learning (pcgml). *arXiv preprint arXiv:1702.00539*.

Thagard, P., and Stewart, T. C. 2011. The aha! experience: Creativity through emergent binding in neural networks. *Cognitive science* 35(1):1–33.

Treanor, M.; Blackford, B.; Mateas, M.; and Bogost, I. 2012. Game-O-Matic: Generating videogames that represent ideas. In *Proceedings of the 3rd FDG working on Procedural Content Generation*.

Wilke, W., and Bergmann, R. 1998. Techniques and knowledge used for adaptation during case-based problem solving. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 497–506. Springer.