Future Research Avenues for Artificial Intelligence in Digital Gaming: An Exploratory Report

Markus Dablander[*]
Commissioned by Beam Foundation[†] as Independent Contract Artificial Intelligence Researcher

Abstract

Video games are a natural and synergistic application domain for artificial intelligence (AI) systems, offering both the potential to enhance player experience and immersion, as well as providing valuable benchmarks and virtual environments to advance AI technologies in general. This report presents a high-level overview of five promising research pathways for applying state-of-the-art AI methods, particularly deep learning, to digital gaming within the context of the current research landscape. The objective of this work is to outline a curated, non-exhaustive list of encouraging research directions at the intersection of AI and video games that may serve to inspire more rigorous and comprehensive research efforts in the future. We discuss (i) investigating large language models as core engines for game agent modelling, (ii) using neural cellular automata for procedural game content generation, (iii) accelerating computationally expensive in-game simulations via deep surrogate modelling, (iv) leveraging self-supervised learning to obtain useful video game state embeddings, and (v) training generative models of interactive worlds using unlabelled video data. We also briefly address current technical challenges associated with the integration of advanced deep learning systems into video game development, and indicate key areas where further progress is likely to be beneficial.

[*] 🎓 Google Scholar | LinkedIn | GitHub
[†] 🏠 Homepage | 𝕏

1 Introduction

In the last decade, the rise of advanced neural network architectures has led to a series of dramatic breakthroughs in the fields of machine learning and artificial intelligence (AI). The GPU-accelerated training of large, carefully designed deep learning models has enabled researchers to tackle previously intractable challenges in diverse areas such as computer vision [1, 2, 3, 4, 5], natural language processing [6, 7, 8, 9], artificial content generation [10, 11, 12, 13], and computational chemistry [14, 15, 16, 17, 18]. One exceptionally promising and natural application area for modern deep learning, which will be explored in this report, is **digital gaming**.

The focus of AI research on games already has a long and important history. In particular, the study of classical board games such as Chess, Checkers, and Go has been formative and instrumental for the AI field as a whole [19, 20]. The highly structured nature of many games allows for the emergence of great complexity and strategic depth from simple rules that can easily be expressed in a computational framework; consequently, games have long been considered ideal testing grounds for the reasoning and planning capabilities of AI agents. A significant milestone was reached in 2016, when the first AI system achieved superhuman performance in the game of Go [21], which, at that time, represented the last major, popular board game in which human experts still outperformed computers.

Particularly since then, digital games have increasingly been recognised as one of the next great frontiers of AI research. In recent years, considerable progress has been made towards developing AI agents capable of mastering real-time strategy video games, such as StarCraft II [22], and multiplayer online battle arena video games, such as Dota 2 [23], both of which pose a far greater challenge to AI systems than classical board games. Simultaneously, the construction of *general* AI models that can learn to play multiple, qualitatively distinct arcade video games has emerged as an active field of research [24, 25, 26], and work in this area may serve as a stepping stone towards the development of more general AI systems in other domains.

Importantly, it is not merely the case that video games have the potential to enrich contemporary AI research; the converse is true as well. The relationship between AI research and digital gaming is mutual and synergistic [19, 20, 27], with video games providing valuable benchmarks, test-beds and virtual environments for novel AI systems, while novel AI systems, in turn, provide a wealth of opportunities for video game developers to enhance their creative products. Partly due to the rapid progress of recent AI technologies, in particular deep learning, many of these opportunities are still underutilised and have yet to be explored.

This report aims to give a concise, preliminary overview of a selection of five potential research avenues for the application of state-of-the-art AI techniques to digital gaming. While our emphasis will mainly be on research directions where contemporary AI methods can enhance digital gaming, the reciprocal connection between AI and video games makes it conceivable that investigating these topics could also drive new insights and advancements in AI itself. The objective of this exploratory report is not to provide a comprehensive set of mature research proposals, or to present novel original research findings. Instead, the focus is on offering a speculative collection of high-level ideas that may serve to inspire more rigorous and focused research efforts in the future. The selected areas are not in any way exhaustive, but rather represent a curated and necessarily subjective collection of ideas deemed particularly intriguing during our examination of the current research landscape.

The foundational book from Yannakakis and Togelius [20], which served as one of the most valuable references for this work, outlines three core applications of AI to video gaming:

- **AI for game playing and agent modelling**, which includes simulating the role of a human player [22, 23], or controlling other game agents in the broadest sense, such as non-player characters (NPC) [28, 29], or hidden agents governing aspects of the game environment [30].
- **AI for procedural content generation** [31], which includes the algorithmic creation of game levels, music, textures, art, dialogues, items, characters, or any other digital content.
- **AI for player modelling** [32], which includes the modelling of human player characteristics, such as player type, predicted in-game behaviour, or emotional state, based on measured gameplay and player data.

All of these three areas are reflected in the research avenues discussed below, with a greater emphasis on the first two.

2 Large Language Models for Game Agent Modelling

Large language models (LLMs) such as OpenAI's GPT-4 [33], Google's Llama 3 [34], and Anthropic's Claude 3 [35] have recently risen to enormous prominence due to their advanced capabilities to maintain realistic conversational arcs and generate flexible solutions to a wide range of language-related tasks. Current state-of-the-art LLMs are based almost exclusively on variants of the transformer architecture, introduced in the seminal work of Vaswani et al. [7] in 2017. Transformer networks rely on the concept of *self-attention*, a deep learning mechanism designed to effectively capture long-range dependencies and contextual information in sequential data. The core training process for many LLMs is self-supervised and autoregressive, meaning that the LLM is trained to generate text by probabilistically predicting the next word (or subword token) in a text based on the preceding words. LLMs regularly contain billions of trainable parameters and are frequently trained on vast corpora of unlabelled textual data collected from publicly available sources such as books and websites [36].

At the moment, LLMs are attracting significant attention within the video game AI research community for their potential applicability to a diverse array of gaming-related tasks [37, 38]. For example, LLMs have recently been explored for the algorithmic creation of new video game levels in *Super Mario Bros* [39], the autonomous playing of *Minecraft* through the generation of code for a suitable game API [40], the systematic extraction of player sentiment from written game reviews [41], and the automatic generation of dynamic audio commentary for *League of Legends* gameplay [42]. Covering all promising use cases of LLMs in digital gaming would be beyond the scope of this exploratory report. However, we briefly highlight one possible research direction we consider to be particularly interesting, namely the use of LLMs for **game agent modelling**.

Game agent modelling includes the development and control of NPCs such as teammates, enemies, sidekick companions, merchants, bystanders, and other virtual characters in the broadest sense. Perhaps one of the most evident and fruitful applications of LLMs in this context would be to equip NPC agents with the ability to have natural and unscripted conversations with each other and with human players. First investigations in this area have already begun [43, 44, 45]; further advancements in integrating LLMs as NPC dialogue systems may be able to markedly enhance the realism of virtual characters, leading to substantially deeper and more immersive video game experiences.

However, the overall potential of LLMs for agent modelling may exceed the already appealing area of dynamic dialogue generation. In 2024, Hu et al. [46] gave a conceptual description of an entire cognitive architecture for general game agents that embeds an LLM as the core thinking component within a network of other submodules covering perception, memory, role-playing, action, and learning. Drawing closely from the work of Hu et al., one might envision an LLM-based cognitive architecture broadly working as follows: the **perception** module translates current game states into textual descriptions; the **thinking** module, powered by an LLM, receives outputs from the perception module and relevant text-based memories retrieved from the **memory** module to output textual action plans; these plans are translated by the **action** module into executable low-level in-game actions; the LLM-based thinking process is additionally biased with character information by the **role-playing** module; and continuously updated with techniques such as reinforcement learning or supervised finetuning by the **learning** module. One may also consider introducing a separate **goal** module that manages the objectives of the agent in a text-based manner and interacts with the other modules.

While each of the above modules could easily warrant its own extensive research programme, first successful attempts to design game agents via the integration of LLMs into broader cognitive architectures have already been made. Most notably, Park et al. [44] created an interactive artificial society consisting of a virtual 2D village with 25 distinct LLM-based game agents with different personalities and professions. Each agent maintains a text-based memory stream that contains a comprehensive list of the agent's perceptions, along with generated action plans and synthesised higher-order reflections. An LLM interacts with the agent's memory stream and current perceptions to generate new reflections and adapt action plans. This approach leads to an impressively complex and convincing set of self-organising emergent social behaviours: agents lead natural dialogues, coordinate actions, spread information, and dynamically update social relationship memories. A simple, schematic overview of an LLM-based cognitive architecture, heavily inspired by the works of Park et al. [44] and Hu et al. [46], is depicted in Figure 1.

Further efforts, such as those by Park et al., to integrate LLMs into a broader network of cognitive modules could not only contribute to the development of more immersive video game characters and

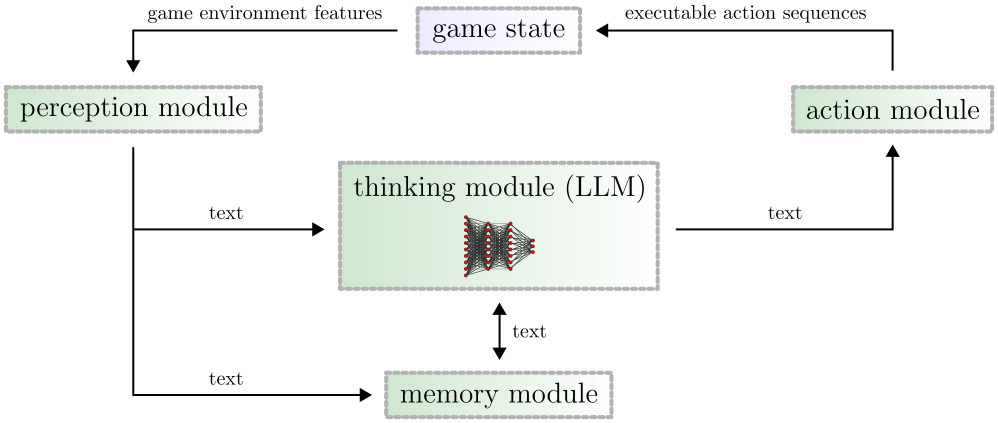

Figure 1: Simple, high-level overview of a conceivable LLM-based cognitive architecture for a video game agent, strongly influenced by the works of Park et al. [44] and Hu et al. [46]. The perception module translates game environment features (pixels, statistical features, vectorial embeddings, etc.) extracted from the game state into textual descriptions. The memory module stores past textual perceptions, as well as other memory items that are either predetermined (fixed character information, basic goals, etc.) or generated by the thinking module (novel knowledge, reflections, goals, procedural skills, etc.). The thinking module, based on a large language model (LLM), processes current textual perceptions and relevant textual memory items retrieved from the memory module, and outputs textual action plans and new memory items. The textual action plans are converted by the action module into low-level sequences of in-game behaviours that are executed to change the game state.

more human-like virtual agents for playtesting, but also advance research on the disputed question of how suitable LLMs truly are as core engines for artificial general intelligence [47, 48].

3 Neural Cellular Automata for Procedural Content Generation

Cellular automata (CA) [49, 50] are a family of extensively investigated and diverse mathematical models represented by grids of cells, whose states evolve in discrete time. At each time point t, each cell has a state represented by a number (or a vector of numbers), and its state at time $t + 1$ is determined by its own state and the states of its neighboring cells at time t, according to a local transition function that defines how the states evolve.

A simple and iconic example of CA that many readers may be familiar with is given by *Conway's Game of Life* [51], which takes place on an infinite 2D orthogonal grid of square cells, each of which can only be in one of two possible states, *dead* or *alive*. Given some initial configuration, cell states start to evolve based on a simple transition function that only takes into account how many dead or alive neighbours a cell has at a given time. In spite of its extreme simplicity, Conway's Game of Life exhibits an impressive set of complex self-organising behaviours.[1]

CA have already been used in video games with considerable success, for instance to grow infinite cave levels for the game *Cave Crawler* [52], automatically generate playable mazes for maze running games [53], model granular media like sand or soil [54], or simulate erosion in virtual environments [55]. CA are highly computationally efficient models that can be used to generate intricate virtual content. At the same time, CA are conceptually simple, intuitive to understand and easy to implement. However, the constructive, emergent nature of CA also makes them difficult to control [20]. In general, given a local transition function, it is very difficult to predict which pattern will arise over time from a specific initial grid state; even extremely similar initial states may quickly diverge in a chaotic manner, leading to entirely different outcomes [56]. Similarly, identifying a local transition function that over time maps a given initial state to a desired pattern is a nontrivial technical problem. These properties limit the utility of CA as procedural content generators for video games by making it challenging to impose essential constraints on generated content. Such constraints may

[1] Video illustration of Conway's Game of Life

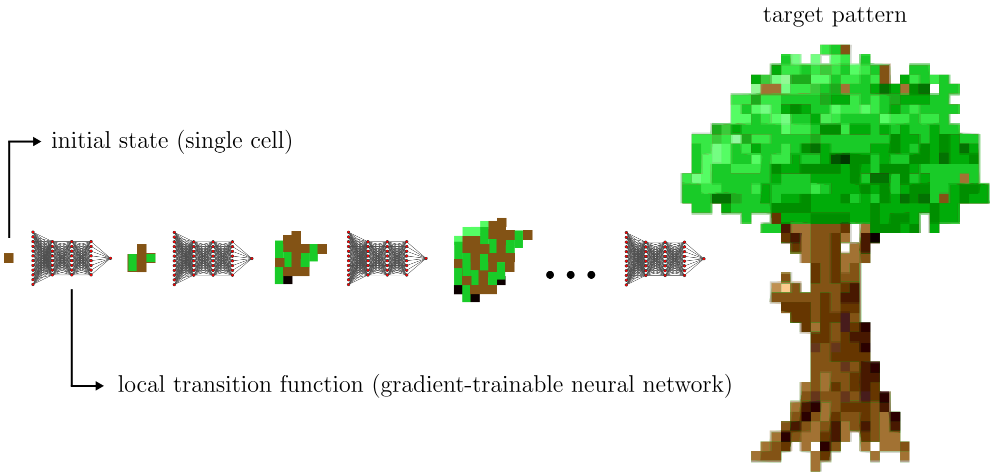
target pattern
initial state (single cell)
local transition function (gradient-trainable neural network)

Figure 2: Conceptual diagram of how a neural cellular automaton (NCA), once trained, could iteratively generate the target image of a tree (image not generated by actual NCA, used for illustrative purposes only). An NCA is a cellular automaton whose local transition function is parametrised by a neural network. Mordvintsev et al. [59] showed how an NCA can be trained with gradient-based methods to organically grow an arbitrary, predefined target pattern from a single initial cell. The NCA can also learn to automatically converge back to its intended target pattern when disturbed in a manner that resembles *self-regeneration*.

include guaranteed solvability for a game level, or a particular shape, connectivity and aesthetic style for a game object.

Recently, **neural cellular automata (NCA)** [57, 58] have been increasingly investigated as a significant way to address some of these shortcomings and allow for substantially greater control over the dynamical processes governing CA. An NCA is a CA whose local transition function is parametrised by a trainable neural network. One of the key contributions in the field of NCA was made by Mordvintsev et al. [59] in 2020, who demonstrated how an NCA parametrised by a convolutional neural network can be effectively trained in a differentiable end-to-end manner via gradient-based methods to iteratively generate any predefined target image from a single cell (see Figure 2 for an illustration of this idea). They moreover showed how NCA can be trained to exhibit *self-regeneration*, or, in the language of dynamical systems theory, how the target image can be turned into an attractor. A self-regenerating NCA automatically converges back to its intended target pattern when perturbed.[2]

The seminal work of Mordvintsev et al. [59] has implications stretching into diverse areas, including morphogenesis, embryonic development, regenerative medicine, self-organisation, and swarm robotics. In addition, first attempts have already been made to apply NCA in video game research [60, 61, 62, 63, 64, 65]: Earle et al. [60] successfully trained NCA to generate levels for 2D tile-based games while taking into account validity and diversity constraints; Sudhakaran et al. [63] used NCA in the virtual world of *Minecraft* for the targeted morphogenetic growth of complex 3D objects such as castles and trees; and Pajouheshgar et al. [64] employed NCA for the virtual synthesis of desired textures on 3D meshes.

These early studies highlight the potential of NCA as a novel deep-learning-based tool for procedural content generation in virtual environments [66]. However, many promising research directions remain to be investigated. Additional work could further explore the capabilities of NCA to be trained via custom loss functions designed to promote specific design constraints for video game content generation. Future studies could also more deeply investigate NCA for the creation of realistic textures for digital objects in a computationally efficient manner, or for accurately simulating organic and regenerative processes, such as the growth of natural ecosystems, aging characters, material degradation, or wound healing. Beyond content generation, it may also be interesting to investigate NCA as efficiently trainable swarm intelligence models to induce emergent behaviours in groups of locally connected NPCs.

[2] Interactive animations of self-regenerating NCA by Mordvintsev et al. [59]

4 Deep Surrogate Modelling to Accelerate Computationally Expensive In-Game Simulations

In their seminal study, Gilmer et al. [14] not only introduced *message-passing* as a unifying framework for graph neural network architectures; in addition to this salient contribution, they also showed that a graph neural network can learn to efficiently predict quantum-chemical properties of small molecules using the QM9 data set [67] for training. The QM9 data set consists of around 134k chemical compounds; each compound comes with a set of numerical labels that represent approximations of relevant quantum-chemical properties. For each compound, each numerical label is the result of a quantum-mechanical simulation based on what is known as *density functional theory* [68]. Density functional theory simulations, while highly useful for elucidating the electronic structure of a molecule, are associated with prohibitive computational costs; for example, generating all the labels in the QM9 data set for a single molecule with nine heavy atoms on a single core of a Xeon E5-2660 processor with 2.2 GHz and commonly used software may take around one hour [14]. In contrast, the graph neural network from Gilmer et al., which was trained on the QM9 data set in a supervised manner, can estimate the outcome of density functional theory simulations for a novel molecule in a fraction of a second. This corresponds to a speed-up of five orders of magnitude, making it computationally feasible to rapidly predict quantum-chemical properties for large molecular libraries.

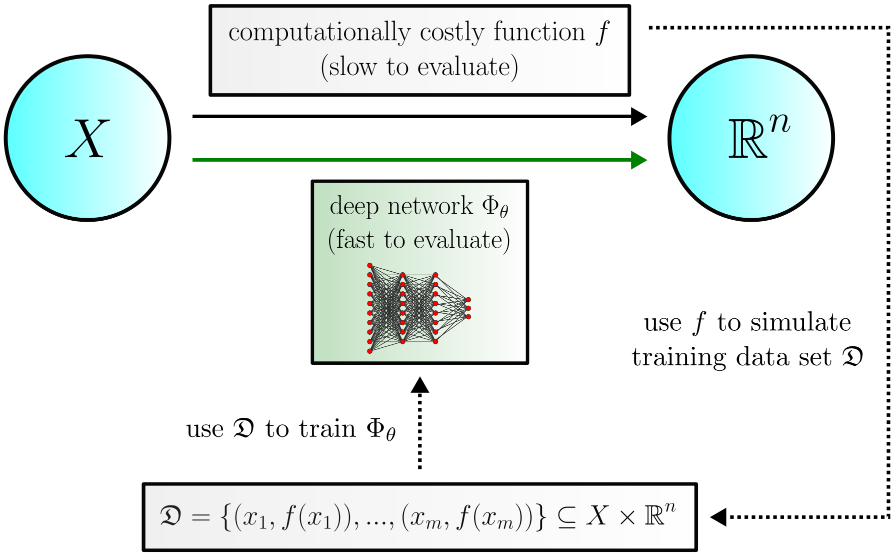
computationally costly function f
(slow to evaluate)
X
$\mathbb{R}^n$
deep network Φ_θ
(fast to evaluate)
use f to simulate training data set $\mathfrak{D}$
use $\mathfrak{D}$ to train Φ_θ
$\mathfrak{D} = \{(x_1, f(x_1)), ..., (x_m, f(x_m))\} \subseteq X \times \mathbb{R}^n$

Figure 3: Illustration of the elementary idea behind deep surrogate modelling: A computationally expensive function f is repeatedly evaluated to generate a training data set $\mathfrak{D}$, which is then used to train a deep network Φ_θ. After training, Φ_θ acts as a computationally fast approximation of f.

The work of Gilmer et al. represents a prime example of what can be referred to as **deep surrogate modelling** [69, 70, 71, 72]. A high-level illustration of the key idea behind deep surrogate modelling is given in Figure 3. In one of its most elementary forms, deep surrogate modelling is a technique used to speed up the evaluation of a computationally expensive function

$$f : X \to \mathbb{R}^n$$

that is of interest for a practical application. For instance, f could represent an expensive numerical computer simulation. In our earlier example from Gilmer et al., the domain X would be a set of molecular graphs, and f would represent a density functional theory simulation that maps molecular graphs to numerical quantum-chemical properties expressed as vectors in $\mathbb{R}^n$. Initially, a (sometimes considerable) computational effort is made to create a data set

$$\mathfrak{D} = \{(x_1, f(x_1)), ..., (x_m, f(x_m))\} \subseteq X \times \mathbb{R}^n,$$

which is then used to train the parameters θ of a suitable deep learning architecture

$$\Phi_\theta : X \to \mathbb{R}^n$$

in a supervised manner. After training, the deep network Φ_θ can be used as a *surrogate* for f, approximating the value of $f(x)$ with $\Phi_\theta(x)$ for novel x outside the training set $\mathfrak{D}$. Furthermore, Φ_θ can also be optimised instead of f when looking for maximisers or minimisers of f. While Φ_θ may be less accurate than the original simulation function f, it can be orders of magnitude faster to evaluate.

Deep surrogate modelling is particularly useful in situations requiring computationally expensive and repetitive simulations [70]. Video games regularly involve a plethora of such simulations, spanning diverse areas like gameplay balancing, difficulty tuning, fluid and particle dynamics, Newtonian mechanics, pathfinding, environmental systems, realistic lighting, sound propagation, game state prediction, and procedural content generation. As such, digital gaming may be well-suited for the application of deep surrogate models to accelerate gameplay, reduce loading times and optimise the game development process. Despite these encouraging possibilities, the number of studies exploring deep surrogate models for video games appears to be relatively limited [73, 74, 75, 76, 77, 78, 79]. However, early work in this field has already shown some success. For example, Bhatt et al. [74] trained a deep surrogate model on simulated data to predict the behaviour of a game agent in novel environments, applying the model to accelerate the algorithmic generation of new environments that lead to diverse agent behaviours. Overall, the utility of deep surrogate modelling for digital gaming may still be underexplored, offering notable opportunities for future research.

5 Self-Supervised Video Game State Representation Learning

Being able to represent the abstract **state** of a video game in terms of a meaningful numerical vector is a key element in a large variety of modern AI applications for digital gaming [20]. In this context, a high-quality vectorial representation technique should be able to condense the essential features of a video game state into an informative embedding that can be effectively used for downstream AI tasks. Such tasks may, for example, include using a game state embedding as a model of perception for an autonomous game agent [26, 80], predicting the emotional state of a player from gameplay video streams [81], predicting future game states from current ones [82], dynamically adapting game music depending on the state of a game [83], or algorithmically translating game states into natural language descriptions [44].

A powerful deep learning paradigm for vectorial data representation that has emerged in recent years is **self-supervised learning** [84], which offers a collection of strategies to learn rich, general-purpose embeddings solely from the internal structure of unlabelled input data. While supervised deep learning is based on the extraction of task-specific features from labelled data sets, self-supervised learning does not rely on data annotation by human subjects, and instead allows one to find flexible feature representations in a task-agnostic manner. Labelled data is frequently scarce and hard to obtain; self-supervised learning methods do not suffer from comparable limits, as they can take advantage of vast corpora of unlabelled data sets, such as curated libraries of images and text extracted from the internet. Representations learned via self-supervised training can be employed in a variety of ways, including clustering [85] and anomaly detection [86]. Importantly, they can also be fine-tuned on downstream supervised tasks [87], an approach that regularly leads to substantial boosts in performance compared to purely supervised techniques.

While self-supervised learning has become a significant area of research in domains like natural language processing and computer vision [88], comparable work in digital gaming is still relatively sparse. A literature search for studies that use concepts from self-supervised learning in digital gaming revealed only 12 instances [80, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]. In a notable article, Anand et al. [80] introduced a systematic benchmark to evaluate self-supervised learning methods for video game states via the prediction of essential internal game variables in Atari 2600 games from learned representations. They employed this benchmark to demonstrate the effectiveness of a mutual-information-based representation learning strategy. In a related study, Trivedi et al. [89] showed that three popular self-supervised learning strategies applied to video game pixels alone can be used to derive game state embeddings that are predictive of key internal game variables, such as enemy positions on the screen in a first-person shooter, or game world coordinates of football players and the ball in a football simulator.

Representing video game states via state-of-the-art self-supervised learning may be an impactful area for future research. One particularly interesting approach could be to further investigate *joint-*

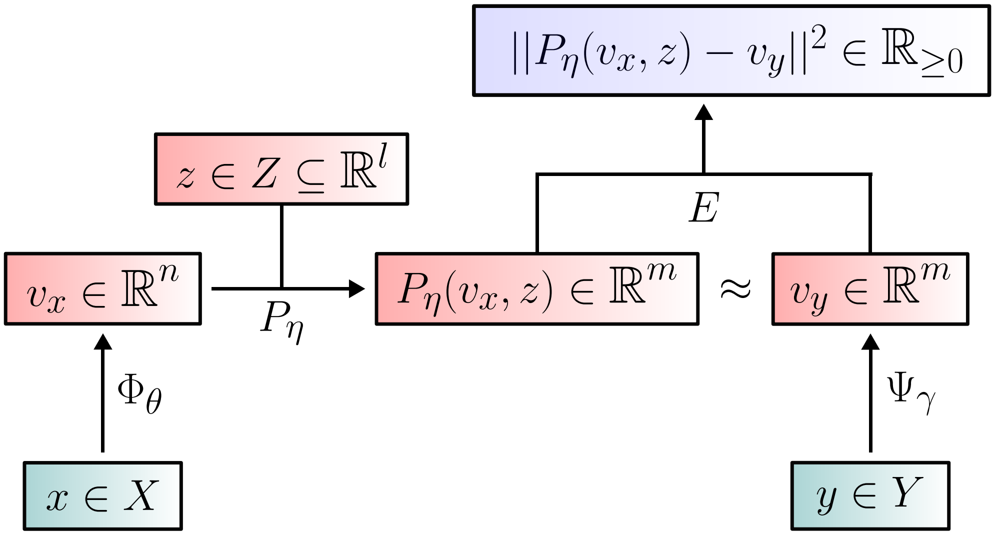
$||P_\eta(v_x, z) - v_y||^2 \in \mathbb{R}_{\geq 0}$
$z \in Z \subseteq \mathbb{R}^l$
E
$v_x \in \mathbb{R}^n$
P_η
$P_\eta(v_x, z) \in \mathbb{R}^m$
$\approx$
$v_y \in \mathbb{R}^m$
Φ_θ
Ψ_γ
$x \in X$
$y \in Y$

Figure 4: Schematic overview of a prototypical joint-embedding predictive architecture (JEPA) [100] for self-supervised learning. The variables x and y could, for instance, represent images of game pixels at times t and $t + \delta$, the encoders Φ_θ and Ψ_γ could be convolutional neural networks that map the images to embeddings v_x, v_y, the latent variable z could symbolise the action taken by the player at time t, and P_η could be a multilayer perceptron whose output $P_\eta(v_x, z)$ aims to approximate v_y.

embedding predictive architectures (JEPAs) [100, 101] for this purpose. JEPAs constitute a novel self-supervised learning framework with attractive technical properties that has recently demonstrated encouraging results in the image domain [102]. Let

$$\Psi_\gamma : Y \to \mathbb{R}^m, \quad \Phi_\theta : X \to \mathbb{R}^n,$$

be two trainable deep learning encoders that map given data entities in the sets X and Y (which could, for example, be collections of images or graphs) to vectors. In essence, a JEPA aims to learn useful representations by training to predict the embedding

$$v_y := \Psi_\gamma(y) \in \mathbb{R}^m$$

of an input entity $y \in Y$ from the embedding

$$v_x := \Phi_\theta(x) \in \mathbb{R}^n$$

of a somehow related input entity $x \in X$, with the help of a latent variable

$$z \in Z \subseteq \mathbb{R}^l$$

that can be used to add additional information about y not contained in x. The prediction of v_y from v_x and z is done via a trainable predictor

$$P_\eta : \mathbb{R}^n \times Z \to \mathbb{R}^m$$

whose aim is to minimise a scalar error function such as

$$E(P_\eta(v_x, z), v_y) := ||P_\eta(v_x, z) - v_y||^2 \in \mathbb{R}_{\geq 0}.$$

A schematic visualisation of this architecture is depicted in Figure 4.

Since the prediction of y from x and z occurs implicitly in abstract representation space, unimportant details of x and y can be eliminated by the encoders Ψ_γ and Φ_θ prior to prediction. This architecture also allows for multiple values of y to be compatible with a single value of x, due to potential invariance properties of the encoder Ψ_γ and the ability to change the output of P_η by varying the latent variable z.

In the context of video games, x and y could, for example, represent game pixels on the player screen at times t and $t + \delta$, the encoders Φ_θ and Ψ_γ could be convolutional neural networks that map the pixels to vectorial embeddings v_x, v_y, the latent variable z could symbolise the action taken by the player at time t, and P_η could be a multilayer perceptron whose output $P_\eta(v_x, z)$ aims to approximate v_y. In other words, P_η could be trained to predict the abstract future game state from the abstract current game state given the player action, a task that may encourage Φ_θ and Ψ_γ to simultaneously learn meaningful abstract game state embeddings. The BYOL method [103], which shares some similarities with the JEPA approach, has already been successfully tested for pixel-based game state representation learning in the previously mentioned study by Trivedi et al. [89]. For a more detailed technical description of JEPAs, including important training considerations to prevent such architectures from collapsing into producing only constant representations, we refer the reader to the original article by LeCun [100].

6 Learning Generative Models of Interactive Worlds from Unlabelled Videos

In early 2024, Google DeepMind introduced **Genie** [104], an 11-billion parameter generative world model trained in a self-supervised manner on a large-scale library of publicly available gameplay videos of 2D platformer games. Genie can automatically generate an infinite variety of novel and action-controllable 2D platformer gaming worlds.[3] Each unique world is created using only a single image as an initial seed.

The neural architecture of Genie consists of three major components, all of which rely on the use of computationally efficient spatiotemporal transformer models [7, 105]: a video tokeniser, a latent action model, and a dynamics model. The video tokeniser is implemented via a VQ-VAE [106] that is trained to translate video game frames into discrete vectorial tokens, and vice versa. The latent action model is trained to infer plausible actions between consecutive pairs of frames. It too is based on a VQ-VAE architecture that naturally allows for limiting the number of possible actions to a small, fixed-size set of discrete vectorial action embeddings. Most notably, the latent action model is trained in an entirely self-supervised way, without the need for human-annotated action labels. The dynamics model [107] is trained autoregressively to predict future tokenised frames from past and current tokenised frames and inferred action embeddings. As a result, the dynamics model is encouraged to learn the consequences of actions on the temporal evolution of the video game world.

Once trained, the Genie system can be operated in the following way:

1. A user first prompts Genie with an image x_1 vaguely resembling a scene from a platformer game. This image serves as the initial game frame. The starting image x_1 could for instance be a screenshot from an actual game, an imagined sketch drawn by a human, or an artificial image created via a text-to-image generator [108] from a natural language description.
2. The image x_1 is compressed into a discrete vectorial token z_1 using the video tokeniser.
3. The player can then input an initial action which is translated into a discrete vectorial action embedding a_1 by a component of the latent action model.
4. The dynamics model uses its acquired world knowledge [109] to predict the next tokenised frame z_2 based on action a_1 and state z_1.
5. The compressed vectorial token z_2 is decoded into the next video game frame x_2 by the video tokeniser. The image frame x_2 is displayed to the user.
6. The last three steps are iteratively repeated to give rise to an interactive sequence of image frames
$$(x_1, x_2, x_3, ...)$$
that constitute a playable platformer game. For example, after the initial iteration, the user specifies another action a_2, the dynamics model uses (z_1, z_2) and (a_1, a_2) to predict z_3, and z_3 is subsequently decoded into the next visible frame x_3.

This process is visualised in Figure 5.

One of the limitations of Genie outlined in the original article [104] is its low frame rate, reported to be around one frame per second. Genie may also sometimes hallucinate unrealistic future scenarios,

[3] Video illustrations of Genie's [104] capabilities

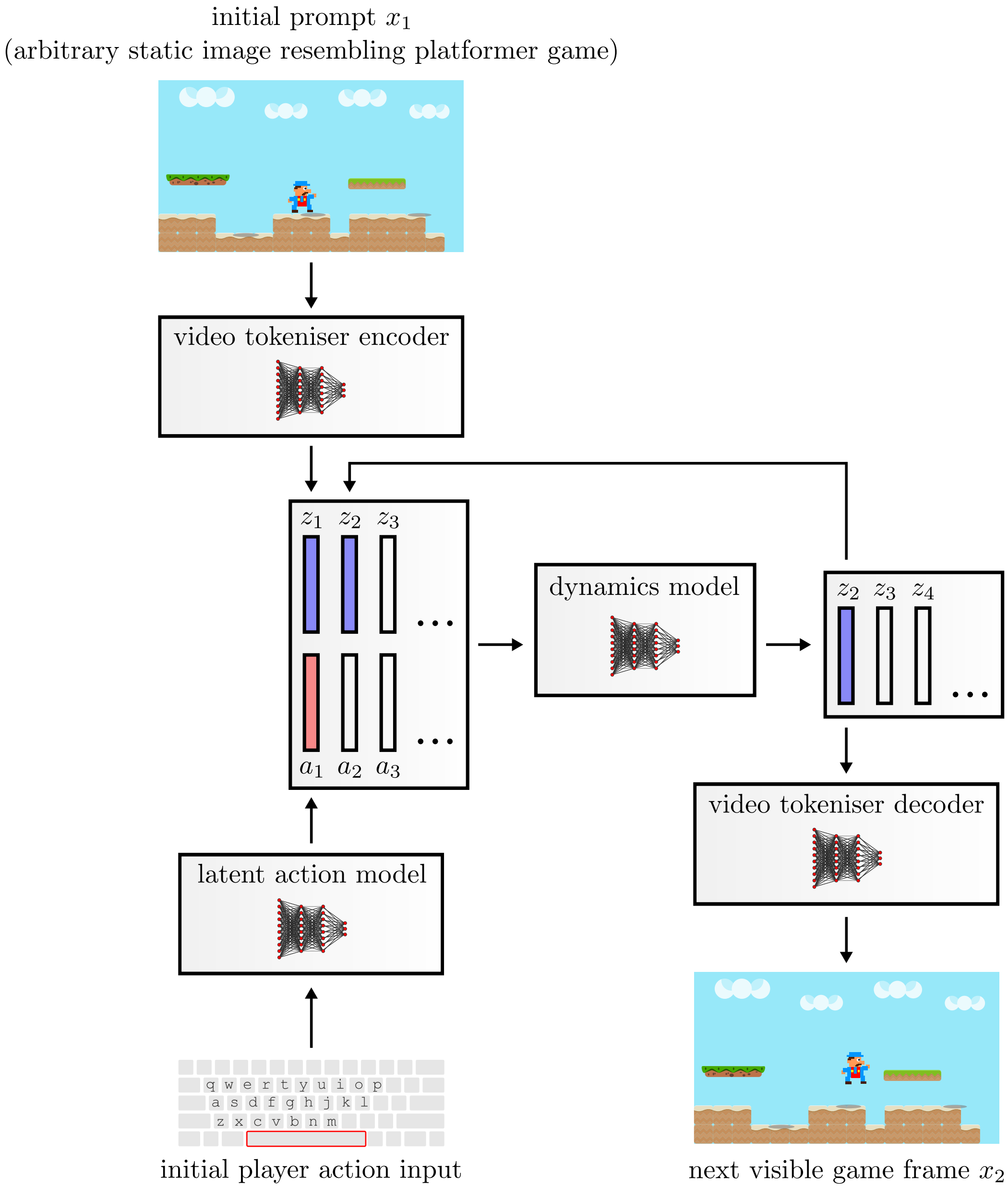

Figure 5: Schematic diagram illustrating the inference process of the trained Genie model [104] to generate a playable platformer game from a given image prompt x_1 (images not generated by actual Genie system, used for illustrative purposes only). The video tokeniser and the latent action model respectively translate the prompt image x_1 and the initial player action input into embeddings z_1 and a_1, which are subsequently used by the dynamics model to predict the next tokenised frame z_2. The compressed representation z_2 is then converted by the video tokeniser into a visible game frame x_2. This process is iteratively repeated using previously generated image tokens and recorded input actions to give rise to a sequence of interactive game frames $(x_1, x_2, x_3, ...)$.

or fail to maintain the stability and consistency of a generated world over time. Despite these shortcomings, Genie's architecture represents a strong proof of concept for the possibility of learning **generative models of interactive worlds** from gameplay video data alone. The fact that Genie is able to effectively infer an operable latent action space entirely without human-annotated action labels is particularly noteworthy.

Note that this article is written only a few days after Google DeepMind's public announcement of Genie 2 [110]. Due to the early stage of this research, a comprehensive technical description of this novel model via an associated research paper is currently lacking. It has been stated, however, that Genie 2 is an autoregressive latent diffusion model [12]. It appears that Genie 2 extends the domain of Genie to the substantially more challenging task of generating complex, interactive 3D game worlds[4] instead of simple 2D platformer games. Genie 2 is reported to exhibit a set of emergent capabilities related to physics, object interactions, character animation and game agents, and to be able to maintain a consistent 3D world for up to a minute.

The potential implications of interactive-world generators like Genie and Genie 2 for general AI research as well as digital gaming are noteworthy. In the future, considerably more mature versions of such systems could conceivably become useful for a wide array of tasks, including training and testing general adaptive agents in maximally diverse virtual environments [111], simulating an infinite stream of realistic training scenarios for robotic systems like autonomous vehicles [112], accelerating video game development via rapid prototyping, allowing non-experts to easily create their own action-controllable video game snippets, procedurally extending existing video games and virtual worlds [66, 113], automatically personalising video game content based on a model of player behaviour or characteristics [32], and simulating virtual environments for human training purposes in fields like medicine [114] and aviation [115].

Future impactful work in this area could for example focus on maintaining the consistency and stability of generated environments over extended periods of time, the prevention of hallucinations [116, 117], the inclusion of audio signals, the generation of game mechanics for less explored genres like bird's-eye strategy games, the identification and mitigation of computational bottlenecks [118] to accelerate low frame rates, and the development of even more capable dynamics and latent action models [119]. In addition, it may be interesting to investigate to what extent combinations of current LLMs with text-to-image models [120, 121, 122] could enable the generation of interactive worlds.

7 Current Technical Challenges for Deep Learning in Digital Gaming: A Critical View

While modern AI techniques, in particular deep learning, hold significant promise to enhance the future of video gaming experiences, serious technical challenges remain. As in other application areas of neural networks, these challenges frequently revolve around computational efficiency and speed, interpretability and predictability, the setting of model constraints, data requirements, model generalisation abilities, privacy considerations, and financial costs. Other, more game-specific issues centre around the complexities of integrating deep learning systems into traditional game development workflows, development time, managing player expectations with regards to AI, maintaining narrative control of video game stories, ensuring model consistency, and preserving debugging options.

One of the main concerns of game developers is the feasibility of training, running and gathering data for advanced deep learning architectures [123, 124]. The large amount of computation time and expensive hardware required to train state-of-the-art models such as LLMs or interactive-world generators represent an important bottleneck, especially for small and moderately-sized studios. Furthermore, if a model does not generalise effectively to new scenarios, it may have to be discarded or retrained. In supervised settings, it may be intractable to obtain sufficient amounts of human-annotated training data. If supervised data relates to in-game behaviour or player analytics, this may also potentially raise concerns regarding privacy ethics. In addition, running large trained models in real-time during gameplay could decelerate frame rate and responsiveness to unacceptably low levels.

Further hurdles for game developers arise in connection to the *black-box* nature of deep learning systems [125, 126], which refers to the difficulty in understanding how such models make predictions

[4] Video illustrations of Genie 2's capabilities

and arrive at decisions. Neural networks consist of inscrutable compositions of large matrices and nonlinear functions, which makes their outputs and working mechanisms notoriously hard to interpret from a human perspective. In particular, this makes it challenging to definitively predict the behaviour of deep networks in novel edge cases [127], to guarantee their consistency, and to debug them in case they produce undesired outcomes. This opacity creates a variety of obstacles for applications in digital gaming: for instance, NPCs controlled by neural networks may exhibit unexpected and inexplicable behaviours that contradict the intended narrative or essential game mechanics; complex debugging procedures may significantly extend game development time; and it may be unclear how to hard-code necessary playability constraints into deep-learning-based procedural game level generators.

In order to reduce justified hesitancy amongst game developers to integrate novel deep learning technologies into their products, concerns like the ones outlined in this section must be addressed. Notably, a variety of currently active research areas could lead to the mitigation of some of the mentioned problems: Advances in self-supervised learning [84, 87] and simulation of synthetic training data [128] could improve generalisation and reduce the need for labelled data sets in video game applications. More powerful model distillation [129] or network pruning techniques [130], possibly based on the exploitation of so-called *super weights* recently discovered in LLMs [131], may reduce the computational costs associated with deep networks. Additionally, large deep learning architectures could potentially be run on distributed cloud computing systems during gameplay to offload the computational burden from local machines. Research on AI explainability [132] and the adversarial robustness of neural networks [133] could make deep learning models more predictable, consistent, and debuggable. And quality-diversity optimisation methods [134, 135] represent a growing family of algorithms that can be combined with deep learning to generate content that respects certain constraints while being diverse in nature.

8 Conclusions

In this work, we illuminated five promising research pathways for the application of state-of-the-art AI techniques to digital gaming: LLMs for game agent modelling, neural cellular automata for procedural content generation, deep surrogate modelling to accelerate expensive in-game simulations, self-supervised game state representation learning, and the use of unlabelled video data to train generative models of interactive worlds. The primary objective of this report is to provide a high-level overview of these areas within the current research landscape, with the aim of sparking intellectual curiosity for more targeted and in-depth research efforts in these or related fields in the future.

Video games are one of the most natural and important research frontiers in the search for general artificial intelligence systems, as they offer an almost limitless abundance of distinct cognitive tasks and simulated environments to challenge virtual agents. Further work in the areas discussed in this report could not only lead to novel technologies that enhance the quality and immersiveness of digital gaming experiences; it could also drive scientific developments in the search for more useful and capable AI models overall. For instance, research on LLM-based video game agents could lead to progress on the debated question to what extent purely language-based systems are in fact suitable as models for general intelligence [47, 48]; and advanced generative models of interactive worlds could supply AI agents with a potentially infinite stream of complex virtual training and testing environments [104, 111].

While we regard the research directions described in this work to hold considerable potential, it is important to note current challenges that may still limit the practical utility of deep learning in digital gaming. Issues remain around topics such as interpretability, predictability, consistency, debuggability, data requirements, generalisation, reusability, the setting of model constraints, as well as financial and computational efficiency. Systematic efforts to address these technical obstacles, alongside experimentation with novel research ideas like those outlined in this report, should be a high priority to accelerate future progress in both AI and video game research.

Acknowledgments and Disclosure of Funding

This exploratory report was commissioned and funded by Beam Foundation[5] as part of their emerging initiative to investigate the future of artificial intelligence in digital gaming.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[3] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *Proceedings of the European Conference on Computer Vision*, pages 818–833, 2014.

[4] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[6] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[9] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[10] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[11] 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.

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33, 2020.

[14] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *International Conference on Machine Learning*, 70, 2017.

[5] 🏠 Homepage | 𝕏

[15] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

[16] David K. Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in Neural Information Processing Systems*, 28, 2015.

[17] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[18] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[19] Chengpeng Hu, Yunlong Zhao, Ziqi Wang, Haocheng Du, and Jialin Liu. Games for artificial intelligence research: A review and perspectives. *IEEE Transactions on Artificial Intelligence*, 2024.

[20] Georgios N. Yannakakis and Julian Togelius. *Artificial intelligence and games*. Springer, 2018. https://gameaibook.org.

[21] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587): 484–489, 2016.

[22] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354, 2019.

[23] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[24] Kuang-Huei Lee, Ofir Nachum, Mengjiao Sherry Yang, Lisa Lee, Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, et al. Multi-game decision transformers. *Advances in Neural Information Processing Systems*, 35, 2022.

[25] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[27] Risto Miikkulainen, Bobby D. Bryant, Ryan Cornelius, Igor V. Karpov, Kenneth O. Stanley, and Chern Han Yong. Computational intelligence in games. *Computational Intelligence: Principles and Practice*, pages 155–191, 2006.

[28] Jeff Orkin. Three states and a plan: The AI of FEAR. *Game Developers Conference*, 2006:4, 2006.

[29] Davide Aversa and Stavros Vassos. Action-based character AI in video-games with CogBots architecture: A preliminary report. *arXiv preprint arXiv:1307.3195*, 2013.

[30] Michael Booth. The AI systems of Left 4 Dead. *Artificial Intelligence and Interactive Digital Entertainment Conference at Stanford*, 2009.

[31] Noor Shaker, Julian Togelius, and Mark J. Nelson. *Procedural content generation in games*. Springer, 2016.

[32] Georgios N. Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André. *Player modeling*. Dagstuhl Publishing, 2013.

[33] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[34] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[35] Maxim Enis and Mark Hopkins. From LLM to NMT: Advancing low-resource machine translation with Claude. *arXiv preprint arXiv:2404.13813*, 2024.

[36] Fei Du, Xin-Jian Ma, Jing-Ru Yang, Yi Liu, Chao-Ran Luo, Xue-Bin Wang, Hai-Ou Jiang, and Xiang Jing. A survey of LLM datasets: From autoregressive model to AI chatbot. *Journal of Computer Science and Technology*, 39(3):542–566, 2024.

[37] Roberto Gallotta, Graham Todd, Marvin Zammit, Sam Earle, Antonios Liapis, Julian Togelius, and Georgios N. Yannakakis. Large language models and games: A survey and roadmap. *arXiv preprint arXiv:2402.18659*, 2024.

[38] Penny Sweetser. Large language models and video games: A preliminary scoping review. *Proceedings of the 6th ACM Conference on Conversational User Interfaces*, pages 1–8, 2024.

[39] Shyam Sudhakaran, Miguel González-Duque, Matthias Freiberger, Claire Glanois, Elias Najarro, and Sebastian Risi. MarioGPT: Open-ended text2level generation through large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[40] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[41] Markos Viggiato and Cor-Paul Bezemer. Leveraging the OPT large language model for sentiment analysis of game reviews. *IEEE Transactions on Games*, 2023.

[42] Noah Ranella and Markus Eger. Towards automated video game commentary using generative AI. *EXAG@ AIIDE*, 2023.

[43] Matthias Müller-Brockhausen, Giulio Barbero, and Mike Preuss. Chatter generation through language models. *IEEE Conference on Games*, pages 1–6, 2023.

[44] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023.

[45] Navapat Nananukul and Wichayaporn Wongkamjan. What if Red can talk? Dynamic dialogue generation using large language models. *arXiv preprint arXiv:2407.20382*, 2024.

[46] Sihao Hu, Tiansheng Huang, Fatih Ilhan, Selim Tekin, Gaowen Liu, Ramana Kompella, and Ling Liu. A survey on large language model-based game agents. *arXiv preprint arXiv:2404.02039*, 2024.

[47] Ben Goertzel. Generative AI vs. AGI: The cognitive strengths and weaknesses of modern LLMs. *arXiv preprint arXiv:2309.10371*, 2023.

[48] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023.

[49] John Von Neumann and Arthur Walter Burks. *Theory of self-reproducing automata*. University of Illinois Press Urbana, 1966.

[50] Palash Sarkar. A brief history of cellular automata. *ACM Computing Surveys*, 32(1):80–107, 2000.

[51] Martin Gardner. Mathematical games. *Scientific American*, 222(6):132–140, 1970.

[52] Lawrence Johnson, Georgios N. Yannakakis, and Julian Togelius. Cellular automata for real-time generation of infinite cave levels. *Proceedings of the Workshop on Procedural Content Generation in Games*, pages 1–4, 2010.

[53] Chad Adams and Sushil Louis. Procedural maze level generation with evolutionary cellular automata. *IEEE Symposium Series on Computational Intelligence*, pages 1–8, 2017.

[54] Jonathan Devlin and Micah D. Schuster. Probabilistic cellular automata for granular media in video games. *The Computer Games Journal*, 10(1):111–120, 2021.

[55] Jacob Olsen. Realtime procedural terrain generation. 2004.

[56] Jesús Urıas, Raúl Rechtman, and Agustın Enciso. Sensitive dependence on initial conditions for cellular automata. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 7(4):688–693, 1997.

[57] N. Wulff and J. A. Hertz. Learning cellular automaton dynamics with neural networks. *Advances in Neural Information Processing Systems*, 5, 1992.

[58] Stefano Nichele, Mathias Berild Ose, Sebastian Risi, and Gunnar Tufte. Ca-NEAT: Evolved compositional pattern producing networks for cellular automata morphogenesis and replication. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):687–700, 2017.

[59] Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. Growing neural cellular automata. *Distill*, 2020. doi: 10.23915/distill.00023. https://distill.pub/2020/growing-ca.

[60] Sam Earle, Justin Snider, Matthew C. Fontaine, Stefanos Nikolaidis, and Julian Togelius. Illuminating diverse neural cellular automata for level generation. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 68–76, 2022.

[61] Alexander Mordvintsev and Eyvind Niklasson. μNCA: Texture generation with ultra-compact neural cellular automata. *arXiv preprint arXiv:2111.13545*, 2021.

[62] Ehsan Pajouheshgar, Yitao Xu, Tong Zhang, and Sabine Süsstrunk. DyNCA: Real-time dynamic texture synthesis using neural cellular automata. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20742–20751, 2023.

[63] Shyam Sudhakaran, Djordje Grbic, Siyan Li, Adam Katona, Elias Najarro, Claire Glanois, and Sebastian Risi. Growing 3D artefacts and functional machines with neural cellular automata. *Artificial Life Conference Proceedings 33*, 2021(1):108, 2021.

[64] Ehsan Pajouheshgar, Yitao Xu, Alexander Mordvintsev, Eyvind Niklasson, Tong Zhang, and Sabine Süsstrunk. Mesh neural cellular automata. *ACM Transactions on Graphics*, 43(4): 1–16, 2024.

[65] Hiroki Sato, Tanner Lund, Takahide Yoshida, and Atsushi Masumori. Automata quest: NCAs as a video game life mechanic. *arXiv preprint arXiv:2309.14364*, 2023.

[66] Jialin Liu, Sam Snodgrass, Ahmed Khalifa, Sebastian Risi, Georgios N. Yannakakis, and Julian Togelius. Deep learning for procedural content generation. *Neural Computing and Applications*, 33(1):19–37, 2021.

[67] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):1–7, 2014.

[68] Axel D. Becke. Density-functional thermochemistry. I. The effect of the exchange-only gradient correction. *The Journal of Chemical Physics*, 96(3):2155–2160, 1992.

[69] Bruno Sudret, Stefano Marelli, and Joe Wiart. Surrogate models for uncertainty quantification: An overview. *11th European Conference on Antennas and Propagation*, pages 793–797, 2017.

[70] Yinhao Zhu, Nicholas Zabaras, Phaedon-Stelios Koutsourelakis, and Paris Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394:56–81, 2019.

[71] Meng Tang, Yimin Liu, and Louis J. Durlofsky. A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. *Journal of Computational Physics*, 413:109456, 2020.

[72] Majdi I. Radaideh and Tomasz Kozlowski. Surrogate modeling of advanced computer simulations using deep Gaussian processes. *Reliability Engineering & System Safety*, 195:106731, 2020.

[73] Daniel Karavolos, Antonios Liapis, and Georgios N. Yannakakis. A multifaceted surrogate model for search-based procedural content generation. *IEEE Transactions on Games*, 13(1): 11–22, 2019.

[74] Varun Bhatt, Bryon Tjanaka, Matthew Fontaine, and Stefanos Nikolaidis. Deep surrogate assisted generation of environments. *Advances in Neural Information Processing Systems*, 35: 37762–37777, 2022.

[75] Daniel Karavolos, Antonios Liapis, and Georgios N. Yannakakis. Pairing character classes in a deathmatch shooter game via a deep-learning surrogate model. *Proceedings of the 13th International Conference on the Foundations of Digital Games*, pages 1–10, 2018.

[76] Daniel Karavolos, Antonios Liapis, and Georgios N. Yannakakis. Using a surrogate model of gameplay for automated level design. *IEEE Conference on Computational Intelligence and Games*, pages 1–8, 2018.

[77] Yulun Zhang, Matthew C. Fontaine, Amy K. Hoover, and Stefanos Nikolaidis. Deep surrogate assisted map-elites for automated hearthstone deckbuilding. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 158–167, 2022.

[78] Panagiotis Migkotzidis and Antonios Liapis. SuSketch: Surrogate models of gameplay as a design assistant. *IEEE Transactions on Games*, 14(2):273–283, 2021.

[79] Daniel Karavolos, Antonios Liapis, and Georgios N. Yannakakis. Learning the patterns of balance in a multi-player shooter game. *Proceedings of the 12th International Conference on the Foundations of Digital Games*, pages 1–10, 2017.

[80] Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R. Devon Hjelm. Unsupervised state representation learning in Atari. *Advances in Neural Information Processing Systems*, 32, 2019.

[81] Konstantinos Makantasis, Antonios Liapis, and Georgios N. Yannakakis. From pixels to affect: A study on games and player experience. *8th International Conference on Affective Computing and Intelligent Interaction*, pages 1–7, 2019.

[82] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L. Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. *Advances in Neural Information Processing Systems*, 28, 2015.

[83] Cale Plut and Philippe Pasquier. Generative music in video games: State of the art, challenges, and prospects. *Entertainment Computing*, 33:100337, 2020.

[84] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2021.

[85] Vivek Sharma, Makarand Tapaswi, M. Saquib Sarfraz, and Rainer Stiefelhagen. Self-supervised learning of face representations for video face clustering. *14th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–8, 2019.

[86] Seungdong Yoa, Seungjun Lee, Chiyoon Kim, and Hyunwoo J. Kim. Self-supervised learning for anomaly detection with dynamic local augmentation. *IEEE Access*, 9:147201–147211, 2021.

[87] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255, 2020.

[88] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[89] Chintan Trivedi, Konstantinos Makantasis, Antonios Liapis, and Georgios N. Yannakakis. Learning task-independent game state representations from unlabeled images. *IEEE Conference on Games*, pages 88–95, 2022.

[90] Chintan Trivedi, Konstantinos Makantasis, Antonios Liapis, and Georgios N. Yannakakis. Towards general game representations: Decomposing games pixels into content and style. *arXiv preprint arXiv:2307.11141*, 2023.

[91] Chintan Trivedi, Antonios Liapis, and Georgios N. Yannakakis. Contrastive learning of generalized game representations. *IEEE Conference on Games*, pages 1–8, 2021.

[92] Young Jae Lee, Insung Baek, Uk Jo, Jaehoon Kim, Jinsoo Bae, Keewon Jeong, and Seoung Bum Kim. Self-supervised contrastive learning for predicting game strategies. *Proceedings of SAI Intelligent Systems Conference*, pages 136–147, 2022.

[93] Hacene Terbouche, Liam Schoneveld, Oisin Benson, and Alice Othmani. Comparing learning methodologies for self-supervised audio-visual representation learning. *IEEE Access*, 10: 41622–41638, 2022.

[94] Nemanja Rašajski, Chintan Trivedi, Konstantinos Makantasis, Antonios Liapis, and Georgios N. Yannakakis. BehAVE: Behaviour alignment of video game encodings. *arXiv preprint arXiv:2402.01335*, 2024.

[95] Chintan Trivedi, Konstantinos Makantasis, Antonios Liapis, and Georgios N. Yannakakis. Game state learning via game scene augmentation. *Proceedings of the 17th International Conference on the Foundations of Digital Games*, pages 1–4, 2022.

[96] DeepMind Interactive Agents Team, Josh Abramson, Arun Ahuja, Arthur Brussee, Federico Carnevale, Mary Cassin, Felix Fischer, Petko Georgiev, Alex Goldin, Mansi Gupta, et al. Creating multimodal interactive agents with imitation and self-supervised learning. *arXiv preprint arXiv:2112.03763*, 2021.

[97] Shreyas Basavatia, Keerthiram Murugesan, and Shivam Ratnakar. STARLING: Self-supervised training of text-based reinforcement learning agent with large language models. *arXiv preprint arXiv:2406.05872*, 2024.

[98] Nazanin Yousefzadeh Khameneh and Matthew Guzdial. Entity embedding as game representation. *arXiv preprint arXiv:2010.01685*, 2020.

[99] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. *International Conference on Machine Learning*, pages 5639–5650, 2020.

[100] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

[101] Anna Dawid and Yann LeCun. Introduction to latent variable energy-based models: A path toward autonomous machine intelligence. *Journal of Statistical Mechanics: Theory and Experiment*, 2024(10):104011, 2024.

[102] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.

[103] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent - a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[104] Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. *International Conference on Machine Learning*, 2024.

[105] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

[106] Aaron Van Den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[107] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[108] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[109] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[110] Google DeepMind. Genie 2: A large-scale foundation world model. https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model. Accessed: 2024-12-11.

[111] Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, Bethanie Brownfield, Gavin Buttimore, Max Cant, Sarah Chakera, et al. Scaling instructable agents across many simulated worlds. *arXiv preprint arXiv:2404.10179*, 2024.

[112] Prabhjot Kaur, Samira Taghavi, Zhaofeng Tian, and Weisong Shi. A survey on simulators for testing self-driving cars. *Fourth International Conference on Connected and Autonomous Driving*, pages 62–70, 2021.

[113] Super Mario as a string: Platformer level generation via LSTMs, author=Summerville, Adam and Mateas, Michael, journal=arXiv preprint arXiv:1603.00930, year=2016.

[114] Greg S. Ruthenbeck and Karen J. Reynolds. Virtual reality for medical training: The state-of-the-art. *Journal of Simulation*, 9(1):16–26, 2015.

[115] Alfred T. Lee. *Flight simulation: Virtual environments in aviation*. Routledge, 2017.

[116] Konstantinos Andriopoulos and Johan Pouwelse. Augmenting LLMs with knowledge: A survey on hallucination prevention. *arXiv preprint arXiv:2309.16459*, 2023.

[117] Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S. M. Tonmoy, Aman Chadha, Amit P. Sheth, and Amitava Das. The troubling emergence of hallucination in large language models–an extensive definition, quantification, and prescriptive remediations. *arXiv preprint arXiv:2310.04988*, 2023.

[118] Gaurav Menghani. Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Computing Surveys*, 55(12):1–37, 2023.

[119] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

[120] Mushui Liu, Yuhang Ma, Yang Zhen, Jun Dan, Yunlong Yu, Zeng Zhao, Zhipeng Hu, Bai Liu, and Changjie Fan. LLM4GEN: Leveraging semantic representation of LLMs for text-to-image generation. *arXiv preprint arXiv:2407.00737*, 2024.

[121] Leigang Qu, Shengqiong Wu, Hao Fei, Liqiang Nie, and Tat-Seng Chua. LayoutLLM-T2I: Eliciting layout guidance from LLM for text-to-image generation. *Proceedings of the 31st ACM International Conference on Multimedia*, pages 643–654, 2023.

[122] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. LLM-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *arXiv preprint arXiv:2305.13655*, 2023.

[123] Niels Justesen, Philip Bontrager, Julian Togelius, and Sebastian Risi. Deep learning for video game playing. *IEEE Transactions on Games*, 12(1):1–20, 2019.

[124] Aiswarya Munappy, Jan Bosch, Helena Holmström Olsson, Anders Arpteg, and Björn Brinne. Data management challenges for deep learning. *45th Euromicro Conference on Software Engineering and Advanced Applications*, pages 140–147, 2019.

[125] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[126] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[127] Andrea Stocco, Michael Weiss, Marco Calzana, and Paolo Tonella. Misbehaviour prediction for autonomous driving systems. *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 359–371, 2020.

[128] Sergey I. Nikolenko. *Synthetic data for deep learning*, volume 174. Springer, 2021.

[129] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[130] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of Machine Learning and Systems*, 2:129–146, 2020.

[131] Mengxia Yu, De Wang, Qi Shan, and Alvin Wan. The super weight in large language models. *arXiv preprint arXiv:2411.07191*, 2024.

[132] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (XAI): A survey. *arXiv preprint arXiv:2006.11371*, 2020.

[133] Samuel Henrique Silva and Peyman Najafirad. Opportunities and challenges in deep learning adversarial robustness: A survey. *arXiv preprint arXiv:2007.00753*, 2020.

[134] Gravina, Daniele and Khalifa, Ahmed and Liapis, Antonios and Togelius, Julian and Yannakakis, Georgios N. Procedural content generation through quality diversity. *IEEE Conference on Games*, pages 1–8, 2019.

[135] Matthew Fontaine and Stefanos Nikolaidis. Differentiable quality diversity. *Advances in Neural Information Processing Systems*, 34:10040–10052, 2021.