Article

# World and Human Action Models towards gameplay ideation

https://doi.org/10.1038/s41586-025-08600-3

Received: 27 February 2024

Accepted: 6 January 2025

Published online: 19 February 2025

Open access

Check for updates

Anssi Kanervisto[1,7], Dave Bignell[1,7], Linda Yilin Wen[1,7], Martin Grayson[1,7], Raluca Georgescu[1,7], Sergio Valcarcel Macua[1,7], Shan Zheng Tan[1,7], Tabish Rashid[1,7], Tim Pearce[1,7], Yuhan Cao[1,7], Abdelhak Lemkhenter[1], Chentian Jiang[2], Gavin Costello[3], Gunshi Gupta[4], Marko Tot[5], Shu Ishida[4], Tarun Gupta[4], Udit Arora[1], Ryen W. White[6], Sam Devlin[1,7], Cecily Morrison[1,7] & Katja Hofmann[1,7]✉

Generative artificial intelligence (AI) has the potential to transform creative industries through supporting human creative ideation—the generation of new ideas[1–5]. However, limitations in model capabilities raise key challenges in integrating these technologies more fully into creative practices. Iterative tweaking and divergent thinking remain key to enabling creativity support using technology[6,7], yet these practices are insufficiently supported by state-of-the-art generative AI models. Using game development as a lens, we demonstrate that we can make use of an understanding of user needs to drive the development and evaluation of generative AI models in a way that aligns with these creative practices. Concretely, we introduce a state-of-the-art generative model, the World and Human Action Model (WHAM), and show that it can generate consistent and diverse gameplay sequences and persist user modifications—three capabilities that we identify as being critical for this alignment. In contrast to previous approaches to creativity support tools that required manually defining or extracting structure for relatively narrow domains, generative AI models can learn relevant structure from available data, opening the potential for a much broader range of applications.

Generative AI, which uses machine learning models to generate text[8,9], images[10,11], audio[12,13], music[14], video[15,16] or gameplay sequences of video games[17–19], has seen rapid uptake across the creative industries[1–3,5]. For example, generated images are used to facilitate communication between creatives on a team with different skill sets or to automate visual production tasks when an artist is not available[4]. However, studies have shown that generative AI capabilities often fall short of the expectations of creatives, raising key challenges in integrating these technologies more fully into creative practices[1,4,5,20,21].

Our work approaches this space through the lens of the gaming industry, as it provides an excellent use case to explore how AI capabilities could be innovated to support creativity[22]. The complexity of 3D game development requires a diverse range of creative skills[23], giving several viewpoints on how generative AI can be architected to enable all creative professions. Further, the richness and diversity of gameplay data offers key opportunities for innovation. This temporally correlated multimodal data affords exploration of increasingly complex tasks, from generating 3D worlds and their mechanics to exploring interactions with non-player characters (also known as NPCs). Not least, gaming is the entertainment industry's largest sector worldwide, at present reaching an audience of more than 3 billion people[24]. As such, game studios are exploring how AI can help them meet the increasing demand and expectations for new content[21].

In this article, we demonstrate that we can make use of an understanding of user needs to devise a methodology for evaluating generative AI models and drive generative AI model development that aligns with these creative practices. We begin with a summary of user study results from 27 creatives working in game development, illustrating the important role of divergent thinking and iterative practice[6,7] to achieve meaningful novelty using generative AI. Building on these insights, we identify a set of generative model capabilities that are probably important to realize creative ideation, namely, consistency, diversity and persistency (see Fig. 1a–c). We introduce a new generative model, WHAM, designed to achieve these capabilities and trained on human gameplay data. We show that WHAM can generate consistent and diverse gameplay sequences and that it can persist user modifications when prompted appropriately. Finally, we describe a concept prototype called the WHAM Demonstrator (Fig. 1d) to support exploration of creative uses and further research into the model capabilities required to support creative practice. We release WHAM's weights, an evaluation dataset and the WHAM Demonstrator as a basis for further research and exploration at https://huggingface.co/microsoft/wham.

Our work builds on a rich tradition of research at the intersection of computational creativity[7,25,26] and procedural content generation[27–32]. Today's generative AI approaches have great potential to complement these previous works because of their broad applicability: they can learn the rich structure of complex domains (such as 3D video games) from appropriate training data, removing the need for time-consuming, manual handcrafting of these structures. At the same time, our findings demonstrate that iterative practice and divergent thinking remain

[1]Microsoft Research, Cambridge, UK. [2]University of Edinburgh, Edinburgh, UK. [3]Ninja Theory, Cambridge, UK. [4]University of Oxford, Oxford, UK. [5]Queen Mary University, London, UK. [6]Microsoft Research, Redmond, WA, USA. [7]These authors contributed equally: Anssi Kanervisto, Dave Bignell, Linda Yilin Wen, Martin Grayson, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Tabish Rashid, Tim Pearce, Yuhan Cao, Sam Devlin, Cecily Morrison, Katja Hofmann. ✉e-mail: katja.hofmann@microsoft.com

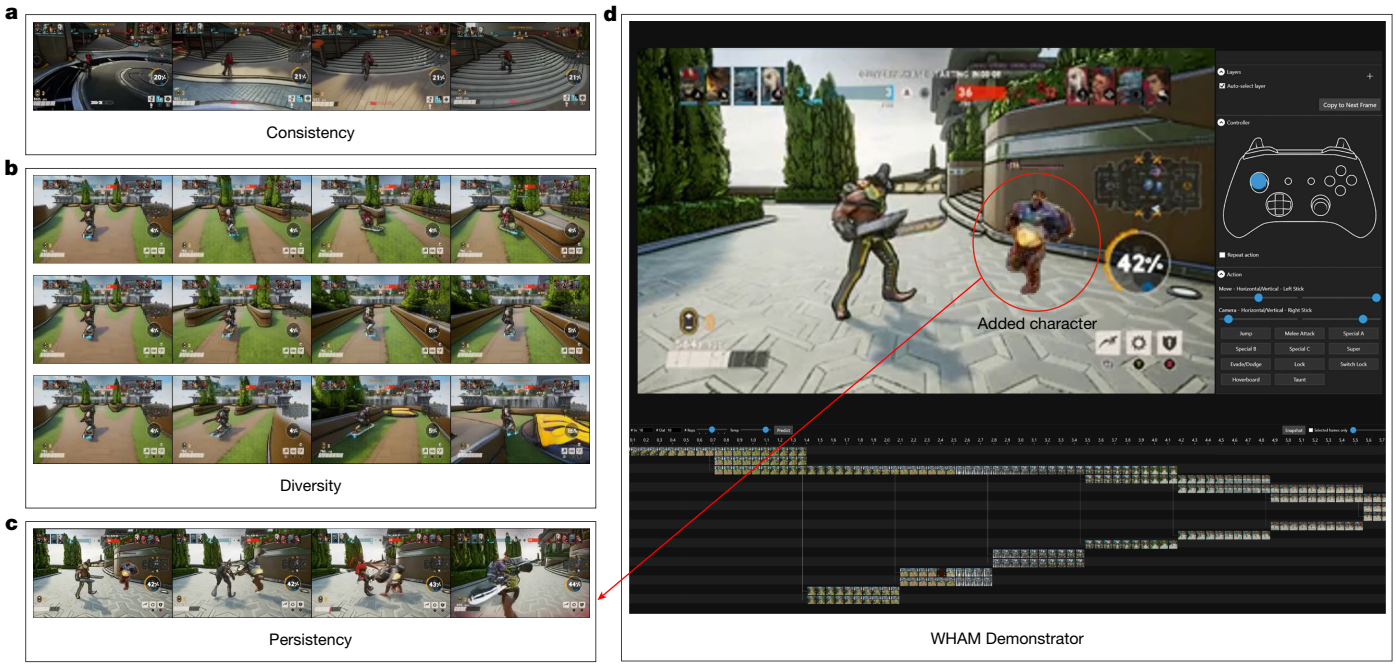



**Fig. 1 | Identified model capabilities.** The three model capabilities derived from our user study with game development creatives ('User needs' section) demonstrated through gameplay sequences generated by WHAM ('WHAM' section) in the WHAM Demonstrator ('WHAM Demonstrator' section). **a**, Consistency: a generated sequence should be consistent over time and with game mechanics. Here the player's character navigates up the stairs, following the established physics of the game world. **b**, Diversity: the model should produce numerous, diverse sequences that reflect different potential outcomes to support divergent thinking. Here the model generated three plausible sequences navigating paths the character could follow. **c**, Persistency: the model should persist user modifications to the game visuals and controller actions, assimilating them into the generated gameplay sequence. Here the character highlighted in the right figure has been added by the user and has then been incorporated into the generated images shown on the left. **d**, Screenshot of the WHAM Demonstrator, a concept prototype that provides a visual interface for interacting with WHAM models, including several ways of prompting the models. See Supplementary Video 1 for video case studies.

crucial in the context of ideation using generative AI models. By optimizing models towards these proposed capabilities, we direct machine learning research towards innovations for the type of human–AI partnership that will empower human creativity and agency.

## User needs

### Interview study

To better understand the needs of creatives working in game development, we carried out semistructured interviews with a diverse set of multidisciplinary creative teams. In each interview session, three to four creatives from the same studio interacted with a design probe[33] (see the 'Design probe' section in Methods and Extended Data Fig. 1a for details) that provided a fictitious but concrete set of potential generative AI capabilities to spur thinking. Participants described several ways in which generative AI could assist in game ideation or pre-production ('Game development process' section in Methods), while maintaining their creative agency.

Focusing specifically on participants' discussions of AI and creative practice, we analysed the discussion transcripts using thematic analysis[34] ('Data analysis' section in Methods and Extended Data Fig. 1b). We identified two themes that have implications for AI model development: (1) creatives need the diversity of their divergent thinking contextualized into a consistent game world to achieve meaningful new experiences ('Divergent thinking' section) and (2) to experience creative agency, creatives need the ability to control the iterative process (iterative practice), for example, with their direct modifications adopted as they guide the model ('Iterative practice' section).

**Divergent thinking.** Creatives in our study had already used generative AI models to seek inspiration and drive divergent thinking to produce new ideas, as also shown in other literature[21]. Nevertheless, the creatives spoke about the need for novelty to be framed within the consistency of professional practice. This remains a challenge for present generative AI models[21]. In game development, for example, consistency includes: upholding game world physics; adhering to the style of the title and the studio; maintaining the specific atmosphere and emotions that the level intends to evoke; and ensuring alignment with the larger narrative of the game[35], whereas diversity might apply to the path a player takes. Without contextual consistency, diversity in generated outputs risks being devoid of meaningful importance[36]. As one participant shares:

> Generative AI still has kind of a limited amount of context. This means it's difficult for an AI to consider the entire experience and kind of generate iteratively on top of that, the AI still isn't very good at kind of keeping generating and then kind of following specific rules and mechanics, you know, because it's inconsistent.
> – Vice President of Experience of an indie studio

In other words, supporting ideation is not just about novelty but about contextualizing that novelty into the coherence of an interactive experience or game. Consequently, generative AI models need to combine diversity with consistency to ensure that outputs are meaningfully new and useful.

**Iterative practice.** The importance of iteration in the ideation process is well described in the literature on creativity support[37,38]. Participants in our study frequently expressed the importance of iterative practice, which highlights that this theme continues to be crucial in the context of creative uses enabled by generative AI.

Specifically, participants spoke of making something that feels 'right', underscoring the intuition that game creators have about the numerous

nuanced elements that make up each design decision. Whether it be the tempo of the character's movements or the arc of a grappling hook swing, creators invested considerable time fine-tuning these seemingly minor details. As one participant said: "details are what make really amazing game experiences". Nevertheless, this feeling of 'rightness' was often nebulous at the outset of the creative process, becoming clearer only as the process evolved:

> It's hard to know what the right output is until we see it, and it takes a lot of finessing it and playing with it. There is a lot of trial and error. As game designers, we're not even conscious of the details where there are thousands of small decisions to be made. But we just know something's off and we tweak.
> – Chief Operating Officer of an indie studio

This description illustrates how creatives usually work in the visual medium, directly manipulating what they are creating through several, small iterations. The iterative process extends beyond a singular output: many participants noted that they engage in a dynamic back-and-forth exploration between different iterations to draw inspiration and experiment with the possibilities of fusing diverse elements. To facilitate ideation through iterative tweaking, generative AI models should move beyond text-based prompts and support direct manipulation of the generated content, have an ability to adopt user-proposed changes and support fusing of different iterations.

### Evaluating model capabilities

Support for divergent thinking and iterative practice has been provided in a range of ways across the rich literature and practice in this area[7,26,37], but when it comes to generative AI, we find important gaps. On the basis of the results of our user study, coupled with insights from existing literature, we distil evaluation criteria, or 'model capabilities', to assess the diversity, consistency and persistency of generative AI models to support the very basics of creative practice.

To provide concrete examples of what the identified evaluation criteria mean and how they can be instantiated, we assume generative AI that operates at the most generic 'human interface' of a video game, in the sense that it is able to generate sequences of game visuals (what the player would see on the screen, referred to as 'frames') and players' controller actions. However, the evaluation criteria are general and could be instantiated in different modalities, such as language, music and so on.

To support iterative practice, a first important criterion is that models provide consistency, even while a user is iterating. This means that a stream of generated frames must be consistent in themselves (for example, frame to frame) and in terms of the game mechanics, for example, solid objects do not pass through walls. Within this consistency, the creative practice of divergent thinking requires diverse generations. For example, if three potential continuations are generated, they should vary in meaningful ways, such as in the generated player actions, or in terms of how teammates or opponent characters might respond to those actions. Finally, users should be able to modify generated sequences and any modifications should be persistent. If a creative wishes to influence the model output by adjusting a frame, the adjustment should be a focus of the generation and not disappear several frames later.

## WHAM

Now that we have established an understanding of the key capabilities required to realize AI systems that enable creatives, we present an initial model that demonstrates how modern AI approaches can make progress towards achieving these capabilities.

Our WHAM models the dynamics of a modern video game over time. WHAM was trained on human gameplay data to predict game visuals ('frames') and players' controller actions ('Model architecture and data' section). The resulting model accurately captures the 3D structure of the game environment ('Model evaluation' section), the effects of controller actions and the temporal structure of the game. The model can be prompted to generate coherent game situations, demonstrating consistency and diversity and the ability to persist some user modifications.

In our model development and evaluation, we focus on the generation of gameplay sequences in the form of game visuals and player actions, as this is a very generic and broadly accessible representation of a video game. We build on the rich line of work on world models[39] that has demonstrated the potential of recurrent networks[40], recurrent state space models[41] and transformers[42] for capturing environment dynamics in settings such as 2D video games and road traffic[43]. Moving beyond these and contemporary works[18,19,44–47], we drive insights about the requirements and capabilities of these models specifically for creative uses and demonstrate advances in modelling a complex 3D video game consistently over time.

### Model architecture and data

Our modelling choices reflect the identified model capabilities as follows. Consistency requires a sequential model that can accurately capture dependencies between game visuals and controller actions. Diversity requires a model that can generate data that preserve the sequential conditional distribution of visuals and controller actions from the dataset. Finally, persistency is afforded through a predictive model that can be conditioned on (modified) images and/or controller actions. Across all three capabilities, we select components that offer scalability in the sense that the model should benefit from training on large amounts of training data and compute resources.

The resulting WHAM design is shown in Fig. 2. It is built on the transformer architecture[48,49] as its sequence prediction backbone. Transformers gained popularity through their application in large language models and have also been adopted by previous world-modelling approaches[42,43,50].

Critical to our approach is our framing of the data as a sequence of discrete tokens. To encode an image into a sequence of tokens, we make use of a VQGAN image encoder[51]. The number of tokens used to encode each image is a key hyperparameter that trades off the quality of predicted images with generation speed and context length. For the Xbox controller actions, although the buttons are natively discrete, we discretize the $x$ and $y$ coordinates of the left and right joysticks into 11 buckets[52]. We then train a decoder-only transformer[49,53] to predict the next token in the sequence of interleaved image and controller actions.

The resulting model can then generate new sequences by autoregressively sampling the next token. We can also modify the tokens during the generation process to allow for modifications to the images and/or actions. This unlocks the ability to control (or prompt) the generation through the controller actions or by directly editing the images themselves, a prerequisite for persistency that we evaluate in the 'Persistency' section.

To demonstrate the potential of this framework for capturing the dynamics of modern video games, we use a large dataset of real human gameplay to train WHAM. We worked with the game studio Ninja Theory and their game Bleeding Edge, a 3D, 4v4 multiplayer combat video game, to render and produce videos of human gameplay. In total, we extracted data from around 500,000 anonymized gaming sessions (over 7 years of continuous play) across all seven Bleeding Edge maps. We refer to this dataset as the 7 Maps dataset. We also filter this dataset to 1 year of anonymized gameplay on only the Skygarden map and refer to this as the Skygarden dataset. See the 'Data' section in Methods for details on data collection for the resulting datasets.

The largest WHAM uses a 1.6B-parameter transformer, with a 1-s context length, trained on the 7 Maps dataset. For this variant, each image is

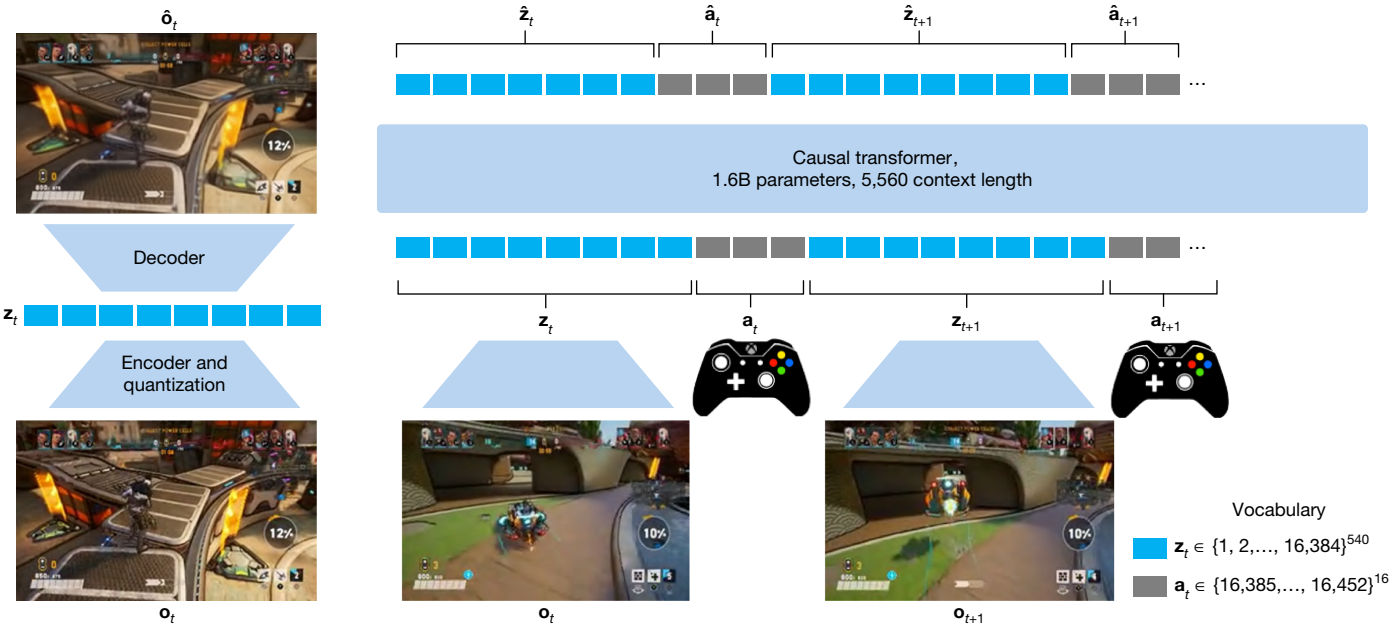



**Fig. 2 | Overview of WHAM.** We formulate human gameplay as sequences of discrete tokens, alternating between image observations and controller actions. We use $\mathbf{z}_t$ to refer to all tokens encoding an observation $\mathbf{o}_t$ at time step $t$ and $\mathbf{a}_t$ for the controller action. Hatted variables denote model predictions. A VQGAN[51] tokenizes the images from observation space, $\mathbf{o}_t \in \mathbb{R}^{H\times W\times 3}$ (in which $H$, $W$ and 3 refer to the height, width and number of channels of the video frames, respectively), to a compact discrete latent space $\mathbf{z}_t \in \{1, 2, \ldots, V_O\}^{d_z}$, for vocabulary size $V_O$ and bottleneck size $d_z$. A causal transformer[53] is then trained to predict the latent observation and discretized action tokens. The VQGAN encoder/decoder is trained using a reconstruction and perceptual loss[61]. No explicit delimiter is provided to distinguish whether an observation or action token should be predicted next—the model must infer this from learned position embeddings.

encoded into 540 tokens at the dataset's native resolution of 300 × 180. We also trained a range of smaller WHAMs: from 15M-parameter to 894M-parameter transformers with a 1-s context length, trained on the filtered Skygarden dataset, with 128 × 128 images encoded into 256 tokens. Further details on modelling choices and hyperparameters are provided in the 'Modelling choices and hyperparameters' section and model scalability is analysed in the 'Model scale' section, both in Methods.

### Model evaluation

We propose a methodology to evaluate models in terms of the three capabilities identified in our user study ('Evaluating model capabilities' section) to support ideation: consistency, diversity and persistency. We use this methodology to evaluate WHAM. The 'Consistency' section evaluates how consistent generated gameplay is with the game mechanics. The 'Diversity' section investigates the diversity of the generated gameplay. Finally, the 'Persistency' section explores the extent to which user modifications persist in the generations.

**Consistency.** Consistency ensures that creatives can effectively iterate and build on the generated sequences and is therefore key to iterative practice. In the game context, this means that a generated sequence should be consistent with the established game dynamics and remain coherent throughout, with no sudden changes to game characters or objects. For example, characters should not pass through walls and objects should not disappear without cause.

An established approach for measuring consistency in video in the field of machine learning is Fréchet Video Distance (FVD)[54], a measure that was designed to capture the quality of the temporal dynamics and visual quality of a video, and that has been shown to correlate with human judgements of video quality. Here we adapt FVD to the task of measuring consistency in generated gameplay by using human gameplay as the ground truth. For this, we use WHAM to generate gameplay visuals, conditioned on 1 s of gameplay, including video and controller actions, followed by conditioning on the controller actions taken by the human player over the course of the following 10 s of gameplay. Generated gameplay that closely matches the ground truth, as indicated by a low FVD score, provides evidence that the model has accurately captured the structure of the underlying game (for details, see the 'Consistency' section in Methods). We have validated the link between low FVD score and high human-perceived consistency in a preliminary analysis using the 894M WHAM ('Consistency' section in Methods and Extended Data Fig. 3).

Figure 3a shows the improvement of FVD with compute (in FLOPS) across model sizes (detailed in Extended Data Fig. 2c), showing improved FVD with more compute for appropriately sized models (see our discussion of model scale in the 'Model scale' section in Methods and results in Extended Data Fig. 2a,b for comparison). Furthermore, we see an improvement in FVD for the 1.6B WHAM, which uses higher-resolution images. This is because the ceiling on reconstruction performance is much higher, allowing for the generated images to much more closely resemble the ground truth data.

Figure 3b shows qualitative results, demonstrating that the 1.6B WHAM can generate highly consistent gameplay sequences of up to 2 min. More examples are shown in Extended Data Fig. 4 and in Supplementary Video 1.

**Diversity.** Providing creatives with diverse options has been shown to support human creative ideation by sparking new ideas[21,55], and the need for meaningful diversity was highlighted by participants in our user study ('Divergent thinking' section). Consequently, generative AI models aimed at supporting creativity should generate material that reflects a range of different potential outcomes. As the space of possibilities is vast[36] (encompassing game mechanics, other players, as well as randomness in the game), we focus our evaluation on the ability of the models to capture the full diversity of a human player's actions. If the model is able to generate this diversity while maintaining consistency (measured separately by FVD as detailed above), then the

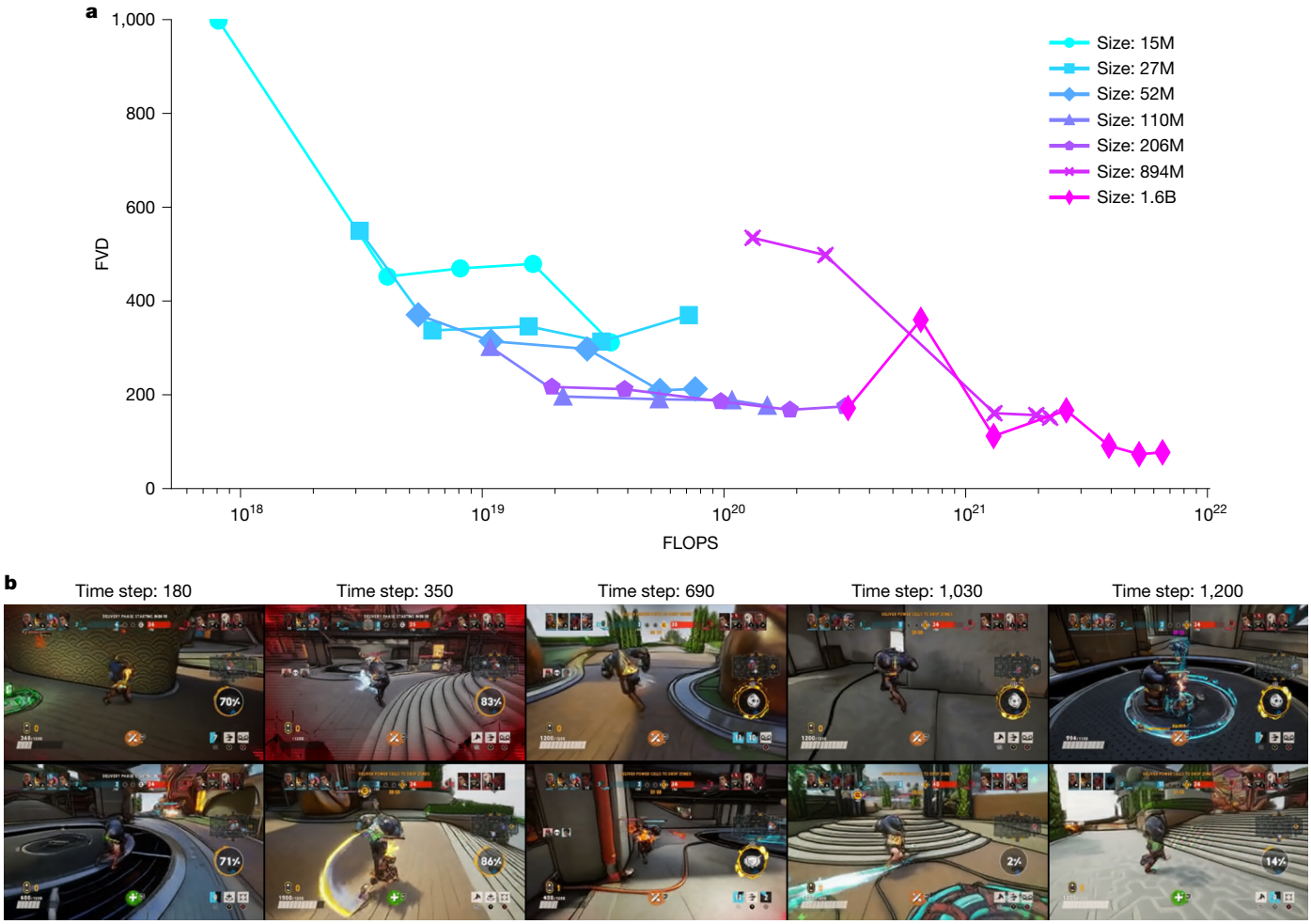



**Fig. 3 | Consistency results. a**, FVD for a range of WHAM sizes over training compute budget (FLOPS). FVD improves for larger models and compute budgets. **b**, Key frames of two example generations (one per row) from the 1.6B WHAM of 2 min each, indicating that the 1.6B WHAM is capable of generating long-term consistent gameplay.

generated gameplay sequences will reflect the full diversity of plausible human gameplay.

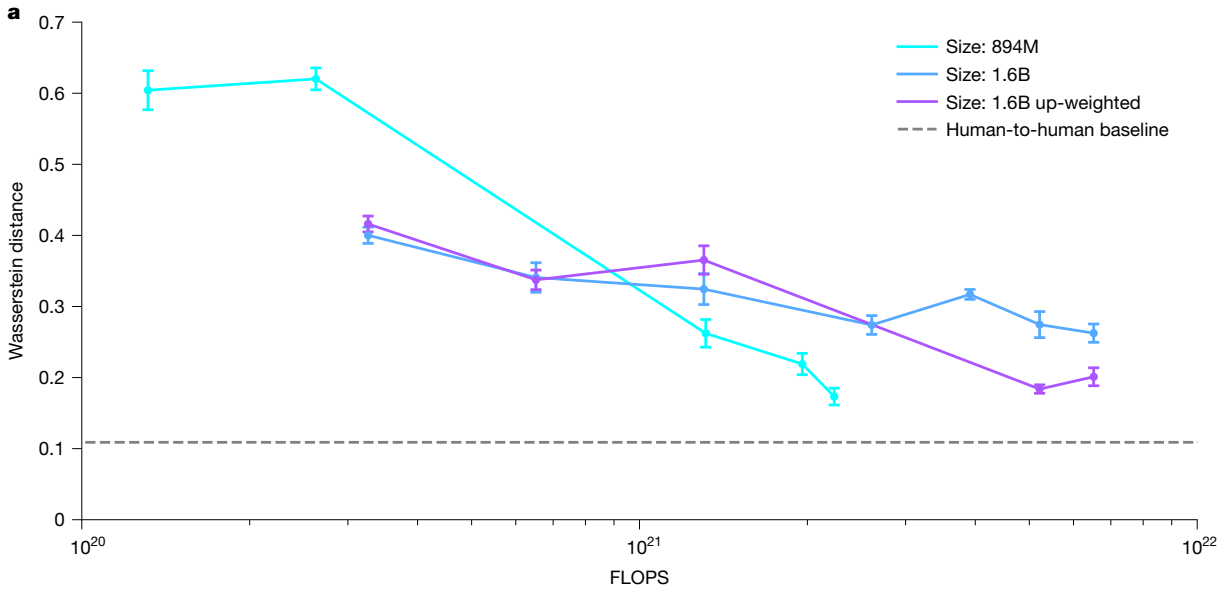
We assess diversity using the Wasserstein distance, a measure of the distance between two distributions previously used to assess whether the actions of a model capture the full distribution of human actions[56]. We compare the marginal distribution over real human actions with those generated by the model. The lower the Wasserstein distance, the closer the generations of the model are to the actions the human players took in our dataset (see the 'Diversity' section in Methods for further details).

Figure 4a shows our quantitative results. Over the course of training, the Wasserstein distance decreases for all models, nearing the human-to-human baseline (computed as the average distance between two random subsets of actions from the human action sequences). Despite using more compute, the 1.6B model is slightly worse compared with the 894M model. One hypothesis for this is that the 1.6B model uses more image tokens (540 compared with 256) and a larger vocabulary size (16,384 compared with 4,096), both of which implicitly put less emphasis on the loss for the tokens representing the actions. To test this, we train another 1.6B model with a ten times increased weight on the action loss ('1.6B up-weighted'). This up-weighting provides an improvement in the Wasserstein distance compared with the 1.6B model.

Figure 4b provides a qualitative assessment of diversity. Conditioned on a single sequence of real gameplay, three possible futures are generated using the 1.6B WHAM, showing that the model can generate a range of behaviourally and visually diverse gameplay sequences. Extended Data Fig. 5 highlights examples of behavioural (Extended Data Fig. 5b) and visual (Extended Data Fig. 5c) diversity in generated gameplay sequences.

**Persistency.** Persistency is aimed at giving creatives control over the generated outputs, thus enabling iterative tweaking ('Iterative practice' section). The model should be flexible enough to allow creative users' modifications to the game state, assimilating these changes into the generated environment.

To evaluate the persistency of WHAM, we manually edited game images by inserting one of three different elements: (1) an in-game object (a 'Powercell'); (2) another player (an allied or opponent character); and (3) a map element (a 'Vertical Jumppad'). We inserted each element into eight plausible but new game locations (shown in Extended Data Fig. 7a). For each element and location, we used the 1.6B WHAM to generate ten images, that is, a 1-s video, conditioned on either one or five of the altered images. To account for diversity in the output of the model, we repeated the generation step ten times per altered image(s). We then manually inspected and labelled whether each element persisted in the generated videos. Figure 5 shows the editing process and examples of generated videos. Extended Data Fig. 6 illustrates the human labelling of successful and unsuccessful persistency examples.



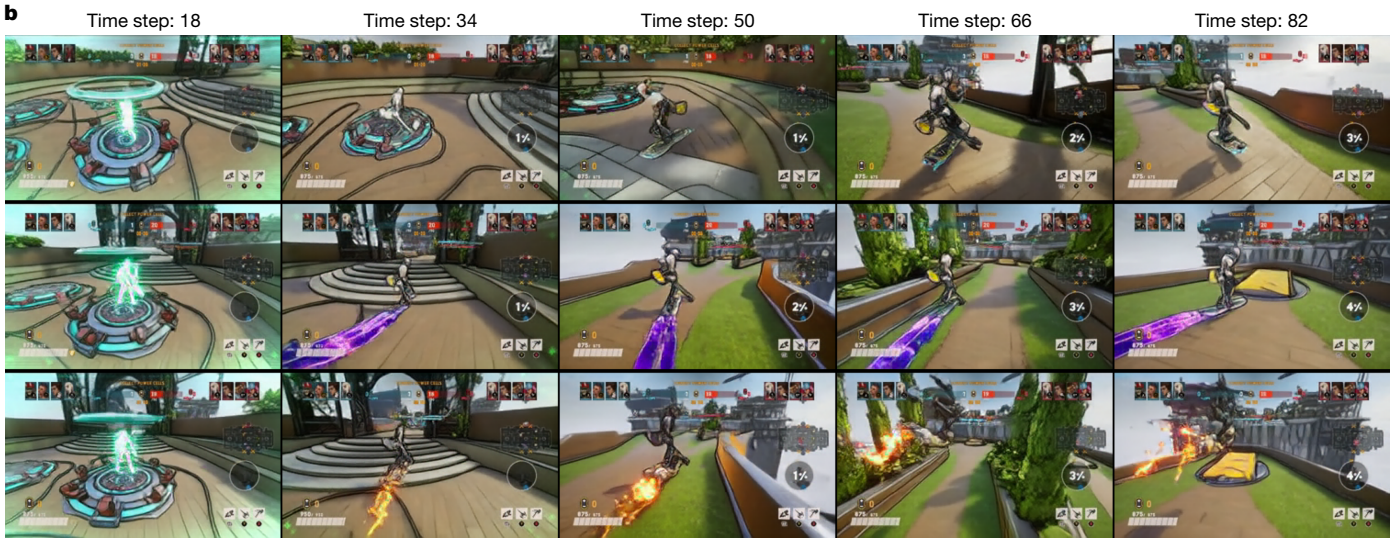




**Fig. 4 | Diversity results. a**, Diversity of three WHAM variants as measured by the Wasserstein distance to human actions. Out of the 102,400 total actions (1,024 trajectories with 100 actions each), we sub-sample 10,000 human and model actions and compute the distance between them. We repeat this ten times and plot the mean ± 1 standard deviation. Closer to the human-to-human baseline is better. Uniform random actions have a distance of 5.3. All models improve through training and can be further improved by up-weighting the action loss. **b**, Three examples of generations from the 1.6B WHAM produced from the same starting context. We see examples of both behavioural diversity (the player character circling the spawn location versus heading straight towards a Jumppad) and visual diversity (the hoverboard the player character has mounted has different skins).

Table 1 presents results showing the proportion of generations that were annotated as successfully persisting. The persistency of WHAM improves substantially when conditioning on five edited images rather than one, reaching 85% and higher for all element types. More detailed analyses and examples of persistency are included in the 'Persistency' section in Methods. Extended Data Fig. 7b left column shows a detailed analysis of persistency by element type and starting location and Extended Data Fig. 7b right column shows an error analysis of starting location, in which persisting elements is more challenging. Supplementary Video 1 shows generated gameplay sequences that include interactions with the inserted elements.

Our results show that the 1.6B WHAM is able to persist common game elements that have been inserted into plausible but new starting locations. We believe that these examples demonstrate the potential for the creative uses of future WHAM versions to incorporate more imaginative elements into generated sequences.

### WHAM Demonstrator

To illustrate how WHAM can support iterative practice and divergent thinking as identified in our user study, we built a concept prototype[57], called the 'WHAM Demonstrator'. Note that concept prototypes are not full-fledged user experiences but rather explorations of specific design patterns. The WHAM Demonstrator provides a visual interface for interacting with WHAM instances, including several ways of prompting the models. This facilitates explorations of WHAM capabilities, as well as interaction patterns supported by these capabilities. To enable creative exploration and follow-up research, we make the following publicly available: trained models (two WHAM sizes), the WHAM Demonstrator and a sample evaluation dataset (see 'Data availability' and 'Code availability' for details).

We demonstrate key features in Supplementary Video 1. First, the video illustrates the identified model capabilities. Consistency is


**Fig. 5 | Editing process and qualitative persistency results.** Examples of successful persistence of Powercell, character and Vertical Jumppad. For our persistency evaluation, the generations of WHAM are all conditioned on no-op actions, hence the player character and camera should not be moving. The examples show the inserted Powercell persisting stably throughout the 1 s of generation and the inserted opponent beginning to attack the player character and inflicting damage. The Vertical Jumppad is inserted into a map area in which it does not appear in the real game and our data. Nevertheless, it is persisted throughout the generations of WHAM.

demonstrated in a case study over the course of training, showing how the ability to generate gameplay sequences that are consistent over time and with a wide range of game mechanics improve with training (00:50–02:10). Diversity is illustrated in a case study of generated gameplay sequences that all start from the same initial spawn location and shows examples of the character navigating across the three available Jumppads (02:11–02:50). Finally, persistency shows case studies of persisted characters and Powercells, corresponding to those aggregated in Table 1 (02:51–03:42).

Second, we illustrate the features of the WHAM Demonstrator in Fig. 1d and in Supplementary Video 1 (from 03:43). A user can choose a set of starting frames to 'prompt' the model[58], enabling visual rather than language-based prompts. WHAM then generates numerous branches of potential gameplay sequences of how the game could evolve, supporting divergent thinking through a diversity of options ('Divergent thinking' section). The user can choose any branch or frame to start (re)generating the next frames, including returning to, and changing, a previous choice to support the fusing of iterations mentioned by participants above ('Iterative practice' section). To enable iteration, the user can modify any generated frames, such as by adding an opponent character (using persistency) or providing input controller data, to influence the next generated sequences. The user can tweak and iterate until they get the 'feel' they are looking for, remaining in control of their creative practice.

## Conclusion

As we navigate the unfolding role of generative AI in the creative industries, there are ways to direct its development to ensure human agency over the creative process. We have presented a user study with diverse game creatives through which we identified three model capabilities that should be given priority when developing AI systems that aim to support creative ideation through iterative practice and divergent thinking: consistency, diversity and persistency. We have also shown that it is possible to develop generative AI models that exhibit these capabilities when trained on appropriate datasets.

Our work suggests new paths of innovation for machine learning researchers that are different from those aimed at models not intended to support creativity. First, model evaluation can, and should, be purposefully informed by the requirements of human creatives to drive innovation in the right direction. This stands in contrast to a predominant focus in the machine learning community on measuring the effectiveness and efficiency of task completion, useful only when human tasks will be automated to support process efficiencies. Second, machine learning models for creativity will unlikely be ends in themselves but, rather, valuable assets within more holistic creative workflows. Model development must fit within these workflows, the need for several iterations of user-modified content being one such example. The literature on computational creativity and creativity support is a rich source of guidance[7,25,26] as the field starts to more fully connect these model innovations with the needs of creatives.

The demonstrated capabilities of WHAM showcase the potential of modern generative AI models to learn increasingly complex structures from relevant data without previous domain knowledge. We show that such models can generate gameplay sequences that are consistent with 3D worlds with appropriate game mechanics and physics. Given that WHAM learned these structures entirely from gameplay data, with no previous domain knowledge, we expect that these results can be replicated across a wide range of existing games and ultimately generalize to new games and genres[18,32]. The key novelty that generative AI models such as WHAM contribute is that they remove the need for handcrafting or learning domain-specific models for individual domains, making it likely that model innovations such as these will broaden creativity support to other domains, such as music[59] or video[60]. Extrapolating from our use case focusing on a single 3D video game, we can also get a first sense of how powerful future models will be in allowing teams of human creators to craft complex new experiences.

**Table 1 | Quantitative persistency results**

| Conditioned frames | Powercell | Character | Vertical Jumppad |
|---|---|---|---|
| 1 | 58% | 45% | 58% |
| 5 | 86% | 85% | 98% |

When conditioning WHAM on one user-edited image, each element persists less than 60% of the time. However, when conditioning on five user-edited images, the persistency of each element increases substantially to 85% or more (binomial tests with a Bonferroni corrected significance level of 0.008).

## Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information,

acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-025-08600-3.


1. Mim, N. J., Nandi, D., Khan, S. S., Dey, A. & Ahmed, S. I. In-between Visuals and Visible: The Impacts of Text-to-image Generative AI Tools on Digital Image-making Practices in the Global South. In *Proc. CHI Conference on Human Factors in Computing Systems* (Association for Computing Machinery, 2024).
2. Eapen, T. T., Finkenstadt, D. J., Folk, J. & Venkataswamy, L. How generative AI can augment human creativity. *Harv. Bus. Rev.* **101**, 76–85 (2023).
3. Guzdial, M., Snodgrass, S. & Summerville, A. J. Procedural Content Generation via Machine Learning: An Overview (Springer, 2022).
4. Ko, H.-K. et al. Large-scale text-to-image generation models for visual artists' creative works. In *Proc. 28th International Conference on Intelligent User Interfaces* 919–933 (Association for Computing Machinery, 2023).
5. Oppenlaender, J. The creativity of text-to-image generation. In *Proc. 25th International Academic Mindtrek Conference* 192–202 (Association for Computing Machinery, 2022).
6. Sternberg, R. J. *Handbook of Creativity* (Cambridge Univ. Press, 1999).
7. Resnick, M. et al. Design principles for tools to support creative thinking. In *National Science Foundation Workshop on Creativity Support Tools* (University of Maryland, 2005).
8. OpenAI. GPT-4 technical report. Preprint at https://arxiv.org/abs/2303.08774 (2023).
9. Touvron, H. et al. Llama 2: open foundation and fine-tuned chat models. Preprint at https://arxiv.org/abs/2307.09288 (2023).
10. Betker, J. et al. Improving image generation with better captions. *OpenAI* https://cdn.openai.com/papers/dall-e-3.pdf (2023).
11. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 10684–10695 (IEEE, 2022).
12. Kreuk, F. et al. AudioGen: textually guided audio generation. In *Proc. 11th International Conference on Learning Representations* (ICLR, 2023).
13. Liu, H. et al. AudioLDM: Text-to-audio generation with latent diffusion models. In *Proc. 40th International Conference on Machine Learning* 21450–21474 (PMLR, 2023).
14. Copet, J. et al. Simple and controllable music generation. In *Proc. Advances in Neural Information Processing Systems 36 (NeurIPS 2023)* (eds Oh, A. et al.) (NeurIPS, 2023).
15. Brooks, T. et al. Video generation models as world simulators. *OpenAI* https://openai.com/research/video-generation-models-as-world-simulators (2024).
16. Blattmann, A. et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. Preprint at https://arxiv.org/abs/2311.15127 (2023).
17. Kim, S. W., Zhou, Y., Philion, J., Torralba, A. & Fidler, S. Learning to simulate dynamic environments with gameGAN. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 1231–1240 (IEEE, 2020).
18. Bruce, J. et al. Genie: generative interactive environments. In *Proc. 41st International Conference on Machine Learning* 4603–4623 (PMLR, 2024).
19. Valevski, D., Leviathan, Y., Arar, M. & Fruchter, S. Diffusion models are real-time game engines. Preprint at https://arxiv.org/abs/2408.14837 (2024).
20. Oppenlaender, J., Silvennoinen, J., Paananen, V. & Visuri, A. Perceptions and realities of text-to-image generation. In *Proc. 26th International Academic Mindtrek Conference* 279–288 (Association for Computing Machinery, 2023).
21. Vimpari, V., Kultima, A., Hämäläinen, P. & Guckelsberger, C. "An adapt-or-die type of situation": perception, adoption, and use of text-to-image-generation AI by game industry professionals. *Proc. ACM Hum. Comput. Interact.* **7**, 131–164 (2023).
22. Yannakakis, G. N. & Togelius, J. *Artificial Intelligence and Games* (Springer, 2018).
23. Schell, J. *The Art of Game Design: A Book of Lenses* (CRC Press, 2008).
24. Buijsman, M. Newzoo's global games market report 2024. *Newzoo* https://newzoo.com/resources/trend-reports/newzoos-global-games-market-report-2024-free-version (2024).
25. Boden, M. A. Creativity and artificial intelligence. *Artif. Intell.* **103**, 347–356 (1998).
26. Lubart, T. How can computers be partners in the creative process: classification and commentary on the special issue. *Int. J. Hum. Comput. Stud.* **63**, 365–369 (2005).
27. Smith, G., Whitehead, J. & Mateas, M. Tanagra: A mixed-initiative level design tool. In *Proc. 5th International Conference on the Foundations of Digital Games* 209–216 (Association for Computing Machinery, 2010).
28. Smith, G. & Whitehead, J. Analyzing the expressive range of a level generator. In *Proc. 2010 Workshop on Procedural Content Generation in Games* (Association for Computing Machinery, 2010).
29. Smith, G. Understanding procedural content generation: a design-centric analysis of the role of PCG in games. In *Proc. SIGCHI Conference on Human Factors in Computing Systems* 917–926 (Association for Computing Machinery, 2014).
30. Guzdial, M. & Riedl, M. Game level generation from gameplay videos. In *Proc. 12th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 44–50 (AAAI, 2016).
31. Guzdial, M., Li, B. & Riedl, M. O. Game engine learning from video. In *Proc. 26th International Joint Conference on Artificial Intelligence* (ed. Sierra, C.) 3707–3713 (AAAI, 2017).
32. Guzdial, M. & Riedl, M. O. Conceptual game expansion. *IEEE Trans. Games* **14**, 93–106 (2021).
33. Wallace, J., McCarthy, J., Wright, P. C. & Olivier, P. Making design probes work. In *Proc. SIGCHI Conference on Human Factors in Computing Systems* 3441–3450 (Association for Computing Machinery, 2013).
34. Cairns, P. & Cox, A. L. *Research Methods for Human-computer Interaction* (Cambridge Univ. Press, 2008).
35. Short, T. X. & Adams, T. *Procedural Generation in Game Design* (CRC Press, 2017).
36. Rabii, Y. & Cook, M. Why oatmeal is cheap: Kolmogorov complexity and procedural generation. In *Proc. 18th International Conference on the Foundations of Digital Games* (eds Lopes, P. et al.) (Association for Computing Machinery, 2023).
37. Frich, J., Nouwens, M., Halskov, K. & Dalsgaard, P. How digital tools impact convergent and divergent thinking in design ideation. In *Proc. 2021 CHI Conference on Human Factors in Computing Systems* (Association for Computing Machinery, 2021).
38. Liapis, A., Yannakakis, G. N., Nelson, M. J., Preuss, M. & Bidarra, R. Orchestrating game generation. *IEEE Trans. Games* **11**, 48–68 (2018).
39. Ha, D. & Schmidhuber, J. World models. Preprint at https://arxiv.org/abs/1803.10122 (2018).
40. Ha, D. & Schmidhuber, J. Recurrent world models facilitate policy evolution. In *Proc. Advances in Neural Information Processing Systems 31 (NeurIPS 2018)* (eds Bengio, S. et al.) (NeurIPS, 2018).
41. Hafner, D., Lillicrap, T. P., Norouzi, M. & Ba, J. Mastering Atari with discrete world models. In *Proc. 9th International Conference on Learning Representations* (ICLR, 2021).
42. Micheli, V., Alonso, E. & Fleuret, F. Transformers are sample-efficient world models. In *Proc. 11th International Conference on Learning Representations* (ICLR, 2023).
43. Hu, A. et al. GAIA-1: a generative world model for autonomous driving. Preprint at https://arxiv.org/abs/2309.17080 (2023).
44. Alonso, E. et al. Diffusion for world modeling: visual details matter in Atari. In *Proc. Advances in Neural Information Processing Systems 37 (NeurIPS 2024)* (eds Globerson, A. et al.) (NeurIPS, 2024).
45. Astolfi, P. et al. Consistency-diversity-realism Pareto fronts of conditional image generative models. Preprint at https://arxiv.org/abs/2406.10429 (2024).
46. Rigter, M., Yamada, J. & Posner, I. World models via policy-guided trajectory diffusion. *TMLR* https://openreview.net/forum?id=9CcgO0LhKG (2024).
47. Yang, S. et al. Learning interactive real-world simulators. In *Proc. 12th International Conference on Learning Representations* (ICLR, 2024).
48. Vaswani, A. et al. Attention is all you need. In *Proc. Advances in Neural Information Processing Systems 30 (NIPS 2017)* (eds Guyon, I. et al.) (NIPS, 2017).
49. Bishop, C. M. & Bishop, H. *Deep Learning: Foundations and Concepts* (Springer, 2024).
50. Yan, W., Zhang, Y., Abbeel, P. & Srinivas, A. VideoGPT: video generation using VQ-VAE and transformers. Preprint at https://arxiv.org/abs/2104.10157 (2021).
51. Esser, P., Rombach, R. & Ommer, B. Taming transformers for high-resolution image synthesis. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 12873–12883 (IEEE, 2021).
52. Kanervisto, A., Scheller, C. & Hautamäki, V. Action space shaping in deep reinforcement learning. In *Proc. 2020 IEEE Conference on Games (CoG)* 479–486 (IEEE, 2020).
53. Radford, A. et al. Language models are unsupervised multitask learners. *OpenAI* https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf (2019).
54. Unterthiner, T. et al. FVD: A new metric for video generation. In *Proc. ICLR 2019 Workshop Deep Generative Models for Highly Structured Data* (ICLR, 2019).
55. Lee, J. H. & Ostwald, M. J. The relationship between divergent thinking and ideation in the conceptual design process. *Des. Stud.* **79**, 101089 (2022).
56. Pearce, T. et al. Imitating human behaviour with diffusion models. In *Proc. 11th International Conference on Learning Representations* (ICLR, 2023).
57. Rogers, Y., Sharp, H. & Preece, J. *Interaction Design: Beyond Human–Computer Interaction* (Wiley, 2001).
58. Brown, T. et al. Language models are few-shot learners. In *Proc. Advances in Neural Information Processing Systems 33 (NeurIPS 2020)* (eds Larochelle, H. et al.) 1877–1901 (NeurIPS, 2020).
59. Cope, D. *Virtual Music: Computer Synthesis of Musical Style* (MIT Press, 2004).
60. Liu, Y. et al. Sora: a review on background, technology, limitations, and opportunities of large vision models. Preprint at https://arxiv.org/abs/2402.17177 (2024).
61. Zhang, R., Isola, P., Efros, A. A., Shechtman, E. & Wang, O. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 586–595 (IEEE, 2018).

# Methods

## User study

**Participant recruitment.** To recruit for the user study ('Interview study' section), game studios were opportunistically sampled from the Microsoft Founders Hub if: (1) they were funded start-ups; (2) they had published at least one game; and (3) they used, or were planning to use, AI tools. We made special efforts to be inclusive in our sampling by approaching studios from the Global South or led by people with disabilities. Eight studios participated (27 individuals), including four indie studios, one AAA studio and three teams of game accessibility developers. Most of the participants came from the USA and the UK, with further representations from Belgium, India and Cameroon. Most sessions had a mix of disciplinary representations, notably from engineering, design and art. There were three female participants in total, indicative of the underrepresentation of women in the industry in general.

The study was reviewed and approved by the Microsoft Research Ethics Review Program and informed consent was collected from all participants. Participants were thanked through invitations to two technical talks or a voucher for £40.

**Design probe.** A design probe[33], a well-established tool for imagining technical futures, was used for idea elicitation. It is a strategy for helping participants move beyond from what they already understand to unexpected ideas. They differ from user studies of prototypes in that the aim is not to systematically evaluate an idea or system but to surface potential opportunities for the future that will help shape a base technology. In this case, we were looking for high-level capabilities that AI models need to possess.

Specifically, we bring together a set of existing mechanisms that allows participants to manipulate AI-generated outcomes in various ways. Participants could: (1) use natural language to modify the generated scene; (2) alter an image through transforming it or drawing on it to direct generation; or (3) use example images or videos to convey a concept to the model. These are all existing interaction mechanisms for users to guide AI generation but the outcomes were scripted, that is, they did not rely on the capabilities of present AI models. To contextualize these ideas, we simulate the experience of creating a new game level (that is, the environment in which a player can interact and complete an objective), as shown in Extended Data Fig. 1a. The design probe was implemented in Unity.

**Session protocol.** Three to four participants from a single creative studio attended each session, which lasted 90 min and took place on a video call. Participants were prompted to think of AI as a new design material, a concept that would be familiar. To support this imaginative exercise, participants were then walked through a pre-specified journey through the design probe (Extended Data Fig. 1a on their own computer (see the 'Design probe' section); they were asked at points to reflect on how the highlighted capabilities might fit into their individual and/or collective creative processes. Team discussion was encouraged.

**Data analysis.** Sessions were recorded, transcribed and analysed thematically[18]. We first conducted an open coding of the transcripts to identify common themes, with a particular emphasis on how these tools might augment creative workflows and how participants imagined that they might support creative practice. See Extended Data Fig. 1b for themes and examples, including potential inputs and outputs, desired human–AI interaction design patterns and characteristics of creative practice that generative models need to support. A second round of coding took a higher-level view to identify suitable application areas for assistance in game ideation. Codes and examples were discussed within the team and iterated. We identified both opportunities to augment workflows (category 1), as well as user requirements for supporting creative practice (category 2). We present only the latter in this article.

Our study was initially designed to probe input and output modalities of generative AI systems for creatives (theme 6). However, our participants found it hard to engage with these specific questions when they were thinking about how generative AI fits within their creative practice more generally, because they saw more urgent blockers in the use of present generative AI systems in their creative practice. Consequently, we focus our analysis on this aspect of the interview sessions, highlighting some large gaps in model capabilities that need addressing to support creative ideation.

**Game development process.** Game development is a time-consuming process, with a single game typically taking two or more years (for indie games[62]) or five or more years (AAA games) to develop. Up to half of this period is spent in the concept and pre-production phases[62], which encompass ideation of the concept for the plot, characters, setting/world and mechanics. We use an example of how a small (indie) games studio created a new level for a new character to illustrate a typical game development process:

> The CEO came up with an idea of a character, a vampire, and conveyed the idea to the character artist. The character artist generated several concept sketches and iteratively tweaked the sketches with the CEO to arrive at a final design. Then the character artist spent several days sculpting a 3D model of the vampire character before passing it on to the animator for rigging. The finished rig was sent to the Head of Game to work with the programmer to define the character behavior. Taking approximately a month, the programmer made test environments, tried out different behavior patterns, and finally programmed the behavior. Once done, the finalized character design along with the behavior tree were passed on to the level designer, who started another round of iterations with the environment artist to craft a level prototype tailored to this new vampire character.
> – Chief Executive Office (CEO) of an indie studio

This example illustrates the numerous rounds of ideation that happen, as well as the complexity of working across several disciplines. Although this process varies with studio size and game genre, extensive iteration and subsequent coordination is needed to deliver a polished game by any game studio[63–65].

Connecting the complexity of the game development process to the contributions of this work, we note that our goal is not to demonstrate a specific tool or workflow that could be readily integrated into game development processes. Rather, our user study highlighted limitations of state-of-the-art generative AI models more broadly, that limit their adoption. We identify support for iterative practice and divergent thinking and derive three capabilities, consistency, diversity and persistency, that can meaningfully drive model development towards more fully supporting creative practice. Our evaluation results and case studies using WHAM and the WHAM Demonstrator show how this progress can enable iterative practice and divergent thinking, paving the way to future tool development and workflow innovation.

## Data

Data for WHAM training ('Model architecture and data' section) were provided through a partnership with Ninja Theory, who collected a large corpus of human gameplay data for their game Bleeding Edge. Data collection was covered by an end-user license agreement and our use of the data was governed by a data-sharing agreement with the game studio and approved by our institution's institutional review board. These data were recorded between September 2020 and October 2022. To minimize risk to human subjects, any personally identifiable information (Xbox user ID) was removed from the data. The resulting data were cleaned to remove errors and data from inactive players.

Image data were stored in MP4 format at 60 fps, alongside binary files containing the associated controller actions. A timecode extracted

from the game was stored for each frame, to ensure actions and frames remained in sync at training time.

We extracted two datasets, 7 Maps and Skygarden, from the data provided to us by Ninja Theory. The 7 Maps dataset comprised 60,986 matches, yielding approximately 500,000 individual player trajectories, totalling 27.89 TiB on disk. This amounted to more than 7 years of gameplay. After downsampling to 10 Hz, this equated to roughly 1.4B frames. This was then divided into training/validation/test sets by dividing the matches with an 80:10:10 split.

Our filtered Skygarden dataset used the same 80:10:10 split and 10-Hz downsampling but focused on just one map, yielding 66,709 individual player trajectories, or approximately 310M frames (about 1 year of game play).

### Modelling choices and hyperparameters

**Training.** We used PyTorch Lightning[66] and FSDP[67] for training.

**Encoder/decoder.** We trained two encoder/decoder models as follows.

15M–894M WHAMs: each image $\mathbf{o}_t$ is of shape 128 × 128 × 3, produced by resizing the frames of the original data from 300 × 180 × 3 (width, height and number of channels). No image augmentations are applied.

We train an approximately 60M-parameter VQGAN convolutional autoencoder using the code provided in ref. 51 to map images to a sequence of $d_z = 256$ discrete tokens with a vocabulary of $V_O = 4{,}096$. The encoder/decoder is trained first with a reconstruction loss and perceptual loss[61] and then further trained using a GAN loss.

1.6B WHAM: each image $\mathbf{o}_t$ is kept at the native shape of the data, 300 × 180 × 3. No image augmentations are applied.

We train an approximately 300M ViT-VQGAN[68] to map images to a sequence of $d_z = 540$ discrete tokens with a vocabulary of $V_O = 16{,}384$. The encoder/decoder is trained first with an $L_1$ reconstruction error, perceptual loss[61] and a maximum pixel loss[69]. It is then also trained with a GAN loss.

**Transformer.** We use a causal transformer for next-token prediction, with a cross-entropy loss. Specifically, we use a modified nanoGPT[70] implementation of GPT-2 (ref. 53). Configurations for all models used in the paper are given in Extended Data Fig. 2c.

894M WHAM: the context length is 2,720 tokens, or equivalently 1 s or ten frames. Each batch contains 2M tokens. The model is trained for 170k updates.

We use AdamW[71] with a constant learning rate of 0.00036 preceded by a linear warm-up. We set $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

1.6B WHAM: the context length is 5,560 tokens, or equivalently 1 s or ten frames. Each batch contains 2.5M tokens. We train for 200k updates.

We use AdamW with a cosine annealed learning rate, which peaks at a max value of 0.0008 and is annealed to a final value of 0.00008 over training, preceded by a linear warm-up over the first 5,000 steps. We set $\beta_1 = 0.9$, $\beta_2 = 0.95$ and use a weight decay of 0.1.

### Model scale

To investigate the scalability of WHAM with model size, amount of data and compute, we conducted analysis similar to that performed on large language models[72–74]. We trained several configurations of WHAM at varying sizes (measured by the number of parameters in the model; see Extended Data Fig. 2c). Extended Data Fig. 2a shows the training curves for these runs and illustrates how training losses improve with model, data and compute. This analysis offers us assurance that the performance of the model reliably improves with compute, as well as providing a means to understand what the optimal model size would be. Using this approach, we were able to accurately predict the final loss of the larger 894M model, based on extrapolations of models in the range 15M to 206M.

This analysis also informed the configuration of the 1.6B WHAM aimed at achieving the lowest possible loss given our compute budget of around $1 \times 10^{22}$ FLOPS. The initial exploration of scaling laws presented here led to a deeper investigation of scaling laws for world and behaviour models[75].

Extended Data Fig. 2b shows a strong correlation ($r = 0.77$, with sample Pearson's correlation coefficient calculated using numpy's corrcoef function[76]) between FVD and the training loss, providing a strong justification for optimizing towards a lower loss (similar observations relating model performance to loss have also been observed in the language domain[73]).

### Model evaluation

This section presents further detail on metrics, as well as further analyses. The 'Consistency' section provides details on the FVD calculation used for the consistency analysis and provides justification by correlating it with human judgement. The 'Diversity' section details the Wasserstein calculation and provides further qualitative results evidencing the diverse generations of WHAM. The 'Persistency' section details the editing and annotation process and provides further examples and insights into the persistency results.

**Consistency.** FVD was calculated by comparing two sets of sequences. The first set is composed of 'ground truth' sequences: 1,024 gameplay videos generated by human players at the resolution of the data of 300 × 180. Each video is 10 s long and was not used during training. For each video in this set, the initial ten frames and the entire action sequence were used as prompts for generating the second set using WHAM. The second set is composed of 10-s videos generated by the WHAM model at resolution 128 × 128 for the 15M to 894M WHAMs and 300 × 180 for the 1.6B WHAM given the prompts.

To ensure that FVD is an appropriate metric to gauge the performance of the model, we conduct a more detailed manual analysis in Extended Data Fig. 3. For this, we use the 894M WHAM. Paired coding was used to mark frames as consistent or inconsistent in 'structure', 'actions' or 'interactions'. The plots are averaged over the per-frame paired consensus of two human annotators, with 1 indicating that all frames are consistent and −1 meaning that all frames are inconsistent. In this case, consistency was framed by three questions. (1) Structure: does the level structure (including geometry and texture of every element in the environment) stay consistent? (2) Actions: does the on-screen character respond to the given actions (for example, when the player moves, jumps or launches an attack)? (3) Interaction: does the character react to the elements in the environment (for example, ascends stairs with the appropriate animation, does not move through solid structures such as walls and the floor)?

Figure 3a shows that FVD for WHAM improves (that is, decreases) with increasing FLOPS (corresponding to later checkpoints). Extended Data Fig. 3 corroborates that human perception of consistency matches our quantitative results. Our manual analysis of the consistency of structure, actions and interaction shows increasing consistency with increased training and lower FVD scores. Hence, we argue that consistency with the ground truth as measured by FVD indicates that game mechanics are modelled correctly and consistently over time.

**Diversity.** To compute the Wasserstein distance in Fig. 4a, we use two sets of inputs: (1) 1,024 human action sequences from recorded gameplay (the same set as used in the FVD consistency analysis) and (2) the 1,024 predicted action sequences generated by WHAM when conditioned on the starting frames that match each sequence in (1).

For each of the 1,024 videos, we generate 100 time steps, both images and actions, using the initial ten frames and actions as prompts. Thus, for the later time steps, the model is conditioning purely on generated frames, which will affect the distributions of sampled actions. The same set of gameplay videos are used as for the FVD calculation in the 'Consistency' section.

**Wasserstein distance:** let $p$ and $q$ be probability distributions on $\mathcal{X}$ and $c$ be a cost function $c : \mathcal{X} \times \mathcal{X} \to [0, \infty)$. Further, let $\Pi$ be the space

of all joint probability distributions with marginals $p$ and $q$. The Wasserstein distance[77] is defined as $\mathcal{W}_c(p,q) \coloneqq \min_{\pi\in\Pi}\int_{\mathcal{X}\times\mathcal{X}} c(x,y)\mathrm{d}\pi(x,y)$. In this paper, $\mathcal{X}\subset\mathbb{R}^{16}$, because we embed each action as a 16-dimensional vector. Each of the 12 action buttons are embedded as 0 or 1 and the $x$ and $y$ axes of both sticks are embedded in $[-1, 1]$ by using the value of the corresponding discretized bin. The cost function is the standard $L_2$.

To calculate the Wasserstein distance, we first sub-sample 10,000 actions from the total set of 102,400 actions and use the emd2 function of the Python Optimal Transport library[78] to calculate the value. We repeat this ten times and report means and one standard deviation.

As well as calculating the Wasserstein distance between the marginal distributions, we also perform more qualitative checks on the generations of the models. We check for both behavioural diversity, in which the player character exhibits a range of behaviours, such as how and where they navigate to, as well as visual diversity, such as the visual modifications to the character's hoverboard. Our qualitative results show possible futures generated from the same starting frames by the 1.6B WHAM (Extended Data Fig. 5). The results show that WHAM can generate a wide range of human behaviours and character appearances. Extended Data Fig. 4 also shows examples from 2-min-long generations by the 1.6B WHAM, demonstrating long-term consistency as well as diversity in the generated outcomes. The initial exploration of behavioural diversity presented here led to a deeper investigation on how to control for more desirable behaviour of similar models in post-training[79].

**Persistency.** Our persistency metric captures how frequently WHAM retains user edits in its generated video frames, in which these edits are objects or characters that have been added to new locations in input frames. We study the effect of the number of input frames by evaluating the persistency of WHAM under a partially filled context window with only one or five input frames, and under a full context window of ten input frames.

To calculate persistency: (1) we prompt WHAM with input frames in which an object or characters has been manually inserted into the frames, and then we generate videos from WHAM; (2) we ask human annotators to categorize the extent to which objects or characters are persisted in the generated frames. Each of these two steps are detailed below.

**Video generation.** Overall, we generate 600 videos from WHAM under the following conditions:

- For Powercell and characters edits, we generate 480 total videos: 8 sequences of input frames × 2 types of edit (Powercell and characters) × 3 input lengths (1, 5, 10) × 10 sampled videos.
- For the Vertical Jumppad edits, we generate 120 videos: 4 sequences of input frames × 1 type of edit (Vertical Jumppad) × 3 input lengths (1, 5, 10) × 10 sampled videos.

To select the sequences of input frames (visualized in Extended Data Fig. 7a), we sampled sequences of ten contiguous frames from a held-out testing set and only kept sequences that satisfied the following conditions. (1) The frames should reflect a variety of locations and characters in the game. (2) They should depend minimally on world modelling capabilities outside persisting an added object or character. This means we picked frames with simpler dynamics, that is, with minimal camera movement, character movement and abilities, special effects and interactions with other characters. (3) Finally, the kept frames are not meant to be particularly adversarial or atypical, meaning that the main character should be visible, on the ground (that is, not in mid-air) and surrounded by an environment that has space for new objects or characters to be added (for example, the main character should not be directly facing a wall).

Next, two of the authors edited a Powercell (in-game object), character (ally or opponent) and a Vertical Jumppad (in-game map element) into each of the selected input sequences (see Fig. 5 for examples of edits). They edited independently to account for some variability in how different users may approach the editing process. Their edits also aimed to place objects or characters in new but plausible locations in the input frames, for example, objects that are usually on the ground should not be placed in the sky.

Finally, we took all of the edited input sequences and created a 1-length, 5-length and 10-length version of each. The 1-length version includes only the last (that is, latest time step) frame, the 5-length version includes only the last five frames and the 10-length version includes all frames. Thus, across the different input lengths, the last frame received by WHAM would be the same edited image and all generated videos would start from the same point. For each edited and length-adjusted sequence of input frames, we generated ten videos from WHAM to obtain some coverage over the stochastic behaviour of the model. WHAM only needed to generate the frames of these videos, whereas actions were given as no-ops. We chose no-ops to minimize the need for world modelling capabilities beyond persistency and to minimize movements that would put the edited element out of the frame (for example, we discouraged the main character from turning away from the edited element, which would make it harder to judge the disappearance of the element as a lack of persistency or as a natural transition).

**Human annotation.** The 600 generated videos were annotated by seven of the authors. These annotators were separate from the authors who edited the input frames and from the authors who generated videos from WHAM. They were blinded to whether videos came from the 1-length, 5-length or 10-length conditions. For each generated video, the annotators saw the last input frame (common to all input length conditions), the object or character that had been edited into the frame and the frames of the video. They independently judged which category a video fell into:

- Persisted: the edited object or character is recognizable for the first ten video frames (1 s).
- Persisted until out of frame (for edited characters only): the edited character is recognizable and moves out of the frame/view in a plausible way (for example, running out of view) within the first ten video frames.
- Unusable: the video is visually distorted or showing implausible continuations within the first ten video frames.
- Did not persist: the video does not fall into any of the above categories.

See Fig. 5 for examples of videos annotated as 'Persisted' and Extended Data Fig. 6 for the remaining categories.

Half (300) of the generated videos were assigned to two random and distinct annotators so that we can evaluate agreement between annotators: there was a 90% agreement between annotators (270/300). Of the 30 disagreements, 26 occurred for the edited character condition, with many arising from noisy labelling owing to the character leaving the frame. For each pair of annotations of the same video, we selected the stricter annotation (that is, 'Unusable' over 'Did not persist' over 'Persisted until out of frame' over 'Persisted') such that we have one annotation per video for the analyses below. We note that, out of the 600 de-duplicated annotations, only seven selected the 'Unusable' category.

**Analysis.** We measure persistency by the percentage of generated videos falling into the 'Persisted' or 'Persisted until out of frame' categories, as opposed to the 'Unusable' or 'Did not persist' categories. In Table 1, we focus on differences in persistency across the 1 and 5 input lengths, for which the persistency for each input length aggregates across variations in the input sequences (that is, different locations and main characters) and is separated for the different types of edit ('Powercell', 'character' and 'Vertical Jumppad'). In Extended Data Fig. 7b, we also compare with the persistency of the 10 input length condition.

For each of the three types of edit, persistency increases substantially from 1 to 5 input lengths but not from 5 to 10 input lengths. Significance

is computed with six one-sided binomial tests at an overall significance level of 0.05, for which each individual test uses a Bonferroni-corrected significance level of 0.008. The six tests compare 1 with 5 input lengths for each of the three edits and 5 with 10 input lengths for each of the three edits.

In Extended Data Fig. 7b, we also show how persistency changes across the different input sequences (each with a different location and character) and types of edit (Powercell, character or Vertical Jump-pad). Notably, the rate of persistency was much lower for some starting locations (of the input sequences). In Extended Data Fig. 7c, we share three examples to illustrate that lower rates of persistency are probably because of the small size of an edit, lack of contrast with the background or unusual location of an edit.

### Inclusion and ethics statement

The gaming industry is heavily centred in the Global North and is dominated by able-bodied men. We made concerted efforts to recruit teams led by those from other perspectives for the user study. We were successful in including a game studio from the Global South as well as people with disabilities. Data used in training the model were collected from players globally. The user study received ethics approval from the Microsoft Ethics Review Program. All participants have consented to participation in the user study and use of their anonymized data in research publications. Data used for training the model were covered by an end-user license agreement to which players agreed when logging in to play the game for the first time. Our use of the recorded human gameplay data for this specific research was governed by a data-sharing agreement with the game studio. To minimize the risk to human subjects, player data were anonymized and any personally identifiable information was removed when extracting the data used for this article. We have complied with all relevant ethical regulations.

### Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

## Data availability

The full dataset used for training the model reported in this paper is owned by Ninja Theory and licensed to us for research purposes only. An impact assessment raised several concerns about sharing the data publicly of a game that is still commercially available and may have active players represented. However, we can provide a sample dataset to enable (re)running the evaluation or seeding the model with game scenes to explore ideation and other creative uses. This comprises the 1,024 trajectories of our Skygarden test dataset. This dataset contains anonymized rendered gameplay sequences of 100 steps (image, controller action pairs). Access to the data is provided under the same terms as to the code, as detailed in the 'Code availability' section. Source data are provided with this paper.

## Code availability

We provide model weights, sample data and the WHAM Demonstrator to support interpreting, validating and extending our results. We will provide public access to the following code (under a Microsoft Research license) and related artefacts on publication of this manuscript: compiled executable for the WHAM Demonstrator, allowing readers to explore creative uses and ideation workflows enabled by the WHAM model; WHAM weights for two WHAM instances: the 1.6B-parameter model trained for 200k updates and a smaller 200M WHAM instance that supports faster iterations; a sample of 1,024 test trajectories as detailed above. Model weights and software will be made publicly available on publication of this paper at https://huggingface.co/microsoft/wham.


62. Fullerton, T. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games* (CRC Press, 2008).
63. Macklin, C. & Sharp, J. *Games, Design and Play: A Detailed Approach to Iterative Game Design* (Addison-Wesley Professional, 2016).
64. Jacob, M., Devlin, S. & Hofmann, K. "It's unwieldy and it takes a lot of time" – Challenges and opportunities for creating agents in commercial games. In *Proc. 16th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 88–94 (AAAI, 2020).
65. Karlsson, T., Brusk, J. & Engström, H. Level design processes and challenges: a cross section of game development. *Games Cult.* **18**, 821–849 (2023).
66. Falcon, W. & The PyTorch Lightning Team. PyTorch Lightning. *GitHub* https://github.com/Lightning-AI/lightning (2019).
67. Zhao, Y. et al. PyTorch FSDP: experiences on scaling fully sharded data parallel. In *Proc. VLDB Endowment* (eds Koutrika, G. & Yang, J.) 3848–3860 (VLDB Endowment, 2023).
68. Yu, J. et al. Vector-quantized image modeling with improved VQGAN. In *Proc. 10th International Conference on Learning Representations* (ICLR, 2022).
69. Anand, A. et al. Procedural generalization by planning with self-supervised world models. In *Proc. 10th International Conference on Learning Representations* (ICLR, 2022).
70. Karpathy, A. nanoGPT. *GitHub* https://github.com/karpathy/nanoGPT (2022).
71. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. In *Proc. 7th International Conference on Learning Representations* (ICLR, 2019).
72. Kaplan, J. et al. Scaling laws for neural language models. Preprint at https://arxiv.org/abs/2001.08361 (2020).
73. Hoffmann, J. et al. An empirical analysis of compute-optimal large language model training. In *Proc. Advances in Neural Information Processing Systems 35 (NeurIPS 2022)* (eds Koyejo, S. et al.) 30016–30030 (NeurIPS, 2022).
74. Pearce, T. & Song, J. Reconciling Kaplan and Chinchilla scaling laws. Preprint at https://arxiv.org/abs/2406.12907 (2024).
75. Pearce, T. et al. Scaling laws for pre-training agents and world models. Preprint at https://arxiv.org/abs/2411.04434 (2024).
76. Harris, C. R. et al. Array programming with NumPy. *Nature* **585**, 357–362 (2020).
77. Villani, C. et al. *Optimal Transport: Old and New* (Springer, 2009).
78. Flamary, R. et al. POT: Python Optimal Transport. *J. Mach. Learn. Res.* **22**, 1–8 (2021).
79. Jelley, A., Cao, Y., Bignell, D., Devlin, S. & Rashid, T. Aligning agents like large language models. Preprint at https://arxiv.org/abs/2406.04208 (2024).



**Acknowledgements** We thank Ninja Theory for the collaboration on unlocking the Bleeding Edge data for this research, A. Shaw for technical support and N. Paleyes for support in onboarding an earlier version of the data. We thank A. Slowey, O. Losinets and the whole Microsoft Research Grand Central Resources (GCR) team for infrastructure support. We are grateful to all participants who joined our user study. We thank A. Jelley, E. Alonso, E. Zuniga, G. Davidson, G. Leroy, H. van Seijen, I. Momennejad, J. Rzepecki, K. Khetarpal, L. Schäfer, M. Hendriksen, M. Carroll, M. Sun, R. E. Turner, S. Milani and S. Sharma for ideas and discussions. We thank S. Parisot for proofreading and feedback on this manuscript. We are grateful to D. Burger and the Microsoft Research Redmond GPU council for GPU resources. We thank D. Tan, H. Zhang, J. Gehrke and the many other colleagues who provided encouragement and support.


**Author contributions** Primary authors listed alphabetically by first name: A.K., D.B., L.Y.W., M.G., R.G., S.V.M., S.Z.T., T.R., T.P., Y.C. Secondary authors listed alphabetically by first name: A.L., C.J., G.C., G.G., M.T., S.I., T.G., U.A. Senior authors: R.W.W., S.D., C.M., K.H. M.G. designed and implemented the concept prototype, with contributions from A.K., S.D., S.Z.T. and L.Y.W.; D.B. generated and ensured the quality of the data, with U.A. supporting the data analysis; S.Z.T. ensured the approval from the institutional review board and G.C. provided support on the dataset; M.G. designed and implemented the design probe for the user study. L.Y.W. conducted the user study (conceptualization, design, execution and analysis), with C.M. supporting. A.K. and R.G. worked on the game environment, with the support of D.B. and G.C.; A.K., S.D., T.R. and T.P. designed the model; T.P. prototyped the model; U.A. and A.L. assisted with model testing; A.K., R.G., S.V.M. and T.R. implemented the training infrastructure; T.R. trained the image encoders. A.K., T.P. and Y.C. trained the lower-resolution models and T.R. trained the higher-resolution model. T.P. conducted the scaling-law analysis; A.K., G.G., S.D., T.R., T.G. and T.P. evaluated the model. C.J., T.R. and K.H. designed and implemented the persistency study; L.Y.W., T.P., S.Z.T., Y.C., A.L., M.T. and S.I. annotated for the persistency study. M.G. contributed to the example-based analysis. K.H. led the project, with S.D. as the co-lead and S.Z.T. providing project management. C.M. conceptualized the manuscript and wrote it with S.D., T.R., L.Y.W., T.P. and K.H., with all authors contributing to revision. A.K. is now at Meta. C.J., G.G., M.T., S.I. and T.G. completed this research while at Microsoft Research.

**Competing interests** A.K., C.M., K.H., L.Y.W., M.G., R.W.W., S.D., S.Z.T., T.G., T.P. and T.R. are inventors of one or more pending patent application(s) in the name of Microsoft Technology Licensing LLC related to machine learning for predictive generative content as related to this paper. The other authors declare no competing interests.

**Additional information**


**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s41586-025-08600-3.

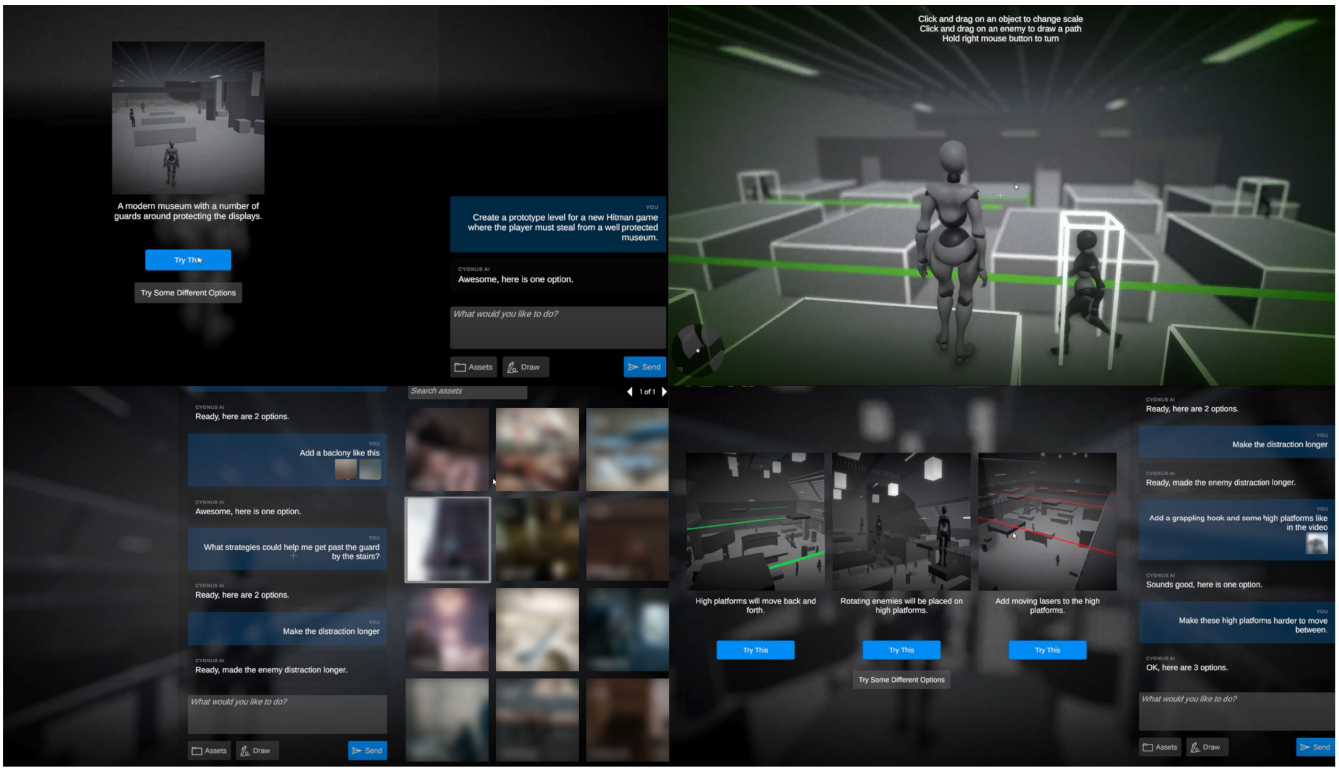


**Correspondence and requests for materials** should be addressed to Katja Hofmann.



**Peer review information** *Nature* thanks Emmanuel Guardiola, Amy Hoover and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

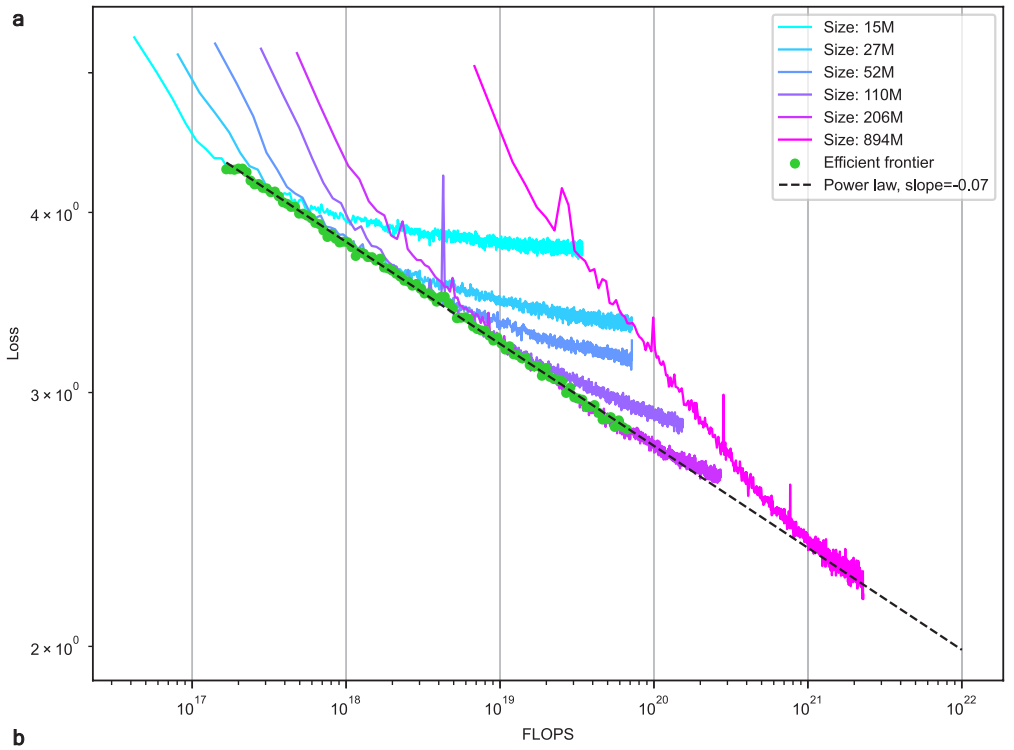
**a**




**b**

| Theme | Example quote |
|---|---|
| | Category 1: Augmenting workflows |
| **1** Assist with concept formation | Seek inspiration: "As game developers, sometimes we want to evoke a certain feeling or a certain atmosphere, but we're not even sure where to start. And again, that's just been a hole that's been very nicely filled with AI." |
| **2** Assist with prototyping | Scope technical requirements: "Through very rapid prototyping, we'd be able to discover all of the things that would emerge throughout the course of development as things we'd have to go and change or address." |
| **3** Facilitate multidisciplinary collaboration | Encourage contribution from all disciplines: "Everybody can try out stuff that could bring something to the debate. Sometimes the issue is that people rely on someone else to (build prototypes and) try out stuff, that sometimes that costs time and sometimes it is not the exact same idea that the (person has in mind)." |
| | Category 2: User requirements for supporting creative practice |
| **4** The need to help creators build a robust mental model to enable effective usage | "The one thing that's really frustrating about generating images with AI is, if it doesn't work, you have to go back to the starting point, redo your prompt, redo another prompt, try another prompt. It is not a fast process, and it doesn't always have a clear path to the outcome that you're looking for because it's so unpredictable." |
| **5** The need to support iterative tweaking | "It's hard to know what the right output is until we see it, and that, I think is one of the tough things. It takes just a lot of finessing it and playing with it." |
| **6** The need to support mixed modality of inputs to allow creators express their ideas in their preferred medium | "I think the mixture of being able to prompt with video and with text would be really beneficial, because sometimes, if you aren't a programmer, it's really difficult to actually work out exactly what you want." |
| **7** The need to provide multiple options to help creators explore and refine their ideas | "It could be nice to save several states of the project, so you can get back to something that you tried out at the very first and get back to this point, or maybe mixed it with the current state and do stuff like iteration mixing" |

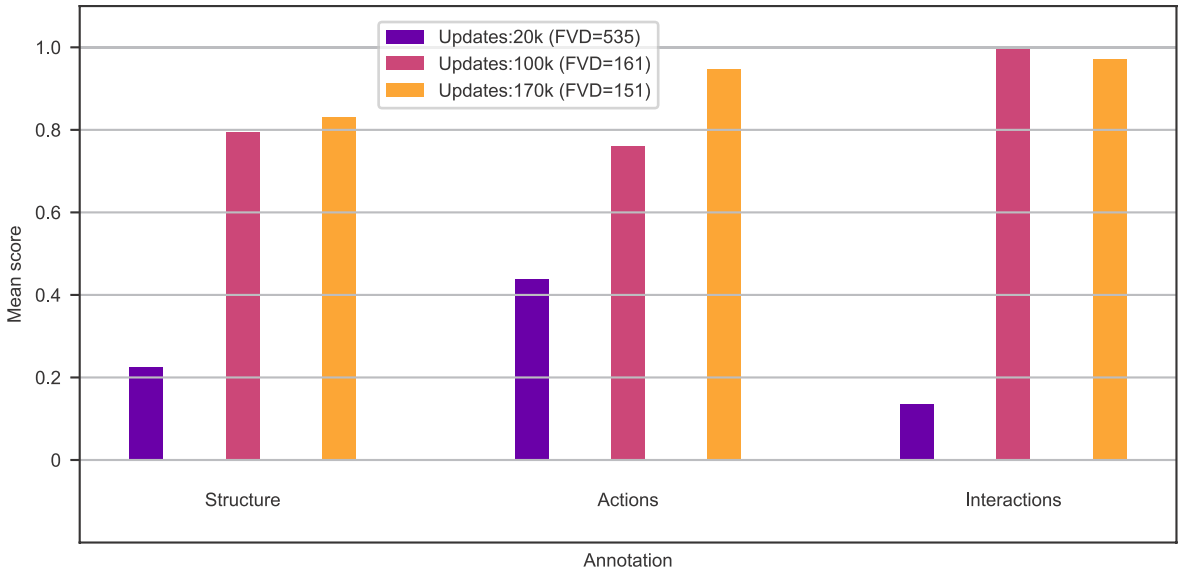
**Extended Data Fig. 1 | Interview study details. a**, Images taken from the design probe used in the semistructured interviews with creative teams. Participants could use natural language to generate a scene (top left), visually tweak the behaviour of non-player characters (also known as NPCs) and the environment by drawing directly onto the frame (top right), use image or video references to influence scene generation and choose from several generations. **b**, Summary of themes identified in the interview study. The focus of reporting of this article is on the themes in category 2.




c

| Number of parameters | Num. Layers | Num. Heads | Model dim |
|---|---|---|---|
| 15M | 4 | 4 | 512 |
| 27M | 8 | 8 | 512 |
| 52M | 10 | 10 | 640 |
| 110M | 15 | 12 | 768 |
| 206M | 16 | 16 | 1024 |
| 894M | 23 | 14 | 1792 |
| 1567M | 24 | 18 | 2304 |

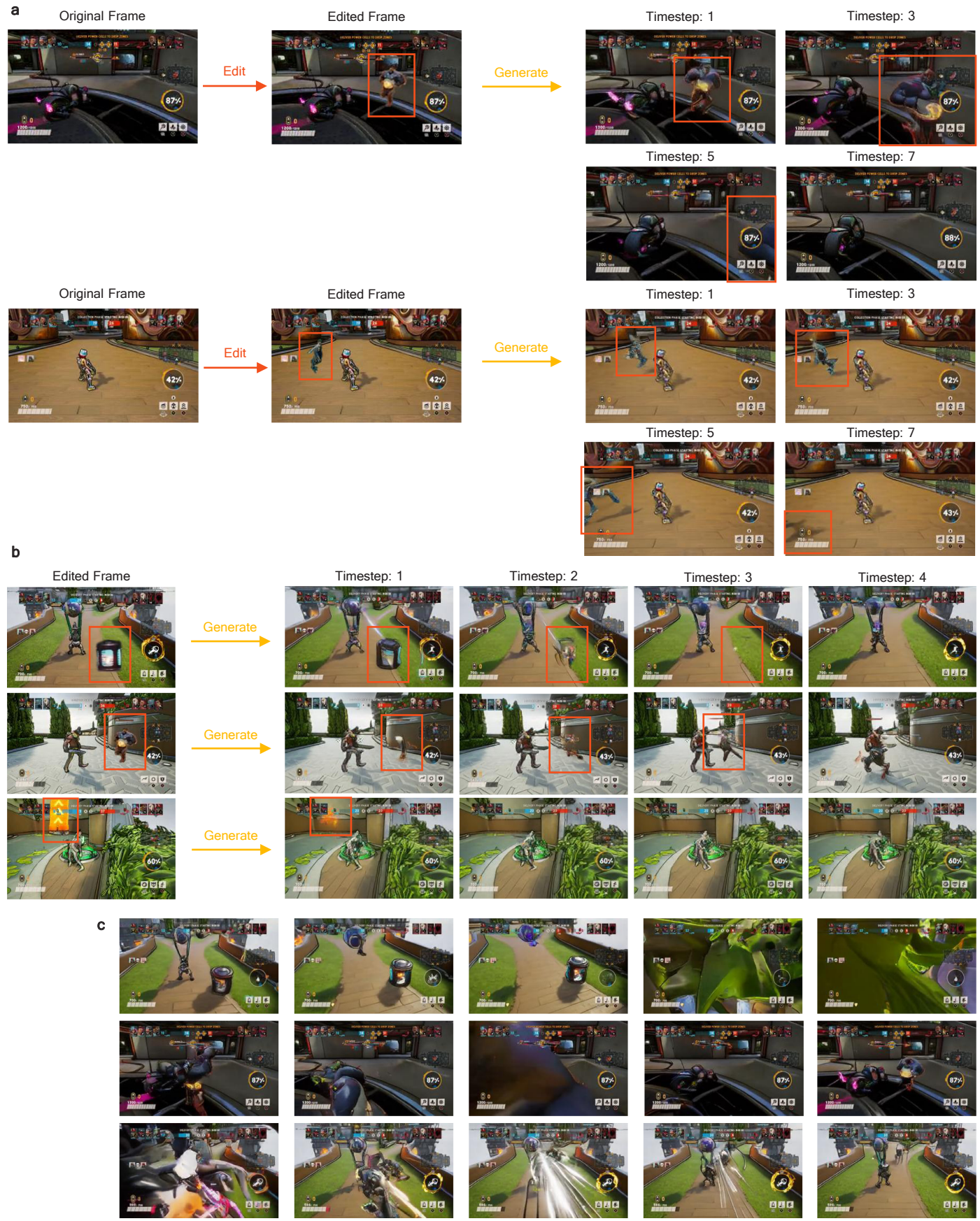
**Extended Data Fig. 2 | Model sizes. a**, Effect of model size and training compute (FLOPS) on the cross-entropy training loss (lower is better). Highlighted in green are models on the 'efficient frontier'—models that minimize training loss for a given compute budget. Note that larger model sizes become efficient at larger compute budgets. **b**, FVD for a range of model sizes each at various numbers of updates, plotted in terms of the training loss. Note that a lower loss usually leads to a lower FVD. **c**, Transformer configurations.



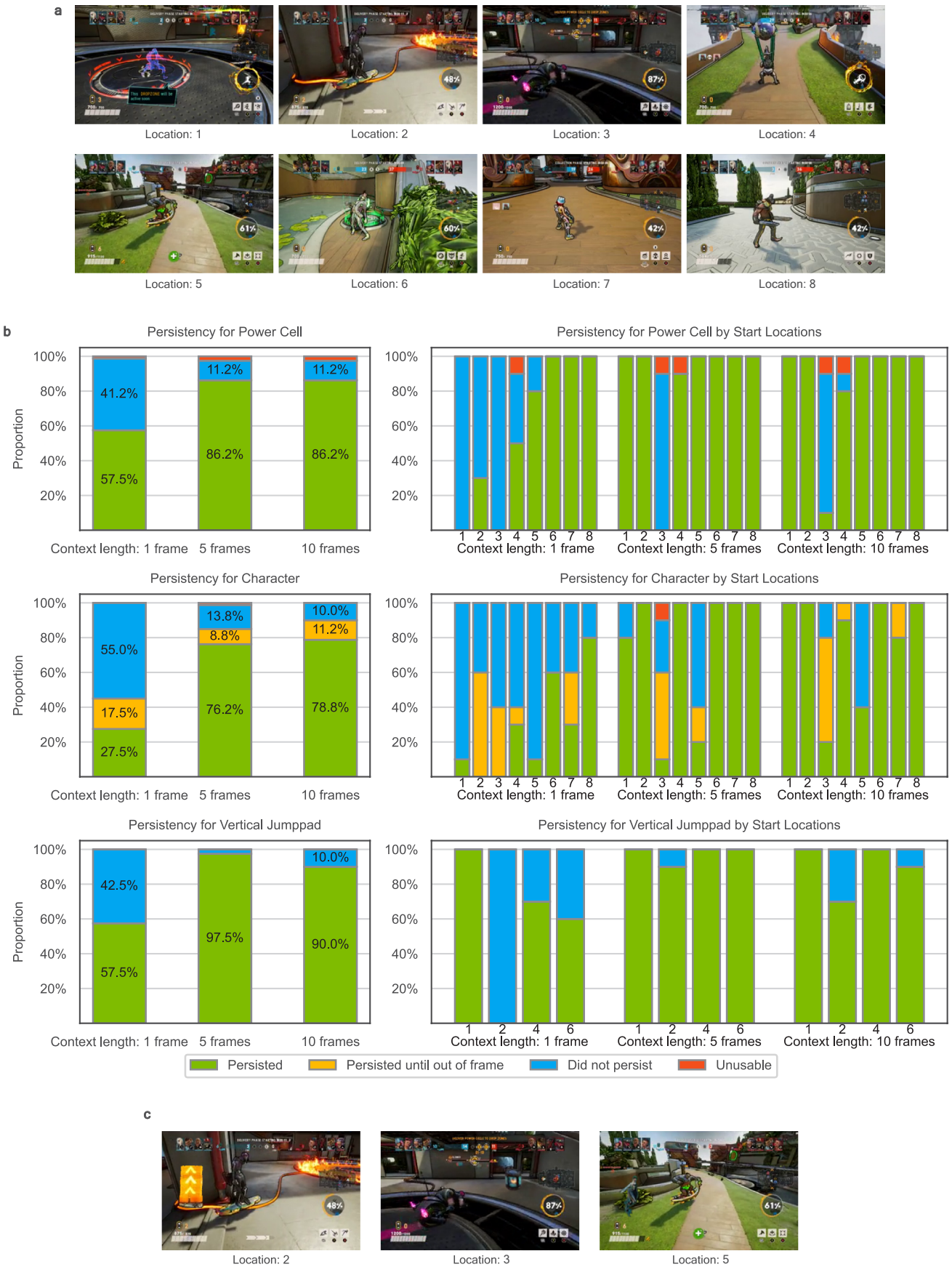

**Extended Data Fig. 3 | Human-perceived consistency.** Detailed analysis of human-perceived consistency for three generated sequences (300 generated frames) across three checkpoints of the 894M WHAM. Increasing training updates and lower FVD scores are typically associated with improved human-perceived consistency.


**Extended Data Fig. 4 | Consistency and diversity in a 2-min-long generated sequence.** Set of 16 examples (one example per row, showing keyframes at specific time steps) of 2-min-long gameplay sequences, generated using the 1.6B WHAM and starting from the same initial sequence, shown at the top (**a**). Generations remain consistent throughout and the examples show the diversity of generated sequences (**b**).


**Extended Data Fig. 5 | Examples of behavioural and visual diversity.** All sequences are generated using the 1.6B WHAM conditioned on ten real frames. **a**, Even-numbered frames. **b**, Five examples of behavioural diversity: camera movement, loitering near the spawn location and different paths navigating to the middle Jumppad. **c**, Five examples of visual diversity in the player's hoverboard.




**Extended Data Fig. 6 | Successful and unsuccessful persistency examples.** **a**, Examples of inserted characters persisting until they are out of frame (annotated as 'Persisted until out of frame'). **b**, Examples of modifications not persisting (annotated as 'Did not persist'). **c**, Examples of unusable generations. Every other frame is shown (annotated as 'Unusable').




**Extended Data Fig. 7** | See next page for caption.

**Extended Data Fig. 7 | Detailed persistency results by item type and starting location. a**, Starting locations 1–8, numbered from left to right and top to bottom and showing the last frame for each of the eight input sequences before they are edited. **b**, Detailed persistency results. Each stacked bar visualizes the percentage of generated videos that were annotated as 'Persisted', 'Persisted until out of frame', 'Did not persist' or 'Unusable'. On the left, we group the bars by the input/context length and the edited element. On the right, we also group by the eight (four for Vertical Jumppad) starting locations. Note that modifications in some starting locations persist less than in others. **c**, Examples of the more challenging start locations. We believe that this is caused by the quality and prominence of a modification. Location 2: example of a Vertical Jumppad edit that is not as persistent as in the other locations. The lower persistency is probably because of how the edit appears in an unusual place on the map and is substantially smaller than it would be in the actual game. Location 3: example of a Powercell edit that is not as persistent as in other starting locations. The inserted Powercell tends to disappear or turn into another object/character, which may be because of its small size. Location 5: example of a character edit that is not as persistent as in the other locations. The edited-in character easily disappears, probably because it is not particularly prominent or contrasting with the background.

nature portfolio

Corresponding author(s): Katja Hofmann (katja.hofmann@microsoft.com)

Last updated by author(s): Dec 3, 2024

# Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

## Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

| n/a | Confirmed | |
|---|---|---|
| ☐ | ☒ | The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement |
| ☐ | ☒ | A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly |
| ☐ | ☒ | The statistical test(s) used AND whether they are one- or two-sided<br>*Only common tests should be described solely by name; describe more complex techniques in the Methods section.* |
| ☒ | ☐ | A description of all covariates tested |
| ☐ | ☒ | A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons |
| ☒ | ☐ | A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals) |
| ☐ | ☒ | For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted<br>*Give P values as exact values whenever suitable.* |
| ☒ | ☐ | For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings |
| ☒ | ☐ | For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes |
| ☐ | ☒ | Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated |

*Our web collection on statistics for biologists contains articles on many of the points above.*

## Software and code

Policy information about availability of computer code

| | |
|---|---|
| Data collection | We provide a Data and Code Availability statement in the manuscript. Sample data and software are available for reviewing and access details have been shared with the editor. Data and code will be publicly released under a Microsoft Research license upon publication of this manuscript. |
| Data analysis | We provide a Data and Code Availability statement in the manuscript. Sample data and software are available for reviewing and access details have been shared with the editor. Data and code will be publicly released under a Microsoft Research license upon publication of this manuscript. |

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

## Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

We provide a Data and Code Availability statement in the manuscript. Sample data and software are available for reviewing and access details have been shared with the editor. Data and code will be publicly released under a Microsoft Research license upon publication of this manuscript.

## Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

| | |
|---|---|
| Reporting on sex and gender | The game development environment is predominantly male. Despite attempts to recruit female-led studios, we were unable to find one. We considered diversity (e.g. disabled creatives) in other ways to help gain multiple perspectives in our findings. |
| Reporting on race, ethnicity, or other socially relevant groupings | We do not report socially relevant groupings, but do state that we attempt to include less "typical" games studios (e.g. disability-led) as a mechanism to diversify results. |
| Population characteristics | See below. |
| Recruitment | See below. |
| Ethics oversight | Microsoft Ethics Review Programme |

Note that full information on the approval of the study protocol must also be provided in the manuscript.

# Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☐ Life sciences ☒ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

# Behavioural & social sciences study design

All studies must disclose on these points even when the disclosure is negative.

| | |
|---|---|
| Study description | Qualitative User Study |
| Research sample | 8 creative teams (27 individuals) |
| Sampling strategy | Participants were recruited opportunistically from Microsoft for Startups Founders Hub if: 1) they were funded startups; 2) had published at least one game; and 3) used, or were planning to use, AI tools (i.e. Bing Image Creator). Special efforts were made to approach studios from the Global South, led by people with disabilities, or led by women to ensure that we might get varied perspectives of need. We were unable to a recruit a studio led by women. |
| Data collection | Each interview session took place with three or four creatives from the same studio and lasted about an hour. The participants interacted with a design probe that simulated the experience of creating a game level with current and envisioned capabilities of generative AI models. Participants were walked through the probe on their own computer; they were asked at points to reflect upon how the highlighted capabilities might fit in their individual and/or collective creative processes. Team discussion was encouraged. Interviews took place on Teams and were recorded and transcribed automatically. |
| Timing | Data was collected between 28 July 2023 and 28 August 2023. |
| Data exclusions | No data was excluded. |
| Non-participation | No participants dropped out. |
| Randomization | N/A |

# Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

## Materials & experimental systems

| n/a | Involved in the study |
|---|---|
| ☒ | ☐ Antibodies |
| ☒ | ☐ Eukaryotic cell lines |
| ☒ | ☐ Palaeontology and archaeology |
| ☒ | ☐ Animals and other organisms |
| ☒ | ☐ Clinical data |
| ☒ | ☐ Dual use research of concern |
| ☒ | ☐ Plants |

## Methods

| n/a | Involved in the study |
|---|---|
| ☒ | ☐ ChIP-seq |
| ☒ | ☐ Flow cytometry |
| ☒ | ☐ MRI-based neuroimaging |

## Plants

Seed stocks

*Report on the source of all seed stocks or other plant material used. If applicable, state the seed stock centre and catalogue number. If plant specimens were collected from the field, describe the collection location, date and sampling procedures.*

Novel plant genotypes

*Describe the methods by which all novel plant genotypes were produced. This includes those generated by transgenic approaches, gene editing, chemical/radiation-based mutagenesis and hybridization. For transgenic lines, describe the transformation method, the number of independent lines analyzed and the generation upon which experiments were performed. For gene-edited lines, describe the editor used, the endogenous sequence targeted for editing, the targeting guide RNA sequence (if applicable) and how the editor was applied.*

Authentication

*Describe any authentication procedures for each seed stock used or novel genotype generated. Describe any experiments used to assess the effect of a mutation and, where applicable, how potential secondary effects (e.g. second site T-DNA insertions, mosiacism, off-target gene editing) were examined.*