

# VIDEOGAMEBUNNY: Towards vision assistants for video games

Mohammad Reza Taesiri  
 University of Alberta  
 mtaesiri@gmail.com

Cor-Paul Bezemer  
 University of Alberta  
 bezemer@ualberta.ca

## Abstract

*Large multimodal models (LMMs) hold substantial promise across various domains, from personal assistance in daily tasks to sophisticated applications like medical diagnostics. However, their capabilities have limitations in the video game domain, such as challenges with scene understanding, hallucinations, and inaccurate descriptions of video game content, especially in open-source models. This paper describes the development of VIDEOGAMEBUNNY, a LLaVA-style model based on Bunny, specifically tailored for understanding images from video games. We release intermediate checkpoints, training logs, and an extensive dataset comprising 185,259 video game images from 413 titles, along with 389,565 image-instruction pairs that include image captions, question-answer pairs, and a JSON representation of 16 elements of 136,974 images. Our experiments show that our high quality game-related data has the potential to make a relatively small model outperform the much larger state-of-the-art model LLaVa-1.6-34b (which has more than 4x the number of parameters). Our study paves the way for future research in video game understanding on tasks such as playing, commentary, and debugging. Code and data are available at: <https://videogamebunny.github.io/>*

## 1. Introduction

The video game industry is projected to be valued at \$321 billion by 2026 [12, 38] and continues to contribute more to the global economy. Generative Artificial Intelligence (GenAI) [41] is rapidly spreading across various sectors, disrupting the ways in which many traditional tasks are executed [6, 11]. In the realm of gaming, GenAI has the potential to enhance many aspects, such as providing a better in-game experience by generating more realistic conversations with non-playable characters (NPCs) [34] or better graphical assets [8, 17, 36, 55].

Large language models (LLMs) and large multimodal models (LMMs) represent advancements in GenAI with the

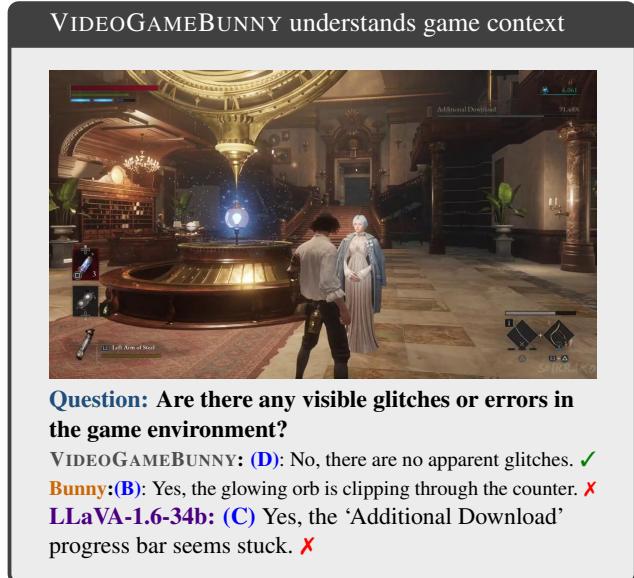


Figure 1. VIDEOGAMEBUNNY is a model specifically fine-tuned on video game content, enabling it to understand game contexts and respond to related questions more accurately.

potential to function as vision assistants and solve complex problems across various domains [1, 5, 45]. In video games, LMMs can offer significant benefits for both in-game experiences and game development. In-game, LMMs can serve as vision assistants, enhancing players' experiences by guiding them through tasks like crafting new items [32]. They also have the potential to narrate the game, summarize events, and highlight critical gameplay moments [4]. For game development, LMMs have the potential to assist in detecting bugs [42], creating bug reports, and deploying automated in-game bots that interact with the environment [44]. These applications require robust models capable of understanding game content.

Despite advances and promises, existing LMMs, particularly open-source models, encounter challenges in accurately understanding game content, such as scenes and

world physics [42] (e.g., see Fig. 1).

In this study, we make the first important step towards addressing these challenges by releasing a suite of datasets specifically designed for video game content and introducing VIDEOGAMEBUNNY, a model trained for video game content understanding. Our study centers on the following research questions:

- (RQ1) Which type of data has the potential to improve the model’s performance?
- (RQ2) Which data type mixture strategy improves the model’s performance the most?
- (RQ3) How does VIDEOGAMEBUNNY perform compared to state-of-the-art (SOTA) open-source models on game understanding tasks?

Our main contributions are as follows:

1. We release VIDEOGAMEBUNNY, a model specifically fine-tuned for video game question-answering tasks.
2. We release a suite of datasets containing 185,259 video game images from 413 games, featuring various game-play elements and graphical styles. Our datasets include 389,565 image-instruction pairs with captions, question-answering tasks, and JSON representations of images (see Sec. 4).
3. We conduct experiments to demonstrate the effectiveness of different instruction datasets and their impact on the model’s performance (see Sec. 6).
4. We release a replication package containing the training logs and intermediate checkpoints at <https://videogamebunny.github.io>.

## 2. Background and Related Work

### 2.1. Large multimodal models

Large multimodal models (LMMs) enhance large language models (LLMs) by incorporating additional modalities such as images or audio, enabling them to process multimodal inputs and generate textual outputs. The role of the language model is to comprehend user instructions and produce responses based on the additional modality inputs provided. Standard approaches to create LMMs involve combining pre-trained models with different modalities via projection layers. These layers can be implemented using simple mechanisms such as multilayer perceptrons (MLP) [24, 27] or transformer layers [25]. Alternatively, a resampler module like Perceiver [3, 20, 21] or Qformer [10, 60] selectively chooses features to reduce the number of visual tokens based on the context and instruction, enhancing efficiency and maintaining performance.

In this study, we focus on LMMs that accept input images and text to produce responses, particularly using the LLaVA-style architecture [29], which is one of the most popular methods [14, 26, 28, 33]. This architecture employs an MLP layer to integrate vision tokens with a language

model.

### 2.2. Instruction following data

Large models trained on massive corpora of text, such as GPT-3 [7], T5 [37], and PaLM [9], are not inherently instruction-following, meaning they do not respond to user queries. To enable these models to follow user instructions and answer queries, they usually undergo a process called instruction tuning [35, 56]. This process involves fine-tuning the models to handle specific user instructions, such as questions or commands, allowing them to respond appropriately based on the given instructions.

In the multimodal context, particularly for models that accept visual inputs, there are various types of visual instruction-following data, such as detailed descriptions, conversational style question answering (Q&A), and complex reasoning. Researchers have explored diverse approaches to generate such data, including the use of academic text-oriented visual Q&A datasets [10]. The LLaVA model [24] demonstrated that leveraging a strong text-only LLM and an image dataset annotated with object names and bounding box information can be converted into effective visual instruction-following data.

### 2.3. LLMs and LMMs in video games

LLMs have shown strong promise for integration with games for a wide range of tasks, from content creation to game-playing agents [13, 19, 30, 39, 43, 47–50, 52, 57, 59]. Large multimodal models (LMMs) can further enhance this integration by providing richer context inputs such as images and videos to enable broader applications. Projects like Cradle [44], which focuses on playing Red Dead Redemption 2 with GPT-4V [1] showcase LMMs’ abilities to identify objects, characters, and environmental features and assist in controlling the game. Beyond gameplay, LMMs have found applications in game testing [42, 43], where they are leveraged for detecting and interpreting video game bugs.

Our study is the first to explore enhancing an LMM’s general game understanding, rather than focusing on a specific game or task. We use screenshots from 413 games, aiming to improve capabilities across various game-related tasks by developing broader game comprehension skills.

### 2.4. Empirical analysis of large multimodal models

Some previous studies have conducted experiments to see how different architectural components or data sources affect the general performance of large multimodal models [22, 23, 33, 46]. For example, McKinzie et al. [33] found that the input resolution of the input image plays a crucial role in improving performance, and Laurençon et al. [22] found that utilizing cross-attention between image and language is more effective than the adapter-based method.

We are the first to systematically investigate the impact of different instruction-following datasets and their combinations on the performance of LMMs in game understanding tasks.

### 3. VIDEOGAMEBUNNY Model Architecture

In this section, we describe the architectural choices and configurations behind our model, VIDEOGAMEBUNNY. VIDEOGAMEBUNNY is based on Bunny [14], a family of efficient and high-performing LLMs known for their competitive or superior performance on various benchmarks compared to many open-source alternatives.

Bunny follows the same principle as LLaVA [24, 27] for the integration of image inputs. Using a shallow network of multilayer perceptrons (MLPs) as the projection layer, vision embeddings extracted from a strong pre-trained vision model are processed and provided as image tokens for the language model. This technique effectively leverages pre-trained vision and language models, allowing them to work together efficiently.

Bunny offers various combinations of vision and language models and supports images with resolutions up to  $1152 \times 1152$  pixels. For creating VIDEOGAMEBUNNY, we selected Bunny configurations that deliver the best performance [14] while being small enough to run on a consumer-grade graphics card. We use LLama-3-8B [2] as the language model and SigLIP [54] with the S<sup>2</sup> wrapper [40] for the vision encoder. The S<sup>2</sup> wrapper extracts features from an input image at various scales to form a multi-scale feature. This is potentially useful since video games often contain visual elements at different scales, from tiny UI icons to large objects. A multi-scale feature could capture these diverse elements. Fig. 2 shows the architecture of VIDEOGAMEBUNNY.

### 4. Instruction-following Data for Video Game Content

One of the main challenges limiting the ability of open-source models to generalize effectively to video game content is the lack of instruction-following data specific to video games in public datasets. Our goal is to collect game-specific data to address this challenge. In this section, we explain the process of collecting and generating game-specific instruction-following data.

#### 4.1. Video game images

We collect images from YouTube by searching for *gameplay walkthroughs* with Full-HD, 4K, and 8K quality. These high-resolution videos ensure that downsampled frames retain more information and details compared to lower quality videos. We randomly sample frames from the downloaded videos and label them with the corresponding game name.

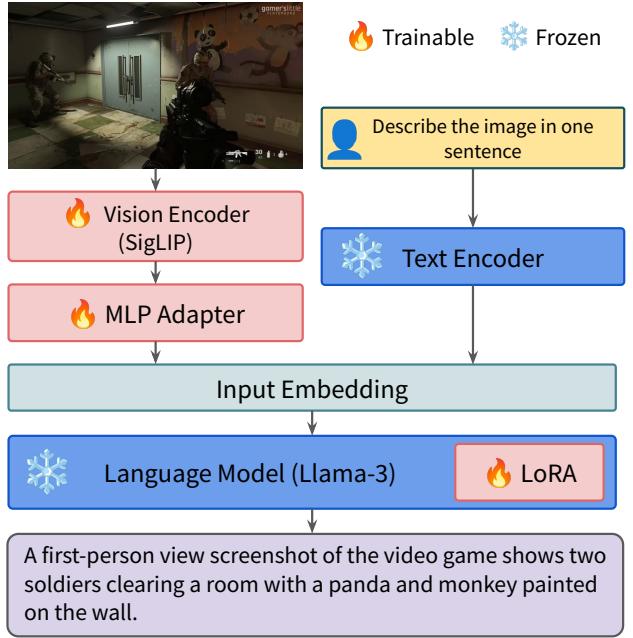


Figure 2. Architecture overview of VIDEOGAMEBUNNY. An image input and a textual instruction are fed into the language model to produce a response. The image is passed through a separate pre-trained vision encoder and a projection layer to align the embedding space between the two models. and icons show trainable and frozen layers respectively

In total, our dataset contains 185,259 images from 413 different video games, encompassing various genres, graphic styles, and gameplay mechanics. Fig. 3 shows some sample images from our dataset, and Fig. A5 shows the distribution of images per game.

#### 4.2. Generating instructions

Following previous studies [27, 29, 51], we employ another robust model to generate instructions in the form of user queries and responses for images in our dataset. We categorize the instructions into four types: *short captions* (70,673 samples), *long captions* (70,799 samples), *image-to-JSON* (136,974 samples) and *image-based question answering* (81,122 samples). In this section, we explain how we generate each type of instructions. Fig. 4 shows an overview of the data generation process.

##### 4.2.1 Image captioning

Image captioning is a basic form of instruction-following that generates a description of the input image. An image caption can be short and concise, providing a high-level overview of the image, or very detailed, covering fine-grain details. Our dataset includes both forms of image captioning to meet user queries, whether they seek a detailed cap-



Figure 3. Our dataset includes sample video game images that showcase a wide range of characters, environments, mechanics, camera viewpoints, and artistic styles. These styles vary from western to contemporary and futuristic, and from realistic to fantasy settings.

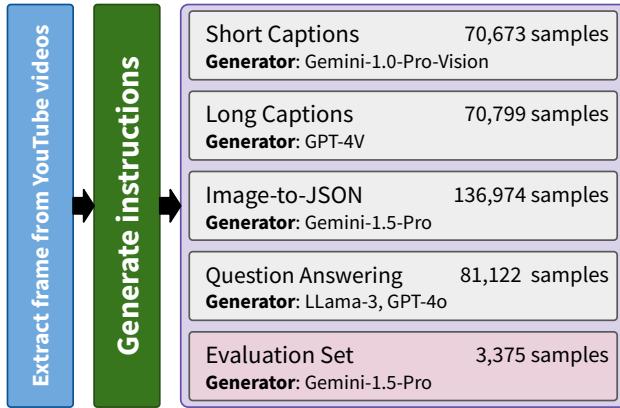


Figure 4. Overview of the dataset generation process.

tion or a short summary. In addition, it includes a structured version where the image is described in 16 predefined fields.

**Short captions:** We use Gemini-1.0-pro-vision to obtain short descriptions of a subset of images in our dataset, which includes 70,673 images. We use the “*Describe the image*” prompt, which generates captions typically consisting of one or a few sentences.

**Long captions:** While short captions provide a high-level overview of the image, it lacks many details in the image which might be useful for the user. To address this, we use GPT-4V to get detailed captions of all images in the previous section (see Fig. A1 for the used prompt).

**Image-to-JSON:** Another comprehensive method for describing images is converting them into a JSON format. This approach summarizes an input image into a JSON structure, with each key describing an element from the image, such as *characters in the image* or *description of Game UI*. Unlike typical captioning, this method provides a template that must be filled. If the image lacks a certain element, that part remains empty, indicating the absence of that element in the image. This ensures a more detailed and structured representation of the image content.

Another benefit of describing an image in JSON is that this structured representation facilitates integration with other systems. JSON is a widely adopted format for sharing information between different software systems. Summarizing the image as JSON can help integrate LMMs in other

systems, such as software testing pipelines, potentially verifying the game output and ensuring that the visual output has the desired properties and information.

To create the image-to-JSON dataset, we use Gemini-1.5-Pro with instructions (Fig. A2) to convert a given image into a JSON file with hierarchical levels of detail and information. The JSON file contains 16 elements that capture both high-level and fine-grained details of the image. These keys are chosen to capture game-specific elements from the image in isolation, which can be used for downstream applications, such as game testing. It starts with an overall summary of the image and then moves to specific aspects such as detailed character descriptions (including facial expressions and clothing), weather information, summaries of UI and player inventory, objects in the scene, and lighting and environmental effects. Tab. A1 shows the keys included in the JSON output. Our dataset contains 136,974 Image-JSON pairs. Fig. 5 shows a sample of information extracted from an image.

#### 4.2.2 Question-Answering conversations

Moving beyond simple image descriptions, a general and capable model should be able to respond to user questions based on the content of the image. Below, we describe how we generate such data for each image (1) from its long caption and (2) directly from the image itself.

**Llama-3-assisted visual instruction data generation:** We use Llama-3 to convert long captions generated by GPT-4V into a series of question-answering conversations. This approach is similar to the original LLaVA [29] method, but instead of using an object’s name and its bounding box information, we directly utilize long captions.

While long captions provide a rich source of information, they lack the structure of question-answer formats. For example, if a caption describes a person in the image with specific details, such as clothing, an LLM can generate a question like, “*What is the color of the dress of the person in the image?*” By utilizing a strong text-only model, we can transform each caption into a multi-turn conversation between a user and an assistant.

We use Llama-3-70B to transform GPT-4V captions into question-answer conversations, with the prompt shown in Fig. A3. The prompt requires questions to directly relate to

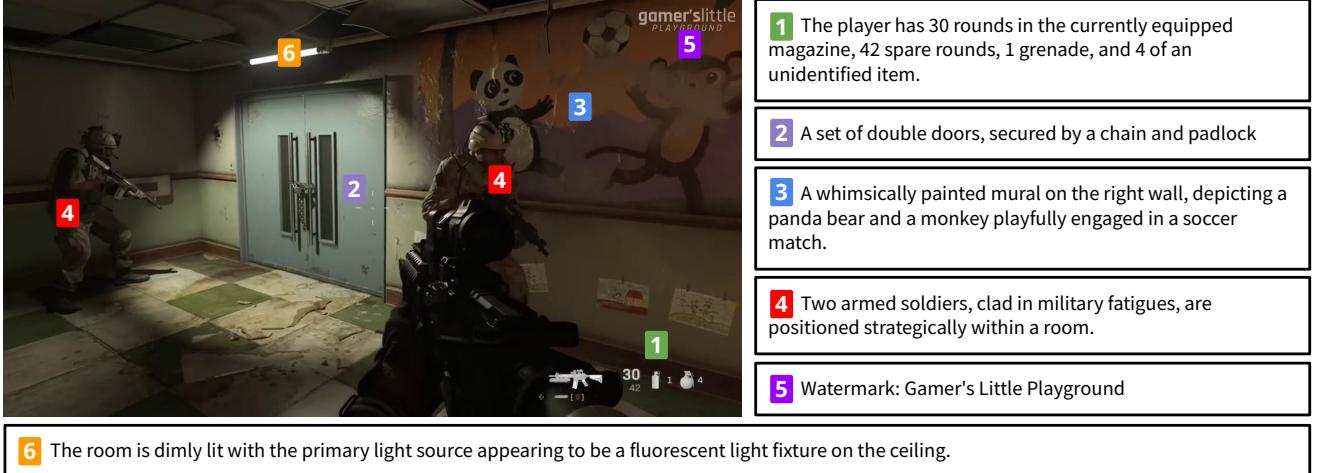


Figure 5. Sample information extracted for the image-to-JSON dataset by Gemini-1.5-Pro. Each sample contains detailed information ranging from minor details to high-level descriptions, such as: 1 player inventory, 2 3 details about the environment, 4 non-player characters, 5 the screenshot’s watermark, and 6 lighting.

the image description. We create 496,469 question-answer pairs for 70,232 images, grouping questions for each image into a multi-turn conversation.

**Image-based question-answering:** We use GPT-4o to generate questions and their answers based on an input image in a *single* prompt. In the prompt (Fig. A4), we first ask GPT-4o to examine the image and provide a detailed description of its content, then to generate relevant questions based on the content of the image and provide answers for each question. In the prompt, we emphasize that the questions should focus on understanding the image to avoid questions that might not be directly relevant to the image.

### 4.3. Evaluation dataset

To assess model performance on video game understanding tasks, we created a multiple-choice question evaluation set using Gemini-1.5-Pro [45]. This approach allows for an efficient comparison of various models. While Gemini-1.5-Pro offers significant advantages over open-source models for data generation, it does have limitations. We reduce noise in the generated questions as follows:

1. **Initial Generation:** We use Gemini-1.5-Pro to create 4,000 questions across 10 categories related to video game content understanding (see Tab. 1).

#### 2. Quality Assessment:

- (a) Self-evaluation: We test Gemini-1.5-Pro on its own questions and found it achieves an accuracy of 84%.
- (b) Manual validation: A random sampling of questions and answers revealed a 14% error rate (incorrect or indeterminate answers).

#### 3. Noise Reduction:

- (a) We remove 625 samples that Gemini-1.5-Pro had answered incorrectly.
- (b) We conduct a second manual analysis and found that the error rate dropped to 9%.

## 5. Experiments

In this section, we describe our experiments to explore how our collected instruction-following datasets can improve a model’s understanding of game context. We focus on three research questions:

**(RQ1) Which type of data has the potential to improve the model’s performance?** In addressing this question, we fine-tune Bunny using a single dataset at a time to observe overall performance trends. Since the primary goal of this experiment is to identify general trends, we fine-tune Bunny on different subset sizes for each dataset only *once*. We increase the subset size from 2K to 60K samples and stop the experiment if we observe a sharp decline in performance.

**(RQ2) Which data type mixture strategy improves the model’s performance the most?** We evaluate different data mixing strategies at various sizes to see how both mixture and subset size change the performance of the model. We use the following four strategies:

1. **Random:** We randomly sample without replacement from the combined dataset pool. This serves as a control group, using no specific selection strategy.
2. **Equal:** We select an equal number of samples from each dataset to ensure a balanced representation.
3. **Stratified:** Datasets are mixed based on *video games*, maintaining the game distribution in the final dataset. This balances game representation and ensures diverse image types. We focus on game variety rather than

Table 1. Categories of questions in our dataset, along with a sample question for each category.

Category	Description	Count
Action Understanding	Recognizing and describing the actions taking place within the image. <i>Sample: What action is the character in the foreground performing?</i>	356
Anomalies and Glitches	Identifying errors, bugs, glitches, or placeholder elements within the game environment. <i>Sample: Describe any anomalies or glitches present in the image.</i>	223
Character Analysis	Recognizing characters, understanding their roles, and interpreting their expressions and poses. <i>Sample: What is Aloy's emotional state based on her facial expression?</i>	312
Common Sense Reasoning	Understanding the image using general knowledge and everyday logic. <i>Sample: Based on the score and time remaining, which team is likely to win the match?</i>	430
Gameplay Mechanics	Understanding the rules and mechanics that govern the game. <i>Sample: What game mechanic is most likely being utilized by the player character?</i>	273
OCR and UI	Reading and interpreting on-screen text and user interface elements. <i>Sample: What is written in the caption box at the bottom of the image?</i>	334
Miscellaneous	Any other type of question that does not fit into the previous categories. <i>Sample: What material are the containers in the image primarily made of?</i>	239
Scene Understanding	Recognizing and interpreting the overall environment or setting in the image. <i>Sample: The racetrack depicted in the image is set in what type of environment?</i>	566
Small Details	Identifying and interpreting small but significant details within the image. <i>Sample: What color is the jacket worn by the character in the foreground?</i>	356
Spatial Reasoning	Testing the ability to understand spatial relationships of objects present in the image. <i>Sample: What is the spatial relationship between the two red markers visible in the image?</i>	286

instruction types. Games with insufficient samples are excluded.

4. **Weighted:** We use the three most effective datasets from **RQ1**: image-based question-answering (GPT-4o), long captions, and image-to-JSON. We assign weights: 30% each for GPT-4o and long captions, 40% for image-to-JSON. This prioritizes valuable datasets to assess their impact on model performance.

We fine-tune Bunny on the above dataset mixture strategies with sizes ranging from 2K to 30K. We repeat each experiment three times, using different samples for each strategy to report the mean performance and standard deviation. We stop at 30K since our smallest dataset (generated by GPT-4o) contains 10K samples, and at 30K, we will exhaust the *Equal* and *Weighted* strategies.

**(RQ3) How does VIDEOGAMEBUNNY perform compared to SOTA open-source models on game understanding tasks?** Building on insights from our experiments, we create VIDEOGAMEBUNNY, a model fine-tuned on a dataset of 50K image-instruction samples compiled from all previously introduced datasets. To assess the effectiveness of fine-tuning a smaller model on game-specific data, we evaluate VIDEOGAMEBUNNY against LLaVA-1.6-34b, a SOTA open-source model with  $4.2 \times$  more parameters.

**Experiment setup:** We instruction tune Bunny with LoRA [18] using the PEFT [31] library. Given that Bunny has been trained on real images, we unfreeze the vision encoder (SigLIP [54]) to adapt to the diverse visual styles of different games. To prevent overfitting and memorization, we fine-tune for only one epoch in all experiments.

Given the importance of reproducibility and accessibility for all researchers, we perform all experiments on a



Figure 6. Relative performance improvement (pp) of Bunny fine-tuned on different subsets of each dataset. The image-to-JSON dataset shows a strong positive trend, while the short captions dataset degrades performance. The best performance achieved in the experiment is highlighted in bold.

single NVIDIA A100 GPU (80GB), ensuring a balance between computational power and accessibility. The total GPU hours needed to conduct all experiments, including some preliminary tests, is approximately 900 hours, which is roughly \$2,000 when using cloud providers.

## 6. Results

**RQ1: Which type of data has the potential to improve the model’s performance?**

**The image-to-JSON dataset has the greatest potential to improve the base model’s performance.** Fig. 6 shows the performance after fine-tuning Bunny using a single dataset at a time. Fine-tuning on a subset of the image-

Table 2. Performance of models fine-tuned on a mixture of data with various strategies. The *Weighted* strategy leads to better performance with smaller dataset sizes, but as size increases, all strategies perform similarly. We use a strategy similar to *Weighted* to train **VIDEOGAMEBUNNY** with 50K samples.

Size	Random	Equal	Stratified	Weighted
2K	$76.7 \pm 0.9$	$77.8 \pm 0.8$	$78.0 \pm 0.2$	$79.0 \pm 0.6$
5K	$79.2 \pm 0.4$	$79.9 \pm 0.4$	$80.0 \pm 0.5$	$79.8 \pm 0.6$
10K	$79.8 \pm 0.8$	$80.8 \pm 0.6$	$80.8 \pm 0.1$	$81.4 \pm 0.5$
20K	$81.5 \pm 0.1$	$81.3 \pm 0.7$	$81.8 \pm 0.8$	$82.3 \pm 0.9$
30K	$81.8 \pm 0.4$	$81.2 \pm 1.1$	$81.6 \pm 0.7$	$82.6 \pm 0.3$
50K	—	—	—	<b>85.1</b>

to-JSON dataset shows the greatest improvements, as this leads to an accuracy above 82% (**+8.7** percentage points (pp) above the baseline of 73.3%) for subset sizes over 10K, with the best performance achieved at 60K (**+11.7** pp).

**While all datasets lead to performance improvement, short captions can degrade it.** Fine-tuning Bunny on a dataset of 10K or 20K short captions degrades performance (**-35.5** pp and **-30** pp), suggesting that short captions do not contain enough signal for the models to improve and can negatively affect the model.

## RQ2: Which data type mixture strategy improves the model’s performance the most?

**There is a general improvement trend as we increase the size across all strategies.** Tab. 2 shows the performance of the models that were fine-tuned using our data mixture strategies. As we increase the dataset size, the mean performance of all mixtures improves. For instance, the *Random* strategy improves from 76.7% at 2K samples to 81.9% at 30K samples and the *Weighted* strategy shows an improvement from 79.0% at 2K samples to 82.6% at 30K samples. This trend demonstrates the value of additional data regardless of the mixing strategy employed.

**As the size of dataset increases, different strategies perform similarly.** The performance difference between various strategies converges as we increase the size of the datasets, and they perform similarly in terms of mean and standard deviations. Yet, the *Weighted* method achieves the highest average among other strategies (82.6%). This convergence suggests that the choice of mixing strategy becomes less critical as more data becomes available. In contrast, smaller dataset sizes such as 2k indicate that the *Weighted* strategy outperforms other mixture strategies, achieving an accuracy of  $79.0 \pm 0.6$ .

**Having a uniform distribution of games does not significantly improve performance.** The *Stratified* strategy, which aims to balance the representation of different games in the dataset, does not significantly enhance performance

Table 3. Average improvement for different sizes for each category

Category/Dataset Size	2K	5K	10K	20K	30K
Action Understanding	1.6	2.5	2.5	3.7	3.9
Anomalies and Glitches	23.4	33.0	33.2	34.0	32.0
Character Analysis	2.6	3.9	4.2	4.7	4.4
Common Sense Reasoning	3.7	4.2	3.8	4.3	4.0
Gameplay Mechanics	4.2	5.0	6.4	8.2	8.9
HUD and UI	9.3	12.9	16.5	18.9	21.0
Miscellaneous	7.2	7.9	9.6	9.9	9.8
Scene Understanding	<b>-0.2</b>	0.6	1.3	2.0	2.0
Small Details	0.3	1.2	2.4	3.4	3.0
Spatial Reasoning	5.3	6.2	7.1	7.8	7.4

Table 4. Performance of various models on the evaluation set (%).

Model	Accuracy	Model	Accuracy
Bunny-1.1-Llama-3-8B	73.3	LLaVA-v1.5-13b	64.6
<b>VIDEOGAMEBUNNY</b>	<b>85.1</b>	LLaVA-v1.6-vicuna-13b	71.7
LLaVA-v1.5-7b	61.3	LLaVA-v1.6-34b	83.9

compared to other strategies. For example, in the 2k dataset, the *Stratified* strategy ( $78.0 \pm 0.2$ ) is outperformed by the *Weighted* strategy ( $79.0 \pm 0.6$ ). Similarly, in the 30k dataset, the performance of both strategies is comparable ( $81.6 \pm 0.7$  vs  $82.6 \pm 0.3$ ).

**Fine-tuning improves performance across all categories, with *Anomalies and Glitches* improving the most.** Fig. A6 shows that fine-tuning improves Bunny’s performance across all categories for almost all dataset sizes. The *Anomalies and Glitches* and *HUD and UI* categories improve the most, with average improvements of **+32.0** and **+21.0**, using a dataset size of 30K (Tab. 3).

## RQ3: How does **VIDEOGAMEBUNNY** perform compared to SOTA open-source models on game understanding tasks?

**VIDEOGAMEBUNNY** achieves **85.1%** (Tab. 2) on the evaluation set, outperforming all trained models and surpassing various open-source models (Tab. 4). It outperforms even LLaVA-1.6-34b, despite its larger parameter count, by **+1.2**. Breakdown of accuracy per category reveals that the most significant benefits come from game-specific categories, such as anomaly and glitch detection (**+16.6**) and *HUD and UI* (**+3.0**) (Tab. A2). Fig. 7 shows a sample from evaluation set where **VIDEOGAMEBUNNY** correctly answers a common-sense reasoning question while both Bunny and LLaVA-1.6-34b fail.

We also conduct an informal qualitative analysis, prompting the models with more open-ended questions, which underlines that **VIDEOGAMEBUNNY** demonstrates an improved understanding of video game content and provides better answers (Fig. 8, and Sec. A3).

### Models comparison in video game understanding



**Question:** Based on the score and time remaining, which team is likely to win the match?

**VIDEOGAMEBUNNY:** (B): The blue team is likely to win ✓

**Bunny:** (C): The red team is likely to win. ✗

**LLaVA-1.6-34b:** (C): The red team is likely to win. ✗

Figure 7. VIDEOGAMEBUNNY correctly utilizes information on the HUD to answer the question, while the basemodel Bunny and the larger model LLaVA-1.6-34b fail to provide a correct answer.

## 7. Discussion

**Potential negative impacts:** Our study initiates the development of a model for understanding game content, with applications in game playing, testing, and commentary. The short-term implications for the gaming industry include enhancing the productivity of game testers and enhancing quality assurance processes. One possible negative impact of such advancement is the facilitation of in-game cheating. As AI models becomes more adept at understanding game contents, there is a risk that they could be used to create sophisticated cheating tools.

**Biases and mistakes in dataset generation:** In this study, we use existing models to annotate our dataset. While this approach aligns with previous studies [29, 61] and follows the same principles as the teacher-student learning paradigm [16], we acknowledge that our data may contain biases and mistakes introduced by these existing models.

**Quality of images:** Images in our dataset, sourced from YouTube videos, may include low-quality frames despite filtering for high-quality videos. Motion blur in video games and sampling during scene transitions can result in some blurry or less-than-ideal images.

**Use of various models for annotation:** In the annotation process, we utilize various models depending on their availability and cost. While this process brings diversity to the labels, it is not meant to compare the different models against each other in terms of performance.

**Use of multiple-choice questions to evaluate game understanding:** We use multiple-choice questions to evaluate

### Models comparison in image captioning



**Question:** Write a short summary of the image

**VIDEOGAMEBUNNY:** The image is a screenshot from a video game, showing a character in mid-air, equipped with a wingsuit, flying over a mountainous landscape. The game's HUD elements are visible, including a distance marker and a weapon indicator. . . .

**Bunny:** A video game scene shows a man in a *green shirt* and blue pants jumping off a cliff. He is *holding a parachute* and is in mid-air. . . .

**LLaVA-1.6-34b:** The image appears to be a screenshot from a video game, showing *a first-person perspective* of a character in mid-air, likely after jumping or falling from a height. The character is *equipped with a parachute, which is open* and slowing their descent. . . .

Figure 8. VIDEOGAMEBUNNY provides a detailed description of video game content in the image, while other models show inaccuracies (highlighted in *red*). Responses are truncated to save space.

a model’s game understanding, as it allows for a clear comparison. While this format has been extensively used for benchmarks [15, 53], it might not be the best proxy for game understanding. Future work needs to focus on human evaluation or the use of LLMs as judges [58].

## 8. Conclusion

We introduce a new instruction-following dataset, with 389,565 image-instruction pairs, specifically designed for video game understanding. We investigate the effectiveness of fine-tuning LMMs on different instruction-following dataset types and mixtures of them, and finally introduce VIDEOGAMEBUNNY, an 8B parameter model that outperforms a SOTA model, LLaVA-1.6-34b, on a game-related question answering benchmark.

## References

- [1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023) 1, 2
- [2] AI@Meta: Llama 3 model card (2024), [https://github.com/meta-llama/llama3/blob/main/MODEL\\_CARD.md](https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md) 3
- [3] Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems **35**, 23716–23736 (2022) 2
- [4] Anonymous: While i wait for gpt-4o with updated voice capabilities, i decided to create a prototype using multiple open source models to simulate an ai commentator who can see your screen and listen to in-game dialogue. Reddit r/OpenAI (2024), [https://www.reddit.com/r/OpenAI/comments/1dm6lg9/while\\_i\\_wait\\_for\\_gpt4o\\_with\\_updated\\_voice/](https://www.reddit.com/r/OpenAI/comments/1dm6lg9/while_i_wait_for_gpt4o_with_updated_voice/), accessed: June 30, 2024 1
- [5] Anthropic: Introducing the next generation of claude (2024), <https://www.anthropic.com/news/clause-3-family> 1
- [6] Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021) 1
- [7] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020) 2
- [8] Chen, Y., He, T., Huang, D., Ye, W., Chen, S., Tang, J., Chen, X., Cai, Z., Yang, L., Yu, G., et al.: Meshanything: Artist-created mesh generation with autoregressive transformers. arXiv preprint arXiv:2406.10163 (2024) 1
- [9] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. Journal of Machine Learning Research **24**(240), 1–113 (2023) 2
- [10] Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P.N., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning. Advances in Neural Information Processing Systems **36** (2024) 2
- [11] Eloundou, T., Manning, S., Mishkin, P., Rock, D.: Gpts are gpts: An early look at the labor market impact potential of large language models. arXiv preprint arXiv:2303.10130 (2023) 1
- [12] Forum, W.E.: Gaming boomed in lockdown and market value will reach \$320bn. World Economic Forum (2023), <https://www.weforum.org/agenda/2023/01/gaming-market-value-320bn-2026/> 1
- [13] Gallotta, R., Todd, G., Zammit, M., Earle, S., Liapis, A., Togelius, J., Yannakakis, G.N.: Large language models and games: A survey and roadmap. arXiv preprint arXiv:2402.18659 (2024) 2
- [14] He, M., Liu, Y., Wu, B., Yuan, J., Wang, Y., Huang, T., Zhao, B.: Efficient multimodal learning from data-centric perspective. arXiv preprint arXiv:2402.11530 (2024) 2, 3
- [15] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J.: Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300 (2020) 8
- [16] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 8
- [17] Holson, B.: Dimension hopper part 1 (Jun 2023), <https://generalrobots.substack.com/p/dimension-hopper-part-1> 1
- [18] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021) 6
- [19] Hu, S., Huang, T., Ilhan, F., Tekin, S., Liu, G., Kompell, R., Liu, L.: A survey on large language model-based game agents. arXiv preprint arXiv:2404.02039 (2024) 2
- [20] Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., Carreira, J.: Perceiver: General perception with iterative attention. In: International conference on machine learning. pp. 4651–4664. PMLR (2021) 2
- [21] Laurençon, H., Saulnier, L., Tronchon, L., Bekman, S., Singh, A., Lozhkov, A., Wang, T., Karamcheti, S.,

- Rush, A., Kiela, D., et al.: Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems* **36** (2024) 2
- [22] Laurençon, H., Tronchon, L., Cord, M., Sanh, V.: What matters when building vision-language models? arXiv preprint arXiv:2405.02246 (2024) 2
- [23] Li, B., Zhang, H., Zhang, K., Guo, D., Zhang, Y., Zhang, R., Li, F., Liu, Z., Li, C.: Llava-next: What else influences visual instruction tuning beyond data? (May 2024), <https://llava-vl.github.io/blog/2024-05-25-llava-next-ablations/> 2
- [24] Li, C., Wong, C., Zhang, S., Usuyama, N., Liu, H., Yang, J., Naumann, T., Poon, H., Gao, J.: Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems* **36** (2024) 2, 3
- [25] Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning. pp. 19730–19742. PMLR (2023) 2
- [26] Lin, B., Zhu, B., Ye, Y., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. arXiv preprint arXiv:2311.10122 (2023) 2
- [27] Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26296–26306 (2024) 2, 3
- [28] Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (January 2024), <https://llava-vl.github.io/blog/2024-01-30-llava-next/> 2
- [29] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. *Advances in neural information processing systems* **36** (2024) 2, 3, 4, 8
- [30] Ma, W., Mi, Q., Yan, X., Wu, Y., Lin, R., Zhang, H., Wang, J.: Large language models play starcraft ii: Benchmarks and a chain of summarization approach. arXiv preprint arXiv:2312.11865 (2023) 2
- [31] Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., Bossan, B.: Peft: State-of-the-art parameter-efficient fine-tuning methods. <https://github.com/huggingface/peft> (2022) 6
- [32] Mashable: You can use microsoft's copilot ai chatbot to learn how to play 'minecraft'. Mashable (2024), <https://mashable.com/article/microsoft-build-2024-copilot-minecraft/>, accessed on June 20, 2024 1
- [33] McKinzie, B., Gan, Z., Fauconnier, J.P., Dodge, S., Zhang, B., Dufter, P., Shah, D., Du, X., Peng, F., Weers, F., et al.: Mml: Methods, analysis & insights from multimodal llm pre-training. arXiv preprint arXiv:2403.09611 (2024) 2
- [34] NVIDIA: Nvidia ace for games sparks life into virtual characters with generative ai (2023), <https://nvidianews.nvidia.com/news/nvidia-ace-for-games-sparks-life-into-virtual-characters-with-generative-ai>, accessed: June 27, 2024 1
- [35] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. *Advances in neural information processing systems* **35**, 27730–27744 (2022) 2
- [36] Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022) 1
- [37] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* **21**(140), 1–67 (2020) 2
- [38] Research, G.V.: Video game market size, share and growth report, 2030. Grand View Research (2023), <https://www.grandviewresearch.com/industry-analysis/video-game-market> 1
- [39] Shao, X., Jiang, W., Zuo, F., Liu, M.: Swarmbrain: Embodied agent for real-time strategy game starcraft ii via large language models. arXiv preprint arXiv:2401.17749 (2024) 2
- [40] Shi, B., Wu, Z., Mao, M., Wang, X., Darrell, T.: When do we not need larger vision models? arXiv preprint arXiv:2403.13043 (2024) 3
- [41] Stokel-Walker, C., Van Noorden, R.: The promise and peril of generative ai. *Nature* **614**(7947), 214–216 (2023). <https://doi.org/10.1038/d41586-023-00340-6> 1

- [42] Taesiri, M.R., Feng, T., Bezemer, C.P., Nguyen, A.: Glitchbench: Can large multimodal models detect video game glitches? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22444–22455 (2024) [1](#), [2](#)
- [43] Taesiri, M.R., Macklon, F., Wang, Y., Shen, H., Bezemer, C.P.: Large language models are pretty good zero-shot video game bug detectors. arXiv preprint arXiv:2210.02506 (2022) [2](#)
- [44] Tan, W., Ding, Z., Zhang, W., Li, B., Zhou, B., Yue, J., Xia, H., Jiang, J., Zheng, L., Xu, X., et al.: Towards general computer control: A multimodal agent for red dead redemption ii as a case study. arXiv preprint arXiv:2403.03186 (2024) [1](#), [2](#)
- [45] Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023) [1](#), [5](#)
- [46] Tong, S., Brown, E., Wu, P., Woo, S., Middepogu, M., Akula, S.C., Yang, J., Yang, S., Iyer, A., Pan, X., et al.: Cambrian-1: A fully open, vision-centric exploration of multimodal llms. arXiv preprint arXiv:2406.16860 (2024) [2](#)
- [47] Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., Anandkumar, A.: Voyager: An open-ended embodied agent with large language models. arXiv preprint arXiv:2305.16291 (2023) [2](#)
- [48] Wang, Z., Cai, S., Chen, G., Liu, A., Ma, X., Liang, Y.: Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. arXiv preprint arXiv:2302.01560 (2023) [2](#)
- [49] Wang, Z., Cai, S., Liu, A., Jin, Y., Hou, J., Zhang, B., Lin, H., He, Z., Zheng, Z., Yang, Y., et al.: Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models. arXiv preprint arXiv:2311.05997 (2023) [2](#)
- [50] Wu, Y., Min, S.Y., Prabhumoye, S., Bisk, Y., Salakhutdinov, R.R., Azaria, A., Mitchell, T.M., Li, Y.: Spring: Studying papers and reasoning to play games. Advances in Neural Information Processing Systems **36** (2024) [2](#)
- [51] Xiao, B., Wu, H., Xu, W., Dai, X., Hu, H., Lu, Y., Zeng, M., Liu, C., Yuan, L.: Florence-2: Advancing a unified representation for a variety of vision tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4818–4829 (2024) [3](#)
- [52] Xu, X., Wang, Y., Xu, C., Ding, Z., Jiang, J., Ding, Z., Karlsson, B.F.: A survey on game playing agents and large models: Methods, applications, and challenges. arXiv preprint arXiv:2403.10249 (2024) [2](#)
- [53] Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9556–9567 (2024) [8](#)
- [54] Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11975–11986 (2023) [3](#), [6](#)
- [55] Zhang, L., Wang, Z., Zhang, Q., Qiu, Q., Pang, A., Jiang, H., Yang, W., Xu, L., Yu, J.: Clay: A controllable large-scale generative model for creating high-quality 3d assets (2024) [1](#)
- [56] Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F., et al.: Instruction tuning for large language models: A survey. arXiv preprint arXiv:2308.10792 (2023) [2](#)
- [57] Zhang, W., Lu, Z.: AdaRefiner: Refining decisions of language models with adaptive feedback. In: Findings of the Association for Computational Linguistics: NAACL 2024. pp. 782–799. Association for Computational Linguistics (jun 2024), <https://aclanthology.org/2024.findings-naacl.50> [2](#)
- [58] Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al.: Judging lilm-as-a-judge with mt-bench and chatbot arena. Advances in Neural Information Processing Systems **36** (2024) [8](#)
- [59] Zheng, S., Feng, Y., Lu, Z., et al.: Steve-eye: Equipping lilm-based embodied agents with visual perception in open worlds. In: The Twelfth International Conference on Learning Representations (2023) [2](#)
- [60] Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023) [2](#)

- [61] Zhu, Y., Zhu, M., Liu, N., Ou, Z., Mou, X., Tang, J.: Llava- $\phi$ : Efficient multi-modal assistant with small language model. arXiv preprint arXiv:2401.02330 (2024) 8

# Supplementary Material

## A1. Additional details

### A1.1. Prompts used to generated datasets

Figure A1. Long caption generation with GPT-4V

*Please provide a detailed description of the image, ensuring that no details are omitted. Describe every element you observe within the image to provide a comprehensive account of its contents. Don't be lazy and it is important to get everything well done.*

Figure A2. image-to-JSON data generation

*First, provide a detailed description of the image, including every small detail possible. Next, create ten multiple-choice questions based on the content of the image. Each question should test the understanding of the image's content. Follow this JSON format:* { "description": "Full Image Description", "short\_description": "Short Image Description", "dialogue": ["Any visible dialogue text as a json list"], "on\_screen\_subtitle": "any subtitle on the image or n/a", "minimap\_details": "Information from the minimap", "inventory\_display": "Information about the player's inventory", "score\_or\_progress": "Details about scores or progress indicators", "NPC\_status": "Information about NPCs", "event\_indicators": "Indicators of any special events", "interaction\_prompts": "Visible prompts for player interactions", "game\_mode": "Current game mode or context", "HUD\_description": "description of the game HUD or n/a if there is no HUD", "on\_screen\_watermark": "any watermark on the image or n/a", "summary\_of\_ui\_values": "summary of the UI values as json or empty json if there is no UI", "scene\_description": "A high-level overview of the entire scene", "character\_list": [ { "name": "Character Name", "appearance": "Description of appearance", "clothing": "Description of clothing", "facial\_expression": "Description of facial expression" } ], "object\_list": [ "Object 1", "Object 2", ... ], "texture\_details": "a json list of object name and texture patterns that they have", "lighting\_details": "Specific information about the light sources and shadows in the scene", "color\_palette": [ "hexadecimal color code", "hexadecimal color code", ... ], "weather\_conditions": "Description of any weather effects present, or say cannot be determined", "environmental\_effects": "Description of any environmental effects like fog, rain, fire, etc.", "animation\_states": "Descriptions of any static poses or actions implied by character positions", "error\_log": "Any noticeable glitches or anomalies in the image", "glitches": "any glitch or buggy aspect of the image or none if there is nothing", "player\_status": { "health": "Player's health value", "equipment": "Player's equipment details", "other\_status": "Other status indicators" } }

Figure A3. LLama-3-based data generation

*Using the image description provided below, create 10 questions and their corresponding answers that pertain exclusively to the details given in the description. Format your response using JSON.*

**Image description:** <image description here >

*Ensure your questions are relevant and directly related to the image description. For example, do not ask about elements not explicitly mentioned in the description.*

Figure A4. GPT-4o-based data generation

*First, provide a detailed description of the image, including every small detail possible. Next, create 10 questions answers based on the content of the image. Each question should test the understanding of the image's content.*

Table A1. Description of entries in the JSON structure

Key	Description
Description	Detailed description of the image
Short description	Concise description of the image
Dialogue	A (JSON) list containing any visible dialogue text
On-Screen subtitle	Subtitles displayed on the image
Inventory display	Details of the player's inventory visible on the image
HUD description	Description of the game's Head-Up Display (HUD)
Scene description	High-level overview of the entire scene
NPC status	High level information about non-playable characters (NPCs)
Character list	List of characters, including their appearances, clothing, and facial expressions
Animation states	Descriptions of static poses or actions suggested by character positions
Object list	A (JSON) list containing all the visible objects in the scene
Texture details	A (JSON) list detailing object names and their texture patterns
Lighting details	Specific information about the light sources and shadows in the scene
Weather conditions	Description of any weather effects present, or state if they cannot be determined
Environmental effects	Description of environmental effects such as fog, rain, or fire
Player status	Player's health, equipment details, and other status indicators

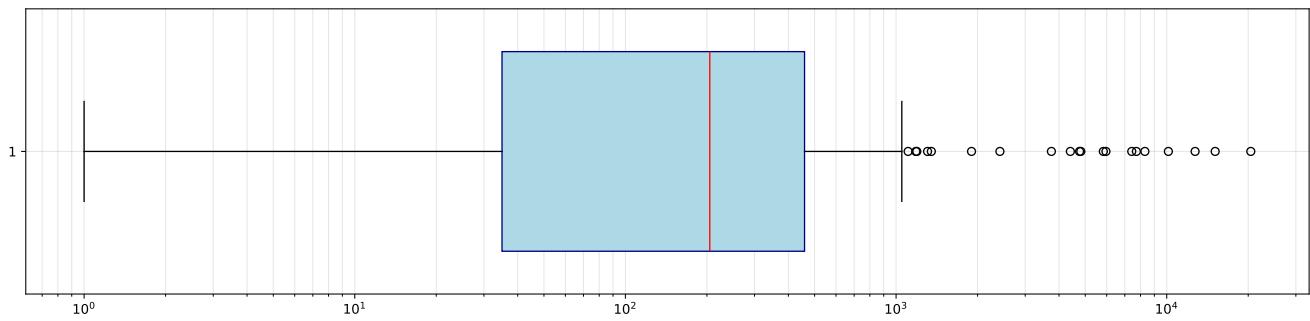


Figure A5. Image distribution across games, with a median of 205 unique images per game.

## A2. Additional results

In this section, we provide complementary results for the experiments conducted in the main text.

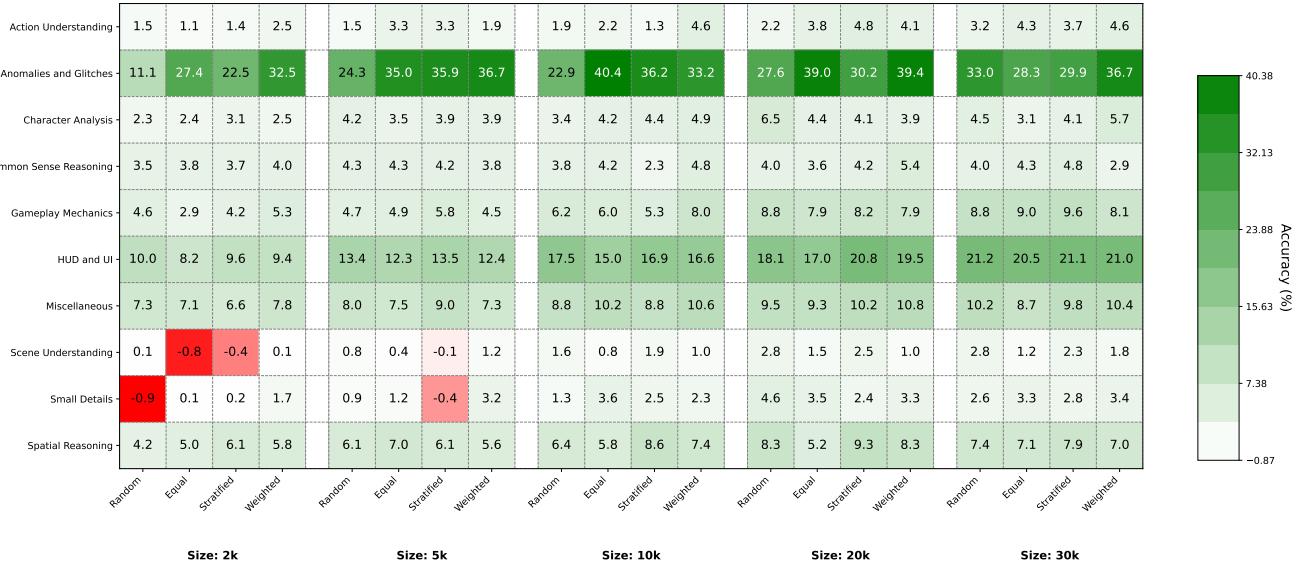


Figure A6. Breakdown of improvement (percentage points) compared to the basemodel Bunny, after fine-tuning it on different mixture strategies by different question categories. As we increase the size of the dataset, all categories show improvement, with *Anomalies and Glitches* showing the greatest gains.

Table A2. Comparison of VIDEOGAMEBUNNY and LLaVA-1.6-34b

Category	VIDEOGAMEBUNNY	LLaVA-1.6-34b
Action Understanding	84.6 ↑(+2.9)	81.7
Anomalies and Glitches	82.1 ↑(+16.6)	65.5
Character Analysis	84.9 ↑(+0.6)	84.3
Common Sense Reasoning	89.1 ↓(-2.8)	91.9
Gameplay Mechanics	80.6 ↓(-5.8)	86.4
HUD and UI	84.1 ↑(+3.0)	81.1
Miscellaneous	86.2 ↑(+1.7)	84.5
Scene Understanding	92.4 ↑(+0.2)	92.2
Small Details	80.1 ↓(-0.2)	80.3
Spatial Reasoning	78.3 ↑(+1.0)	77.3

### A3. Qualitative samples

In this section, we provide qualitative results from VIDEOGAMEBUNNY.

Qualitative results where VIDEOGAMEBUNNY provides a detailed description of the image.



**Question:** Briefly describe the image.

**VIDEOGAMEBUNNY:** The image is a screenshot from a video game, showing a character standing in front of a large, ornate painting. The painting depicts a man with long hair and a beard, wearing a period outfit with a high collar and a sash. The character is facing the painting, with their back to the viewer, and appears to be observing it. The room has a dimly lit, old-fashioned ambiance, with a sconce on the wall and a large, framed painting on the wall behind the character.

Figure A7. VIDEOGAMEBUNNY provides a detailed description of the image

Qualitative results where VIDEOGAMEBUNNY provides a detailed description of the image.



**Question:** Briefly describe the image.

**VIDEOGAMEBUNNY:** The image is a still from a video game, depicting two characters in a dimly lit wooden structure. The character on the left is a man with a stern expression, wearing a dark robe with a patterned design. He is facing the other character, who is partially obscured and has their back to the viewer. The second character appears to be wearing a similar robe. The scene is set in a rustic environment with wooden walls and a thatched roof.

Figure A8. VIDEOGAMEBUNNY provides a detailed description of the image

Qualitative results where VIDEOGAMEBUNNY provides a correct answer with minor inaccuracies



**Question:** Describe the characters in the image

**VIDEOGAMEBUNNY:** The image shows two characters on horseback. The character in the foreground is holding a lantern, and the character in the background *is following them*. Both characters are wearing hats.

Figure A9. VIDEOGAMEBUNNY provides a detailed description of the image with minor inaccuracies. Details that are incorrect are highlighted and emphasized in *red*.

Qualitative results where VIDEOGAMEBUNNY provides a correct answer with some inaccuracies



**Question:** Describe the characters in the image

**VIDEOGAMEBUNNY:** The image shows Batman, who is in the center of the frame, flying through the air. He is wearing his iconic black suit *with a yellow utility belt*. *Below him, there is a character named Penguin, who is wearing a tuxedo and a bowler hat.*

Figure A10. VIDEOGAMEBUNNY hallucinates extra characters and a Batman utility belt in the image, which are not actually present. Details that are incorrect are highlighted and emphasized in *red*.

Qualitative results where VIDEOGAMEBUNNY provide a **wrong** answer for a query



**Question:** What is written on the image?

**VIDEOGAMEBUNNY:** The text on the image reads: 'Tch... Tch... Tch... You way around it, I guess. Tch... Tch... Tch... I'll tell you, without stepping this guy away...'

Figure A11. A sample showing that VIDEOGAMEBUNNY struggles to convert the image to text.

Qualitative results where VIDEOGAMEBUNNY provide a **wrong** answer for a query



**Question:** What is written on the left?

**VIDEOGAMEBUNNY:** The word 'RELOAD' is written on the left side of the gun.

Figure A12. A sample showing that VIDEOGAMEBUNNY fails to provide a correct answer. While the word *RELOAD* is written in the center of the image, the query asks, *What is written on the left?* which should be answered with *nothing*.

## A4. Sample image-to-JSON conversion

Sample image-to-JSON conversion done by VIDEOGAMEBUNNY



```
1 {
2   "description": "The image shows a first-person perspective of a video game, likely a survival horror title. The player is holding a large,
3   → futuristic-looking rifle with a circular scope. The rifle has a white and black camouflage pattern. The player's left hand is visible, adorned with
4   → a tattoo of a skull and what appears to be a snake. The player is standing in a dark, dilapidated wooden structure, possibly a barn or shed. The
5   → walls are made of weathered wooden planks, and the floor is made of wooden planks as well. There is a doorway on the left side of the image,
6   → leading to a darker area. A small, white silhouette of a person holding a rifle is visible on the left side of the screen, indicating the player's
7   → position on a minimap. The minimap shows the player's location as a white dot within a circular area. The player's health bar is visible at the
8   → bottom of the screen, represented by a green bar with a white outline. The health bar is nearly full. The player's ammunition count is displayed in
9   → the bottom right corner of the screen, showing '0' bullets remaining in the current magazine and '41' bullets in reserve. A small icon of a skull
10  → with a knife is visible next to the ammunition count, possibly indicating a special ability or weapon. The overall atmosphere of the image is dark,
11  → gritty, and suspenseful, typical of the survival horror genre.",
12   "dialogue": [],
13   "on_screen_subtitle": "n/a",
14   "minimap_details": "The minimap shows the player's location as a white dot within a circular area. The minimap is located in the bottom left corner of
15   → the screen.",
16   "inventory_display": "The image does not show the player's full inventory.",
17   "score_or_progress": "The image does not display any score or progress indicators.",
18   "NPC_status": "The image does not provide any information about NPCs.",
19   "event_indicators": "The image does not display any specific event indicators.",
20   "interaction_prompts": "The image does not show any interaction prompts.",
21   "game_mode": "The image suggests the game is in a single-player campaign mode, based on the presence of a minimap and health bar.",
22   "HUD_description": "The HUD consists of a circular minimap in the bottom left corner, a circular health bar in the bottom left corner, and an
23   → ammunition counter in the bottom right corner. The ammunition counter displays the current magazine capacity, the remaining bullets in the
24   → magazine, and the total reserve ammunition.",
25   "on_screen_watermark": "MKiceAndFire",
26   "summary_of_ui_values": {
27     "health": "Nearly full",
28     "ammunition": "0/41"
29   },
30   "scene_description": "The scene takes place inside a dilapidated wooden structure, likely a barn or shed. The structure is dark and appears to be
31   → abandoned. The walls and floor are made of weathered wooden planks. There is a doorway on the left side of the image, leading to a darker area.",
32   "character_list": [
33     {
34       "name": "Player",
35       "appearance": "Cannot be fully determined, but the player's left hand is visible, showing a tattoo of a skull and a snake.",
36       "clothing": "Cannot be fully determined",
37       "facial_expression": "Cannot be determined"
38     }
39   ],
40   "object_list": [
41     "Rifle",
42     "Scope",
43     "Wooden planks (walls and floor)",
44     "Doorway"
45   ],
46   "texture_details": [
47     {
48       "object": "Rifle",
49       "texture": "White and black camouflage pattern"
50     },
51     {
52       "object": "Wooden planks",
53       "texture": "Weathered wooden planks"
54     }
55   ]
56 }
```

```

40     "texture": "Weathered, with visible grain and knots"
41   },
42 ],
43 "lighting_details": "The scene is dimly lit, with the primary light source appearing to be coming from the doorway on the left. This creates a sense of
→ depth and shadows within the structure.",
44 "color_palette": [
45   "#181818",
46   "#303030",
47   "#505050",
48   "#FFFFFF"
49 ],
50 "weather_conditions": "Cannot be determined",
51 "environmental_effects": "None",
52 "animation_states": "The player is holding the rifle in a ready position, suggesting a state of alertness and preparedness for combat.",
53 "error_log": "None",
54 "glitches": "None",
55 "player_status": {
56   "health": "Nearly full",
57   "equipment": "The player is equipped with a large, futuristic-looking rifle with a circular scope.",
58   "other_status": "The player has 0 bullets remaining in the current magazine and 41 bullets in reserve."
59 }
60 }

```

### Sample image-to-JSON conversion done by VIDEOGAMEBUNNY



```

1 {
2   "description": "The image shows a third-person perspective of a video game, likely Grand Theft Auto V, based on the HUD elements and graphics style.
→ The player is driving a white van on a multi-lane road. The van is positioned slightly to the right of the center of the road, heading towards the
viewer. The road is a typical city street with sidewalks, buildings, and other vehicles. The time of day appears to be midday with clear skies. The
→ HUD elements visible include a minimap in the bottom left corner, a money counter in the top right corner, and a small portion of the player's
→ health bar in the bottom right corner. The minimap shows the player's location on a simplified map of the surrounding area. The money counter
→ displays the amount of in-game currency the player has, which is 9969. The health bar is partially obscured, but it appears to be mostly full.",
3   "dialogue": [],
4   "on_screen_subtitle": "n/a",
5   "minimap_details": "The minimap shows the player's location on a simplified map of the surrounding area. The player is represented by a white arrow
→ pointing upwards, indicating their direction. The map shows roads, buildings, and a body of water, likely a river or lake.",
6   "inventory_display": "Not visible in the image.",
7   "score_or_progress": "The player's current in-game money is displayed as 9969 in the top right corner.",
8   "NPC_status": "Not applicable, as there are no visible NPCs in the image.",
9   "event_indicators": "None visible in the image.",
10  "interaction_prompts": "None visible in the image.",
11  "game_mode": "Cannot be determined from the image.",
12  "HUD_description": "The HUD consists of a minimap in the bottom left corner, a money counter in the top right corner, and a small portion of the
→ player's health bar in the bottom right corner.",
13  "on_screen_watermark": "The watermark \"Elegy Ave\" is visible in the bottom right corner.",
14  "summary_of_ui_values": {
15    "money": "9969"
16  },
17  "scene_description": "The scene depicts a typical urban environment with buildings, roads, and vehicles. The player is driving a van on a multi-lane
→ road, heading towards a large, modern-looking building with a blue light display.",
18  "character_list": [],
19  "object_list": [
20    "Van",
21  ]
}

```

```

21     "Buildings",
22     "Road",
23     "Sidewalk",
24     "Trees",
25     "Vehicles",
26     "Streetlights",
27     "Blue Light Display"
28 ],
29 "texture_details": [
30   {
31     "object": "Buildings",
32     "texture": "Brick, concrete, glass"
33   },
34   {
35     "object": "Road",
36     "texture": "Asphalt"
37   },
38   {
39     "object": "Sidewalk",
40     "texture": "Concrete"
41   }
42 ],
43 "lighting_details": "The scene is brightly lit, suggesting midday sun. The lighting casts soft shadows from the buildings and trees onto the road.",
44 "color_palette": [
45   "#87CEEB",
46   "#FFFFFF",
47   "#A9A9A9",
48   "#696969",
49   "#B0B0B0"
50 ],
51 "weather_conditions": "Clear and sunny.",
52 "environmental_effects": "None visible in the image.",
53 "animation_states": "The van's wheels suggest it is in motion, and the other vehicles on the road imply traffic.",
54 "error_log": "None visible in the image.",
55 "glitches": "None visible in the image.",
56 "player_status": {
57   "health": "Partially obscured, but appears to be mostly full.",
58   "equipment": "Not visible in the image.",
59   "other_status": "The player has 9969 in-game currency."
60 }
61 }

```

### Sample image-to-JSON conversion done by VIDEOGAMEBUNNY



```

1 {
2   "description": "The image captures a scene from the video game Red Dead Redemption 2, showcasing the protagonist, Arthur Morgan, riding his horse,
2   ↪ Black Jack, along a dirt path. The path leads towards a small, industrial settlement built on a cliffside. The settlement features wooden buildings
2   ↪ with corrugated metal roofs, characteristic of the late 19th-century American frontier. A plume of black smoke rises from a chimney in the
2   ↪ background, suggesting industrial activity. The surrounding environment is a mix of rocky terrain and lush vegetation, with a dense forest visible
2   ↪ in the distance. The sky is overcast, with a hazy, diffused light suggesting either early morning or late afternoon. The overall atmosphere is one
2   ↪ of quiet anticipation, as if Arthur is about to enter a new chapter in his journey.",
3   "dialogue": [],
4   "on_screen_subtitle": "n/a",

```

```

5   "minimap_details": "n/a",
6   "inventory_display": "n/a",
7   "score_or_progress": "n/a",
8   "NPC_status": "n/a",
9   "event_indicators": "n/a",
10  "interaction_prompts": "n/a",
11  "game_mode": "n/a",
12  "HUD_description": "n/a",
13  "on_screen_watermark": "n/a",
14  "summary_of_ui_values": {},
15  "scene_description": "Arthur Morgan riding his horse Black Jack towards a small industrial settlement on a cliffside.",
16  "character_list": [
17    {
18      "name": "Arthur Morgan",
19      "appearance": "A rugged man with a beard and mustache, wearing a dark hat and a dark coat.",
20      "clothing": "Dark hat, dark coat, dark pants, and boots.",
21      "facial_expression": "Cannot be determined as his back is to the camera."
22    }
23  ],
24  "object_list": [
25    "Horse",
26    "Dirt path",
27    "Wooden buildings",
28    "Corrugated metal roofs",
29    "Smoke plume",
30    "Chimney",
31    "Forest",
32    "Rocks",
33    "Vegetation",
34    "Trees"
35  ],
36  "texture_details": [
37    {
38      "object": "Horse",
39      "texture": "Short, dark brown fur"
40    },
41    {
42      "object": "Dirt path",
43      "texture": "Uneven, dusty, with patches of grass"
44    },
45    {
46      "object": "Wooden buildings",
47      "texture": "Weathered wood planks"
48    },
49    {
50      "object": "Corrugated metal roofs",
51      "texture": "Rusted and dented metal sheets"
52    },
53    {
54      "object": "Rocks",
55      "texture": "Rough, jagged, with moss and lichen"
56    },
57    {
58      "object": "Vegetation",
59      "texture": "A mix of grass, shrubs, and trees"
60    }
61  ],
62  "lighting_details": "The scene is lit by diffused sunlight, suggesting an overcast sky. The light is soft and even, with no strong shadows.",
63  "color_palette": [
64    "#A0A0A0",
65    "#808080",
66    "#606060",
67    "#404040",
68    "#202020",
69    "#008000",
70    "#006400",
71    "#004000"
72  ],
73  "weather_conditions": "Overcast",
74  "environmental_effects": "Smoke plume rising from a chimney",
75  "animation_states": "Arthur Morgan is sitting upright on his horse, suggesting a calm and controlled ride.",
76  "error_log": "None",
77  "glitches": "None",
78  "player_status": [
79    "health": "n/a",
80    "equipment": "n/a",
81    "other_status": "n/a"
82  ]
83 }

```

## Sample image-to-JSON conversion done by VIDEOGAMEBUNNY



```
1 {
2   "description": "The image showcases a close-up of a man in a World War II-era military uniform. He is positioned slightly to the right of the center, with his gaze directed slightly upwards and to the left. His expression is serious, bordering on somber. He has short, dark hair and a prominent nose. The man's uniform is a muted green, with a noticeable dirt smudge on his left cheek. A dark green helmet with a netting pattern covers his head, partially obscuring his hair. The helmet's chin strap is visible, fastened with a metal buckle. A thick, dark green strap crosses his chest, likely part of a backpack or equipment harness. The background is a blurred depiction of a battlefield, with other soldiers in similar uniforms visible in the distance. The overall color palette is muted and desaturated, reflecting the grim atmosphere of war.",
3   "dialogue": [],
4   "on_screen_subtitle": "n/a",
5   "minimap_details": "n/a",
6   "inventory_display": "n/a",
7   "score_or_progress": "n/a",
8   "NPC_status": "n/a",
9   "event_indicators": "n/a",
10  "interaction_prompts": "n/a",
11  "game_mode": "n/a",
12  "HUD_description": "n/a",
13  "on_screen_watermark": "n/a",
14  "summary_of_ui_values": {},
15  "scene_description": "The scene depicts a soldier in the midst of a battlefield, likely during World War II, given the uniforms and equipment.",
16  "character_list": [
17    {
18      "name": "Unknown",
19      "appearance": "The man appears to be in his late twenties to early thirties. He has a rugged appearance with short, dark hair and a prominent nose.",
20      "clothing": "He is wearing a World War II-era military uniform in a muted green color. A dark green helmet with a netting pattern covers his head, and a thick, dark green strap crosses his chest, likely part of a backpack or equipment harness.",
21      "facial_expression": "His expression is serious and slightly somber, suggesting he is focused on the situation at hand."
22    }
23  ],
24  "object_list": [
25    "Military uniform",
26    "Helmet",
27    "Backpack strap",
28    "Backpack (partially visible)",
29    "Soldiers (in the background)"
30  ],
31  "texture_details": [
32    {
33      "object": "Helmet",
34      "texture": "Netting pattern"
35    },
36    {
37      "object": "Uniform",
38      "texture": "Camouflage pattern"
39    }
40  ],
41  "lighting_details": "The lighting is soft and diffused, likely due to overcast skies or fog. It casts subtle shadows on the man's face, accentuating his features.",
42  "color_palette": [
43    "#5C6B5A",
44    "#3D4A3B",
45    "#7A8A76",
46    "#B2B4A9"
47  ],
48  "weather_conditions": "The weather appears to be overcast or foggy, as evidenced by the soft lighting and lack of harsh shadows.",
```

```

49 "environmental_effects": "The background suggests a battlefield, with smoke or dust particles visible in the air.",
50 "animation_states": "The man appears to be in a still pose, likely captured in a moment of pause or reflection during the chaos of battle.",
51 "error_log": "None",
52 "glitches": "None",
53 "player_status": {
54   "health": "n/a",
55   "equipment": "n/a",
56   "other_status": "n/a"
57 }
58 }

```

### Sample image-to-JSON conversion done by VIDEOGAMEBUNNY



```

1 {
2   "description": "The image shows a male character in a video game, seemingly in the middle of a climbing or jumping action. He is positioned on a stone
3   ↪ platform within a dimly lit, ancient-looking structure. The platform is part of a larger architectural complex, with visible stone walls and a
4   ↪ barred gate in the background. The character is dressed in a sleeveless top and pants, suggesting a warrior or adventurer archetype. He is equipped
5   ↪ with a sword sheathed on his back and appears to be in motion, with his body angled towards the platform's edge. The overall atmosphere is one of
6   ↪ adventure and exploration, with the character's pose hinting at a moment of action or transition within the game.",
7   "dialogue": [],
8   "on_screen_subtitle": "n/a",
9   "minimap_details": "Cannot be determined",
10  "inventory_display": "Not visible",
11  "score_or_progress": "Not visible",
12  "NPC_status": "No NPCs present",
13  "event_indicators": "None visible",
14  "interaction_prompts": "None visible",
15  "game_mode": "Cannot be determined",
16  "HUD_description": "n/a",
17  "on_screen_watermark": "SHIRRAKO in the bottom right corner",
18  "summary_of_ui_values": {},
19  "scene_description": "The scene is set within a large, ancient-looking structure, possibly a temple or palace. The architecture suggests a Middle
20  ↪ Eastern or Mediterranean influence. The environment is dimly lit, with the primary light source appearing to be natural light filtering in from an
21  ↪ unseen opening above.",
22  "character_list": [
23    {
24      "name": "Unknown",
25      "appearance": "Male, athletic build",
26      "clothing": "Sleeveless top, pants, possibly leather or fabric",
27      "facial_expression": "Cannot be determined"
28    }
29  ],
30  "object_list": [
31    "Stone platform",
32    "Stone walls",
33    "Barred gate",
34    "Sword",
35    "Pottery jug"
36  ],
37  "texture_details": [
38    {
39      "object": "Stone walls",
40      "texture": "Rough, uneven stone with visible cracks and weathering"
41    }
42  ]
}

```

```

35     },
36     {
37       "object": "Stone platform",
38       "texture": "Similar to the walls, but with a slightly smoother appearance"
39     }
40   ],
41   "lighting_details": "The scene is dimly lit, with the primary light source appearing to be natural light coming from an unseen opening above. This
42   ↪ creates a sense of depth and highlights the textures of the stone surfaces.",
43   "color_palette": [
44     "#5C4A3A",
45     "#3B322A",
46     "#7A6D5B",
47     "#A1887F"
48   ],
49   "weather_conditions": "Cannot be determined",
50   "environmental_effects": "None visible",
51   "animation_states": "The character's pose suggests a dynamic action, either in the middle of a jump or a climbing maneuver.",
52   "error_log": "None visible",
53   "glitches": "None",
54   "player_status": {
55     "health": "Not visible",
56     "equipment": "Sword visible on back",
57     "other_status": "Not visible"
58   }

```

---