

A. Fewer Samples for PA-ICVL

We demonstrate how the number of samples N for PA-ICVL impacts on results. Including 5 samples for each class used in the main experiments, we gradually reduced the number of samples to 1 and 3. When we applied in-context learning, we saw that different values of N resulted in different detection results, which is shown in Tab. 3

Model (D-5)	$N = 1$	$N = 3$	$N = 5$
GPT-4 Vision	71%	73%	78%
Gemini 1.5 Pro	72%	73%	80%

Table 3. Results of final model according to the number of N .

B. Results on Image Transformation

Based on the appearance of our dataset, it can be inferred that VLM may focus on the leg or arm regions. Thus, detecting hallucination might be easy in our settings. To address this conjecture, we separately evaluate the hallucination detection performance of two VLMs by flipping and rotating images. The evaluation results for character images flipped horizontally relative to the base model (D5) did not differ significantly. When evaluated using character images rotated 0.5π , there was a significant difference in accuracy. The evaluation results are shown in Tab. 1. This gap could mean that the VLM does not recognise rotated character images correctly, as opposed to forward facing characters.

Model (D-5)	Base	Horizontal-Flip	0.5π Rotation
GPT-4 Vision	78%	76%	54%
Gemini 1.5 Pro	80%	77%	61%

Table 4. Results according to image transformation

C. Visual Hallucination in Cartoon Domain

Cartoon domain has unique appearance. The cases and level of visual hallucination are quite different from realistic domain. When we generate many cartoon image from TTI, we found that there are two classes about hallucination tendency as shown in Fig. 11: one is uncompleted whole-body having one arm, one leg or even no head as shown in Fig. 11(a), other one is over-depiction of body components such as three arms, three legs as shown in Fig. 11(b). These hallucination types led us utilize pose estimation for visual hallucination detection in cartoon image.



Figure 11. Hallucination classes in cartoon domain.

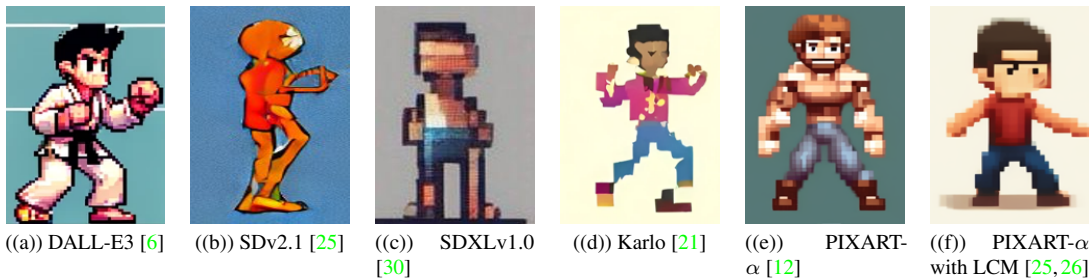


Figure 12. Comparison for cartoon rendering quality with SOTA TTI models

D. Cartoon Rendering Comparison with Various Large TTI Models

We evaluate TTI models which include DALL-E3 [6], SDv2.1⁵ [25], SDXLv1.0⁶ [30], Karlo⁷ [21], PIXART- α ⁸ [12], PIXART- α with LCM⁹. For comparison in terms of cartoon rendering quality, we used same input prompt for every TTI models. For Karlo, the input prompts we used were also converted to combinations of words because, in Karlo website, we found that user typically used word format rather than sentences.

As shown in Fig. 12, we found that all the models generate non-plausible appearance for cartoon-pixel character except DALL-E3. Due to that, we used DALL-E3 for our TTI model.

E. ChatGPT-4 Vision vs DALL-E3 API

We found that there are some gaps about appearance tendency between ChatGPT-4 Vision through ChatGPT site¹⁰ and pure DALL-E3 API¹¹. We conducted experiments by feeding the TTI prompts into both ChatGPT-4 Vision and the DALL-E3 API to generate images. Empirically, we observed that ChatGPT-4 Vision created images that were closer to our desired output with clean apparent structure compared to DALL-E3 API. We conjecture that this result is derived from ChatGPT’s capability to refine and analyze the provided prompt, which in turn elevates its comprehension of the prompt, leading to the production of superior images.

F. Non-Human-like Cartoon Character

We would like to extend the applicability of ours to non-human-like cartoon characters. To do so, we wanted to evaluate these samples by finding suitable input prompts for them. However, it was observed that the pose estimator was unable to extract accurate pose joints as shown in Fig. 13. Thus we cannot conduct evaluations with non-human-like character under the conclusion that pose guidance would not provide useful information to VLMs in PA-ICVL step.

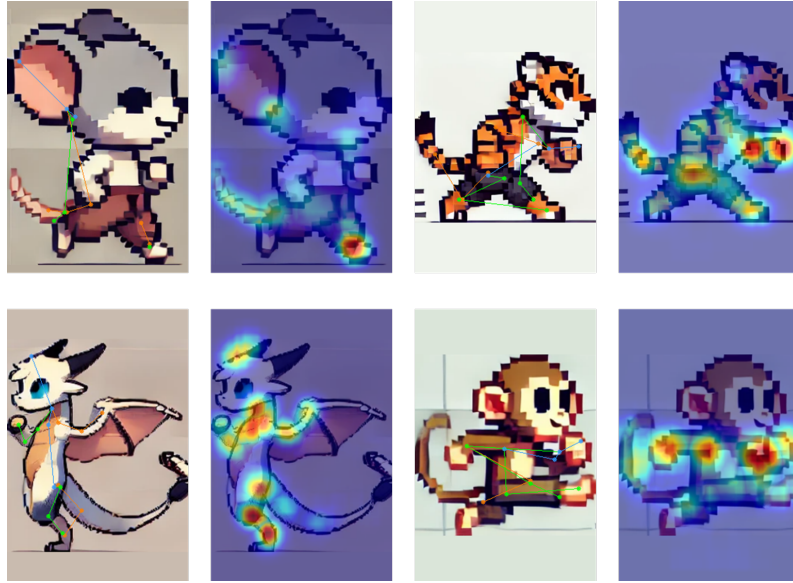


Figure 13. Failed pose estimation on non-human-like character.

⁵<https://huggingface.co/spaces/stabilityai/stable-diffusion>

⁶<https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0>

⁷<https://karlo.ai/>

⁸<https://huggingface.co/spaces/PixArt-alpha/PixArt-alpha>

⁹<https://huggingface.co/spaces/PixArt-alpha/PixArt-LCM>

¹⁰<https://openai.com/gpt-4>

¹¹<https://openai.com/dall-e-3>

G. Pose Estimator

Here, we provide some details for our pose estimator including fine-tuning scheme and comparison on other off-the-shelf pose estimators.

G.1. Comparison on Various Pose Estimators

To use visually precise pose joint about cartoon character to VLM, we used our fine-tuned pose estimator (see Appendices G.2 for details of fine-tuning). Here, we show the pose estimation performance with off-the-shelf pose estimators which include PoseAnything [14], OpenPose [10]. For PoseAnything, we used 1-Shot split-1 small model from official repository¹². For OpenPose, we used pose_iter_160000.caffemodel for MPII format from official repository¹³.



Figure 14. Comparison of pose estimation.

As shown in Fig. 14, we found that there is no one which can predict perceptually plausible joint. This is conjectured that cartoon domain distributed far away from pre-trained weights, leading requirement of fine-tuning with this domain.

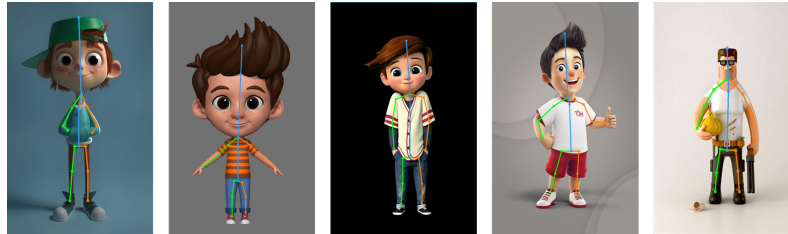


Figure 15. Example of training dataset for pose estimation fine-tuning.

G.2. Fine-tuning

We fine-tune the pose estimator based on HRNet-w48 [35] (Top-down approach) upon MMPose library [27] with collected 3D cartoon dataset as shown in Fig. 15. Our training scheme is based on [9]. For pose estimation settings, we used MPII-TRB [13] keypoint format (16 joint for whole-body) with inference image size as 384 by 256 using boundary padding. Including 2D illustration and rendered 2D images from 3D model shape, totalling 2400 images in animation, illustration and cartoon domains are used for fine-tuning for about 16K iterations with 32 batch size, achieving 0.8902 PCKh (Percentage of Correct Key-points head) [42] at threshold 0.5.

H. Used prompts

H.1. TTI Input prompt list

1. Please design a 2D motion frame pixel style character with a size of 256x384 pixels. Each action should be displayed on a separate row, with the first row being a {kicking, punching, jumping, running, walking .. etc} action (composed of 5 frames) and the second row being a {kicking, punching, jumping, running, walking .. etc} action (composed of 5 frames) that appears

¹²<https://github.com/orhir/PoseAnything>

¹³<https://github.com/CMU-Perceptual-Computing-Lab/openpose>

to be smoothly connected. The generated actions should not overlap, and the character's color should be simple. The entire sprite sheet should be 1792x1024 in size.

2. (Without detailed color & overlap prompt) Please design a 2D pixel style character with a size of 256x384 pixels. Each action should be displayed on a separate row, with the first row being a {kicking, punching, jumping, running, walking .. etc} action (composed of 5 frames) and the second row being a {kicking, punching, jumping, running, walking .. etc} action (composed of 5 frames) that appears to be smoothly connected. The entire sprite sheet should be 1792x1024 in size.

3. (Without pixel size prompt) Please design a 2D motion frame pixel style character. Each action should be displayed on a separate row, with the first row being a kicking, punching, jumping, running, walking .. etc} action and the second row being a kicking, punching, jumping, running, walking .. etc} action that appears to be smoothly connected. The generated actions should not overlap, and the character's color should be simple. The entire sprite sheet should be 1792x1024 in size.

H.2. Instruction prompt list

System Prompt & Hallucination Definition : You are a hallucination detector, and your mission is to detect if the image has hallucinations. Here I define hallucination as when a character is missing an arm, leg, or has an abnormal number of them (three legs, three arms .. etc). So you need to detect the hallucination I defined in the image and visually describe the hallucination. So as a sample of how to detect this well, we'll provide the following prompts with a hallucination image, a normal image {and joint image, joint file, heatmap image}.

1. **(Model C)** Using RGB image - Correct class : This character is performing a {kicking, punching, jumping, running, walking .. etc} motion with an image of a correct human body with two arms and two legs. This image of a correct human anatomy will be classified as C (correct class) in the future. Your task is to recognize images with correct human anatomy as C images.

2. **(Model C)** Using RGB image - Hallucination class : This character is performing a {kicking, punching, jumping, running, walking .. etc} motion with an abnormal character with {three legs, three arms, no head, no arms, no legs, only one arm, only one leg}. This image of abnormal human anatomy will be classified as H (hallucination class) in the future. Your task is to recognize images with abnormal human anatomy as H images.

3. **(Model D-1)** Using RGB & Gaussian heatmap image - Correct class : This image is a pose heatmap obtained from the pose estimator. This character is performing a {kicking, punching, jumping, running, walking .. etc} motion with an image of a correct human body with two arms and two legs. This image of a correct human anatomy will be classified as C (correct class) in the future. Your task is to recognize images with correct human anatomy as C images.

4. **(Model D-1)** Using RGB & Gaussian heatmap image - Hallucination class : This image is a pose heatmap obtained from the pose estimator. This character is performing a {kicking, punching, jumping, running, walking .. etc} with {three legs, three arms, no head, no arms, no legs, only one arm, only one leg}. This image of abnormal human anatomy will be classified as H (hallucination class) in the future. Your task is to recognize images with abnormal human anatomy as H images.

5. **(Model D-2)** Using Overlapped heatmap image - Correct class : This image is a pose heatmap obtained from the pose estimator. This character is performing a {kicking, punching, jumping, running, walking .. etc} motion with an image of a correct human body with two arms and two legs. This image of a correct human anatomy will be classified as C (correct class) in the future. Your task is to recognize images with correct human anatomy as C images.

6. **(Model D-2)** Using Overlapped heatmap image - Hallucination class : This image is a pose heatmap obtained from the pose estimator. This character is performing a {kicking, punching, jumping, running, walking .. etc} with {three legs, three arms, no head, no arms, no legs, only one arm, only one leg}. This image of abnormal human anatomy will be classified as H (hallucination class) in the future. Your task is to recognize images with abnormal human anatomy as H images.

7. **(Model D-3)** Using RGB & overlapped heatmap image - Correct class : The first image is an RGB image of the character and the second image is a heatmap of the character's pose using the pose estimator. This character is performing a {kicking, punching, jumping, running, walking .. etc} motion with an image of a normal human body with two arms and two legs. This image of a normal human anatomy will be classified as C (correct class) in the future. Your task is to recognize images with

normal human anatomy as C images.

8. **(Model D-3)** Using RGB & overlapped heatmap image - Hallucination class : The first image is an RGB image of the character and the second image is a heatmap of the character's pose using the pose estimator. This character is performing a {kicking, punching, jumping, running, walking .. etc} motion with {three legs, three arms, no head, no arms, no legs, only one arm, only one leg}. This image of abnormal human anatomy will be classified as H (hallucination class) in the future. Your task is to recognize images with abnormal human anatomy as H images.

9. **(Model D-4)** Using RGB image & Joint (image) - Correct class : The first image is an RGB image of the character and the second image is a keypoint of the character's pose using the pose estimator. This character is performing a {kicking, punching, jumping, running, walking .. etc} motion with an image of a correct human body with two arms and two legs. This image of a correct human anatomy will be classified as C (correct class) in the future. Your task is to recognize images with correct human anatomy as C images.

10. **(Model D-4)** Using RGB image & Joint (image) - Hallucination class : The first image is an RGB image of the character and the second image is a keypoint of the character's pose using the pose estimator. This character is performing a {kicking, punching, jumping, running, walking .. etc} motion with an abnormal character with {three legs, three arms, no head, no arms, no legs, only one arm, only one leg}. This image of abnormal human anatomy will be classified as H (hallucination class) in the future. Your task is to recognize images with abnormal human anatomy as H images.

11. **(Model D-5)** Using RGB image & Joint (text) - Correct class : The first image is an RGB image of the character and the second file is a keypoint of the character's pose using the pose estimator. This character is performing a {kicking, punching, jumping, running, walking .. etc} motion with an image of a correct human body with two arms and two legs. This image of a correct human anatomy will be classified as C (correct class) in the future. Your task is to recognize images with correct human anatomy as C images.

12. **(Model D-5)** Using RGB image & Joint (text) - Hallucination class : The first image is an RGB image of the character and the second file is a keypoint of the character's pose using the pose estimator. This character is performing a {kicking, punching, jumping, running, walking .. etc} motion with an abnormal character with {three legs, three arms, no head, no arms, no legs, only one arm, only one leg}. This image of abnormal human anatomy will be classified as H (hallucination class) in the future. Your task is to recognize images with abnormal human anatomy as H images.