ChatHuman: Chatting about 3D Humans with Tools

Supplementary Material

Contents

1. Additional Method Details **1**
1.1. Tool Results Integration 1
1.2. Model Training 1
1.3. Training Data Construction 1

2. Evaluation Metric and Benchmark Details **2**
2.1. Evaluation Metric 2
2.2. Benchmarks 3

3. Ethnical Statement **3**

4. Additional Results **4**
4.1. RAG Qualitative Results. 4
4.2. GPT4Tool and Visual ChatGPT-4 4
4.3. K-fold Validation of Tool Usage 4
4.4. GPT4 replaces LLaVA for comparison . . . 4
4.5. Paper Components. 4
4.6. Base Model for Value Extraction. 5
4.7. Text Embedding Model in Retrieval. 5
4.8. Tool Use in Multi-turn Dialogue. 5
4.9. Tool Graph Construction. 5

As promised in the main paper, we provide additional details here on the method for integrating tool results and the model training process. This is followed by an explanation of the training data construction pipeline. We also present further details about the tool evaluation metric, prompts for body shape measurement data, and the MixPose benchmark. Furthermore, we show more ablation studies.

1. Additional Method Details

1.1. Tool Results Integration

To utilize the tool results and improve the LLMs' understanding of 3D humans, which, in turn, enhances the LLMs' ability to apply its world knowledge to problem-solving - we introduce a tool-conditioned transformation $\Psi(\cdot)$. As shown in Figure S1, this transformation converts the varied tool outcomes Y_m into textual or visual formats that the LLM can process more easily. For example, we transform the vertex-wise contact label predicted by DECO [17] into a body part-level description based on the vertex-to-part mapping dictionary of SMPL [11], and we render the mesh generated by PoseScipt [4] into an RGB image using rendering techniques.

1.2. Model Training

ChatHuman comprises a multimodal LLM $f_\phi(\cdot)$, along with a set of 3D human-related functions. During training, the tool functions are kept fixed, and only the LLM $f_\phi(\cdot)$ is finetuned using instruction-following data. Specifically, we employ LoRA [7] with a rank of 128 and an alpha value of 256 to finetune the LLM. The trainable parameters in this setup are represented as ϕ_{lora}. Given a user query X_q, the model generates a textual description of the tool invocation Y_{tool} and a final textual response Y_t after integrating the tool results. With the ground truth tool invocation labels $\hat{Y}_{tool}$ and response label $\hat{Y}_t$, we optimize the model using the following objective function: $\mathcal{L} = \mathbf{CE}(\hat{Y}_{\text{tool}}, Y_{\text{tool}}) + \mathbf{CE}(\hat{Y}_t, Y_t)$, where $\mathbf{CE}$ denotes the cross-entropy loss.

1.3. Training Data Construction

Tool Usage Instruction-following Data. To teach the LLM-based agent to correctly use tools, we construct 90K instruction-response pairs about tool usage. Following GPT4Tools [20], we provide GPT-4 [13] with a textual description of an image from the COCO training set [9] and a tool-related prompt containing a tool description. One of our key observations is that human-related tools often come with an academic paper containing rich background knowledge and varied applications, which are useful for the generation of user queries covering a wide range of application scenarios. Thus, we also incorporate the paper content into GPT-4 to generate the tool usage instruction-following data. To improve efficiency, we first prompt GPT-4 to summarize the paper content, re-articulate the tool functions and enumerate 50 potential user queries for tool activation (see main paper Fig. 6 (a)). The details of the prompt are represented in Table S11. The summarized tool description and user queries are fed to GPT-4 along with the image description to generate the instruction-following data about tool usage. Table S12 illustrates the prompt for the second step.

Tool Feedback Instruction-following Data. To help the multimodal LLM model discriminate and integrate the tool results, we construct 88K pairs of instruction-following data based on existing 3D human datasets.

- **Pose Estimation Results Discrimination.** To teach the LLM-based model to discriminate the pose estimation results from different tools, we build 17K pairs of instruction-following data based on the 3DPW [19] and MOYO [18] training sets. Specifically, we use HMR2.0 [6] and CLIFF-SMPLify [2, 8] to predict the human mesh and calculate the reconstruction error between the pre-

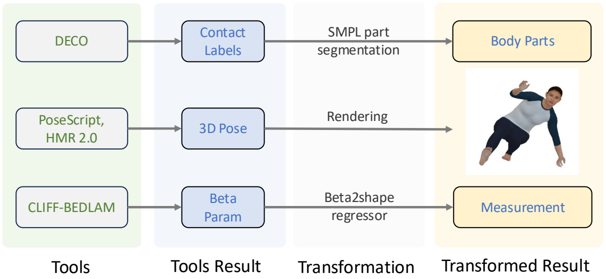


Figure S1. Illustration of tool-conditioned transformation process. We convert the varied tool outcomes into textual or visual formats that the LLMs can more readily process.

dicted mesh and ground truth mesh. Based on MPVPE, we determine which tool is better for each image and construct instruction-following data as shown in main paper Figure 6(b). Pose visualization results are rendered with Pyrender [12].

- **Pose Generation Results Discrimination.** The human pose generation tool, PoseScript [4], has multiple outcomes for each text input. Here we construct 44K pairs of instruction-following data to teach the multimodal LLM-based model to discriminate the multiple pose generation results. Specifically, we use PoseScript training data as the source and construct the data in two formats. The first one is about text-to-pose selection, as shown in Figure S2(a). Given a textual description, we visualize the corresponding pose and three other different poses from the training data and ask the agent to discriminate and choose the one that best aligns with the textual description. The second one is about pose-to-text matching, as shown in Figure S2(b). Given a 3D pose, we visualize it as an image by rendering the 3D body mesh in that pose. Then, we combine it with the corresponding text description and three other pose descriptions in the format of a multiple choice question. Finally, we ask the agent to choose the one that best describes the pose shown in the image.
- **Human Contact Detection Results Integration.** The outcome of the human contact prediction tool, DECO [17], is a vertex-wise contact prediction in a vector representation $y_c \in \mathbb{R}^{6890\times 1}$, which can not be directly used as input for our multimodal LLM baseline, LLaVA. To solve this problem, we transform the vertex-wise contact label of ground-truth and DECO's result into a textual description based on the vertex-to-part mapping dictionary of the SMPL model [11]. Subsequently, we feed the textual descriptions along with the RGB image from the DECO training set [17] into GPT-4V and prompt GPT4 [13] to generate instruction-following data about human-object interaction as shown in Figure S4. Notably, the transformed tool result is merged with the user query as a clue. The details of the prompt are shown in Table S13.
- **Body Shape Measurement Integration.** Similar to human contact prediction, the outcome of the body shape measurement tool is the SMPL body shape parameter $\beta \in \mathbb{R}^{10}$, which is also in a vector representation and can not be used by the LLM directly. Thus, we first convert the shape parameter into measurements based on the shape-to-measurement module from SHAPY [3] and represent it in a textual format. Subsequently, we feed the body measurement description along with attribute labels from the SHAPY training set into GPT-4 and prompt it to generate instruction-following data about human body shape as shown in Figure S3. Similarly, we merge the body measurement predicted by the tool with the user query as a clue. The prompt for GPT-4 is detailed in Table S14.

2. Evaluation Metric and Benchmark Details

2.1. Evaluation Metric

Tool Usage. We use the metrics proposed in GPT4Tools [20] to measure the tool usage accuracy, including:

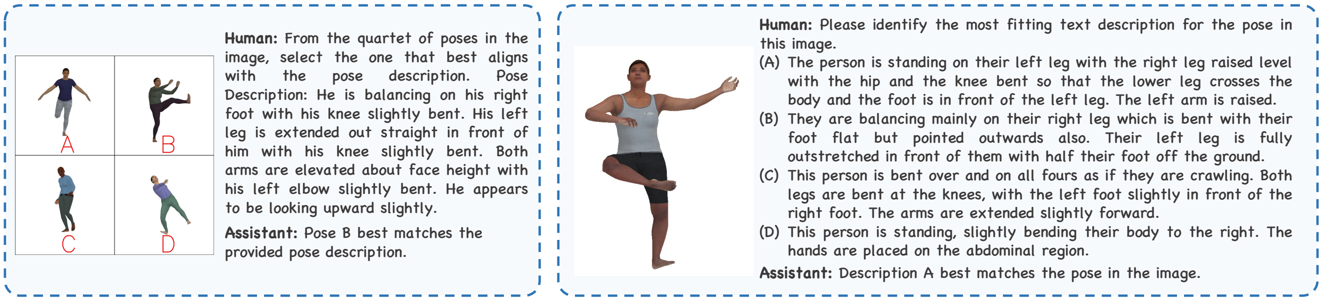
- **Successful Rate of Thought** ($\mathrm{SR_t}$), which measures the decision accuracy, calculated as $\mathrm{SR_t} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\tau_i)$, where N is the number of instructions and τ_i is a singular process. When the predicted thought is the same as the ground-truth thought, $\mathbb{I}(\tau_i)$ is equal to 1, and 0 otherwise.
- **Successful Rate of Action** ($\mathrm{SR_{act}}$), which measures the tool name prediction accuracy, calculated as $\mathrm{SR_{act}} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\alpha_i)$, where α_i is the matching process of the tool name. If the predicted tool name is correct, $\mathbb{I}(\alpha_i)$ is equal to 1, and 0 otherwise.
- **Successful Rate of Arguments** ($\mathrm{SR_{args}}$), which measures the tool arguments prediction accuracy, calculated as:

$$\mathrm{SR_{args}} = \frac{1}{N}\sum_{i=1}^{N}\eta_i, \quad \eta_i = \frac{1}{K}\sum_{j=1}^{K}\eta_{i,j}, \tag{1}$$

  where K is the number of tool arguments. When the argument is a file name, $\eta_{i,j}$ equals 1 if the predicted file name is the same as the ground-truth file name, and 0 otherwise. When the argument is text, $\eta_{i,j}$ equals the BLEU score between the predicted and ground-truth text.
- **Intersection over Union** (IoU), which quantifies the percent overlap between the predicted text and ground-truth text.

Human Understanding. We use the following evaluation metrics to measure the performance of ChatHuman in human-related tasks:

- **Pose Estimation.** We adopt the same evaluation metrics as ChatPose [5] to evaluate the 3D pose estimation accuracy, including Mean Per-Joint Position Error (MPJPE), Mean Per-Joint Position Error after Procrustes alignment (PA-MPJPE), Mean Per-Joint Rotation Error (MPJRE), and Mean Per-Vertex Position Error (MPVPE).



Human: From the quartet of poses in the image, select the one that best aligns with the pose description. Pose Description: He is balancing on his right foot with his knee slightly bent. His left leg is extended out straight in front of him with his knee slightly bent. Both arms are elevated about face height with his left elbow slightly bent. He appears to be looking upward slightly.

Assistant: Pose B best matches the provided pose description.

Human: Please identify the most fitting text description for the pose in this image.

(A) The person is standing on their left leg with the right leg raised level with the hip and the knee bent so that the lower leg crosses the body and the foot is in front of the left leg. The left arm is raised.

(B) They are balancing mainly on their right leg which is bent with their foot flat but pointed outwards also. Their left leg is fully outstretched in front of them with half their foot off the ground.

(C) This person is bent over and on all fours as if they are crawling. Both legs are bent at the knees, with the left foot slightly in front of the right foot. The arms are extended slightly forward.

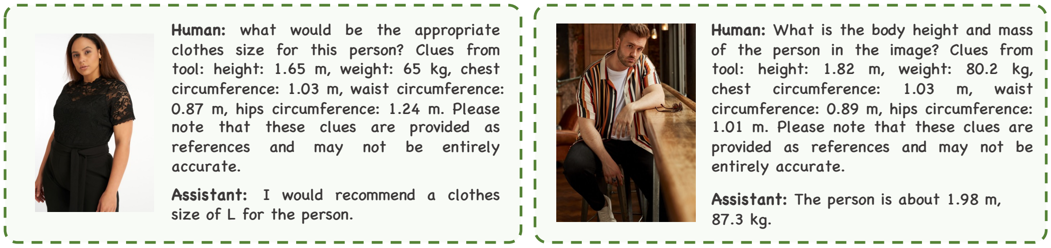
(D) This person is standing, slightly bending their body to the right. The hands are placed on the abdominal region.

Assistant: Description A best matches the pose in the image.

(a) Text-to-Pose Selection

(b) Pose-to-Text Selection

Figure S2. Examples of the instruction-following data for discriminating pose generation and pose description results.



Human: what would be the appropriate clothes size for this person? Clues from tool: height: 1.65 m, weight: 65 kg, chest circumference: 1.03 m, waist circumference: 0.87 m, hips circumference: 1.24 m. Please note that these clues are provided as references and may not be entirely accurate.

Assistant: I would recommend a clothes size of L for the person.

Human: What is the body height and mass of the person in the image? Clues from tool: height: 1.82 m, weight: 80.2 kg, chest circumference: 1.03 m, waist circumference: 0.89 m, hips circumference: 1.01 m. Please note that these clues are provided as references and may not be entirely accurate.

Assistant: The person is about 1.98 m, 87.3 kg.

Figure S3. Instruction-following data for integrating results from human body estimation. Measurements of the estimated body shape from BEDLAM-CLIFF [1] are added to the user query as clues.

- **Pose Generation.** We use the evaluation metrics established in PoseScript [4], including the text-to-pose recall rate R^{P2T} and pose-to-text recall rate R^{T2P} of the retrieval models trained on real poses and evaluated on generated poses. We use the retrieval model from the jounal-version of PoseScript [4] and ChatPose [5] for the classical pose generation and speculative pose generation tasks, respectively.

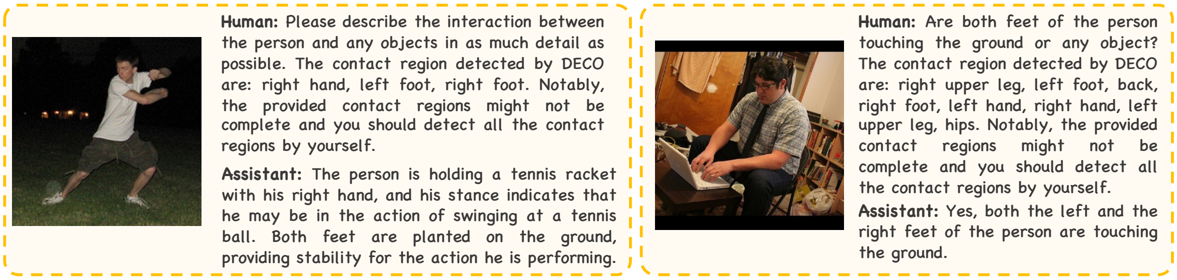
2.2. Benchmarks

Tool Usage Benchmark. To evaluate the tool usage accuracy of our method, we construct a validation and test set. The validation set has 1000 samples with the same tools as the training set, while the test set includes 689 samples related to 3 tools unseen during training. Similar to the training data construction, we feed a textual description of an image from the COCO validation set, a tool description, and some examples summarized from the tool paper into GPT-4 and prompt it to generate instruction-following data about tool usage. We use the image description captioned by LLaVA [10] instead of the original image captions to ensure a difference between training and test sets. Finally, we manually check the question-answering pairs to ensure the accuracy of the benchmark.

MixPose Benchmark. To validate whether the multimodal LLM-based agent can discriminate the pose estimation results from different tools, we build a new benchmark, MixPose. Considering that different tools excel in different scenarios, a benchmark covering diverse scenarios and corner cases is needed. To construct this benchmark, we selected 100 images featuring extreme camera angles from the MoYo [18] test set, 100 full-body images from the 3DPW test set, and 100 images with significant truncation also from the 3DPW test set. This approach ensures our benchmark includes typical in-the-wild images, shots taken from extreme views, and images with heavy truncation. This diverse distribution allows us to test whether the agent can accurately choose the right tool based on sceneries of the image and the tool performance. To get the truncated images from 3DPW, we resize the human bounding box by 2/3 and crop the human image based on the rescaled human bounding box.

3. Ethnical Statement

Our work, ChatHuman, explores utilizing tools for human-centric tasks. All 3D human data and tools used is publicly available intended for research. No identifiable or biometric data is used. We acknowledge the potential risks of deploying such systems in real-world applications, including surveillance, biased inferences, or non-consensual modeling. We strongly discourage such uses. We encourage future users of our system and dataset to consider fairness, inclusivity, and transparency in their work. Our code and data are released under licenses that restrict commercial use and require proper attribution, to encourage responsible and ethical research.



Human: Please describe the interaction between the person and any objects in as much detail as possible. The contact region detected by DECO are: right hand, left foot, right foot. Notably, the provided contact regions might not be complete and you should detect all the contact regions by yourself.

Assistant: The person is holding a tennis racket with his right hand, and his stance indicates that he may be in the action of swinging at a tennis ball. Both feet are planted on the ground, providing stability for the action he is performing.

Human: Are both feet of the person touching the ground or any object? The contact region detected by DECO are: right upper leg, left foot, back, right foot, left hand, right hand, left upper leg, hips. Notably, the provided contact regions might not be complete and you should detect all the contact regions by yourself.

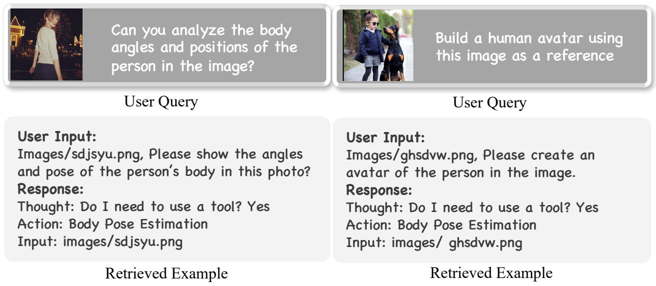
Assistant: Yes, both the left and the right feet of the person are touching the ground.

Figure S4. Instruction-following data about integrating results from human contact detection. The contact labels detected by DECO [17] are combined with the user query.

4. Additional Results

4.1. RAG Qualitative Results.

As mentioned in main paper Sec. 3.2, many tools require background knowledge and have various application scenarios, which can be derived from the scientific paper. Fig. S5 shows some retrieved examples for the "Body Pose Estimation" tool from our RAG Mechanism.



Can you analyze the body angles and positions of the person in the image?

User Query

User Input:
Images/sdjsyu.png, Please show the angles and pose of the person's body in this photo?
Response:
Thought: Do I need to use a tool? Yes
Action: Body Pose Estimation
Input: images/sdjsyu.png

Retrieved Example

Build a human avatar using this image as a reference

User Query

User Input:
Images/ghsdvw.png, Please create an avatar of the person in the imgage.
Response:
Thought: Do I need to use a tool? Yes
Action: Body Pose Estimation
Input: images/ ghsdvw.png

Retrieved Example

Figure S5. Qualitative examples of the scientific-paper-based RAG mechanism. A single tool can have multiple application scenerios.

	3DPW			RPE Benchmark		
Method	MPJPE ↓	PA-MPJPE ↓	MPJRE ↓	MPJPE ↓	PA-MPJPE ↓	MPJRE ↓
GPT4Tools-FT	114.1	71.0	10.2	190.5	100.6	11.6
Visual ChatGPT-4	103.8	63.1	10.1	168.4	82.3	10.7
ChatHuman	**91.3**	**58.7**	**9.2**	**147.2**	**79.1**	**10.3**

Table S1. Comparison of vanilla human pose estimation and reasoning-based pose estimation on 3DPW and RPE.

4.2. GPT4Tool and Visual ChatGPT-4

Tab. S1 shows a comparison of ChatHuman, GPT4Tools [20], and Visual ChatGPT-4 [14] on the task of pose estimation, following the metrics in main paper Tab. 3. We change the LLM agents while keeping the other setting unchanged. For GPT4Tools, we finetune it with our training data about tool usage for a fair comparison. When the method selects a wrong tool and fails to get a pose output, we calculate the error using a T-pose as the result. ChatHuman outperforms these baselines in the standard pose estimation task and the more complex reasoning-based pose estimation task.

	SR_t	SR_act	SR_args	SR	IoU
Trial 1	0.999	0.967	0.893	0.954	0.953
Trial 2	1.000	0.971	0.895	0.955	0.954
Trial 3	1.000	0.965	0.890	0.951	0.951
Average	1.000	0.968	0.893	0.953	0.953

Table S2. K-fold cross-validation with 3 trials.

4.3. K-fold Validation of Tool Usage

To further verify the robustness of tool utlizaiton, we conducted K-fold validation by splitting 26 tools into 9 folds, each containing 2-3 tools. The experiment was repeated three times, each time using a random fold as the unseen tools for evaluation and the rest for training. As shown in Tab. S2, consistent performance across trials shows strong generalization to unseen tasks. Tab. S3 shows that our method maintains robust performance as the number of unseen tools grows.

Unseen Tool	SR_t	SR_act	SR_args	SR	IoU
3	1.000	0.965	0.890	0.951	0.951
5	1.000	0.962	0.885	0.939	0.946
7	0.999	0.948	0.881	0.929	0.944

Table S3. Accuracy on more unseen tools.

4.4. GPT4 replaces LLaVA for comparison

We replace LLaVA with GPT4 and evaluate performance on the tool-use benchmark. As in Tab. S2, we use K-fold validation with 3 repeats, reporting average accuracy in Tab. S4. We will include it in Tab. 2 of the main paper.

	Seen Tools					Unseen Tools				
Method	SR_t	SR_act	SR_args	SR	IoU	SR_t	SR_act	SR_args	SR	IoU
Ours w/ GPT-4	0.953	0.920	0.732	0.751	0.875	0.969	0.924	0.734	0.746	0.876
ChatHuman	**1.000**	**0.973**	**0.951**	**0.966**	**0.974**	**1.000**	**0.968**	**0.893**	**0.953**	**0.953**

Table S4. Comparison of tool use accuracy.

4.5. Paper Components.

To improve the tool usage accuracy, we propose a paper-based RAG mechanism. Here we conduct an ablation study to analyze the effects of each component of the paper for instructing tool usage. The baseline model is derived by removing the RAG operation and is trained with the

T	Paper					C	Seen Tools					Unseen Tools				
T	A	I	R	M	E	C	SR_t	SR_{act}	SR_{args}	SR	IoU	SR_t	SR_{act}	SR_{args}	SR	IoU
✓						✓	1.0	0.97	0.93	0.96	0.96	0.95	0.89	0.78	0.82	0.87
✓	✓						1.0	0.97	0.95	0.97	0.97	0.99	0.94	0.85	0.90	0.93
✓	✓	✓					1.0	0.97	0.95	0.97	0.97	1.0	0.97	0.86	0.91	0.94
✓	✓	✓	✓				1.0	0.98	0.95	0.97	0.97	1.0	0.97	0.84	0.91	0.93
✓	✓	✓	✓	✓			1.0	0.98	0.94	0.97	0.97	0.99	0.95	0.82	0.87	0.92
✓	✓	✓	✓	✓	✓		1.0	0.98	0.95	0.97	0.97	1.0	0.96	0.86	0.91	0.94
✓	✓	✓				✓	1.0	0.97	0.95	0.97	0.98	1.0	0.97	0.89	0.95	0.95

Table S5. Ablation study on the impact of each paper component in the paper-based RAG mechanism. T denotes tool description, A, I, R, M, E are abstract, introduction, related work, method, experiment section from the paper, and C denotes the data after manually cleaning. Successful rate of thought (SR_t), action (SR_{act}), arguments (SR_{args}), execution (SR), and IoU are reported.

Evaluator	Precision ↑	Recall ↑	F1 Score ↑
GPT-3.5	0.67	0.67	0.63
GPT-4	0.69	0.69	0.64

(a) HOI Contact Detection.

Evaluator	Height ↓	Weight ↓	Chest ↓	Waist ↓	Hip ↓
GPT-3.5	6.7	10.4	6.1	13.0	6.4
GPT-4	6.7	10.4	6.1	13.0	6.4

(b) Body Shape Measurement.

Table S6. Ablation study on the base model for value extraction.

instruction-following data constructed based on the manually defined tool descriptions and examples (T), without referring to paper content. We then add the tool descriptions and examples summarized from different paper components into the prompt and request GPT-4 to generate new instruction-following data. The paper components include the abstract (A), introduction (I), related work (R), method (M), and experiment section (E). The results are presented in Table S5. As shown, incorporating the paper content consistently improves the accuracy of tool usage. Interestingly, feeding the abstract and introduction of the paper achieves the best performance, and adding additional paper components like the experiments does not result in further improvement. This is expected since the abstract and introduction have covered the tool function and potential applications in most cases. We also find that examples generated based on paper content can be noisy in some cases, and thus we manually check the examples and remove the incorrect samples before feeding them to GPT-4. This data cleaning operation improves the accuracy.

4.6. Base Model for Value Extraction.

During our body shape measurement and human contact detection experiments, the answer of ChatHuman is a sentence and thus could not be directly used to compute the evaluation metrics. Thus, we need to prompt an LLM to extract the value from the sentence and compare the extracted value with the ground truth label to calculate the metric. The prompts for body shape measurement and contact estimation are shown in Table S10. Here we use two different LLM models for value extraction and report the results in Table S6. As shown, the choice of the LLM model for value extraction does not introduce a significant difference.

Method	Seen Tools					Unseen Tools				
Method	SR_t	SR_{act}	SR_{args}	SR	IoU	SR_t	SR_{act}	SR_{args}	SR	IoU
instructor-base	1.000	**0.975**	0.947	**0.972**	0.974	0.997	0.950	0.884	0.949	0.949
instructor-xl	**1.000**	0.974	**0.950**	0.970	**0.975**	**0.999**	**0.967**	**0.893**	**0.954**	**0.953**

Table S7. Ablation study of text embedding model for RAG. Successful rate of thought (SR_t), action (SR_{act}), arguments (SR_{args}), execution (SR), and IoU are reported.

	SR_{args}	SR	IoU	SR_t	SR_{act}
GPT4Tools [21]	0.582	0.551	0.553	0.513	0.612
Visual ChatGPT-3.5 [14]	0.438	0.203	0.162	0.173	0.691
Visual ChatGPT-4 [14]	0.860	0.794	0.711	0.744	0.789
ChatHuman	**1.000**	**0.959**	**0.927**	**0.955**	**0.962**

Table S8. Comparison of tool usage accuracy within multi-turn dialogue. Successful rate of thought (SR_t), action (SR_{act}), arguments (SR_{args}), execution (SR), and IoU are reported.

4.7. Text Embedding Model in Retrieval.

During the paper-based retrieval mechanism, we retrieve a relevant example by matching the text embedding of the query and those within the tool documents using a pretrained text embedding model [16]. Here we conduct an ablation study of the text embedding model to analyze the robustness of our RAG mechanism. We experiment with two models, i.e., instructor-xl and instructor-base. As shown in Table S7, the performance of our method is not greatly affected by the text embedding model, which demonstrates the robustness of ChatHuman.

4.8. Tool Use in Multi-turn Dialogue.

In many real-world scenarios, the models need to correctly use the tool based on the context within a multi-turn dialogue. Here, we build a benchmark to evaluate the tool use accuracy within multi-turn conversations. We randomly select several single-turn question-answer pairs from the original multi-turn tool use benchmark and ask GPT-4 to merge them into a multi-turn dialogue. We then evaluate the performance of GPT4Tools [20], Visual ChatGPT [14], and ChatHuman on the built benchmark, which contains 1000 instruction-response pairs. As shown in Table S8, ChatHuman achieves a superior performance in the multi-turn setting, demonstrating its capacity to understand the comprehensive context information within the multi-turn dialogue and correctly use the tool to solve the problems.

4.9. Tool Graph Construction.

We prompt GPT-4 to construct a tool graph with three structure types: nodes (single tool calls for simple tasks), chains (tool sequences for dependent tasks), and directed acyclic graphs (DAGs) [15] for complex multi-branch operations. Table S9 shows the detail of the prompt.

Given an image whose image path is ”example.jpg”. Image caption: ”{caption}”. The image caption includes detail image description and each object paired with the bounding box [x1, y1, x2, y2]. For the bounding box, (x1, y1) refers to the top left, and (x2, y2) refers to the bottom right. x1 less than x2, and y1 less than y2.

Below are 26 visual tools. Each tool is defined as ”tool name: usage scenario, and arguments to tool”.

Please generate 10 instructions that will need multiple tools to finish. The generated instructions should follow the format of ”instruction content, [[tool name1, arguments to tool1], [tool name2, arguments to tool2], ...]”. Each instruction must relate to the caption and can be solved by the tool. You can not revise the ”tool name”, or add any other fake tools that is not defined. You must keep the correct ”arguments to tool”.

Tools: {tool description}

Note that you should use 1-5 tools in each instruction and your generated visual instructions should be highly related to the image caption. Directly reply to me with the list, here are some examples:
{examples}

Diversify the instructions to cover a wide range of possible user queries. Feel free to adapt and rephrase the examples provided to generate diverse, complex, and deceptive instructions as much as possible.
For example, you can also change the subject position or the person and pose description positions. Don't use too much imperative sentence, you should also use interrogative sentence.

Table S9. Prompt to request GPT-4 to construct tool graph based on the image description, tool description, and tool usage examples summarized from tool paper.

(a) You are an AI assistant. Your input will be a description of body measurements, including height, weight, chest circumference, hip chest circumference, and waist circumference. Your task is to extract the value of each attribute and return a result like:
“height: 1 m, weight: 1 kg, chest circumference: 1 m, waist circumference: 1 m, hip circumference: 1 m”
If there is no measurement value, return “There is no measurement value.”

(b) You are an AI assistant. Your input will be a description of the human-object interaction information of a person. Your task is to extract the body parts that contact with objects and return a list. Consider the following possible body parts: right hand, right upper leg, left arm, left leg, left foot, back, left shoulder, right shoulder, right foot, head, right arm, left hand, right leg, left forearm, right forearm, neck, left upper leg, hips.

Table S10. Prompt to extract the target values from a sentence generated by ChatHuman for metric computation. Prompt (a) is used to request GPT-3.5 for body measurement values. Prompt (b) instructs GPT-3.5 to extract the body part names.

You are an AI visual assistant tasked with analyzing a paper on a method in the field of 3D human modeling. Your goal is to extract key information about the method—its name, purpose, uses, and potential application scenarios. Based on this, you need to succinctly define the method in the following formats “Method name is a tool to do something. Useful when you want to do something. Like: user query.”

Subsequently, craft 50 diverse, realistic user prompts that indirectly pertain to using this method. These queries should be framed as questions, demands, or scenarios from consumers who are unaware of the method’s name but whose needs align with its capabilities. Assume that consumers have an image and seek assistance in achieving a task related to the image using this method. Each prompt should introduce the task of the user in an imperative tone. The prompt should specify and refer to the image.

Here is one example:

Method definition:
name=“HMR2.”,
description=“HMR2.0 is a tool to estimate the 3D pose and shape of the person in the image. Useful when you want to detect poses of the humans in the image. Like: estimate the human poses in the image.”

Possible queries:
1. Can you help me estimate the pose of the person in the photo?
2. Please reconstruct a 3D avatar for the person in the image.
3. Could you please estimate the SMPL parameter of the man in the picture?
4. Someone in the image is riding a bicycle, could you please help me estimate her pose?

Table S11. Prompt to request GPT-4 to summarize paper content, rearticulate tool functions, and enumerate possible user queries for tool activation.

Given an image whose image path is “example.jpg”. Image caption: “{caption}”. The image caption includes detail image description and each object paired with the bounding box [x1, y1, x2, y2]. For the bounding box, (x1, y1) refers to the top left, and (x2, y2) refers to the bottom right. x1 is less than x2, and y1 is less than y2.

Below are 26 visual tools. Each tool is defined as “tool name: usage scenario, and tool arguments”.

Please generate 3 visual instructions for each tool, so you need to generate 66 visual instructions in total.
The generated instructions should follow the format of “instruction content, [tool name, tool arguments]”. Each instruction must relate to the caption and can be solved by the tool.
You can not revise the “tool name”, or add any other fake tools that are not defined. You must keep the correct “tool arguments”.

Tools: {tool description}

Note that your generated visual instructions should be highly related to the image caption. Directly reply to me with the list, here are some examples: {examples}

Diversify the instructions to cover a wide range of possible user queries. Feel free to adapt and rephrase the examples provided to generate diverse, complex, and deceptive instructions as much as possible.
For example, you can also change the subject position or the person and pose description positions. Don’t use too much imperative sentence, you should also use interrogative sentences.

Table S12. Prompt to request GPT-4 to generate instruction-following data about tool usage based on the image description, tool description, and tool usage examples summarized from tool paper.

You are an AI visual assistant, and you are seeing a single image and a sentence about the human-object contact regions of the person in the image. The sentence include the human-object contact body parts of the person. Notably, the provided contact regions might not be complete and you should detect all the contact regions by yourself.

Design a conversation between you and a person asking about the human-object contact information of the person. The answers should be in a tone that a visual AI assistant is seeing the image and answering the question. Ask diverse questions and give corresponding answers.

Include questions asking about the person's human-object contact information, etc. Only include questions that have definite answers: (1) one can see the content in the image that the question asks about and can answer confidently; (2) one can determine confidently from the image that it is not in the image. Do not ask any questions that cannot be answered confidently.

Provide detailed answers when answering complex questions. In your answer, you should imitate as if you see the image and the contact regions are estimated by you. You should only ask questions about the human-object interaction. The answer should be as detailed as possible. Don't mention any other irrelevant information! Directly reply to me with a list, here are some examples:

1. Please help me detect the contact regions of the person in the image. [The person's contact region includes his feet, hands, and back. His feet touch the ground and his hands are holding a mobile phone.]
2. Does the person's hand contact any objects? [Yes, her right hand holds an umbrella.]
3. Describe the human-object interaction information of the person, as detailed as possible. [The person holds a phone with his left hand and stands on a skateboard with both their left and right feet.]

Notably, at least one question is to ask all the contact regions of the person. In your answer, you should distinguish and specify the left and right body parts. Notably, you should distinguish based on the body pose and orientation. If the person is facing the camera, the hand, foot, and ear on the left side of the image is the person's right hand, right foot, and right ear, and the one on the right side of the image is the person's left hand, left foot, and left ear. If the person has their back to the camera, the one on the left side of the image is the person's left body part, and the one on the right side of the image is the person's right body part.

Table S13. Prompt to request GPT-4V to generate instruction-following data about human-object interaction based on the textual contact description and RGB image.

You are an AI visual assistant, and you are seeing a single image. What you see are provided with a sentence, describing the body shape of the person in the image. Answer all questions as you are seeing the image.

The sentence includes information about the person's gender, body mass, height, chest circumference, waist circumference, and hip circumference. Besides, it includes 15 linguistic shape attributes scale from 1 (strongly disagree) to 5 (strongly agree).

Design a conversation between you and a person asking about the body shape of the person. The answers should be in a tone that a visual AI assistant is seeing the image and answering the question.

Ask diverse questions and give corresponding answers.

Include questions asking about the visual content of the image, including the person's overall body fit, shape, height, mass, etc. Only include questions that have definite answers:

(1) one can see the content in the image that the question asks about and can answer confidently;

(2) one can determine confidently from the image that it is not in the image. Do not ask any questions that cannot be answered confidently.

Provide detailed answers when answering complex questions. When the question is about the measurement, provide an explicit and concrete metric number in the answer.

In your answer, you should imitate as if you see the image and the measurements and linguistic attributes are estimated by you. The linguistic attribute score is only used to help you understand and don't mention it in your answer.

Directly reply to me with a list, here are some examples:

1. How tall is the person in the image? [The person looks quite tall. He is about 1.85 m.]
2. Please help me estimate the body measurements of the man in the image. [The man is about 1.74 m and 60 kg. His chest circumference is about 0.9 m.]
3. What's the waist circumference of the person? [The chest circumference is about 0.95 m.]

Table S14. Prompt to request GPT-4 to generate instruction-following data about human body shape based on the textual description about human body.

References

[1] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8726–8737, 2023. 3

[2] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016. 1

[3] Vasileios Choutas, Lea Müller, Chun-Hao P Huang, Siyu Tang, Dimitrios Tzionas, and Michael J Black. Accurate 3d body shape regression using metric and semantic attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2718–2728, 2022. 2

[4] Ginger Delmas, Philippe Weinzaepfel, Thomas Lucas, Francesc Moreno-Noguer, and Grégory Rogez. Posescript: 3d human poses from natural language. In *ECCV*, 2022. 1, 2, 3

[5] Yao Feng, Jing Lin, Sai Kumar Dwivedi, Yu Sun, Priyanka Patel, and Michael J. Black. ChatPose: Chatting about 3d human pose. In *CVPR*, 2024. 2, 3

[6] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4D: Reconstructing and tracking humans with transformers. In *ICCV*, 2023. 1

[7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv:2106.09685*, 2021. 1

[8] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. CLIFF: Carrying location information in full frames into human pose and shape estimation. In *ECCV*, 2022. 1

[9] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[10] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 3

[11] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. In *ACM TOG*, 2015. 1, 2

[12] Matthew Matl. Pyrender. `https://github.com/mmatl/pyrender`, 2019. 2

[13] OpenAI. GPT-4 technical report. 2023. 1, 2

[14] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023. 4, 5

[15] Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueting Zhuang. Taskbench: Benchmarking large language models for task automation. *arXiv preprint arXiv:2311.18760*, 2023. 5

[16] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*, 2022. 5

[17] Shashank Tripathi, Agniv Chatterjee, Jean-Claude Passy, Hongwei Yi, Dimitrios Tzionas, and Michael J Black. Deco: Dense estimation of 3d human-scene contact in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8001–8013, 2023. 1, 2, 4

[18] Shashank Tripathi, Lea Müller, Chun-Hao P Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. 3d human pose estimation via intuitive physics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4713–4725, 2023. 1, 3

[19] Timo von Marcard, Roberto Henschel, Michael J. Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3D human pose in the wild using IMUs and a moving camera. In *ECCV*, 2018. 1

[20] Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. GPT4Tools: Teaching llm to use tools via self-instruction. *arXiv preprint arXiv:2305.18752*, 2023. 1, 2, 4, 5

[21] Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. Gpt4tools: Teaching large language model to use tools via self-instruction, 2023. 5