Text2HOI: Text-guided 3D Motion Generation for Hand-Object Interaction -Supplementary-

Junuk Cha[1] Jihyeon Kim[1,2†] Jae Shin Yoon[3*] Seungryul Baek[1*]

[1]UNIST [2]KETI [3]Adobe Research

In this supplementary material, we provide detailed descriptions of text annotation process along with data summary; qualitative results for ablation studies; refiner design; the effect of the refiner; inference speed; the summary of notations; network architectures utilized in our pipeline; masking process; implementation for baselines; articulation angle; and limitation. Additionally, for further results (the qualitative comparison and the additional video results), please refer to the accompanying supplementary video. The video is also available in: https://youtu.be/YBRsu0pnTeA.

1. Dataset

Text prompt annotation. The datasets H2O [4], GRAB [7], and ARCTIC [1] do not provide the text prompts which describe the hand-object interactions. To facilitate the generation of 3D hand-object interactions from text prompts, collecting such text prompts is necessary. Text prompts should include details about the interacting hand type (*e.g.*, left hand, right hand, and both), the action involved (*e.g.*, grab, place, lift), and the object category or name (*e.g.*, apple, airplane, microwave). The basic format for text prompts is as follows: "{action} {object category} with {hand type}." (*e.g.*, "Place a book with both hands."). For the H2O and GRAB datasets, we automatically annotate text using provided action labels, which include both the action and object category. However, these labels do not specify the interacting hand type. We determine the interacting hand type based on the proximity between the hand and the object in global 3D space; if the distance during interaction is less than a predefined threshold (2cm), we decide that the hand is involved in the interaction motion. For the ARCTIC dataset, we conduct manual text annotation by observing the video, noting the action, hand type, and object category. We further augment the text prompt with additional descriptors using passive form, subject modifications, and gerunds. For instance, prompts are augmented to formats like "A {object category} is {action} by {hand type}." (*e.g.* "A book is placed by both hands."), or "{hand type} {action} {object category}." (*e.g.* "Both hands place a book.") or "{action}+ing {object category} with {hand type}." (*e.g.* "Placing a book with both hands.").

H2O. The H2O dataset [4] consists of 660 interactions, each involving two hands and one of 8 distinct objects. The dataset is annotated with 11 unique verb labels. Using these object and verb labels, and identifying which hand is interacting with the object, we automatically generate 272 distinct sentences to represent these interactions. Additionally, the dataset provides MANO hand parameters, object meshes, as well as information on object rotation and translation.

GRAB. The GRAB dataset [7] consists of 1,335 motions involving two hands and one of 51 distinct objects of varying shape and size. The dataset has 29 action labels. Using these object and action labels, and identifying which hand is interacting with the object, we automatically generate 1,104 distinct sentences to represent these motions. Additionally, the dataset provides MANO hand parameters, object meshes, as well as information on object rotation and translation.

ARCTIC. The ARCTIC dataset [1] is released for reconstructing hands and objects from RGB images. We annotate the data with our defined 11 action labels. We manually create 644 sentences to describe the motions, and these sentences contain information about the action, the type of object, and which hand is interacting. We annotate a total of 4,597 motions. Essentially, this dataset provides MANO hand parameters, object meshes, as well as information on object rotation, translation, and articulation angle with the pre-defined axis.

2. Qualitative results for ablation studies

We present the qualitative results in Fig. 1, to demonstrate the effectiveness of geometry losses (distance map loss L_{dm} and relative orientation loss L_{ro}), conditions (contact map $\hat{\mathbf{m}}_{\text{contact}}$ and object's scale s_{obj}), and our proposed

This research was conducted when Jihyeon Kim was a graduate student (Master candidate) at UNIST†. Co-last authors∗.

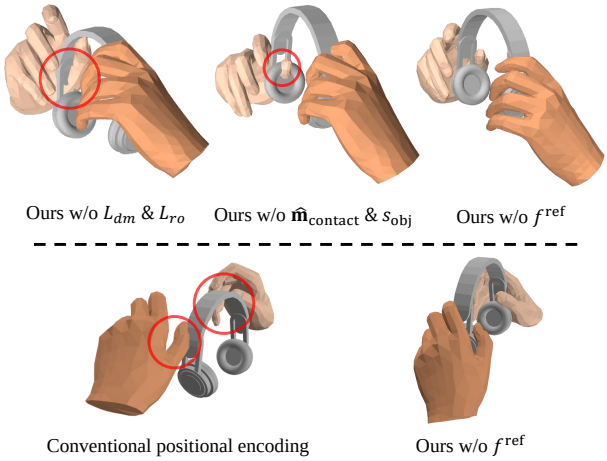

Figure 1. In the first row, the comparisons of geometry losses and conditions are presented. In the second row, the comparison focuses on positional encodings.

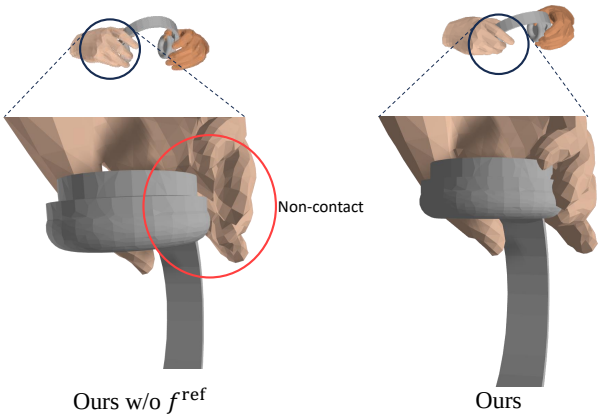

Figure 2. Our hand refinement network refines the contacts.

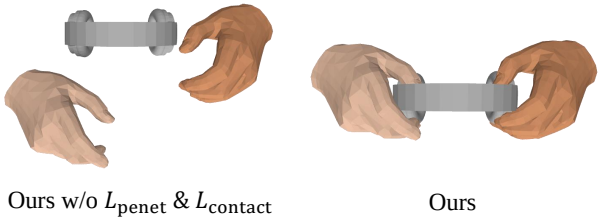

Figure 3. Without the penetration loss and contact loss, the generated motions result in the hands and object not interacting.

positional encodings (frame-wise and agent-wise positional encodings). In addition, we present the qualitative results related to hand refinement in Figs. 2 and 3, to show the effectiveness of refiner f^{ref}, and additional losses (penetration loss L_{penet} and contact loss L_{contact}), respectively. In Figs. 1, 2, and 3, the text prompt employed is *"Use headphones with both hands."*.

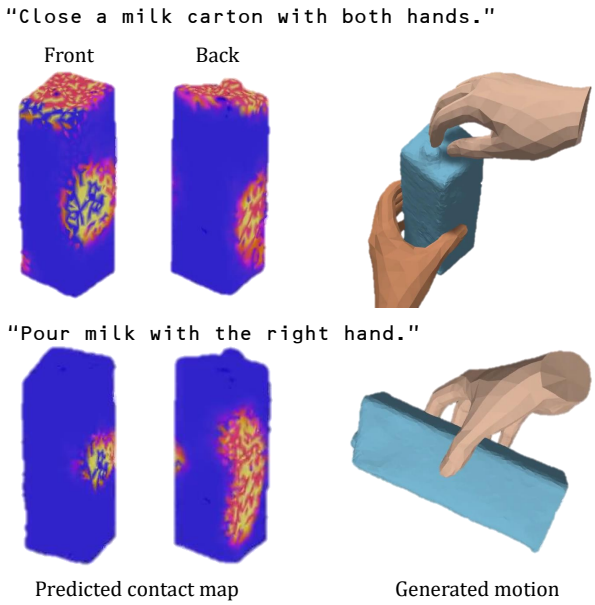

Figure 4. We display the predicted contact map and the generated motion, focusing on their variations in response to different text prompts.

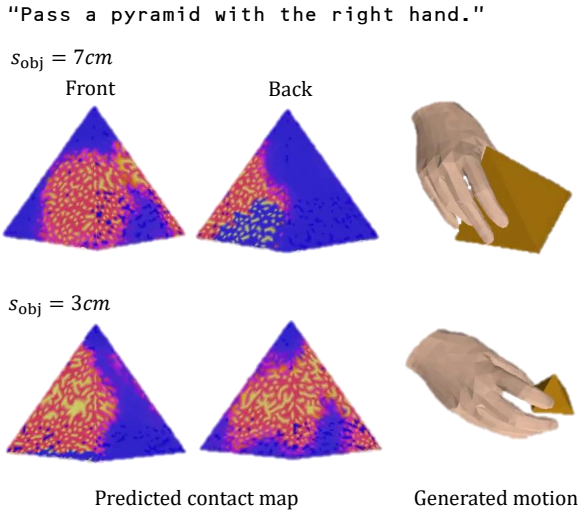

Figure 5. We illustrate how the predicted contact map and generated motion vary across different object's scales.

We demonstrate the variant predicted contact maps and generated motions, as shown in Figs. 4 and 5. The predicted contact maps accurately reflect the text prompts. The motions vary depending on the predicted contact maps and the prompts. In addition, the contact maps are predicted differently for different object's scales, influencing the number of fingers involved and the manner of grasping depending on the object's scale.

Table 1. Comparative physical realism scores for the different refiner designs.

Method	Physical realism
Diffusion-based hand refiner	0.1682 ± 0.0006
Ours	$\mathbf{0.8839 \pm 0.0005}$

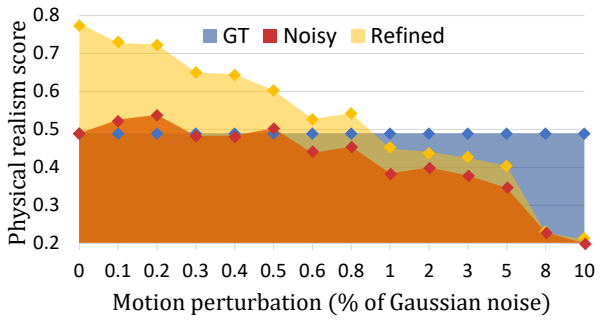

Figure 6. The physical realism score with ground-truth (GT), noisy, and refined motion parameters. Synthetically created noisy motion parameters are produced by incorporating the function of Gaussian noise, which distorts the ground-truth motion parameters.

3. Refiner design

We compare our hand refinement network with the diffusion-based hand refinement network. It is counterintuitive to apply the diffusion-based approach for refining already generated motions by f^{THOI}. The diffusion model necessitates adding noise to the already generated motions through a diffusion forward process, and then denoising them via a backward process. Given this inefficiency, we propose a refiner that does not rely on the diffusion-based method. Additionally, in terms of performance, our refiner f^{ref} demonstrates superior results as shown in Tab. 1.

4. Effect of refiner

We demonstrate the effectiveness of our hand refinement network f^{ref} as shown in Fig. 6. The physical realism score is highly increased by our hand refinement network even it outperforms that of ground-truth motion.

5. Inference speed

We measure the time required to generate 150 frames of hand-object interaction using an RTX 4090, with the detailed results presented in Tab. 2. Notably, our method demonstrates faster performance compared to the diffusion-based method MDM [8].

Table 2. Inference speed.

Method		Time
MDM [8]		28.5s
IMoS [2]		101s
Ours	Contact map f^{contact}	0.011s
	Text2HOI f^{THOI}	4.9s
	Refinement f^{ref}	0.013s

Table 3. Notations.

Symbol	Definition
$\mathbf{T}$	Text prompt
$f^{\text{CLIP}}(\mathbf{T})$	Text features
$\mathbf{M}_{\text{obj}}$	Canonical object mesh
$\mathbf{H}^*$	Hand type (left, right, both)
$\mathbf{x}$	Motion
$\mathbf{x}^l$	Motion at l-th frame
$\mathbf{x}_t$	Noised motion at t-th diffusion time-step
$\mathbf{x}_0$	Clean motion
$\mathbf{x}_{\text{lhand}}$	Left hand motion
$\mathbf{x}_{\text{rhand}}$	Right hand motion
$\mathbf{x}_{\text{hand}}$	Hand motion
$\mathbf{x}_{\text{obj}}$	Object motion
T	The number of diffusion steps
L	Motion length
L_{max}	Maximum motion length (=150)
s_{obj}	Object's scale
$\mathbf{P}$	Object point cloud
$\mathbf{P}_{\text{def}}$	Deformed object point cloud
$\mathbf{F}_{\text{obj}}$	(Global) object features
$\mathbf{V}$	Hand vertices
V	The number of hand vertices (=778)
$\mathbf{J}$	Hand joints
J	The number of hand joints (=21)
$\mathbf{m}_{\text{contact}}$	Contact map
$\mathbf{X}$	Embedded value (motion, condition)
$\hat{\cdot}$	Estimated value (output)
$\tilde{\cdot}$	Refined value (output)
f	Network

6. Notations

We summarize notations used in main paper and supplementary material in Tab. 3.

7. Network

Several networks are involved in our framework: contact prediction network f^{contact}, text-to-3D hand-object interaction generator f^{THOI}, and hand refinement network f^{ref}.

Contact prediction network. We predict the contact map

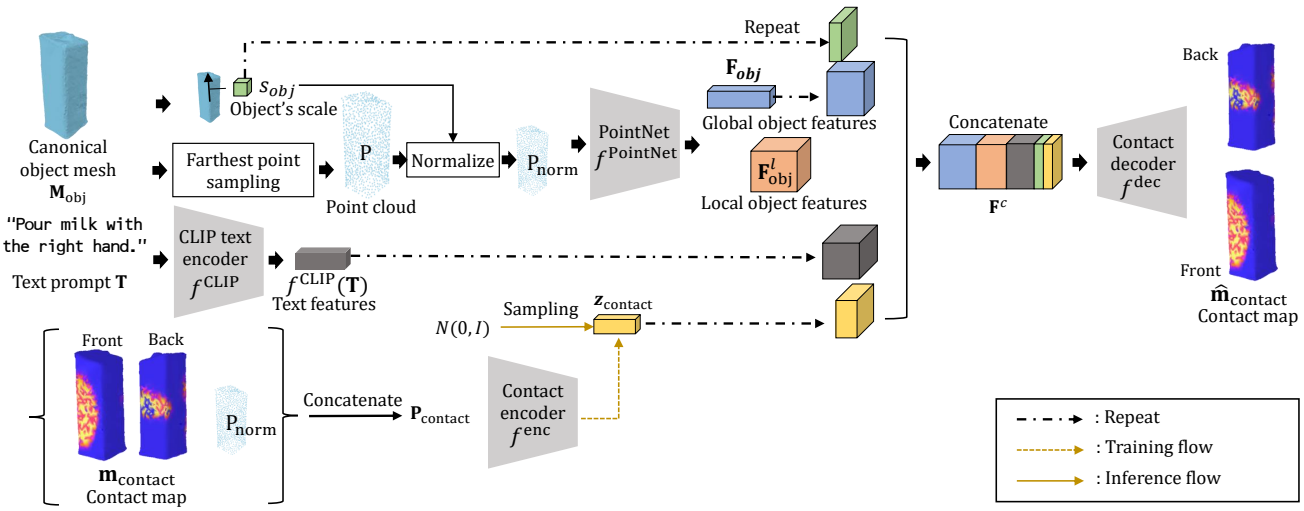

Figure 7. Predicting the contact map consists of 4 steps. 1) From a canonical object mesh $\mathbf{M}_{\text{obj}}$, we compute the object's scale s_{obj}. Then, we sample the point cloud $\mathbf{P}$ using the farthest point sampling algorithm, and normalize $\mathbf{P}$ to $\mathbf{P}_{\text{norm}}$ by dividing it with s_{obj}. The PointNet f^{PointNet} receives $\mathbf{P}_{\text{norm}}$ as input, to extract the global object features $\mathbf{F}_{\text{obj}}$ and local object features $\mathbf{F}^{l}_{\text{obj}}$. 2) From a text prompt $\mathbf{T}$, CLIP text encoder f^{CLIP} extracts the text features $f^{\text{CLIP}}(\mathbf{T})$. 3) At training time, we extract the vector $\mathbf{z}_{\text{contact}}$ using the contact encoder f^{enc} from $\mathbf{P}_{\text{contact}}$ that is concatenated with the ground-truth contact map $\mathbf{m}_{\text{contact}}$ and the normalized point cloud $\mathbf{P}_{\text{norm}}$. At inference time, we sample $\mathbf{z}_{\text{contact}}$ from Gaussian distribution. 4) We concatenate the diverse features $\mathbf{F}_{\text{obj}}$, $\mathbf{F}^{l}_{\text{obj}}$, $f^{\text{CLIP}}(\mathbf{T})$, s_{obj}, and $\mathbf{z}_{\text{contact}}$, to produce $\mathbf{F}^{c}$. Finally, the contact decoder f^{dec} predicts the contact map $\hat{\mathbf{m}}_{\text{contact}}$ from $\mathbf{F}^{c}$.

$\mathbf{m}_{\text{contact}}$ from a text prompt $\mathbf{T}$ and a canonical object mesh $\mathbf{M}_{\text{obj}}$. The contact map prediction procedure involves four steps: 1) extracting object features from the object; 2) extracting text features from the text prompt; 3) sampling a Gaussian random noise vector; and 4) predicting the contact map, as illustrated in Fig. 7.

We first compute an object's scale $s_{\text{obj}} \in \mathbb{R}^1$ that represents the maximum distance from the center of object mesh $\mathbf{M}_{\text{obj}}$ to its vertices. Then, we sample N-point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$ from the vertices of $\mathbf{M}_{\text{obj}}$ ($N = 1,024$) using the farthest point sampling (FPS) algorithm [5]. Subsequently, $\mathbf{P}$ is normalized to $\mathbf{P}_{\text{norm}}$ by dividing it with s_{obj}. We utilize PointNet f^{PointNet} [5] to extract local object features $\mathbf{F}^{l}_{\text{obj}} \in \mathbb{R}^{N\times 64}$ and global object features $\mathbf{F}_{\text{obj}} \in \mathbb{R}^{1,024}$ from the normalized point cloud $\mathbf{P}_{\text{norm}}$.

Second, we extract the text features $f^{\text{CLIP}}(\mathbf{T}) \in \mathbb{R}^{512}$ from the text prompt $\mathbf{T}$ using CLIP text encoder f^{CLIP} [6].

Third, at training time, the vector $\mathbf{z}_{\text{contact}} \in \mathbb{R}^{64}$ is encoded from the concatenated input $\mathbf{P}_{\text{contact}} \in \mathbb{R}^{N\times 4}$ with $\mathbf{P}_{\text{norm}} \in \mathbb{R}^{N\times 3}$ and the ground-truth contact map $\mathbf{m}_{\text{contact}} \in \mathbb{R}^{N\times 1}$ using the contact encoder f^{enc}. At inference time, $\mathbf{z}_{\text{contact}}$ is sampled from Gaussian distribution.

Fourth, we concatenate $\mathbf{F}_{\text{obj}}$, $\mathbf{F}^{l}_{\text{obj}}$, s_{obj}, $f^{\text{CLIP}}(\mathbf{T})$, and $\mathbf{z}_{\text{contact}}$. Since they have varying feature dimensions, we duplicate these features N times except $\mathbf{F}^{l}_{\text{obj}}$, to align the dimension shape: $\text{Repeat}(\mathbf{F}_{\text{obj}}) : \mathbb{R}^{1,024} \rightarrow \mathbb{R}^{N\times 1,024}$, $\text{Repeat}(s_{\text{obj}}) : \mathbb{R}^{1} \rightarrow \mathbb{R}^{N\times 1}$, $\text{Repeat}(f^{\text{CLIP}}(\mathbf{T})) : \mathbb{R}^{512} \rightarrow \mathbb{R}^{N\times 512}$, and $\text{Repeat}(\mathbf{z}_{\text{contact}}) : \mathbb{R}^{64} \rightarrow \mathbb{R}^{N\times 64}$. Finally, the concatenated features $\mathbf{F}^{c} \in \mathbb{R}^{N\times 1,665}$ are fed to the contact decoder f^{dec} to predict the contact map $\hat{\mathbf{m}}_{\text{contact}} \in \mathbb{R}^{N\times 1}$. The architectures of f^{PointNet}, f^{enc}, and f^{dec} are detailed in Tabs. 4, 5, and 6, respectively.

Text-to-3D hand-object interaction generator. Our generator f^{THOI} receives several inputs: time-step t, text features $f^{\text{CLIP}}(\mathbf{T})$, object features $\mathbf{F}_{\text{obj}}$, contact map $\hat{\mathbf{m}}_{\text{contact}}$, object's scale s_{obj} as condition, and noised motion $\mathbf{x}_t$ as input. Then, it outputs the denoised motions $\hat{\mathbf{x}}_0$. The structure of f^{THOI} includes various layers: input embedding layers ($f^{\text{in, lhand}}$, $f^{\text{in, rhand}}$, $f^{\text{in, obj}}$), condition embedding layers (f^{ts}, f^{text}, f^{obj}), a Transformer encoder, and output embedding layers ($f^{\text{out, lhand}}$, $f^{\text{out, rhand}}$, $f^{\text{out, obj}}$). The specific architectures of the input and condition embedding layers (see Tab. 7), Transformer encoder (see Tab. 8), and output embedding layers (see Tab. 9) are detailed in their respective tables.

Hand refinement network. Our hand refinement network f^{ref} receives the generated hand motion $\hat{\mathbf{x}}_{0,\text{hand}} \in \mathbb{R}^{2\hat{L}\times 99}$ ($\hat{\mathbf{x}}_{0,\text{hand}} = \{\hat{\mathbf{x}}^{l}_{0,\text{lhand}} \in \mathbb{R}^{99}, \hat{\mathbf{x}}^{l}_{0,\text{rhand}} \in \mathbb{R}^{99}\}^{\hat{L}}_{l=1}$), hand joints $\hat{\mathbf{J}}_{\text{hand}} \in \mathbb{R}^{2\hat{L}\times J\times 3}$ ($\hat{\mathbf{J}}_{\text{hand}} = \{\hat{\mathbf{J}}^{l}_{\text{lhand}} \in \mathbb{R}^{J\times 3}, \hat{\mathbf{J}}^{l}_{\text{rhand}} \in \mathbb{R}^{J\times 3}\}^{\hat{L}}_{l=1}$), contact map $\hat{\mathbf{m}}_{\text{contact}} \in \mathbb{R}^{N\times 1}$, deformed object's point cloud $\hat{\mathbf{P}}_{\text{def}} \in \mathbb{R}^{\hat{L}\times N\times 3}$, and distance-based attention map $\mathbf{m}_{\text{att}} \in \mathbb{R}^{2\hat{L}\times J\times 3}$ ($\mathbf{m}_{\text{att}} = \{\mathbf{m}^{l}_{\text{att, left}} \in \mathbb{R}^{J\times 3}, \mathbf{m}^{l}_{\text{att, right}} \in \mathbb{R}^{J\times 3}\}^{\hat{L}}_{l=1}$) as input, where $\hat{L}$ is an es-

Table 4. Architecture of PointNet f^{PointNet}. N represents the number of points of point cloud. k denotes the kernel size. BN denotes a batch normalization. I represents an identity matrix. x denotes the output of the previous layer.

Layer	Operation	Output Features
input	$\mathbf{P}_{\text{norm}}$	$N \times 3$
transpose	transpose	$3 \times N$
STN-conv1 (k=1)	Conv 1D + BN + ReLU	$64 \times N$
STN-conv2 (k=1)	Conv 1D + BN + ReLU	$128 \times N$
STN-conv3 (k=1)	Conv 1D + BN + ReLU	$1,024 \times N$
max (axis=1)	max	$1,024$
STN-fc1	Linear + BN + ReLU	512
STN-fc2	Linear + BN + ReLU	256
STN-fc3	Linear	9
reshape	reshape	3×3
identity	$x = x + I$	3×3
multiplication	$\mathbf{P}_{\text{norm}} \times x$	$N \times 3$
transpose	transpose	$3 \times N$
conv1 (k=1)	Conv 1D + BN + ReLU	$64 \times N$
transpose	transpose	$N \times 64$
assign	$\mathbf{F}^l_{\text{obj}} = x$	$N \times 64$
transpose	transpose	$64 \times N$
conv2 (k=1)	Conv 1D + BN + ReLU	$128 \times N$
conv3 (k=1)	Conv 1D + BN	$1,024 \times N$
max (axis=1)	max	$1,024$
assign	$\mathbf{F}_{\text{obj}} = x$	$1,024$

Table 5. Architecture of contact encoder f^{enc}. N represents the number of points of point cloud. k denotes the kernel size. BN denotes the batch normalization. x denotes the output of previous layer.

Layer	Operation	Output Features
input	Concatenate($\mathbf{P}_{\text{norm}}, \mathbf{m}_{\text{contact}}$)	$N \times 4$
transpose	transpose	$4 \times N$
STN-conv1 (k=1)	Conv 1D + BN + ReLU	$64 \times N$
STN-conv2 (k=1)	Conv 1D + BN + ReLU	$128 \times N$
STN-conv3 (k=1)	Conv 1D + BN + ReLU	$1,024 \times N$
max (axis=1)	max	$1,024$
STN-fc1	Linear + BN + ReLU	512
STN-fc2	Linear + BN + ReLU	256
STN-fc3	Linear	16
reshape	reshape	4×4
identity	$x = x + I$	4×4
multiplication	$\mathbf{P}_{\text{norm}} \times x$	$N \times 4$
transpose	transpose	$4 \times N$
conv1 (k=1)	Conv 1D + BN + ReLU	$64 \times N$
conv2 (k=1)	Conv 1D + BN + ReLU	$128 \times N$
conv3 (k=1)	Conv 1D + BN	$1,024 \times N$
max (axis=1)	max	$1,024$
fc1	Linear + BN + ReLU	512
fc2	Linear + BN + ReLU	256
fc3	Linear	128
split	mean $\mathbf{z}_{\mu}$	64
	variance $\mathbf{z}_{\sigma^2}$	64
reparameterize	$\mathbf{z}_{\text{contact}}$	64

timated motion length. We concatenate them to use them as input. First, the hand joints are reshaped: $\hat{\mathbf{J}}_{\text{hand}} \in \mathbb{R}^{2\hat{L} \times 3J}$.

Table 6. Architecture of contact decoder f^{dec}. 'slope' denotes the negative slope of LeakyReLU.

Layer	Operation	Output Features
input	$\mathbf{F}^c$	$N \times 1,665$
fc1	Linear + LeakyReLU(slope=0.2)	$N \times 512$
fc2	Linear + LeakyReLU(slope=0.2)	$N \times 256$
fc3	Linear + LeakyReLU(slope=0.2)	$N \times 128$
fc4	Linear	$N \times 1$

Table 7. Architectures of input and condition embedding layers in text-to-3D hand-object interaction generator f^{THOI}. {} denotes a concatenation.

Layer	Operation	Output Features
input	$\mathbf{x}^l_{t,\text{lhand}}$	99
$f^{\text{in, lhand}}$	Linear	512
input	$\mathbf{x}^l_{t,\text{rhand}}$	99
$f^{\text{in, rhand}}$	Linear	512
input	$\mathbf{x}^l_{t,\text{obj}}$	10
$f^{\text{in, obj}}$	Linear	512
input	t	scalar
f^{ts}-pe	Positional encoding	512
f^{ts}-fc1	Linear + SiLU	512
f^{ts}-fc2	Linear	512
input	$f^{\text{CLIP}}(\mathbf{T})$	512
f^{text}	Linear	512
input	$\{\mathbf{F}_{\text{obj}}, \hat{\mathbf{m}}_{\text{contact}}, s_{\text{obj}}\}$	2049
f^{obj}	Linear	512

Second, the contact map is duplicated $2\hat{L}$ times and reshaped: $\hat{\mathbf{m}}_{\text{contact}} \in \mathbb{R}^{2\hat{L} \times N}$. Third, the deformed object's point cloud is duplicated 2 times, and the computation of the norm is applied across the last dimension: $\hat{\mathbf{P}}_{\text{def}} \in \mathbb{R}^{2\hat{L} \times N}$. Fourth, the distance-based attention map is reshape: $\mathbf{m}_{\text{att}} \in \mathbb{R}^{2\hat{L} \times 3J}$. We concatenate $\hat{\mathbf{x}}_{0,\text{hand}}$, $\hat{\mathbf{J}}_{\text{hand}}$, $\hat{\mathbf{m}}_{\text{contact}}$, $\hat{\mathbf{P}}_{\text{def}}$, and $\mathbf{m}_{\text{att}}$ to create the input $\mathbf{x}_{\text{ref}} = \{\mathbf{x}^l_{\text{ref, lhand}}, \mathbf{x}^l_{\text{ref, rhand}}\}^{\hat{L}}_{l=1} \in \mathbb{R}^{2\hat{L} \times (99+3J+N+N+3J)}$. $(99 + 3J + N + N + 3J)$ is 2,273, where $J = 21$ and $N = 1,024$. The hand inputs $\mathbf{x}^l_{\text{ref, lhand}}$, and $\mathbf{x}^l_{\text{ref, rhand}}$ are fed to hand input embedding layers $f^{\text{in, lhand}}_{\text{ref}}$ and $f^{\text{in, rhand}}_{\text{ref}}$, respectively, to obtain the embeddings $\mathbf{X}^l_{\text{ref, lhand}} \in \mathbb{R}^{512}$ and $\mathbf{X}^l_{\text{ref, rhand}} \in \mathbb{R}^{512}$. Then, they are applied the frame-wise and agent-wise positional encodings and masked using $\mathbf{H}^*$, and fed to Transformer encoder. Then, Transformer encoder outputs the refined embeddings $\tilde{\mathbf{X}}^l_{\text{lhand}} \in \mathbb{R}^{99}$ and $\tilde{\mathbf{X}}^l_{\text{rhand}} \in \mathbb{R}^{99}$. These embed-

Table 8. Architecture of Transformer encoder in text-to-3D hand-object interaction generator f^{THOI}. LN denotes a layer normalization. n=8 denotes that the layer repeat 8 times. x denotes the output of previous layer. $\hat{L}$ represenst the estimated motion length.

Layer		Operation	Output Features
input		$\mathbf{X}_t$	$(1+3\hat{L}) \times 512$
n=8	Multi-Head Attention (h=4)	Self attention (SA)	$(1+3\hat{L}) \times 512$
	Residual	$\mathbf{X}_t = \mathbf{X}_t$+SA	$(1+3\hat{L}) \times 512$
	Normalize1	LN	$(1+3\hat{L}) \times 512$
	fc1	Linear + GeLU	$(1+3\hat{L}) \times 1024$
	fc2	Linear	$(1+3\hat{L}) \times 512$
	Residual	$\mathbf{X}_t = \mathbf{X}_t + x$	$(1+3\hat{L}) \times 512$
	Normalize2	LN	$(1+3\hat{L}) \times 512$

Table 9. Architectures of output embedding layers in text-to-3D hand-object interaction generator f^{THOI}.

Layer	Operation	Output Features
input	$\mathbf{X}^l_{0,\text{lhand}}$	512
$f^{\text{out, lhand}}$	Linear	99
input	$\mathbf{X}^l_{0,\text{rhand}}$	512
$f^{\text{out, rhand}}$	Linear	99
input	$\mathbf{X}^l_{0,\text{obj}}$	512
$f^{\text{out, obj}}$	Linear	10

dings are passed through $f^{\text{out, lhand}}_{\text{ref}}$ and $f^{\text{out, rhand}}_{\text{ref}}$ and converted to refined hand motions $\tilde{\mathbf{x}}^l_{\text{lhand}}$ and $\tilde{\mathbf{x}}^l_{\text{rhand}}$. The final refined hand motions is expressed as follows: $\tilde{\mathbf{x}}_{\text{hand}} = \{\tilde{\mathbf{x}}^l_{\text{lhand}}, \tilde{\mathbf{x}}^l_{\text{rhand}}\}^{\hat{L}}_{l=1}$. The architecture of f^{ref} is detailed in Tab. 10.

8. Masking inputs, outputs, and losses

Using the hand-type variable $\mathbf{H}^*$, we implement masking in three areas: 1) the inputs of the Transformer encoder, 2) the outputs of the Transformer decoder, and 3) the losses (L_{dm}, L_{ro}, L_{penet}, L_{contact}). The masking process depends on the representation of $\mathbf{H}^*$ as follows:

- If $\mathbf{H}^*$ represents the 'left hand', then the inputs, outputs, and losses pertaining to the 'right hand' are masked.
- Conversely, if $\mathbf{H}^*$ represents the 'right hand', the corresponding components for the 'left hand' are masked.
- If $\mathbf{H}^*$ indicates 'both hands', no masking is applied to the inputs, outputs, or losses.

The indicator functions $\mathbb{1}_{\text{left}}$ and $\mathbb{1}_{\text{right}}$ are defined according to $\mathbf{H}^*$ as follows:

$$\{\mathbb{1}_{\text{left}}, \mathbb{1}_{\text{right}}\} = \begin{cases} \{1,0\}, & \text{if } \mathbf{H}^* \text{ is 'left hand'} \\ \{0,1\}, & \text{if } \mathbf{H}^* \text{ is 'right hand'} \\ \{1,1\}, & \text{if } \mathbf{H}^* \text{ is 'both hands'} \end{cases} \quad (1)$$

Masking the inputs means that the attention mechanism is inhibited for those inputs. Masking the outputs results in the visualization of those outputs being blocked. Masking the losses implies that backpropagation for those losses is restricted.

Table 10. Architecture of hand refinement network f^{ref}. $\mathbf{X}_{\text{ref}}$ denotes $\{\mathbf{X}^l_{\text{ref, lhand}}, \mathbf{X}^l_{\text{ref, rhand}}\}^{\hat{L}}_{l=1}$.

Layer		Operation	Output Features
Input		$\mathbf{x}_{\text{ref}}$	$2\hat{L} \times 2,273$
$f^{\text{in, lhand}}_{\text{ref}}$		Linear	$\hat{L} \times 512$
$f^{\text{in, rhand}}_{\text{ref}}$		Linear	$\hat{L} \times 512$
Transformer encoder			
n=8	Multi-Head Attention (h=4)	Self attention (SA)	$2\hat{L} \times 512$
	Residual	$\mathbf{X}_{\text{ref}} = \mathbf{X}_{\text{ref}}$+SA	$2\hat{L} \times 512$
	Normalize1	LN	$2\hat{L} \times 512$
	fc1	Linear + GeLU	$2\hat{L} \times 1024$
	fc2	Linear	$2\hat{L} \times 512$
	Residual	$\mathbf{X}_{\text{ref}} = \mathbf{X}_{\text{ref}} + x$	$2\hat{L} \times 512$
	Normalize2	LN	$2\hat{L} \times 512$
$f^{\text{out, lhand}}_{\text{ref}}$		Linear	$\hat{L} \times 99$
$f^{\text{out, rhand}}_{\text{ref}}$		Linear	$\hat{L} \times 99$

9. Implementation details for baselines

We maintained the baselines' model architecture, training scheme, and inference process, and just adjusted the model's output dimension to obtain parameters for two hands and for object [2, 3, 8]. We used their pre-estimated length if they had a length estimator; otherwise, we used their predefined fixed length. We employed hand-type selection for masking the hand input and hand output, following the approach of our method.

10. How to use the articulation parameter

The articulation parameter is generated for all objects, regardless of their type. However, its actual application depends on datasets using articulation indicator. For articulated objects in ARCTIC dataset, the indicator is set as true

and the articulation is reflected. For rigid objects in H2O and GRAB datasets, the indicator is set as false and the articulation parameter essentially acts as a placeholder and is not applied.

11. More qualitative results

Fig. 8 shows the diverse hand-object interactions from the same prompt *'Type a laptop with both hands.'*. Fig. 9 demonstrates the hand pose change in canonical coordinate. Fig. 10 illustrates the varying results according to different hand types. Each figure illustrates a sequence of key frames extracted from a video, displayed in a grid format of M rows and N columns. The frames are organized to represent the temporal progression of the video from left to right and top to bottom, simulating the temporal order of the events depicted in the video.

12. Limitation.

Hand-object interacting motions are generated from the text prompt, considering the relative 3D location and contact between hands and an object; while we are missing forces between them, which may provide better physical understanding. Future works may need to consider such new aspects.

References

[1] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. Arctic: A dataset for dexterous bimanual hand-object manipulation. In *CVPR*, 2023. 1

[2] Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. Imos: Intent-driven full-body motion synthesis for human-object interactions. In *Computer Graphics Forum*, 2023. 3, 6

[3] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *CVPR*, 2022. 6

[4] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *ICCV*, 2021. 1

[5] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 4

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4

[7] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *ECCV*, 2020. 1

[8] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *ICLR*, 2023. 3, 6

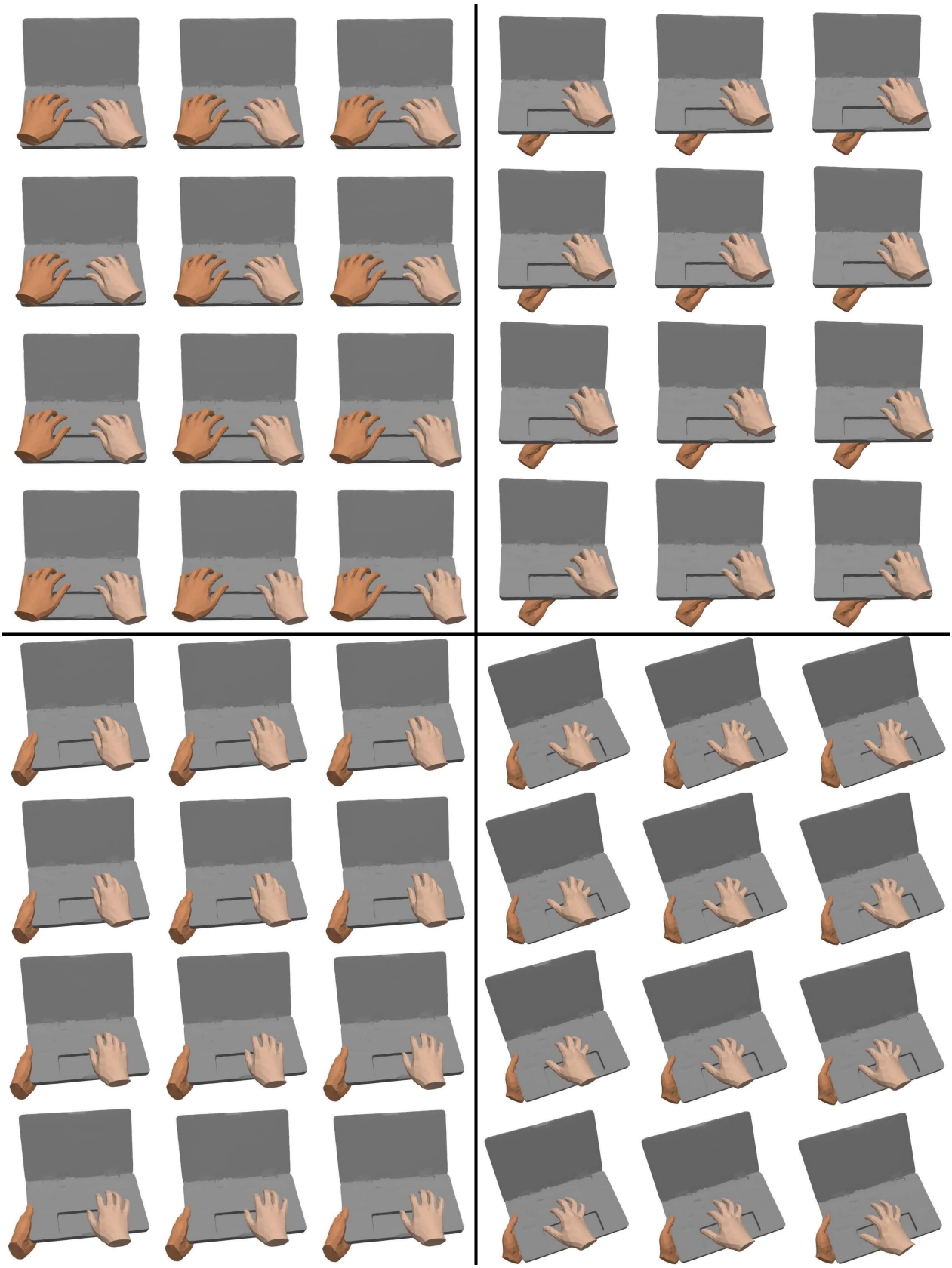

Figure 8. The diverse hand-object interactions from the same prompt *'Type a laptop with both hands.'*. The sequence is from left to right and top to bottom.

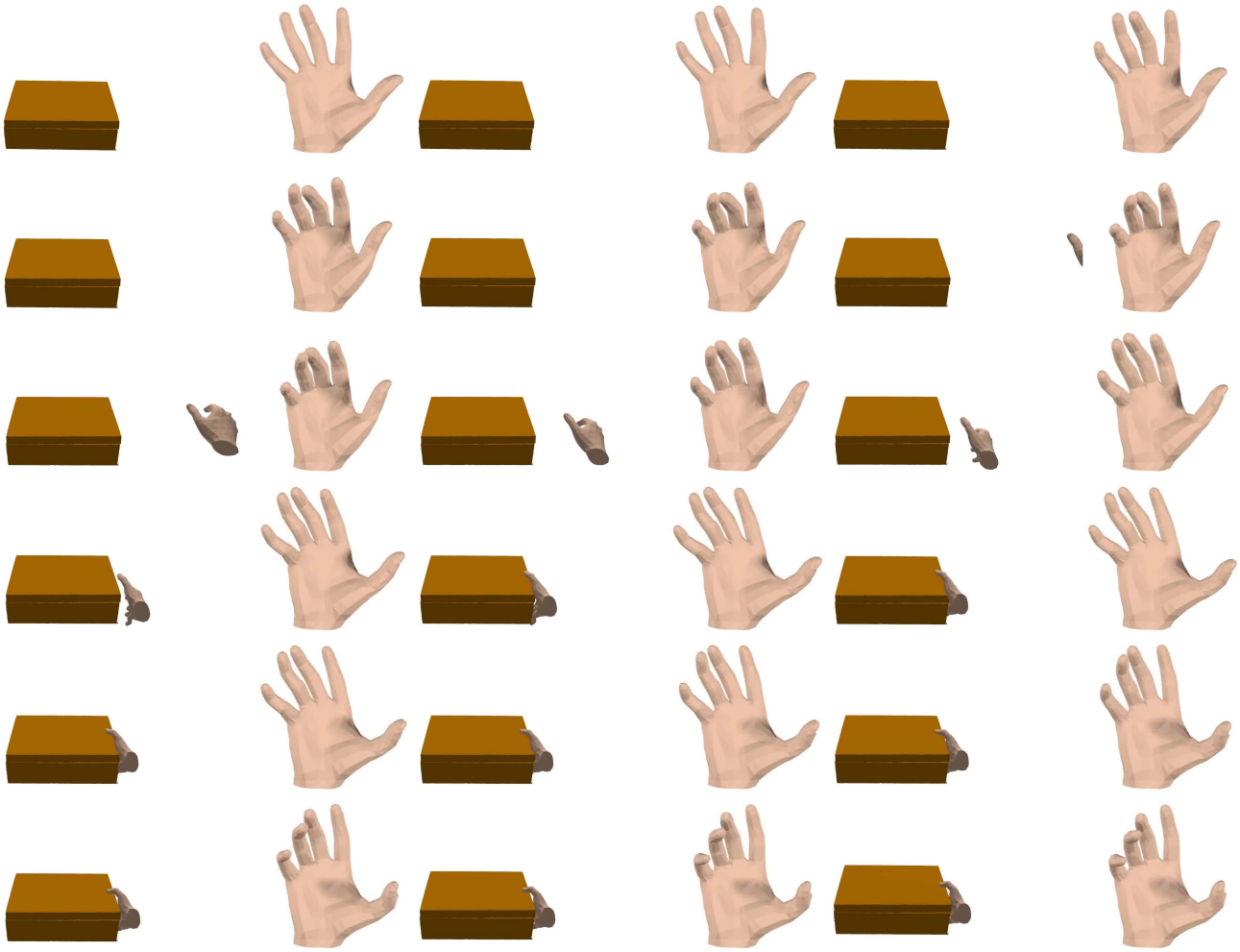

Figure 9. Generated motions (left) and hands in MANO canonical space (right) for the text prompt, *'Grab a box with the right hand.'*. The sequence is from left to right and top to bottom.

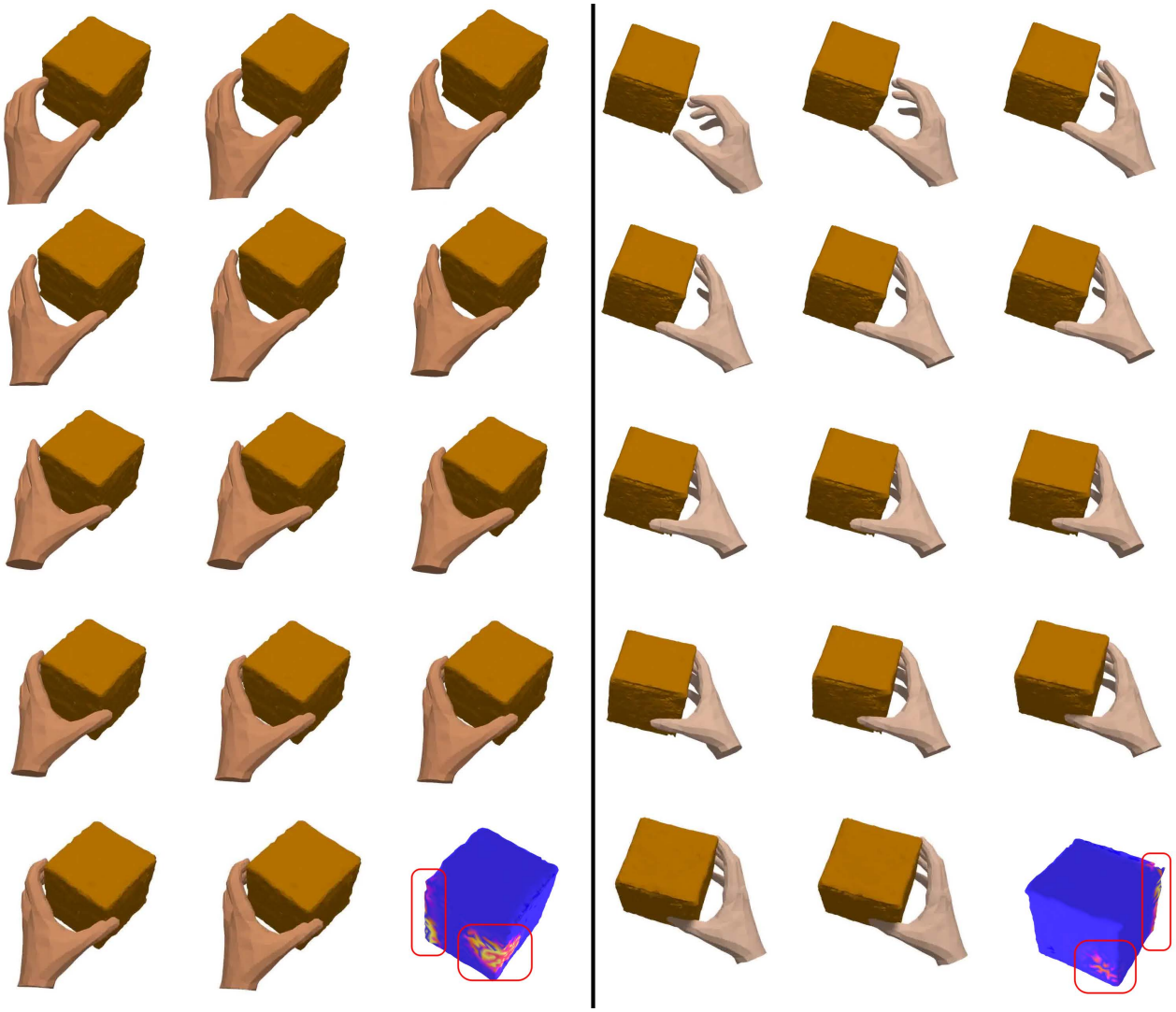

Figure 10. The different hand type results for the text prompts *'Grasp a cappuccino with the left hand.'* and *'Grasp a cappuccino with the right hand.'*. The sequence is from left to right and top to bottom.