Towards Natural Prosthetic Hand Gestures: A Common-Rig and Diffusion Inpainting Pipeline

Seungyup Ka[1], Taemoon Jeong[1], Sunwoo Kim[3], Sankalp Yamsani[2], Joohyung Kim[2], Sungjoon Choi[1]*

Abstract— **Existing works on prosthetic hands focus on increasing dexterity by carrying out functional tasks. Achieving specific hand movements, such as pointing the index finger, are desired but research on generating the hand movement itself has yet to be widely explored. In this work, we propose a pipeline for generating hand motion from body motion via using the Common-Rig, a kinematic rig representation for effective motion representation, and a diffusion-based inpainting method, which has shown strengths in generalization and stability. Common rigging is applied to a motion capture dataset with both body and hands information, and hand motions are generated while conditioned on the body motions of a hand-zeroed test set. The generated results of our proposed method, compared to two baseline methods, attain smaller fingertip positional errors and diversity closer to that of the ground truth. In addition, the generated motions are implemented on a real robotic system with prosthetic hands for evaluation.**

I. INTRODUCTION

The hands provide great competence to carry out various functions in life, ranging from precision-grip and power-grip tasks to performing delicate movements with one's fingers. The hand is comprised of 21 Degrees of Freedom (DoF) and fully utilizes such a complicated structure to execute the wide variety of functions [1]. Losing a hand can be devastating as it greatly limits a person's ability to perform such tasks. Prosthetic hands, while not perfect, can offer a solution for upper limb amputees to regain some of these abilities.

Research on prosthetic hands has focused on two main goals: increasing functionality to mimic the dexterity of human hands and improving hardware for better user convenience [2]–[7]. This includes consideration of various grasp tasks and objects needed for daily activities or reducing the weight of the hand by modeling the hands as underactuated systems [2], [7]. However, despite these efforts, 50% of unilateral limb amputees and 34.4% of bilateral limb amputees, at some point in time, have abandoned the use of upper limb prosthetics due to the limited functionality, discomfort, and fatigue [8]. Actions requiring delicate movements in the distal extremities [9] or the control of individual fingers, such as pointing the index finger, are highly desired [1], yet research on creating the prosthetic hand movement itself remains underexplored.

In this work, we focus on the problem of generating hand motions from body motions in order to effectively create hand movements that are feasible to be applied to actual prosthetic hands. We propose a common rigging method, which utilizes the kinematic priors of the prosthetic hands and the human skeleton, for the pre-processing of the dataset. Additionally, in order to generate hand motions conditioned on body motions, we utilize a diffusion-based inpainting method, which has shown great strengths in generating diverse results in the domain of images.

To be specific, the Common-Rig, which contains a rigid body structure with pre-defined link lengths, is used to retarget motions from a motion capture dataset that contains both body and hand movements. From the motions of the Common-Rig, the diffusion-based inpainting method is used to generate hand motion from body motion and a hand coupling method, that accounts for an underactuated system in simulation, is applied to restrict joint movements within the same finger. To evaluate the fidelity of the generated hand motion, we compare the total distance of the fingertip positions from the ground truth and the diversity of the generated motions to two baseline models.

Our proposed method is extensively evaluated on a real robotic system, which consists of two prosthetic hands connected to a dual-arm base. We demonstrate the applicability of our method on the PSYONIC ability hand [10], by obtaining the joint values of the Common-Rig through the generation process in simulation and transferring the values of the corresponding joints to the real robotic system.

The main contributions of our method are threefold. Firstly, we present the Common-Rig, an effective kinematic rig representation that reduces the number of joints representation and infeasibility of the generated motion. Secondly, to tackle the inherent one-to-many property of the hand motion generation task, we utilized a diffusion model, which has shown its great strengths in handling such problems. Thirdly, to evaluate the feasibility of the generated motions on a real prosthetic hand, we visualize our generated motions obtained from simulation on a real robotic system.

II. PRELIMINARIES

Generative methods have been widely used to make hand motions from body motions (e.g., Body2Hands [11]). While GAN-based methods have been mainly used, our proposed method uses a diffusion-based method which is known to be easier to train and have better performances. In particular, we treat our problem as an inpainting problem where we aim to fill in the missing hand motion from the body-only motion.

[1] Seungyup Ka, Taemoon Jeong and Sungjoon Choi are with the Department of Artificial Intelligence, Korea University, Seoul, Korea (email: seungyup-ka, taemoon-jeong, sungjoon-choi@korea.ac.kr)

[2] Sankalp Yamsani and Joohyung Kim are with the Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Champaign, IL, USA (email: yamsani2,joohyung@illinois.edu)

[3] Sunwoo Kim is with the Graphics AI Lab., NC Research, NCSOFT Corporation, Seongnam, Korea (email: sunwookim@ncsoft.com)

*Corresponding author

A. Body2Hands

Body2Hands [11] adopts a GAN-based structure to generate hand gestures from upper body movements in a conversational setting. The GAN-based model is trained using a large-scale dataset that contains upper body and hand poses extracted from in-the-wild internet videos using a monocular 3D pose estimation algorithm. From a sequence of body poses $\mathbf{B} = \{\mathbf{b}\}_{1:L}$ where L refer to the length of the sequence, the generator model $\mathcal{G}$ aims to generate a sequence of hand poses $\mathbf{H} = \{\mathbf{h}\}_{1:L}$.

$$\mathbf{H} = \mathcal{G}(\mathbf{B}) \tag{1}$$

The generator model $\mathcal{G}$ of Body2Hands consists of a body encoder, a UNet-based encoder and a hand decoder. The body encoder outputs a body embedding from the body joint rotation input and the UNet-based encoder learns the dynamics of the body. With the output of the UNet-based encoder as the input, the hand decoder recovers the hand joint rotation. The generator model $\mathcal{G}$ aims to regress the hand joint rotation of the original dataset $\hat{H}$ using the following loss term:

$$\mathcal{L}_{L1}(\mathcal{G}) = \|\hat{\mathbf{H}} - \mathcal{G}(\mathbf{B})\|_1 \tag{2}$$

The discriminator model $\mathcal{D}$ aims to differentiate the realistic hand movements from the unrealistic ones, in order to create more natural hand gestures. The discriminator $\mathcal{D}$ maximizes the following adversarial loss, while the generator $\mathcal{G}$ aims to minimize it:

$$\mathcal{L}_{GAN}(\mathcal{G}, \mathcal{D}) = \mathbb{E}_{\mathbf{H}}[\log\mathcal{D}(\mathbf{H})] + \mathbb{E}_{\mathbf{B}}[\log\mathcal{D}(1 - \mathcal{G}(\mathbf{B}))] \tag{3}$$

Overall, the full objective of Body2Hands is as follows:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}_{GAN}(\mathcal{G}, \mathcal{D}) + \lambda\mathcal{L}_{L1}(\mathcal{G}) \tag{4}$$

B. Denoising Diffusion Probabilistic Models (DDPM)

Identical to other generative models, DDPM [12] aims to learn the distribution of the training dataset. A diffusion process, with a predetermined variance schedule β_t, is implemented to transform the original data x_0 to an isotropic Gaussian noise $x_T \sim \mathcal{N}(0, 1)$ by injecting noise every step for T time steps. The forward diffusion process at each step is defined by:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t x_{t-1} I) \tag{5}$$

DDPM is trained to learn the reverse diffusion process, which predicts the reverse of (5) at each time step t via a Gaussian distribution with mean μ_θ and variance Σ_θ:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{6}$$

A distinct property of the forward diffusion process (5) is that x_t at an arbitrary time step t can be sampled using x_0 since the noising step is Gaussian and independent at each time step. By accumulating the variance schedule as $\bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s)$, the sampled x_t can be obtained as follows:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \tag{7}$$

C. RePaint

The goal of RePaint [13] is to inpaint the missing pixels of an image when the mask m that denotes the missing pixels is identified. RePaint leverages an pretrained DDPM [12] and samples the masked region by conditioning on the unmasked region during the reverse diffusion process. During the generation process, when predicting x_{t-1} from x_t, the unmasked regions are predicted from by sampling using x_0 and the masked regions are obtained from going through a single reverse diffusion process step using the pretrained model. The mask matrix m has value 1 for the known pixels and 0 for the missing pixels, whereas $*$ represents pointwise multiplication.

$$x_{t-1}^{unmasked} \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \tag{8}$$

$$x_{t-1}^{masked} \sim \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{9}$$

$$x_{t-1} = m * x_{t-1}^{unmasked} + (1 - m) * x_{t-1}^{masked} \tag{10}$$

If the equations (8) to (10) are applied directly from time step T to 0, only the content type of the generated masked region matches the unmasked region (i.e., furry texture of the dog is generated when the face of the dog is masked). In order to harmonize the masked region with the unmasked region, a resampling strategy, that repeats forward diffusion processes in between reverse diffusion processes, is applied.

III. PROPOSED METHOD

The main objective of our proposed method is to recover the full body and hands sequence from the partial body-only sequence. Fig. 1 shows the overall pipeline of our generation process. In the 'Dataset Rigging' process, the joint positions of the Common-Rig is matched with the joint positions of the motion capture skeleton to extract the motions of the Common-Rig. The rigging process allows for the consideration of the kinematic priors of the human motion and reduces the total DoFs required for the motion representation.

During the 'Hand Motion Generation' process, diffusion-based inpainting, which has shown great strengths in generalization and being easy to train, is used to fill in the missing hand motion from the body-only motion. To do so, the rigged motions is first split into a train set and a test set. DDPM [12] is used to train a diffusion model on the train set and RePaint [13], along with the pretrained diffusion model, is used to generate the hand motions from the body-only motions of the test set. Finally, a Gaussian filter is applied to the generated results to ensure smoother transitions.

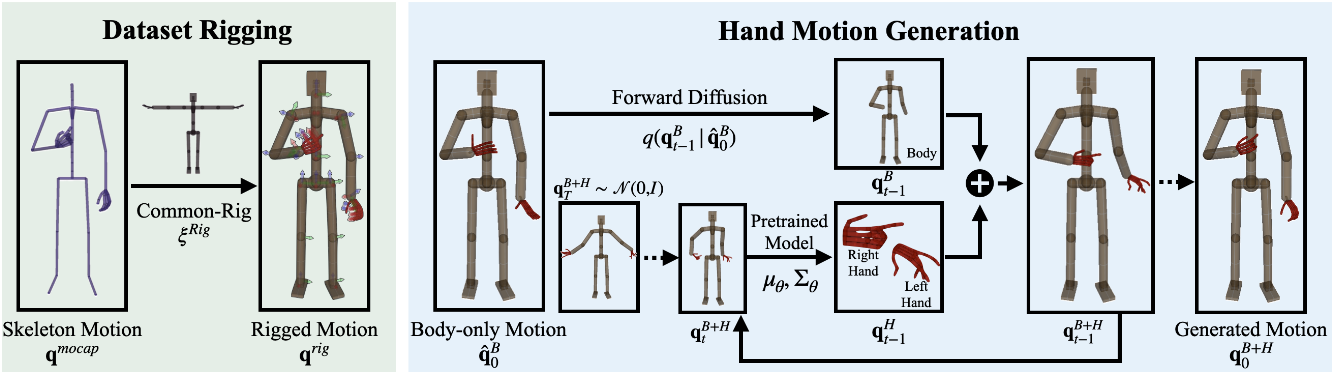


Fig. 1: The overall pipeline of our proposed method. The 'Dataset Rigging' process converts the motion from the motion capture dataset to the motion of the Common-rig and each arrow indicate a single revolute axis. The 'Hand Motion Generation' process inpaints the missing hand motion of the body-only motion using a diffusion-based inpainting method.

A. Notations

The notations used in this paper are as follows:

- $\mathbf{q}_{1:L}^{\text{mocap}} \in \mathbb{R}^{L \times N_m \times 3}$: The joint angles sequence of the motion capture skeleton. L refers to the length of the sequence, N_m refers to the number of joints of the rig and each joint is represented using three Euler angles.
- $\mathbf{q}_{1:L}^{\text{rig}} \in \mathbb{R}^{L \times N_r \times 2}$: The joint angles sequence of the Common-Rig. L refers to the length of the sequence, N_r refers to the number of joints of the rig and each joint value in represented using a trigonomical embedding, containing of a pair of cosine value and sine value.
- $\mathbf{T}_{1:L}^{\text{mocap}} \in \text{SE}(3)^{L \times N_m}$: The homogeneous transformation matrix sequence of the motion capture skeleton. L refers to the length of the sequence and N_m refers to the number of joints of the motion capture skeleton.

B. Common Rigging

The pose of a skeleton is usually represented by a combination of a root pose (i.e., root position and orientation) and the local joint offsets. As the link lengths between joints are usually preserved, a single skeleton pose can be represented by N_J rotation matrices where N_J is the number of joints. However, in the case of the elbow joint or a single finger joint, the rotation occurs in single axis and does not require multiple rotational values. A limitation of the rotation matrix representation is that such constraints cannot be considered. By utilizing the Common-Rig, the rotations are represented with a single value per axis, preventing infeasible human motions and reducing the number of joint angles for representation.

Using the joint offsets of the motion capture skeleton and $\mathbf{q}_{1:L}^{\text{mocap}}$, a forward kinematic process is carried out to obtain the homogeneous transformation matrix sequence $\mathbf{T}_{1:L}^{\text{mocap}}$. For the inverse kinematic process, positional targets of 52 joints of the motion capture skeleton were used. The joints include 14 joints for the body (Right Pelvis (RP), Right Knee (RK), Right Ankle (RA), Left Pelvis (LP), Left Knee (LK), Left Ankle (LA), Spine, Neck, Right Shoulder (RS), Right Elbow (RE), Right Wrist (RW), Left Shoulder (LS), Left Elbow (LE), Left Wrist (LW)) and 19 positional joints for each hand (Thumb: Metacarpophalangeal joint(MP), Interphalangeal joint (IP), tip / Remaining 4 fingers: MP, Distal Interphalangeal joint (DIP), Proximal Interphalangeal joint (PIP), tip). The task space position of the corresponding joints of the Common-Rig was matched to the aforementioned positional targets to obtain $\mathbf{q}_{1:L}^{\text{rig}}$. Both the forward kinematics process and inverse kinematics process are executed in the MuJoCo [14] simulator.

After the joint values of the Common-Rig are obtained, joints within the same finger that accounts for flexion/extension are coupled via a rule-based method. The angle values of the associated joints are added up and distributed via a specific ratio shown in Table I. The ratios were selected in a heuristic manner to imitate the human hand motion.

TABLE I: Ratio for hand coupling

Thumb	MP:IP = 1:1
Index	MP:DIP:PIP = 1:3:2
Middle	MP:DIP:PIP = 1:3:2
Ring	MP:DIP:PIP = 1:3:2
Pinky	MP:DIP:PIP = 1:3:2

C. Hand Motion Generation

Following the main assumption of Body2Hands [11] that upper body movements are sufficient for the inference of the hand movements, we only use the joint values of the shoulder, elbow, and wrist for the body representation. From the finger-coupled rigged dataset, only the upper body and hands joints are retrieved to form the subset $\mathbf{q}_0^{B+H}$, where the subscript 0 indicates the diffusion time step. For the convenience of notation, the variables related to the frame number and sequence length would be omitted during the diffusion processes and specified in the Experiments section.

Subsequently, the dataset is split into a train set and a test set, and a diffusion model is trained on the train set using DDPM [12] to obtain the mean μ_θ and variance Σ_θ. Using the pretrained diffusion model and the hands-zeroed test set, hand motion inpainting from the body motion is executed using RePaint [13]. We particularly selected the diffusion-based method for our hand generation task due to its ability to capture more diversity while requiring less parameters to train compared to GAN-based methods.

Throughout the diffusion time steps, the body joint values are obtained by sampling from the body-only motion $\hat{\mathbf{q}}_0^B = m * \mathbf{q}_0^{B+H}$ and the hand joint values are obtained through the pretrained model μ_θ and Σ_θ. The mask m with values 1 at the upper body joints and 0 for the hand joints is specified in advance.

$$\mathbf{q}_{t-1}^B \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\hat{\mathbf{q}}_0^B, (1-\bar{\alpha}_t)I) \tag{11}$$

$$\mathbf{q}_{t-1}^H \sim \mathcal{N}(\mu_\theta(\mathbf{q}_t^{B+H}, t), \Sigma_\theta(\mathbf{q}_t^{B+H}, t)) \tag{12}$$

$$\mathbf{q}_{t-1}^{B+H} = m * \mathbf{q}_{t-1}^B + (1-m) * \mathbf{q}_{t-1}^H \tag{13}$$

The resampling strategy (Fig. 2) is included to ensure the coherence of the body and hand motions, whereby the parameters are as follows: diffusion time steps $t_T = 1000$, jump length $j = 10$, resample $r = 10$. Starting from t_T steps, after every j reverse diffusion steps, j steps of forward diffusion steps are applied, and the reverse and forward steps are repeated for r times.

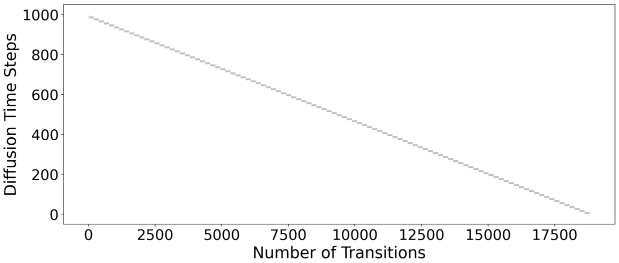


Fig. 2: Resampling time steps during the generation process. In order to increase the coherence of the generated hand motion with the original body motion, forward diffusion steps are alternated in between the reverse diffusion steps.

D. Hand Motion Smoothing

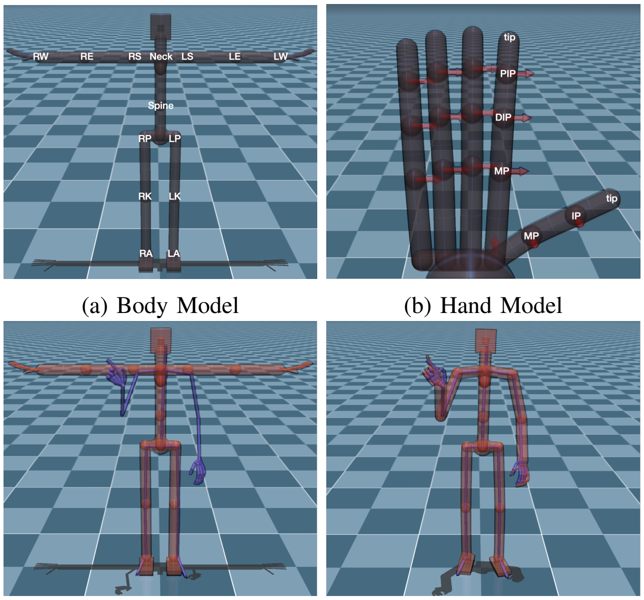
Since RePaint [13] is originally applied on the domain of images, the dimensions of the generated results are fixed to that of the train set. As such, during our generation process, motion sub-sequence of a fixed length of 64 frames is generated each time. Since the generation of a single sub-sequence does not affect the generation of its neighboring sub-sequences, abrupt changes of the joint values may occur at the boundaries when concatenating the generated results. In order to ensure smooth transitions between the generated sub-sequences, a Gaussian filter with a standard deviation of two is applied to the concatenated sequence.

IV. EXPERIMENTS

A. Dataset Preparation

In this paper, we utilized the NCSOFT Mocap dataset, a dataset containing both body and hands movements captured using motion capture devices at 60Hz. Each frame of the dataset contains XYZ positions and XYZ rotations of $N_m =$ 76 joints, while the Common-Rig is composed of $N_r = 70$ joints. The body limb lengths of the Common-Rig are matched with the positional offsets of the NCSOFT Mocap dataset, and the hand limb lengths are matched with that of the PSYONIC Ability hand [10], with an additional limb per finger to match the number of joints in the dataset.

For the joint representation of the hand model, there are a total of 30 joints (15 joints for each hand: one joint for thumb opposition, two joints for flexion/extension of the thumb, three joints for flexion/extension of each of the remaining four fingers). The default T-pose of the skeletal rig and the hand model with the corresponding joints used in the inverse kinematic process is visualized in MuJoCo [14], as shown in Fig. 3A and 3B. In addition, an example of the dataset rigging process, starting from the default T-pose to a specific position of the dataset, is shown in Fig. 3C and 3D.



(a) Body Model

(b) Hand Model

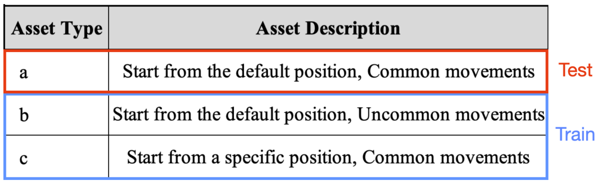
(c) Before Rigging

(d) After Rigging

Fig. 3: Common-Rig and Common rigging process. (a) The default T-pose of the Common-Rig and the body joints used for the inverse kinematics process. (b) The hand model of the Common-Rig and the hand joints used for the inverse kinematics process. (c) The Common-Rig pose before the rigging process. The purple skeleton indicates a specific pose of the motion capture dataset and the orange rig indicates the Common-Rig. (d) The Common-Rig pose after the rigging process.

The NCSOFT Mocap dataset is comprised of 80 motion sequences, with three motion sequences for each of 27 instruction labels (with the exception of the one instruction label that has 2 motion sequences). The labels include explanations of actions such as ‘Point a specific location with the finger’ and ‘Cover the mouth with the hand’, and an actor was guided to perform such an action. The same instruction was given for motion sequences of different assets, and the descriptions of the assets are shown in Fig. 4.

After applying hand coupling, the total dataset is split



Asset Type	Asset Description	
a	Start from the default position, Common movements	Test
b	Start from the default position, Uncommon movements	Train
c	Start from a specific position, Common movements	

Fig. 4: Train-test split of the dataset and the description of the three assets. Motions of asset ‘b’ and ‘c’ comprise the train set and motions of asset ‘a’ comprise the test set.

into a train set and a test set, where motions of assets ‘b’ and ‘c’ are classified as the train set and the body-only motions of asset ‘a’ are classified as the test set. Following Body2Hands [11], the motion sequences are divided into sub-sequences of 64 frames with an overlap of 32 frames in between two consecutive sub-sequences.

B. Generation Settings

For the generation setting of our method, the motion of the motion capture skeleton is converted into the motion of the Common-Rig. The upper body motion is represented by 14 joints (7 joints per arm: 3 shoulder joints, 1 elbow joint, 3 wrist joints), while the hand motion is represented by 30 joints, with each joint angle represented using a trigonometrical embedding. Overall, the train set comprises of 407 sub-sequences $\{\mathbf{q}_0^{B+H}\}_{i:i+64} \in \mathbb{R}^{64\times44\times2}$ and the test set comprises of 188 sub-sequences $\{\hat{\mathbf{q}}_0^{B}\}_{i:i+64} \in \mathbb{R}^{64\times44\times2}$, for $i = 0, 32, 64, \cdots$.

After the hand motion have been inpainted on the body motion sub-sequences in the test set, the generated sub-sequences of the same sequence are concatenated and smoothing is applied to the entire sequence. The smoothing process alleviates the jitters and abrupt changes, increasing the feasibility of the generated motion to be applied on the real robotic system.

In order to validate the fidelity of our generated results, we compare the generated results of our pipeline to two baselines: a Supervised Learning (SL) approach, that consists of a Multi-Layer Perceptron with two hidden layers of dimension size 256 and Body2Hands (B2H) [11]. For the generation setting of SL and B2H, the motions of the motion capture skeleton are used. The upper body motion is represented with 6 joints (shoulder joint, elbow joint, wrist joint for each arm) and the hand motion is represented by 28 (Thumb: MP joint, IP joint / Remaining 4 fingers: MP joint, DIP joint, PIP joint) joints, with each joints having three Euler angle values.

For SL and B2H, only the train set is split into sub-sequences of 64 frames and the entire body sequence is used to generate the entire hand sequence at once for the test set. Overall, the train set comprises of 407 sub-sequences $\{\mathbf{q}^{mocap}\}_{i:i+64} \in \mathbb{R}^{64\times34\times3}$, for $i = 0, 32, 64, \cdots$. During the inference phase, the upper-body sequences $\{\mathbf{b}^{mocap}\}_{1:L} \in \mathbb{R}^{L\times6\times3}$ are used to generate the hand sequences $\{\mathbf{h}^{mocap}\}_{1:L} \in \mathbb{R}^{L\times28\times3}$, where L is the total length of each sequence.

C. Experimental Results

Two metrics are used to compare the generated results of the three frameworks: the total distance of the 10 fingertip positions from the ground truth in task space and the diversity of the generated motions. The results are summarized in Table II and Table III. For the comparison of the distance from the ground truth (GT), the distance is accumulated throughout each sequence, then divided by the total number of frames and averaged between the 10 fingers. For the diversity of the generated results, each motion sequence is split into K sub-sequences of 64 frames, and the distances of the fingertip positions from the wrist position $\{\mathbf{d}_1, \cdots, \mathbf{d}_K\}$ is calculated. The diversity measure of each sequence is defined as follows:

$$\text{Diversity} = \frac{1}{N_K}\sum_{i=1}^{K}\sum_{j=i+1}^{K}\|\mathbf{d}_i - \mathbf{d}_j\| \tag{14}$$

where $N_K = \frac{K(K+1)}{2}$.

TABLE II: Distance from the Ground Truth
(Mean ± Standard Deviation)

	Ours	SL	B2H
L2 Distance (mm)	16.15±8.91	41.49±9.30	42.09±8.71

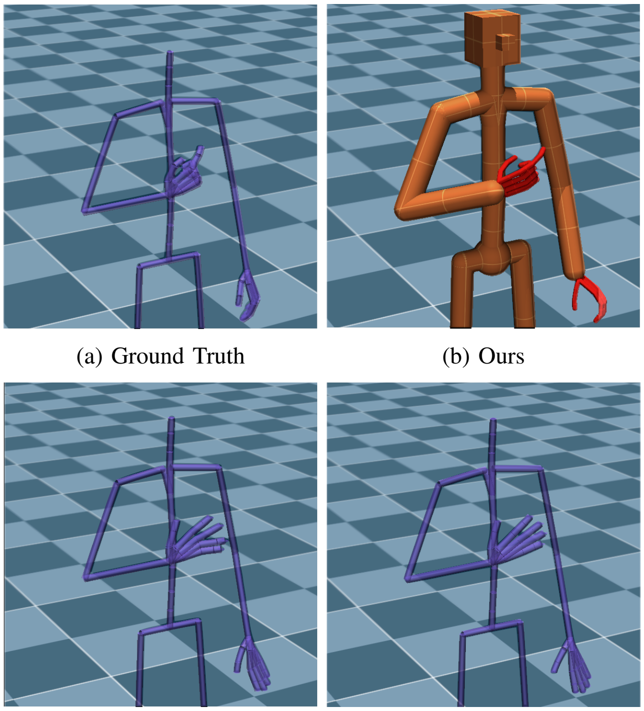
TABLE III: Diversity (Mean ± Standard Deviation)

	GT	Ours	SL	B2H
Diversity	2.47±2.75	3.43±2.84	0.25±0.19	0.11±0.09

As shown in Table II, our proposed method outperformed the two baselines, whereby the average finger distance from the GT throughout the motion sequence was lesser than half. Whereas our method could achieve movements of individual fingers, SL and B2H had a tendency of generating fairly minimal movements throughout the sequence, due to the small size of the dataset. As such, a bigger difference in fingertip positions was induced for the two baselines, while our method was able to capture more semantic information about the hand motion.

The shortcomings of SL and B2H producing minimal movements, resulting in a relatively constant hand motion, can also be observed in the diversity measure shown in Table III. Whereas the original GT motion contains diverse motions where the distance of the fingertips from the wrist varies in between frames within the same sequence, the distance values were almost constant for SL and B2H, resulting in a low diversity measure. Our method produced an even higher diversity compared to the ground truth, as the diffusion model created minute jiggly hand motions of each fingers even when the hand was relatively still. Nevertheless, the diversity measure was closer to the ground truth compared to the two baselines, showing that our method was able to generate diverse hand movements similar to that of the ground truth.

The largest difference was observed in motion 15a with the instruction label “Point oneself”. Starting from a hand pose with open palms, our method was the only one that was able to bend the middle, ring, and pinky fingers to produce an index-pointing pose, whereas the two baselines had their fingers open throughout the motion. The generated results of motion 15a are visualized in Fig. 5.


(a) Ground Truth

(b) Ours

(c) Supervised Learning

(d) Body2Hands

Fig. 5: Ground truth and the generated results of motion 15a, with the instruction label "Point oneself", for our proposed method and the two baselines. Amongst the three generated results, our method was the only one that could produce a index finger pointing motion.

D. Real Robotic System

The dual-arm system consists of a base with two PAPRAS [15] arms each connected to a PSYONIC Ability hand [10]. In order to validate the generated motions of our pipeline, the generated joint angles are transferred to the robotic system to playback the motion. However, PAPRAS arms have six DoFs while our body representation have seven DoFs for each arm, resulting in slightly varied transferred motion. In particular, one out of the three joint angles of the shoulder is excluded while transferring the joint angles. The results of the transferred real robot motion and the corresponding generated poses in simulation are shown in Fig. 6.

E. Limitations

The biggest limitation of the current study is the lack of sufficient training motions. NCSOFT Mocap Dataset only contains 80 sequences of motions, in which we only use 53 motions for training and the remaining 27 motions are used for the generation process. This deficiency of training motions greatly limits the full capacity of generative models that are effective in generating diverse results. Since the pretrained diffusion model is biased to the train dataset that contains mainly hand motion at its default position (all five fingers spread out), the probability of the inpainted motion including specific hand motions with diverse finger movements is low. This problem can be mitigated by having a larger dataset or introducing conditioning during the generation process, which is more feasible to implement in diffusion models compared to other generative models.

V. CONCLUSIONS

In this work, we proposed a method that focuses on generating hand motions from body motions, using the CommonRig and a diffusion-based inpainting method. By applying hand coupling and smoothing, feasible motions applicable to prosthetic hands were generated. We compared our generated results to two baselines and showed our method could achieve the movements of individual fingers with adequate diversity. In addition, we visualized the generated motion in a real robotic system to evaluate the effectiveness of our method. For future work, we plan on devising a new generation network to produce hand motions close to real-time and validating the whole system on upper limb amputees to evaluate the usefulness of our system.

ACKNOWLEDGMENT

This work was supported by NC Research (Number NCSOFT2022-2292-K), Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program (Korea University), No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration, No. 2022-0-00612, Geometric and Physical Commonsense Reasoning based Behavior Intelligence for Embodied AI, and No. 2022-0-00480, Development of Training and Inference Methods for Goal-Oriented Artificial Intelligence Agents).

REFERENCES

[1] F. Cordella, A.L. Ciancio, R. Sacchetti, A. Davalli, A.G. Cutti, E. Guglielmelli and L. Zollo, "Literature Review on Needs of Upper Limb Prosthesis Users," *Frontiers in Neuroscience*, vol. 10, no. 209, May. 2016.

[2] K. Scott and A. Perez-Gracia, "Design of a Prosthetic Hand with Remote Actuation," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2012)*, pp. 3056-3060, 2012.

[3] M. Yoshikawa, R. Sato, T. Higashihara, T. Ogasawara and N. Kawashima, "Rehand: Realistic Electric Prosthetic Hand Created with a 3D Printer," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2015)*, pp. 2470-2473, 2015.

[4] Y. Nemoto, K. Ogawa and M. Yoshikawa, "F3Hand: A Five-Fingered Prosthetic Hand Driven with Curved Pneumatic Artificial Muscles," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2018)*, pp. 1668-1671, 2018.

[5] N. Odagaki, M. Yoshikawa, Y. Tanaka and N. Kawashima, "Rehand II: Wire-Driven Five-Fingered Electric Prosthetic Hand Utilizing Elasticity of a Cosmetic Glove," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2019)*, pp. 6661-6664, 2019.

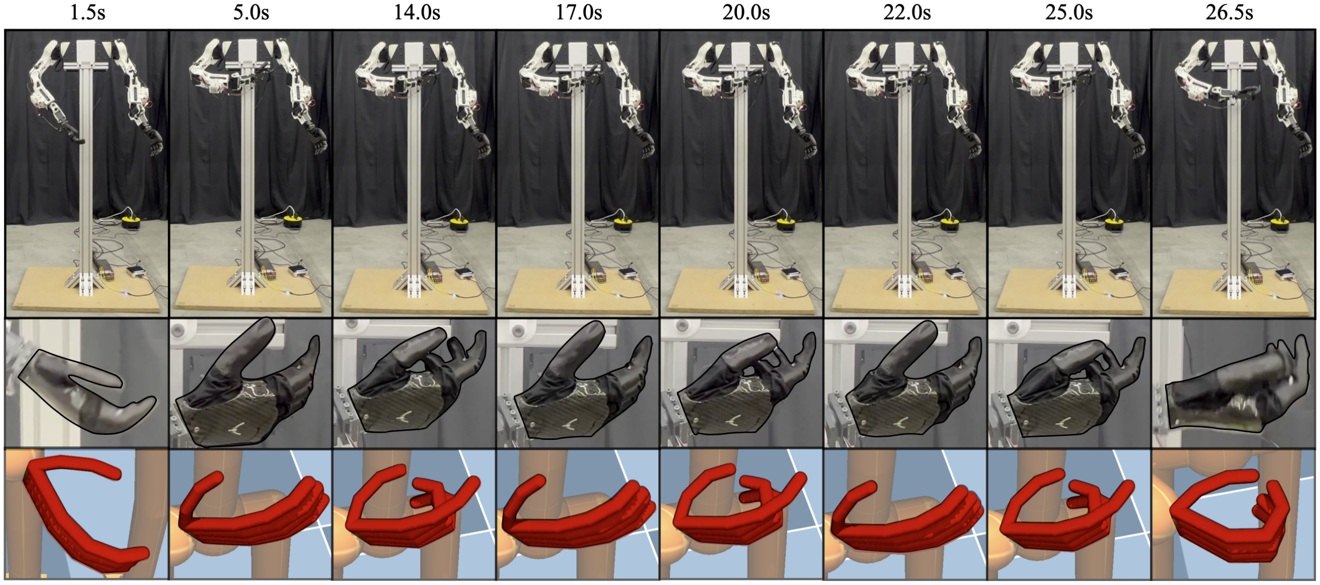


Fig. 6: Real robot results with their corresponding generated results. Row 1: Whole pose of the real robot. Row 2: Hand-zoomed poses of the real robot. Row 3: Hand-zoomed poses of the generated results visualized in simulation.

[6] Y. Yan, Y. Wang, X. Chen, C. Shi, J. Yu and C. Cheng, "A tendon-driven prosthetic hand using continuum structure," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2020)*, pp. 4951-4954, 2020.

[7] B. Busby, G. Gao and M. Liarokapis, "An Adaptive, Lightweight, Body-Powered System for Prosthetic Hands Equipped with a Selectively Lockable Differential Mechanism," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2023)*, pp. 1-7, 2023.

[8] L. Resnik, S. Ekerholm, M. Borgia and M.A. Clark, "A national study of Veterans with major upper limb amputation: Survey methods, participants, and summary findings," *PLos One*, vol. 14, no. 3, Mar. 2019.

[9] C.H. Jang, H.S. Yang, H.E. Yang, S.Y. Lee, J.W. Kwon, B.D. Yun, J.Y. Choi, S.N. Kim and H.W. Jeong, "A Survey on Activities of Daily Living and Occupations of Upper Extremity Amputees," *Ann Rehabil Med.*, vol. 35, no. 6, pp. 907–21, Dec. 2011

[10] A. Akhtar, J. Cornman, J. Austin and D. Bala, "Touch feedback and contact reflexes using the PSYONIC Ability Hand," MEC20 Symposium, 2020

[11] E. Ng, S. Ginosar, T. Darrell and H. Joo, "Body2Hands: Learning to Infer 3D Hands from Conversational Gesture Body Dynamics," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, pp. 11865-11874, 2021.

[12] J. Ho and T. Salimans, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems (NeurIPS 2020)*, pp. 6840–6851, 2020.

[13] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte and L. Van Gool, "RePaint: Inpainting using Denoising Diffusion Probabilistic Models," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)*, pp. 11461-11471, 2022.

[14] E. Todorov, T. Erez and Y. Tassa, "MuJoCo: A physics engine for model-based control," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, pp. 5026-5033, 2012.

[15] J. Kim, D. Mathur, K. Shin, S. Taylor, "PAPRAS: Plug-And-Play Robotic Arm System," *arXiv preprint arXiv:2302.09655*, 2023.