

SyncDiff: Synchronized Motion Diffusion for Multi-Body Human-Object Interaction Synthesis —Supplementary Material

Contents

A Details of Our Two Synchronization Mechanisms	1
A.1. Diffusion Model Basics	1
A.2. Complete Proof of Our Two Synchronization Mechanisms	1
A.3. Algorithm for Explicit Synchronization	5
B Supplementary Experiments	5
B.1. Results on BEHAVE [1] Dataset	5
B.2. User Study On Five Datasets	5
B.3. Determine the cutoff boundary L for Frequency Decomposition	6
B.4. Determine the Interval s for Explicit Synchronization and Computational Cost for the Process	6
B.5. Check the Reconstruction Quality of High-frequency Components	8
C Eliminated Details in the Main Text	9
C.1. Formulas for rel and comb	9
C.2. Post Process: Mesh Reconstruction	9
C.3. A Brief Introduction to BPS Algorithm	10
C.4. Details for Ablation Studies	10
C.5. Pseudocode for CSR, CRR, and CSIoU	10
D Important Statistics	11
D.1. Dataset Statistics	11
D.2. Hyperparameters in Model Architecture	11
D.3. Training and Inference Hyperparameters	12
D.4. Time and Space Cost, Hardware Configurations	12
E Limitations and Discussions	12
E.1. Current Limitations and Potential Solutions	12
E.2. Discussions and Some Extensions	13

A. Details of Our Two Synchronization Mechanisms

A.1. Diffusion Model Basics

Diffusion models [21] and its variants [4, 22], especially latent diffusion, have been widely applied in different tasks, such as high-resolution image generation [15] or video generation [5]. They simulate the data distribution by introducing a series of variables $\{x_i\}_{i=1}^T$ with different noise levels. The forward noise process can be represented as

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

where $0 < \beta_t < 1$. We can derive that

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon,$$

where $\epsilon \sim \mathcal{N}(0, I)$, and $\bar{\alpha}_t = \prod_{i=1}^t (1 - \beta_i)$.

For the inference process, from T that is large enough so that $p(x_T) \approx \mathcal{N}(0, I)$, do reverse sampling step by step, following

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I),$$

where μ_θ is the mean distribution center predicted by network, and σ_t^2 is pre-defined constant variance. Then

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t z (z \sim \mathcal{N}(0, I)).$$

Finally we can get x_0 , which is the denoised sample.

A.2. Complete Proof of Our Two Synchronization Mechanisms

The goal in this section consists of two aspects:

1) Illustrate how to define a set of alignment scores featuring synchronization analogous to the commonly used data sample scores in diffusion models, and derive the corresponding loss term $\mathcal{L}_{\text{align}}$ in main text **Section 3.5**;

2) Prove that the explicit synchronization formulas in main text **Section 3.6** are equivalent to maximum total likelihood sampling on the newly computed Gaussian distribution, where data sample scores and alignment scores are jointly considered.

Before we start, let's derive some commonly used formulas in diffusion models that we will need in our proof, which the readers might not be familiar with. If you are familiar with the step-by-step denoising formula of diffusion, you can skip directly to Eq 1 and 2.

Suppose the noise-adding process has a total of T steps, with each step's amplitude denoted by $\beta_t (t \in [T])$, we define $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$.

Then by basic principles of diffusion models, we have

$$\begin{aligned} x_t &\sim \mathcal{N}(\sqrt{\alpha_t}x_{t-1}, (1 - \alpha_t)I) \\ x_{t-1} &\sim \mathcal{N}(\sqrt{\bar{\alpha}_{t-1}}x_0, (1 - \bar{\alpha}_{t-1})I) \\ x_t &\sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \end{aligned}$$

According to Bayesian's Formula,

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1})q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

Taking negative logarithmic,

$$\begin{aligned} &-\log q(x_{t-1}|x_t, x_0) \\ &= \frac{(x_t - \sqrt{\alpha_t}x_{t-1})^2}{2(1 - \alpha_t)} + \frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)^2}{2(1 - \bar{\alpha}_{t-1})} \\ &\quad - \frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{2(1 - \bar{\alpha}_t)} + \text{Const} \\ &= \left[\frac{\alpha_t}{2(1 - \alpha_t)} + \frac{1}{2(1 - \bar{\alpha}_{t-1})} \right] x_{t-1}^2 \\ &\quad - 2 \left[\frac{\sqrt{\alpha_t}}{2(1 - \alpha_t)} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{2(1 - \bar{\alpha}_{t-1})} x_0 \right] x_{t-1} + C(x_t, x_0) \\ &\triangleq Ax_{t-1}^2 + Bx_{t-1} + C \\ &= A \left(x_{t-1} + \frac{B}{2A} \right)^2 + C' \end{aligned}$$

so

$$x_{t-1} \sim \mathcal{N}\left(-\frac{B}{2A}, \frac{1}{2A}I\right) \triangleq \mathcal{N}(\mu_t, \sigma_t^2 I)$$

where

$$\begin{aligned} \mu_t &= -\frac{B}{2A} \\ &= \frac{\frac{\sqrt{\alpha_t}}{1-\alpha_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0}{\frac{\alpha_t}{1-\alpha_t} + \frac{1}{1-\bar{\alpha}_{t-1}}} \\ &= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{\alpha_t(1 - \bar{\alpha}_{t-1}) + 1 - \alpha_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{\alpha_t(1 - \bar{\alpha}_{t-1}) + (1 - \alpha_t)}x_0 \\ &= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{1 - \bar{\alpha}_t}x_0 \end{aligned}$$

$$\text{Since } x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}x_t - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\epsilon,$$

$$\begin{aligned} \mu_t &= \left[\frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} + \frac{1 - \alpha_t}{\sqrt{\alpha_t}(1 - \bar{\alpha}_t)} \right] x_t - \frac{1 - \alpha_t}{\sqrt{\alpha_t}(1 - \bar{\alpha}_t)}\epsilon \\ &= \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}}\epsilon \right) \end{aligned} \tag{1}$$

$$\begin{aligned} \sigma_t^2 &= \frac{1}{2A} \\ &= \frac{1}{\frac{\alpha_t}{1-\alpha_t} + \frac{1}{1-\bar{\alpha}_{t-1}}} \\ &= \frac{1 - \alpha_t - \bar{\alpha}_{t-1} - \bar{\alpha}_t}{1 - \bar{\alpha}_t} \\ &= \frac{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \\ &= \beta_t \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \end{aligned} \tag{2}$$

To define alignment scores and derive $\mathcal{L}_{\text{align}}$ from it, we first need to review how traditional diffusion models derive the reconstruction loss term from data sample scores. In fact, due to the overly simple form of the reconstruction loss, its profound mathematical background is often overlooked. Like most generative models, the essence of the reconstruction loss lies in optimizing the **negative log-likelihood** $-\log p_\theta(x_0)$, where θ is the model parameters, and $p_\theta(x_0)$ is the probability of the model reconstructing the data x_0 . However, due to the step-by-step denoising mechanism for diffusion models, it is challenging to directly optimize $-\log p_\theta(x_0)$. The common approach is to optimize the ELBO (Evidence Lower Bound) [21]. Note that

$$\begin{aligned}
& -\mathbb{E}_{q(x_0)} \log p_\theta(x_0) \\
& = -\mathbb{E}_{q(x_0)} \log \left(\int p_\theta(x_{0:T}) dx_{1:T} \right) \\
& = -\mathbb{E}_{q(x_0)} \log \left(\int q(x_{1:T}|x_0) \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} dx_{1:T} \right) \\
& = -\mathbb{E}_{q(x_0)} \log \left(\mathbb{E}_{q(x_{1:T}|x_0)} \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right) \\
& \leq -\mathbb{E}_{q(x_{0:T})} \log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \\
& = \mathbb{E}_{q(x_{0:T})} \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} \right] \\
& = \mathbb{E}_{q(x_{0:T})} \left[\log \frac{q(x_T|x_0) \prod_{t=2}^T q(x_{t-1}|x_t, x_0)}{p_\theta(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t)} \right] \\
& = \mathbb{E}_{q(x_{0:T})} \left[\log \frac{q(x_T|x_0)}{p_\theta(x_T)} \right. \\
& \quad \left. + \sum_{t=2}^T \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} - \log p_\theta(x_0|x_1) \right] \\
& = \mathbb{E}_q [D_{\text{KL}}(q(x_T|x_0) \| p_\theta(x_T)) - \log p_\theta(x_0|x_1) \\
& \quad + \sum_{t=2}^T D_{\text{KL}}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t))] \tag{3}
\end{aligned}$$

Our loss is primarily the error between the distribution p_θ predicted by the model during the reverse denoising process and the true distribution q . It is well known that the KL divergence between two Gaussian distributions $\mathcal{N}(\mu_1, \sigma_1^2)$, $\mathcal{N}(\mu_2, \sigma_2^2)$ is given by

$$\begin{aligned}
& D_{\text{KL}}(\mathcal{N}(\mu_1, \sigma_1^2) \| \mathcal{N}(\mu_2, \sigma_2^2)) \\
& = \log \left(\frac{\sigma_2}{\sigma_1} \right) + \frac{\sigma_1^2 + \|\mu_1 - \mu_2\|_2^2}{2\sigma_2^2} - \frac{1}{2} \tag{4}
\end{aligned}$$

In our derivation, $q(x_{t-1} | x_t, x_0)$ means the ground-truth reverse process distribution, while $p_\theta(x_{t-1} | x_t)$ is our predicted distribution in stepwise denoising. Their mean values are referred to as μ_t and $\hat{\mu}_t$, and the standard variance σ_t is a predefined constant in DDPM [4]. Plug in $q(x_{t-1} | x_t, x_0) = \mathcal{N}(\mu_t, \sigma_t^2)$, $p_\theta(x_{t-1} | x_t) = \mathcal{N}(\hat{\mu}_t, \sigma_t^2)$, we have

$$D_{\text{KL}}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)) = \frac{1}{2\sigma_t^2} \|\hat{\mu}_t - \mu_t\|_2^2$$

In general, diffusion models predict the noise ϵ , but in our setting, we need to define the alignment loss later. Thus, it's more convenient to directly predict \hat{x}_0 , which is the result after complete denoising. Note that

$$\epsilon = \frac{1}{\sqrt{1 - \bar{\alpha}_t}} (x_t - \sqrt{\bar{\alpha}_t} x_0),$$

so

$$\begin{aligned}
\mu_t & = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon \right) \\
& = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{(1 - \alpha_t)(x_t - \sqrt{\bar{\alpha}_t} x_0)}{1 - \bar{\alpha}_t} \right) \\
& = \frac{1}{\sqrt{\alpha_t}} \left(\frac{\alpha_t(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + \frac{(1 - \alpha_t)\sqrt{\bar{\alpha}_t}}{1 - \bar{\alpha}_t} x_0 \right) \\
& = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} x_0. \tag{5}
\end{aligned}$$

Similarly,

$$\hat{\mu}_t = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \hat{x}_0.$$

Therefore,

$$D_{\text{KL}}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)) = \frac{\bar{\alpha}_{t-1}\beta_t^2}{2\sigma_t^2(1 - \bar{\alpha}_t)^2} \|x_0 - \hat{x}_0\|_2^2.$$

By removing the coefficient, it becomes our simple reconstruction loss. Note that in our implementation, we split it into \mathcal{L}_{dc} and \mathcal{L}_{ac} for separate supervision. Similarly, in order to derive the alignment loss, our approach is still to express the loss function as some form of negative log-likelihood. Consider a triplet $(\hat{x}_{\mathbf{b}_1}, \hat{x}_{\mathbf{b}_2}, \hat{x}_{\mathbf{b}_2 \rightarrow \mathbf{b}_1})$, where $\mathbf{b}_1, \mathbf{b}_2$ are two different bodies. We know that the definition of the score function is $\nabla \log p(x)$, which is just the negative gradient of the negative log-likelihood. As stated in the introduction part, we hope that this score function can guide $\hat{x}_{\mathbf{b}_2 \rightarrow \mathbf{b}_1}$ towards $\text{rel}(\hat{x}_{\mathbf{b}_1}, \hat{x}_{\mathbf{b}_2})$. A simple solution is to let $\hat{x}_{\mathbf{b}_2 \rightarrow \mathbf{b}_1}$ follow a distribution $\mathcal{N}(\mu, \sigma^2)$, where $\mu = \text{rel}(\hat{x}_{\mathbf{b}_1}, \hat{x}_{\mathbf{b}_2})$, and σ is a parameter that we can tune. Note that a vector \hat{x} satisfying $\hat{x} \sim \mathcal{N}(\hat{\mu}, \sigma^2 I)$ has the probability density function

$$p(\hat{x}) = \frac{1}{(2\pi)^{d/2} |\sigma^2|^{1/2}} \exp \left(-\frac{1}{2} (\hat{x} - \hat{\mu})^\top (\sigma^2)^{-1} (\hat{x} - \hat{\mu}) \right)$$

where d is the dimension of \hat{x} . The negative log-likelihood of $p(\hat{x})$, $-\log p(\hat{x})$, can be written as

$$-\log p(\hat{x}) = \frac{1}{2\sigma^2} \|\hat{x} - \hat{\mu}\|_2^2$$

It is not difficult to obtain that the negative log-likelihood is $\frac{1}{2\sigma^2} \|\hat{x}_{\mathbf{b}_2 \rightarrow \mathbf{b}_1} - \text{rel}(\hat{x}_{\mathbf{b}_1}, \hat{x}_{\mathbf{b}_2})\|_2^2$. Summing up for all such pairs $(\mathbf{b}_1, \mathbf{b}_2)$, and removing the coefficients, we can derive the alignment loss

$$\begin{aligned}\mathcal{L}_{\text{align}} = & \sum_{j_1, j_2 \in [1, m], j_1 \neq j_2} \|\hat{x}_{o_{j_2} \rightarrow o_{j_1}} - \text{rel}(\hat{x}_{o_{j_1}}, \hat{x}_{o_{j_2}})\|_2^2 \\ & + \sum_{i \in [1, n], j \in [1, m]} \|\hat{x}_{h_i \rightarrow o_j} - \text{rel}(\hat{x}_{o_j}, \hat{x}_{h_i})\|_2^2.\end{aligned}\quad (6)$$

Next, let's prove the equivalence between the explicit synchronization formulas and maximum total likelihood sampling in inference. In our task, during the inference process, suppose we want to derive $\hat{x}' = \hat{x}_{t-1}$ from $\hat{x} = \hat{x}_t$, we first use the denoising backbone to predict the mean value of distribution $\hat{\mu} = \hat{\mu}_t$.

In our task, $\hat{x}' = [\hat{x}'_{o_1}, \hat{x}'_{o_2}, \dots, \hat{x}'_{o_m}, \hat{x}'_{h_1}, \hat{x}'_{h_2}, \dots, \hat{x}'_{h_n}, \hat{x}'_{o_1 \rightarrow o_2}, \hat{x}'_{o_1 \rightarrow o_3}, \dots, \hat{x}'_{o_m \rightarrow o_{m-1}}, \hat{x}'_{h_1 \rightarrow o_1}, \hat{x}'_{h_1 \rightarrow o_2}, \dots, \hat{x}'_{h_n \rightarrow o_m}]$, and $\hat{\mu} = [\hat{\mu}_{o_1}, \hat{\mu}_{o_2}, \dots, \hat{\mu}_{o_m}, \hat{\mu}_{h_1}, \hat{\mu}_{h_2}, \dots, \hat{\mu}_{h_n}, \hat{\mu}_{o_1 \rightarrow o_2}, \hat{\mu}_{o_1 \rightarrow o_3}, \dots, \hat{\mu}_{o_m \rightarrow o_{m-1}}, \hat{\mu}_{h_1 \rightarrow o_1}, \hat{\mu}_{h_1 \rightarrow o_2}, \dots, \hat{\mu}_{h_n \rightarrow o_m}]$, so

$$\begin{aligned}\frac{1}{2\sigma^2} \|\hat{x}' - \hat{\mu}\|_2^2 = & \frac{1}{2\sigma^2} \left(\sum_{i=1}^n \|\hat{x}'_{h_i} - \hat{\mu}_{h_i}\|_2^2 \right. \\ & + \sum_{j=1}^m \|\hat{x}'_{o_j} - \hat{\mu}_{o_j}\|_2^2 \\ & + \sum_{j_1, j_2 \in [1, m], j_1 \neq j_2} \|\hat{x}'_{o_{j_2} \rightarrow o_{j_1}} - \hat{\mu}_{o_{j_2} \rightarrow o_{j_1}}\|_2^2 \\ & \left. + \sum_{i \in [1, n], j \in [1, m]} \|\hat{x}'_{h_i \rightarrow o_j} - \hat{\mu}_{h_i \rightarrow o_j}\|_2^2 \right)\end{aligned}$$

On the other side, the negative logarithm of alignment likelihood is defined as

$$\mathcal{P}_{\text{align}} = \sum_{v=1}^{|V|} \lambda_v \|c_v - a_v \circ b_v\|_2^2$$

where $\lambda_{v \in [1, V]}$ are a hyperparameters, $\{(a_v, b_v, c_v)\}_{v=1}^{|V|}$ is the set consisting of all triplets $(\hat{x}'_{b_1}, \hat{x}'_{b_2}, \hat{x}'_{b_2 \rightarrow b_1})$, where c_v can be computed by a_v and b_v through combination operation or relative operation. For example, if $a_v = \hat{x}'_{o_1}$ and $b_v = \hat{x}'_{o_2}$, then $c_v = \hat{x}'_{o_2 \rightarrow o_1}$, and $a_v \circ b_v = \text{rel}(\hat{x}'_{o_1}, \hat{x}'_{o_2})$. If $a_v = \hat{x}'_{o_1}$ and $b_v = \hat{x}'_{h_1 \rightarrow o_1}$, then $c_v = \hat{x}'_{h_1}$, and $a_v \circ b_v = \text{comb}(\hat{x}'_{o_1}, \hat{x}'_{h_1 \rightarrow o_1})$. The alignment likelihoods encompass the likelihoods of all binary computational relationships. Note that the alignment negative log-likelihood here is different from alignment loss $\mathcal{L}_{\text{align}}$ in the main text, which only considers relative operations rel , without combination operations comb . The intrinsic mathematical meaning of one item $\lambda_v \|c_v - a_v \circ b_v\|_2^2$ is similar to the above

derivations of $\mathcal{L}_{\text{align}}$, where we let c_v follow a Gaussian distribution with a mean of $\hat{\mu} = a_v \circ b_v$ and a variance of $\sigma^2 = \frac{1}{2\lambda_v}$.

Now consider fixing some $c_v = \hat{x}''$ (This is the motion of one single body, which is different from \hat{x} and \hat{x}' , the representation comprising of all individual/relative motions). It might be computed by K pairs of (a_k, b_k) . Take $\hat{x}'' = \hat{x}'_{o_1}$ as an example. Here $K = m - 1$, and $(a_1, b_1) = (\hat{x}'_{o_2}, \hat{x}'_{o_1 \rightarrow o_2})$, $(a_2, b_2) = (\hat{x}'_{o_3}, \hat{x}'_{o_1 \rightarrow o_3})$, \dots , $(a_{m-1}, b_{m-1}) = (\hat{x}'_{o_m}, \hat{x}'_{o_1 \rightarrow o_m})$. We need to simultaneously make \hat{x}'' as close as possible to the corresponding part $\hat{\mu}'' = \hat{\mu}_{o_1}$ in $\hat{\mu}$ predicted by the diffusion model, while ensuring that \hat{x}'' aligns with each predicted pair (a_k, b_k) . Add these terms together, maximizing total likelihood (the combination of data sample likelihood and alignment likelihoods) is equivalent to minimizing

$$\mathcal{P}_{\hat{x}''} = \frac{1}{2\sigma^2} \|\hat{x}'' - \hat{\mu}''\|_2^2 + \sum_{k=1}^K \lambda_k \|\hat{x}'' - a_k \circ b_k\|_2^2$$

A problem here is that $\hat{\mu}$ is predicted by the model based on the result of step t , but $(a_1, b_1), \dots, (a_K, b_K)$ all belong to step $(t-1)$ along with \hat{x}' . Here, we make an assumption that \hat{x}_t and \hat{x}_{t-1} are close, so that we can take $(a_1, b_1), (a_2, b_2), \dots, (a_K, b_K)$ from $\hat{x}_t = \hat{x}$.

Denote $\hat{\mu}''$ as f_0 , and $a_1 \circ b_1, a_2 \circ b_2, \dots, a_K \circ b_K$ as f_1, f_2, \dots, f_K . Here f_0, f_1, \dots, f_K are all deterministic values calculated from some certain parts of \hat{x}_t . Also let $\lambda_0 = \frac{1}{2\sigma^2}$, then

$$\begin{aligned}\mathcal{P}_{\hat{x}''} = & \sum_{k=0}^K \lambda_k \|\hat{x}'' - f_k\|_2^2 \\ = & \sum_{k=0}^K \lambda_k (\hat{x}''^\top \hat{x}'' - 2f_k^\top \hat{x}'' + f_k^\top f_k) \\ = & \left(\sum_{k=0}^K \lambda_k \right) \|\hat{x}''\|_2^2 - 2 \left(\sum_{k=0}^K \lambda_k f_k \right)^\top \hat{x}'' + \sum_{k=0}^K \lambda_k \|f_k\|_2^2\end{aligned}$$

This can be viewed as the negative log-likelihood of a new Gaussian distribution $\hat{x}'' \sim \mathcal{N}(\hat{\mu}', \sigma'^2)$, where

$$\begin{aligned}\hat{\mu}' = & \sum_{k=0}^K \frac{\lambda_k}{\sum_{k=0}^K \lambda_k} f_k \\ \sigma'^2 = & \frac{1}{2 \left(\sum_{k=0}^K \lambda_k \right)}\end{aligned}$$

Finally, it's time to consider the specific body types for calculation.

1. **For individual motions of rigid body** $o_j (j \in [1, m])$ (Here we assume that $m > 1$, otherwise there is no need for explicit synchronization on this part), relevant pairs of (a_k, b_k) consist of $(\hat{x}_{o_{j'}}, \hat{x}_{o_j \rightarrow o_{j'}}) (j' \neq j)$. $\lambda_0 = \frac{1}{2\sigma^2}$, $\lambda_1 = \lambda_2 = \dots = \lambda_{m-1} = \frac{\bar{\lambda}}{m-1}$. Here $\bar{\lambda}$ is an empirical value, satisfying

$$\bar{\lambda} = \frac{\lambda_{\text{exp}}}{R} \sum_{r=1}^R \frac{1}{2\sigma_{t_r}^2}$$

where $1 \leq t_1 < t_2 < \dots < t_R \leq T$ are the synchronization timesteps, and $\sigma_{t_1}, \sigma_{t_2}, \dots, \sigma_{t_R}$ are the original correspondent standard variances (without synchronization). The value of hyperparameter λ_{exp} can be found in Table 15. Therefore,

$$\begin{aligned} \hat{\mu}'_{o_j} &= \frac{1}{\frac{1}{2\sigma^2} + \bar{\lambda}} \left(\lambda_0 \hat{\mu}_{o_j} + \sum_{j' \neq j} \frac{\bar{\lambda}}{m-1} \text{comb}(\hat{x}_{o_{j'}}, \hat{x}_{o_j \rightarrow o_{j'}}) \right) \\ &= \frac{1}{1 + 2\sigma^2 \bar{\lambda}} \hat{\mu}_{o_j} + \frac{\frac{2}{m-1} \sigma^2 \bar{\lambda}}{1 + 2\sigma^2 \bar{\lambda}} \sum_{j' \neq j} \text{comb}(\hat{x}_{o_{j'}}, \hat{x}_{o_j \rightarrow o_{j'}}) \end{aligned}$$

2. **For individual motions of articulated skeleton** $h_i (i \in [1, n])$, relevant pairs of (a_k, b_k) consist of $(\hat{x}_{o_j}, \hat{x}_{h_i \rightarrow o_j}) (j \in [1, m])$. $\lambda_0 = \frac{1}{2\sigma^2}$, $\lambda_1 = \lambda_2 = \dots = \lambda_m = \frac{\bar{\lambda}}{m}$. Therefore,

$$\begin{aligned} \hat{\mu}'_{h_i} &= \frac{1}{\frac{1}{2\sigma^2} + \bar{\lambda}} \left(\lambda_0 \hat{\mu}_{h_i} + \sum_{j \in [1, m]} \frac{\bar{\lambda}}{m} \text{comb}(\hat{x}_{o_j}, \hat{x}_{h_i \rightarrow o_j}) \right) \\ &= \frac{1}{1 + 2\sigma^2 \bar{\lambda}} \hat{\mu}_{h_i} + \frac{\frac{2}{m} \sigma^2 \bar{\lambda}}{1 + 2\sigma^2 \bar{\lambda}} \sum_{j \in [1, m]} \text{comb}(\hat{x}_{o_j}, \hat{x}_{h_i \rightarrow o_j}) \end{aligned}$$

3. **For relative motions**, there is only one relevant pair of (a_k, b_k) , where a_k and b_k are both individual motions, which can obtain the relative motion through relative composition. Here $\lambda_0 = \frac{1}{2\sigma^2}$, $\lambda_1 = \bar{\lambda}$. Therefore,

$$\begin{aligned} \hat{\mu}'_{o_j \rightarrow o_{j'}} &= \frac{1}{\frac{1}{2\sigma^2} + \bar{\lambda}} \left(\lambda_0 \hat{\mu}_{o_j \rightarrow o_{j'}} + \lambda_1 \text{rel}(\hat{x}_{o_{j'}}, \hat{x}_{o_j}) \right) \\ &= \frac{1}{1 + 2\sigma^2 \bar{\lambda}} \hat{\mu}_{o_j \rightarrow o_{j'}} + \frac{2\sigma^2 \bar{\lambda}}{1 + 2\sigma^2 \bar{\lambda}} \text{rel}(\hat{x}_{o_{j'}}, \hat{x}_{o_j}) \\ \hat{\mu}'_{h_i \rightarrow o_j} &= \frac{1}{\frac{1}{2\sigma^2} + \bar{\lambda}} \left(\lambda_0 \hat{\mu}_{h_i \rightarrow o_j} + \lambda_1 \text{rel}(\hat{x}_{o_j}, \hat{x}_{h_i}) \right) \\ &= \frac{1}{1 + 2\sigma^2 \bar{\lambda}} \hat{\mu}_{h_i \rightarrow o_j} + \frac{2\sigma^2 \bar{\lambda}}{1 + 2\sigma^2 \bar{\lambda}} \text{rel}(\hat{x}_{o_j}, \hat{x}_{h_i}) \end{aligned}$$

For the derivation of \hat{x}' , we only need to add noise $\sigma' \epsilon (\epsilon \sim \mathcal{N}(0, I))$ to $\hat{\mu}'$, where

$$\sigma' = \sqrt{\frac{1}{2(\frac{1}{2\sigma^2} + \bar{\lambda})}} = \sqrt{\frac{\sigma^2}{1 + 2\sigma^2 \bar{\lambda}}}$$

Thus, we have completed the proof. \square

A.3. Algorithm for Explicit Synchronization

To help readers better understand the process of explicit synchronization in inference, we have specially prepared Algorithm 1 here.

B. Supplementary Experiments

B.1. Results on BEHAVE [1] Dataset

Due to space constraints in the main text, we do not demonstrate the results on the BEHAVE dataset. Here, we provide a more detailed explanation. Since the default human skeleton representation in BEHAVE is SMPL-H [17], we first converted it to SMPL-X [10] using the method provided in the official code repository. After conversion, we are able to preprocess, train, and infer in a manner similar to CORE4D.

The comparison results between SyncDiff and the two baselines on CRR, FID, and RA are shown in Table 1.

Method	CRR(% , \uparrow)	FID(\downarrow)	RA (% , \uparrow)
Ground-truth	13.02	0.03	97.78
OMOMO [7]	8.81	6.19	68.89
CG-HOI [3]	8.64	5.50	70.00
Ours	10.29	4.45	81.11

Table 1. Results on BEHAVE [1] dataset. The best in each column is highlighted in bold.

It can be observed that the comparison between our method and baselines on BEHAVE [1] is similar to that on CORE4D [24], with our method leading in both contact accuracy and semantic realism compared to OMOMO [7] and CG-HOI [3]. For visual results, please refer to our static website.

B.2. User Study On Five Datasets

As is mentioned in the experiment part of the main text, in order to evaluate semantic correctness of synthesized interactions from a perceptual aspect, we conduct a user study, comparing our method with all other baselines.

There are 10 different splits in our experiments (TACO splits 1-4, CORE4D splits 1-2, OAKINK2 test split, GRAB splits of unseen subjects/objects, BEHAVE test split). We randomly draw 15 samples per split, resulting in a total of 150 different questions (15×10). Each question requires participants to choose the best option among **Ours** and all other baselines, based primarily on ‘‘Completion, Realism, and Naturalness’’. If the decision is difficult to make, participants are also instructed to consider ‘‘additional misalignments such as interpenetration, contact loss, jitter’’. 150

Algorithm 1 Explicit Synchronization

```

1: procedure EXP_SYNC(cond)
2:    $\lambda \leftarrow 0, R \leftarrow 0$ 
3:   for  $t \leftarrow T$  to 1 do
4:     if  $t \bmod s = \lfloor s/2 \rfloor$  then
5:        $\lambda \leftarrow \lambda + \frac{1}{2\sigma_t^2}, R \leftarrow R + 1$ 
6:     end if
7:   end for
8:    $\bar{\lambda} \leftarrow \lambda_{\text{exp}} \cdot \frac{\lambda}{R}$ 
9:    $\hat{x}_T \sim \mathcal{N}(0, I)$ 
10:  for  $t \leftarrow T$  to 1 do
11:     $\hat{\mu} \leftarrow \text{Denoise}(\hat{x}_t)$ 
12:    if  $t \bmod s = \lfloor s/2 \rfloor$  then
13:       $\lambda_0 \leftarrow \frac{1}{1+2\sigma_t^2\bar{\lambda}}, \lambda_1 \leftarrow \frac{2\sigma_t^2\bar{\lambda}}{1+2\sigma_t^2\bar{\lambda}}$ 
14:       $\hat{x}' \leftarrow \lambda_0 \cdot \hat{\mu}$ 
15:      for  $j \leftarrow 1$  to  $m$  do
16:        if  $m = 1$  then
17:           $\hat{x}'_{o_j} \leftarrow \hat{x}'_{o_j} + \lambda_1 \cdot \hat{\mu}_{o_j}$ 
18:        else
19:          for  $j' \neq j$  do
20:             $\hat{x}'_{o_j} \leftarrow \hat{x}'_{o_j} + \frac{\lambda_1}{m-1} \cdot \text{comb}(\hat{x}_{o_{j'}}, \hat{x}_{o_j \rightarrow o_{j'}})$ 
21:             $\hat{x}'_{o_j \rightarrow o_{j'}} \leftarrow \hat{x}'_{o_j \rightarrow o_{j'}} + \lambda_1 \cdot \text{rel}(\hat{x}_{o_{j'}}, \hat{x}_{o_j})$ 
22:          end for
23:        end if
24:      end for
25:      for  $i \leftarrow 1$  to  $n$  do
26:        for  $j \leftarrow 1$  to  $m$  do
27:           $\hat{x}'_{h_i} \leftarrow \hat{x}'_{h_i} + \frac{\lambda_1}{m} \cdot \text{comb}(\hat{x}_{o_j}, \hat{x}_{h_i \rightarrow o_j})$ 
28:           $\hat{x}'_{h_i \rightarrow o_j} \leftarrow \hat{x}'_{h_i \rightarrow o_j} + \lambda_1 \cdot \text{rel}(\hat{x}_{o_j}, \hat{x}_{h_i})$ 
29:        end for
30:      end for
31:       $\sigma' \leftarrow \sqrt{\frac{\sigma_t^2}{1+2\sigma_t^2\bar{\lambda}}}$ 
32:      else
33:         $\hat{x}' \leftarrow \hat{\mu}$ 
34:         $\sigma' \leftarrow \sigma_t$ 
35:      end if
36:       $\epsilon \sim \mathcal{N}(0, I)$ 
37:       $\hat{x}' \leftarrow \hat{x}' + \sigma' \cdot \epsilon$ 
38:       $\hat{x}_{t-1} \leftarrow \hat{x}'$ 
39:    end for
40:  end procedure

```

different individuals are involved, and each is assigned 10 different questions, where the option orders are randomly

shuffled. Our mechanism guarantees that each question is answered by exactly 10 different agents. Figure 1 shows the times each method voted as the “best” in each split.

From the data of User Study (which is also corroborated by other metrics such as contact-based metrics and metrics regarding semantics), it can be observed that for datasets with fewer bodies, like GRAB or BEHAVE, the gap between our method and the baselines is not as significant as in datasets with more bodies. This is because, as the number of bodies increases, the requirements for synchronization become higher, and the interaction patterns between rigid objects become more complex, necessitating explicit modeling of high-frequency components through frequency decomposition, as well as those synchronization mechanisms. In such scenarios, the advantages of our method are more prominently demonstrated.

B.3. Determine the cutoff boundary L for Frequency Decomposition

In frequency decomposition (main text Section 3.3), we discard signals with frequencies higher than $\phi_L/2\pi$, where $\phi_L = L/N$, and L is the cutoff boundary. We analyzed the impact of different L on the maximum error ϵ , which is averaged on the dimension of N (number of frames) and takes the maximum across all degrees of freedom (the dimension of D_{sum}). The errors listed in Table 2 are averaged over all data samples in the datasets and are measured in millimeters.

L	TACO	CORE4D	OAKINK2	GRAB	CG-HOI
6	8.3	33.1	6.1	8.4	11.5
8	6.8	15.3	4.9	6.9	7.8
12	4.9	13.4	3.7	5.1	6.8
16	3.9	11.4	3.0	4.0	5.9
20	3.2	9.0	2.6	3.2	5.2
25	2.6	7.2	2.2	2.6	4.8

Table 2. Results for the impact of different L on maximum error.

Our goal is to minimize L , provided that the error between filtered and raw motions remains negligible relative to the original signal amplitude. To strike a balance between motion representation fidelity and simplicity, as well as filtering of too high-frequency noise in mocap datasets, in practice, we take $L = 16$.

B.4. Determine the Interval s for Explicit Synchronization and Computational Cost for the Process

In explicit synchronization (main text Section 3.6), to balance inference speed and performance, we choose a hyperparameter s ($s \ll T$), where $T = 1000$ is the total number of denoising steps, which means we only perform explicit synchronization operations every s step, $R = T/s$ times in total.

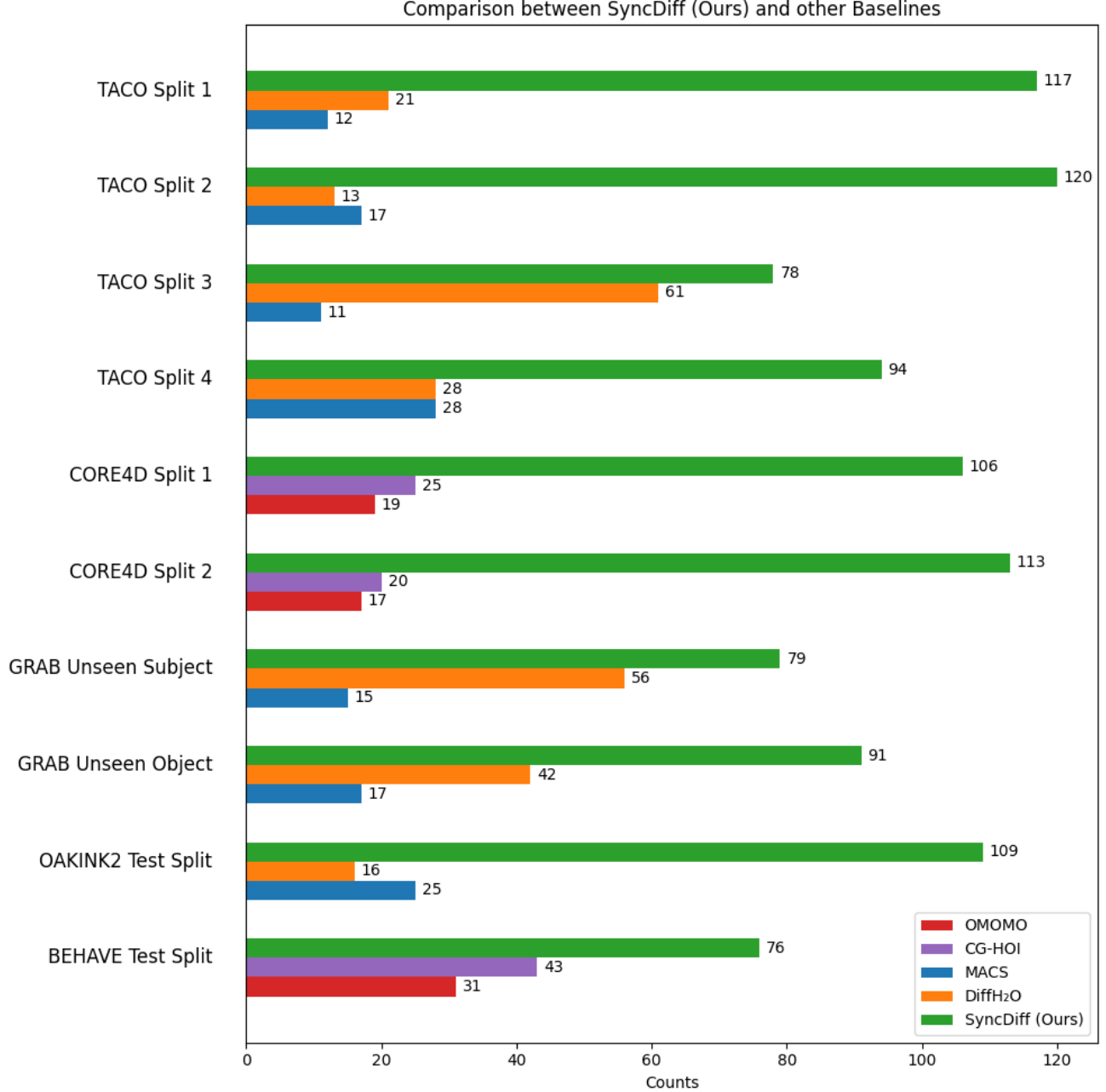


Figure 1. User study results on different dataset splits.

We conduct experiments on TACO Split 1 using different s , as shown in Table 3. RA refers to recognition accuracy. To reduce computational cost without affecting the semantic accuracy of synthesized motions too much, in practice, we choose $s = 50$.

Here we also list the computational cost comparison between SyncDiff and each baselines. We test inference speed on a single NVIDIA A40 GPU for synthesizing a 200-frame

sequence. Time consumption across different methods are shown in Tables 4 and 5.

It can be observed that computational cost of **SyncDiff** is no worse than most baselines, since specific designs like contact guidance (CG-HOI) or multi-stage synthesis (OMOMO / MACS / DiffH₂O) are circumvented. Comparison to “w/o exp sync” shows that the exp sync operation applied every $s = 50$ steps adds modest time cost of less

s	Inference time per sample(s)	RA(%, \uparrow)
1	88.5	74.04
5	22.2	73.91
10	14.2	74.09
50	7.6	73.28
100	6.7	72.15
500	6.1	70.52
w/o exp	5.9	67.27

Table 3. Results for different s on TACO Split 1.

Method \ Dataset	CORE4D	BEHAVE
OMOMO	7.1s	5.9s
CG-HOI	10.7s	7.6s
SyncDiff (Ours)	6.5s	4.2s
w/o exp sync	5.4s	3.6s

Table 4. Inference time for a 200-frame human-object-interaction sequence.

Method \ Dataset	TACO	OAKINK2	GRAB
MACS	7.4s	7.0s	5.7s
DiffH ₂ O	8.6s	8.9s	7.4s
SyncDiff (Ours)	7.7s	7.3s	6.6s
w/o exp sync	5.9s	5.6s	5.4s

Table 5. inference time for a 200-frame hand-object-interaction sequence.

than 25%.

B.5. Check the Reconstruction Quality of High-frequency Components

From the comparison between **Ours** and “w/o decompose” in the main text and supplementary videos, we can clearly see that if high-frequency components are not explicitly modeled, the motion trajectory tends to miss subtle high-frequency details, which are crucial for the accuracy of motion semantics. For example, the spatula merely contacts and gets stuck on the plate instead of moving back and forth to complete the *scrape off* motion; similarly, when a person walks, his/her legs fail to alternate properly and instead slide forward like “ice skating”.

However, frequency decomposition consists of 1) the filter of too high-frequency noise, and 2) the explicit frequency domain representation (x_F) of high-frequency components with semantics (x_{ac}), where the former is relatively trivial. We need to decouple the contributions of them in further ablation studies. Additionally, a noteworthy phenomenon is that, compared to the baselines, our method exhibits less jitter (See videos of CORE4D/OAKINK2/GRAB “comparison to baselines”), which could be attributed to ei-

ther the filter of too high-frequency noise or the two synchronization mechanisms. It is essential to investigate the core source of this effect.

Identifying the source of less jitter is relatively straightforward. It is noted that the motion trajectories for training in the “w/o decompose” scenario includes those too high-frequency noise (See Section C.4). However, as observed in the “w/o decompose” videos of TACO, there is almost no large-scale jitter in the synthesized motions, with the only issue being the absence of high-frequency interactions between objects. This is because, in the “w/o decompose” experiments, both synchronization mechanisms are effective, ensuring that the generated motions are sufficiently aligned and thus reducing jitter. Conversely, in the OAKINK2 ablation studies of synchronization, whether it’s “w/o align loss” or “w/o exp sync,” the use of frequency decomposition also removes too high-frequency noise, yet it still results in greater jitter compared to the complete SyncDiff (Ours). This sufficiently demonstrates that the contribution to less jitter is primarily due to the synchronization mechanisms, rather than the filter of too high-frequency noise. A more straightforward method of verification is to observe that the errors in Table 2 caused by the filter of too high-frequency noise are almost negligible compared to the scales of the motions, indicating that this operation theoretically has minimal impact on the model’s performance.

To decouple the effects of the frequency-domain based representation and the filter of too high-frequency noise from a more strict aspect, we conduct the following experiments. Besides “w/o decompose”, we add “**simply filter**”, which replaces x_t in “w/o decompose” with $x_{t,dc} + x_{t,ac}$, simulating the effect of filtering too high-frequency noise. “**Only dc**” replaces x_t with $x_{t,dc}$, simulating the effect of using no high-frequency signals (including those with semantics).

Experiment 1: On the four test splits of TACO, we only add noise for no larger than 200 steps (where the full train pipeline needs 1000 steps), and examine the reconstruction ability of different methods. For any predicted motion trajectory \hat{x} , we decompose it as described in main text **Section 3.3**, to obtain the high-frequency component \hat{x}_{ac} . We then examine the discrepancy between \hat{x}_{ac} and the ground truth x_{ac} . The experimental results on the four splits of TACO are shown in Table 6, proving that explicitly representing high-frequency signals with semantics in the frequency domain indeed improves the effect of reconstruction.

Experiment 2: We test the semantic quality of synthesized motions on the four test splits of TACO. The experimental setting is the same as Table 1 in the main text. Results are shown in Table 7. We can clearly see that frequency decomposition enhances the semantic quality of the synthesized motion, and this can not be achieved by merely

	Error (cm, ↓)			
Method	Test1	Test2	Test3	Test4
MACS [20]	2.43	2.37	3.11	2.74
DiffH ₂ O [2]	1.74	2.08	3.12	3.31
Ours	1.99	1.63	2.24	2.25
w/o decomp	2.68	3.28	4.12	3.72
simply filter	3.23	3.28	3.64	3.77
only dc	5.42	4.87	6.17	5.89

Table 6. Reconstruction of high-frequency components on TACO [9] dataset. The best in each column is highlighted in bold.

	FID (↓)				RA (% , ↓)			
Method	Test1	Test2	Test3	Test4	Test1	Test2	Test3	Test4
MACS [20]	10.56	23.24	32.18	42.37	58.40	53.08	33.00	19.02
DiffH ₂ O [2]	4.34	17.04	24.92	39.20	61.40	56.70	43.67	28.15
Ours	2.70	2.68	22.96	30.23	73.28	85.92	46.90	40.12
w/o decomp	6.44	21.21	28.67	49.58	56.60	51.85	40.02	22.18
simply filter	7.87	20.38	29.35	37.69	54.71	53.08	39.66	22.34
only dc	42.83	54.62	88.47	85.02	30.26	31.44	25.86	19.74

Table 7. Semantic quality of ablation studies on TACO [9] dataset. The best in each column is highlighted in bold.

removing too high-frequency noise.

C. Eliminated Details in the Main Text

C.1. Formulas for rel and comb

In main text Section 3.5, 3.6, for two distinct bodies \mathbf{a} and \mathbf{b} , we define $\text{rel}(x_{\mathbf{a}}, x_{\mathbf{b}})$ as \mathbf{b} 's motion relative to \mathbf{a} , and let $\text{comb}(x_{\mathbf{a}}, x_{\mathbf{b} \rightarrow \mathbf{a}})$ utilize the individual motion of \mathbf{a} and relative motion between \mathbf{b} and \mathbf{a} to compute \mathbf{b} 's motion. The detailed expressions are shown below:

If $\mathbf{a} = o_{j_1}$ and $\mathbf{b} = o_{j_2}$ are two rigid objects, whose motions are $x_{o_{j_1}} = [\mathbf{t}_{o_{j_1}}, \mathbf{q}_{o_{j_1}}]$ and $x_{o_{j_2}} = [\mathbf{t}_{o_{j_2}}, \mathbf{q}_{o_{j_2}}]$, denote $x_{o_{j_2} \rightarrow o_{j_1}} = [\mathbf{t}_{o_{j_2} \rightarrow o_{j_1}}, \mathbf{q}_{o_{j_2} \rightarrow o_{j_1}}]$, then $\text{rel}(x_{o_{j_1}}, x_{o_{j_2}}) = [\mathbf{q}_{o_{j_1}}^{-1}(\mathbf{t}_{o_{j_2}} - \mathbf{t}_{o_{j_1}}), \mathbf{q}_{o_{j_1}}^{-1}\mathbf{q}_{o_{j_2}}]$, $\text{comb}(x_{o_{j_1}}, x_{o_{j_2} \rightarrow o_{j_1}}) = [\mathbf{q}_{o_{j_1}}\mathbf{t}_{o_{j_2} \rightarrow o_{j_1}} + \mathbf{t}_{o_{j_1}}, \mathbf{q}_{o_{j_1}}\mathbf{q}_{o_{j_2} \rightarrow o_{j_1}}]$.

If $\mathbf{a} = o_j$ is rigid object, and $\mathbf{b} = h_i$ is an articulated skeleton, let $x_{o_j} = [\mathbf{t}_j, \mathbf{q}_j]$, and denote the motion for one of the joints in x_{h_i} as \mathbf{p}_{h_i} , the motion for one of the joints in $x_{h_i \rightarrow o_j}$ as $\mathbf{p}_{h_i \rightarrow o_j}$. We have $\text{rel}(x_{o_j}, \mathbf{p}_{h_i}) = \mathbf{q}_{j_1}^{-1}(\mathbf{p}_{h_i} - \mathbf{t}_{j_1})$, $\text{comb}(x_{o_j}, \mathbf{p}_{h_i \rightarrow o_j}) = \mathbf{q}_{o_j}\mathbf{p}_{h_i \rightarrow o_j} + \mathbf{t}_{o_j}$.

Note that \mathbf{a} must be a rigid object, in order to define a coordinate system based on its transformation matrix. More details can be found in Section E.1. Also, $\text{rel}(\hat{x}_{\mathbf{a}}, \hat{x}_{\mathbf{b}})$ is different from $\hat{x}_{\mathbf{b} \rightarrow \mathbf{a}}$. The former is calculated based on the predicted motion of \mathbf{a} and \mathbf{b} , which should approach $\hat{x}_{\mathbf{b} \rightarrow \mathbf{a}}$ in order to foster synchronization. The latter is directly predicted by the diffusion model, which should fit the distribution of ground-truth $x_{\mathbf{b} \rightarrow \mathbf{a}}$, in order to increase data fidelity. Similar arguments hold for comb .

C.2. Post Process: Mesh Reconstruction

For tasks such as robot learning, the joint information for articulated skeletons (hands/humans) is already sufficient.

Post processing methods like Reinforcement Learning (RL) or Imitation Learning (IL) can further guide the model to generate physically realistic data with the help of physics-based simulations. Therefore, in such scenarios, we can directly use $\hat{x}_{h_i \in [1, n]}$ as predicted results. However, for tasks like animation production or VR/AR, it is necessary to reconstruct the full meshes. In this section, we introduce how to reconstruct MANO hand mesh or SMPL-X human body mesh of natural shape from the predicted joint positions $\mathbb{R}^{N \times D \times 3}$, where N and D are the number of frames and joints.

MANO Hand Mesh Reconstruction. For one single hand, suppose the predicted joint positions are $\hat{x} \in \mathbb{R}^{N \times 21 \times 3}$. We start with tunable MANO [16] parameters for joint pose $\theta \in \mathbb{R}^{N \times 45}$, global orientation $R \in \mathbb{R}^{N \times 3}$ and translation $l \in \mathbb{R}^{N \times 3}$, which are all set to zero tensors initially. We freeze hand shape $\beta \in \mathbb{R}^{10}$. During the optimizing process, let the current calculated joint positions be $K \in \mathbb{R}^{N \times 21 \times 3}$ based on θ , R , and l . We want to minimize

$$\mathcal{L}_{\text{MANO}} = \lambda_{\text{pos}}\mathcal{L}_{\text{pos}} + \lambda_{\text{angle}}\mathcal{L}_{\text{angle}} + \lambda_{\text{vel}}\mathcal{L}_{\text{vel}}$$

Here

$$\mathcal{L}_{\text{pos}} = \sum_{t=1}^N \sum_{i=1}^{21} \|\hat{x}_{t,i} - K_{t,i}\|_2^2$$

which minimizes the difference between joint positions K from MANO calculation and our predicted results \hat{x} .

$$\mathcal{L}_{\text{angle}} = \sum_{t=1}^N \sum_{i=1}^{45} [\max(\theta_{t,i} - u_i, 0) + \max(d_i - \theta_{t,i}, 0)]$$

where (d_i, u_i) is the permitted rotation range for the i -th degree of freedom. This loss item ensures that the rotation angle of each joint remains within permissible limits, preventing unreasonable distortions.

$$\mathcal{L}_{\text{vel}} = \sum_{t=1}^{N-1} \|l_t - l_{t+1}\|_2^2$$

which keeps the positions at adjacent timesteps close to ensure a smooth trajectory without abrupt changes.

SMPL-X Human Body Mesh Reconstruction. Similar to MANO in hand pose representation, for human body pose representation, we also have SMPL-X [10] representation. It is composed of a set of parameters $(\theta, R, l, \theta_{\text{left}}, \theta_{\text{right}})$, where the body pose $\theta \in \mathbb{R}^{N \times 21 \times 3}$ represents the 21 joint orientations(except root), $R, l \in \mathbb{R}^{N \times 3}$ are global orientation and translation, and $\theta_{\text{left}}, \theta_{\text{right}} \in \mathbb{R}^{N \times 12}$ are compressed representations of two hands. Human shape parameters $\beta \in \mathbb{R}^{10}$ are given, while $\hat{x} \in \mathbb{R}^{N \times 22 \times 3}$ denotes the joint positions we predict.

Due to the flexibility of human body poses, to reconstruct the human body mesh, we not only need joint positions but also joint orientations. Therefore, we need to introduce additional $N \times 21 \times 3$ degrees of freedom to every x_{h_i} ($i \in [1, n]$), representing the predicted human body pose $\hat{\theta}$ under SMPL-X. These degrees of freedom are also predicted by the diffusion model, but they only participate in the decomposition process and do not get involved in our two synchronization mechanisms.

Similar to hand mesh reconstruction, we start with tunable SMPL-X parameters $R, l, \theta_{\text{left}}, \theta_{\text{right}}$, which are all set to zero tensors. We freeze hand shape $\beta \in \mathbb{R}^{10}$ and $\theta = \hat{\theta}$. During the optimizing process, let the current calculated joint positions be $K \in \mathbb{R}^{T \times 22 \times 3}$ based on $R, l, \theta_{\text{left}}, \theta_{\text{right}}$. We want to minimize

$$\mathcal{L}_{\text{SMPL-X}} = \lambda_{\text{pos}} \mathcal{L}_{\text{pos}} + \lambda_{\text{vel}} \mathcal{L}_{\text{vel}}$$

Here

$$\mathcal{L}_{\text{pos}} = \sum_{t=1}^T \sum_{i=1}^{22} \|\hat{x}_{t,i} - K_{t,i}\|_2^2$$

which minimizes the difference between joint positions from SMPL-X calculation and our predicted results.

$$\mathcal{L}_{\text{vel}} = \sum_{t=1}^{T-1} \|l_t - l_{t+1}\|_2^2$$

which keeps the trajectory smooth.

The hyperparameters involved in the mesh reconstruction process can be found in Table 8 and 9.

Parameter	TACO	OAKINK2	GRAB
Optimizer	AdamW, lr = 0.01		
Epoch	5k	8k	5k
λ_{pos}	1	1	1
λ_{angle}	0.2	0.2	0.05
λ_{vel}	0.03	0.03	0.02

Table 8. Hyperparameters for MANO hand mesh reconstruction.

Parameter	CORE4D	BEHAVE
Optimizer	AdamW, lr = 0.001	
Epoch	5k	2k
λ_{pos}	3	1
λ_{vel}	0.1	0.1

Table 9. Hyperparameters for SMPL-X human body mesh reconstruction.

C.3. A Brief Introduction to BPS Algorithm

As is discussed in the main text, we use the Basis Point Set (BPS) [11] algorithm to encode the geometric features of rigid bodies. Compared to pretrained models like PointNet [12] and PointNet++ [13], BPS is more lightweight and compact. It does not rely on any data-driven methods and places greater emphasis on object surface features, which is crucial in our relative motion synthesis.

Its working principle is as follows: First, a large enough sphere is chosen such that when its center coincides with any object’s centroid, it can fully contain the object. In practice, we choose radius $r = 1\text{m}$. Then, 1024 points are randomly sampled from the sphere’s volume. BPS representation is computed by calculating the difference from each sampled point to the nearest point on the object’s surface. This results in a vector of size $\mathbb{R}^{1024 \times 3}$.

C.4. Details for Ablation Studies

In our experiments in main text **Section 4**, there are three categories of ablation studies: “w/o decompose”, “w/o $\mathcal{L}_{\text{align}}$ ”, and “w/o exp sync”.

“w/o decompose” means that we direct concatenate the condition vector with x_t , which substitute the position of $x_{t,\text{dc}}$ in the pipeline figure, and eliminate the branch of x_{F} or x_{ac} . Note that $x_t \neq x_{t,\text{dc}} + x_{t,\text{ac}}$, as x_t includes the components with frequencies higher than $\phi_L/2\pi$, where $\phi_L = L/N$, and L is the cutoff boundary.

“w/o $\mathcal{L}_{\text{align}}$ ” means that we eliminate the term $\lambda_{\text{align}} \mathcal{L}_{\text{align}}$ in total loss.

“w/o exp sync” means that we perform normal denoising steps every time, without explicit synchronization steps. This can be equivalently interpreted as $s = +\infty$.

C.5. Pseudocode for CSR, CRR, and CSIoU

In this section, we will provide a detailed description for three contact-based metrics: CSR, CSIoU, and CRR.

First, we define two types of contact: surface contact and root contact. Here o is one single object mesh sequence of size $\mathbb{R}^{N \times M \times 3}$, where M is the number of vertices on its mesh. h is hand mesh sequence of size $\mathbb{R}^{N \times 778 \times 3}$ in Contact.Surface, while it denotes trajectories of root joints of two hands of size $\mathbb{R}^{N \times 2 \times 3}$ in Contact.Root, as shown in Algorithm 2.

Based on these contact definitions, it comes to the calculation of the three metrics. Here o is the object mesh sequence list of length m , and h is the hand mesh sequence or human hand root joint sequence list of length n . o' and h' are corresponding ground-truth versions. The calculation is shown in Algorithm 3.

Algorithm 2 Contact Definitions

```

1: procedure CONTACT_SURFACE( $o, h$ )
2:    $\mathbf{c} \leftarrow \mathbf{0}$ 
3:   for  $t \leftarrow 1$  to  $T$  do
4:      $d \leftarrow \min_{v_1 \in [1, M], v_2 \in [1, 778]} \|o_{t, v_1} - h_{t, v_2}\|_2$ 
5:     if  $d \leq 5\text{mm}$  then
6:        $\mathbf{c}_t \leftarrow 1$ 
7:     end if
8:   end for
9:   return  $\mathbf{c}$ 
10: end procedure
11: procedure CONTACT_ROOT( $o, h$ )
12:    $\mathbf{c} \leftarrow \mathbf{0}$ 
13:   for  $t \leftarrow 1$  to  $T$  do
14:      $d_1 \leftarrow \min_{v \in [1, M]} \|o_{t, v} - h_{t, 1}\|_2$ 
15:      $d_2 \leftarrow \min_{v \in [1, M]} \|o_{t, v} - h_{t, 2}\|_2$ 
16:     if  $\max(d_1, d_2) \leq 3\text{cm}$  then
17:        $\mathbf{c}_t \leftarrow 1$ 
18:     end if
19:   end for
20:   return  $\mathbf{c}$ 
21: end procedure

```

D. Important Statistics**D.1. Dataset Statistics**

The sizes of each dataset split are shown in Table 10.

Dataset	Statistics
TACO	train:test1:test2:test3:test4 =1035:238:260:403:610
CORE4D	train:test1:test2=483:197:195
OAKINK2	train:val:test=1884:167:723
GRAB	train:val:test=992:198:144 (Unseen Subject) train:test=1126:208 (Unseen Object)
BEHAVE	train:test=231:90

Table 10. Dataset statistics.

D.2. Hyperparameters in Model Architecture

Action/object Label Feature Extraction Branch. Action/object label features are first encoded by pretrained CLIP [14], and then pass through a 2-layer MLP. Specific parameters are shown in Table 11.

Object Geometry Feature Extraction Branch. Object geometry features are first encoded by pretrained BPS [11], and then pass through a 2-layer MLP. Specific parameters are shown in Table 12.

Noise Timestep Embedding Module. Detailed architecture is shown in Table 13.

Transformer Encoder-Decoder. Concatenate action

Algorithm 3 Metric Calculation

```

1: procedure CSR( $o, h$ )
2:    $\text{CSR} \leftarrow 0$ 
3:   for  $i \leftarrow 1$  to  $n$  do
4:      $\mathbf{c} \leftarrow \mathbf{0}$ 
5:     for  $j \leftarrow 1$  to  $m$  do
6:        $\mathbf{c} \leftarrow \mathbf{c} \vee \text{Contact\_Surface}(o_j, h_i)$ 
7:     end for
8:      $\text{CSR} \leftarrow \text{CSR} + \frac{1}{T} \sum_{t=1}^T \mathbf{c}_t$ 
9:   end for
10:  return  $\text{CSR}/n$ 
11: end procedure
12: procedure CSIoU( $o, h, o', h'$ )
13:    $\text{CSIoU} \leftarrow 0$ 
14:   for  $i \leftarrow 1$  to  $n$  do
15:      $\mathbf{c}_1 \leftarrow \mathbf{0}, \mathbf{c}_2 \leftarrow \mathbf{0}$ 
16:     for  $j \leftarrow 1$  to  $m$  do
17:        $\mathbf{c}_1 \leftarrow \mathbf{c}_1 \vee \text{Contact\_Surface}(o_j, h_i)$ 
18:        $\mathbf{c}_2 \leftarrow \mathbf{c}_2 \vee \text{Contact\_Surface}(o'_j, h'_i)$ 
19:     end for
20:      $\text{CSIoU} \leftarrow \text{CSIoU} + \text{IoU}(\mathbf{c}_1, \mathbf{c}_2)$ 
21:   end for
22:  return  $\text{CSIoU}/n$ 
23: end procedure
24: procedure CRR( $o, h$ )
25:    $\text{CRR} \leftarrow 0$ 
26:   for  $i \leftarrow 1$  to  $n$  do
27:      $\mathbf{c} \leftarrow \mathbf{0}$ 
28:     for  $j \leftarrow 1$  to  $m$  do
29:        $\mathbf{c} \leftarrow \mathbf{c} \vee \text{Contact\_Root}(o_j, h_i)$ 
30:     end for
31:      $\text{CRR} \leftarrow \text{CRR} + \frac{1}{T} \sum_{t=1}^T \mathbf{c}_t$ 
32:   end for
33:  return  $\text{CRR}/n$ 
34: end procedure

```

Component	Description
CLIP Type	ViT-B/32
Raw Feature Space Dimension	512
MLP Architecture	Linear(512, 512) ReLU() Linear(512, 128)

Table 11. Hyperparameters in label feature extraction branch.

label features, object label features (for m rigid objects), object geometry features (for m rigid objects), and the shape parameters β of n articulated skeletons to form a condition vector, whose shape is $\mathbb{R}^{128 \times (2m+1) + 10n}$, as is mentioned in main text Section 3.4. After replicating and concatenating with padding mask, x_{dc} and x_{F} , the dimension becomes

Component	Description
Raw Feature Space Dimension	1024×3
MLP Architecture	Linear($1024 \times 3, 512$) ReLU() Linear($512, 128$)

Table 12. Hyperparameters in rigid body geometry feature extraction branch.

Component	Description
Timestep Embedder	SinusoidalPosEmbedding(64) Linear(64, 256) GeLU() Linear(256, 1024)

Table 13. Hyperparameters in noise timestep embedding.

$\mathbb{R}^{N \times (128(2m+1) + 10n + 7m + 3Dn + 7m(m-1) + 3Dmn + 1)}$ = $\mathbb{R}^{N \times (128(2m+1) + 10n + D_{\text{sum}} + 1)}$ = $\mathbb{R}^{N \times \mathcal{C}}$. The architecture of the transformer encoder-decoder is shown in Table 14.

Component	Description
Encoder	Conv1D($\mathcal{C}, 512$)
Latent Transformer	4-layer, 8-head, 1024-dim
Decoder	Linear($512, D_{\text{sum}}$)

Table 14. Hyperparameters in the transformer encoder-decoder.

Note that after x_{dc} and x_{F} (concatenating with the condition vector) passes through the encoder, their shapes becomes $N \times 512$. Concatenating them together results in a latent vector of shape $N \times 1024$, which can be affiliated by the noise embedding, whose dimension is 1024.

D.3. Training and Inference Hyperparameters

Other important hyperparameters for training and inference process are shown in Table 15.

Parameter	TACO	CORE4D	OAKINK2	GRAB	BEHAVE
Batch Size	32				
Optimizer	Adam, lr = 0.0001, ema_decay = 0.995				
Epoch	250k	140k	280k	100k	100k
λ_{DC}	1	1	1	1	1
λ_{AC}	2.5	0.8	0.8	0.3	0.8
λ_{norm}	0.1	0.1	0.1	0.1	0.1
λ_{align}	0.3	0.3	0.3	0.15	0.15
λ_{exp}	0.3	0.3	0.3	0.3	0.3
Diffusion	$\alpha \in [0.0001, 0.01]$, Uniform				

Table 15. Hyperparameters for training and inference process in SyncDiff.

D.4. Time and Space Cost, Hardware Configurations

We conduct experiments on NVIDIA A40. All operations can be performed on a single GPU.

Time and Space Cost for Training and Inference. The training time, average inference time per sample, and GPU memory usage during training are detailed in Table 16.

Cost	TACO	CORE4D	OAKINK2	GRAB	BEHAVE
Training	20.7h	9.5h	40.1h	7.9h	3.7h
Inference	7.7s	6.5s	7.3s	6.6s	4.2s
Memory	6.11G	5.07G	8.59G	4.93G	3.67G

Table 16. Time and space costs for training and inference process in SyncDiff on different datasets.

Time and Space Cost for Mesh Reconstruction. Time and space cost of performing mesh reconstruction for a motion sequence of $N = 200$ frames are shown in Table 17.

Cost	TACO	CORE4D	OAKINK2	GRAB	BEHAVE
Time	130s	144s	206s	130s	74s
Memory	492M	826M	492M	492M	430M

Table 17. Time and space costs for mesh reconstruction in SyncDiff on different datasets.

Although the mesh reconstruction operation seems to take much more time than the inference process, due to parallelized calculation, the amortized time complexity is relatively low. In addition, the mesh reconstruction process is also optional, since in tasks like robot planning, only joint positions are enough.

E. Limitations and Discussions

E.1. Current Limitations and Potential Solutions

Some current limitations of SyncDiff and their potential solutions are as follows:

1. Lack of Articulation-Aware Modeling. Our method models articulated objects (such as those in OAKINK2 [23]) as part-wise rigid body individuals directly and coordinates their motions without leveraging their intrinsic articulations. Integrating these articulations into multi-body likelihood modeling could be an interesting future direction. The potential solution may be unify them with the articulated skeletons (hands/humans), and treat their intrinsic articulations as conditions.

2. High-cost of Explicit Synchronization Step. As body number increases, the time consumption for the calculation of alignment loss and explicit synchronization step grows quadratically. Note that for multi-body HOI synthesis, not all pairwise relationships are necessary. A possible solution is to use human priors or another algorithm to filter out the relationships that truly require synchronization, and perform synchronization only across them.

3. Lack of Physically Accurate Guarantees. Unlike methods that utilize true physical simulations, our approach cannot guarantee physical truthfulness. In many cases, minor errors can be observed in the supplementary videos, but

these small discrepancies may be sufficient to cause visible failures in real tasks. In robot manipulation tasks, we prefer to treat SyncDiff as a robot planning method that requires downstream integration with physically accurate optimization (RL/IL) to ensure practical usability in real-world applications.

4. Limitations in Pure Multi-human Interaction Synthesis. Since the relative representations need to be generated in the coordinate systems of rigid bodies, whose motions can be represented by translations and rotations, our method may not be directly adapted for pure multi-human interaction synthesis. To address this limitation, more complex relative representations is required. A potential solution is use representations similar to MANO [16], representing the motions of an articulated skeleton as the transformation matrix and local relative positions of its joints. Although this approach sacrifices the homogeneity of the articulated skeleton motion representation, it allows for a more convenient computation of the relative representation between two skeletons. This idea is manifested in a bunch of multi-human interaction synthesis works like ComMDM [18] and InterGen [8].

E.2. Discussions and Some Extensions

This section aims to enumerate some points where readers may have questions or misconceptions, providing detailed explanations. For scenarios where SyncDiff can be extended, we will also offer intuitive methods for expansion.

1. Does SyncDiff lose the flexibility of “one model for all datasets”?

Due to the scarcity of mocap datasets, as well as the need to adopt models into multiple scenarios, sometimes it is necessary to merge data from multiple datasets for training. However, due to the fixed number of parameters, a trained model of a specific size can only handle a pair of (number of rigid objects, number of articulated skeletons). For instance, the data from TACO can only be merged with the 2-hand 2-object samples in OAKINK2 for training, but not directly with GRAB.

Our response is divided into three aspects.

First, the extension from a unified framework to “one model for all datasets” is basically trivial, only increasing some computational load. It is noted that any graphical model is a subgraph of a sufficiently large complete graph. By using masks to cover the individual bodies that do not need to be synthesized and the relative relations that do not need to be adopted, this can be achieved. Such an idea is also reflected in [19].

Second, existing sota methods can only train for specific (number of rigid objects, number of articulated skeletons), and even require the type of articulated skeleton to be given (for example, the carefully designed grasp guid-

ance in DiffH2O is almost ineffective when applied to the human-object interaction dataset CORE4D; the cross-attention between bodies and contact maps in CG-HOI is too coarse-grained for dexterous hand-object interaction). Our method, as a unified framework, significantly reduces the cost of manually trying different pipelines and designing different guidance/representations for different settings, which is a giant leap forward.

Third, even if our method is treated as a single-track method for every specific configuration, it surpasses many sota methods in the corresponding tracks. As body number increases, motion semantics become more complex, especially between multiple rigid objects. For example, the object trajectories in the GRAB/BEHAVE dataset mainly consist of basic units like picking up, putting down, and lateral movements. However, in TACO, there may be actions such as rubbing back and forth, tapping, and pouring between two items. If OAKINK2 involves modeling the rigid parts of articulated objects, it also needs to ensure coordination between these different parts, such as a test tube passing precisely between the two halves of a test tube holder. As the first work to adopt synchronization and frequency decomposition to model complex motions, we have also significantly outperformed existing state-of-the-art methods in settings with a larger number of bodies.

2. Why doesn’t SyncDiff use the 6 DoF rotation representation that is widely employed in the current motion synthesis methods?

To represent the rotation $R \in SO(3)$ of an object, the 6 DoF representation proposed in [25] concatenates the first two columns of the 3×3 rotation matrix into a 6-dimensional vector r as the representation. After the model generates the predicted result \hat{r} , it is split into two column vectors, normalized, and subjected to Gram-Schmidt orthogonalization, and then completed into the rotation matrix \hat{R} in $SO(3)$.

In our method, since the relative representations need to be solved frequently, if we use such 6 DoF representation, it is necessary to complete them into matrices for operations such as inversion and composition, which significantly increases the computational cost. Therefore, we adopt quaternions, which are more convenient for normalization and calculation.

3. Is the explicit synchronization during inference in SyncDiff inspired by the guidance strategy in Guided Motion Diffusion (GMD) [6]?

As is stated in main text **Section 2.2**, several prominent works inject external priors or constraints into synthesized results by performing linear fusions, imputations or inpainting during the inference process of diffusion models. The term $-\nabla_{\mathbf{x}_0} G_z(P_x^z \mathbf{x})$ in GMD and the gradient of negative log-likelihood (actually equivalent to the score functions) in SyncDiff (Refer to **Section A.2**) both provide the direction

of denoising (for both), adhering to constraints (for GMD), or synchronization (for SyncDiff).

The distinction lies in that different application scenarios have led to different concrete operations for the same idea. GMD aims to constrain the generation of high-dimensional human pose x controlled by the goal function G_z , where z is low-dimensional pelvis trajectory. Hence, it employs an inpainting/imputation strategy to project (by the projector P_x^z) and complete the sparse trajectory during inference. In contrast, in SyncDiff, “synchronization” is a concept that is difficult to quantify. Therefore, we introduce the probabilistic modeling on graphical models, and ensure that synchronization and data denoising are simultaneously established in the inference process.

4. Can the articulated skeletons in SyncDiff be other body parts besides hands and the full body?

In main text **Section 3.1**, our definition of articulated skeletons is “a set of joints whose motion can be reconstructed based on joint information and shape parameter β ”. To extend SyncDiff to HOI motion synthesis of other body parts, **two issues need to be addressed**:

1) Collect relevant high-precision mocap data. For example, if we want to perform HOI motion synthesis for feet, we need to collect high-quality data such as kicking a ball, putting on shoes, ice skating, etc.

2) Design a data representation for motion reconstruction based on joint information and shape parameter β , similar to MANO for hands or SMPL-X for human bodies.

5. Does SyncDiff’s ignorance of affordance result in the item being manipulated incorrectly? For example, the cup was not picked up from the handle position?

SyncDiff does not consider affordance because the five datasets we use do not provide affordance for rigid objects at all. If the affordance needs to be considered, the relevant data with affordance map needs to be collected first, and the next potential solution may be to encode it in some way as part of the object’s geometric features.

In addition, the data-driven generative model should fit the data distribution. As long as the dataset used for training ensures high-quality and correct interactions in a wide range, the model should learn to generate similar samples with correct manipulation by itself.

6. Is the metric RA (Recognition Accuracy) fair? Since the discriminator is trained on the combination of train, val, and test folders, is there any risk of overfitting?

Training the classifier solely on the training split would introduce a more serious issue: motion synthesis models overfitting to training distribution would achieve inflated RA scores. In fact, experiments reveal that this approach causes ground-truth test splits to underperform some baselines. This is because the distributional difference between the ground-truth test splits and train splits is greater than

that between the distribution fitted by the generative model on train split and the train split itself.

The fundamental challenge is that any classifier—regardless of its training data—will inherently favor in-domain motions. While a huge and more diverse dataset could mitigate this bias, it is currently impractical due to the acquisition cost of mocap data. A motion recognition foundation model might help mitigate this issue. However, currently there are no such foundation models capable of performing high-quality action recognition based solely on trajectories rather than RGB video inputs.

Besides, the relative rankings produced by our RA metric (Tables 1-4 in main text, Table 1) align with perceptual judgments in user studies (Figure 1), supporting its validity.

References

- [1] Bharat Lal Bhatnagar, Xianghui Xie, Ilya A. Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15914–15925, 2022. 1, 5
- [2] Sammy Christen, Shreyas Hampali, Fadime Sener, Edoardo Remelli, Tomas Hodan, Eric Sauser, Shugao Ma, and Bugra Tekin. Diffh2o: Diffusion-based synthesis of hand-object interactions from textual descriptions. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 9
- [3] Christian Diller and Angela Dai. Cg-hoi: Contact-guided 3d human-object interaction generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19888–19901, 2024. 5
- [4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 3
- [5] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022. 1
- [6] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2151–2162, 2023. 13
- [7] Jiaman Li, Jiajun Wu, and C Karen Liu. Object motion guided human motion synthesis. *ACM Transactions on Graphics (TOG)*, 42(6):1–11, 2023. 5
- [8] Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. InterGen: Diffusion-based multi-human motion generation under complex interactions. *International Journal of Computer Vision*, 132(9):3463–3483, 2024. 13
- [9] Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. Taco: Benchmarking generalizable bimanual tool-action-object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21740–21751, 2024. 9

- [10] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 5, 9
- [11] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4332–4341, 2019. 10, 11
- [12] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 10
- [13] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 10
- [14] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 11
- [15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1
- [16] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):1–17, 2017. 9, 13
- [17] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022. 5
- [18] Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. Human motion diffusion as a generative prior. *arXiv preprint arXiv:2303.01418*, 2023. 13
- [19] Mengyi Shan, Lu Dong, Yutao Han, Yuan Yao, Tao Liu, Ifeoma Nwogu, Guo Jun Qi, and Mitch Hill. Towards open domain text-driven synthesis of multi-person motions. In *European Conference on Computer Vision*, 2025. 13
- [20] Soshi Shimada, Franziska Mueller, Jan Bednarik, Bardia Doosti, Bernd Bickel, Danhang Tang, Vladislav Golyanik, Jonathan Taylor, Christian Theobalt, and Thabo Beeler. Macs: Mass conditioned 3d hand and object motion synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 1082–1091. IEEE, 2024. 9
- [21] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 1, 2
- [22] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 1
- [23] Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 445–456, 2024. 12
- [24] Chengwen Zhang, Yun Liu, Ruofan Xing, Bingda Tang, and Li Yi. Core4d: A 4d human-object-human interaction dataset for collaborative object rearrangement. *arXiv preprint arXiv:2406.19353*, 2024. 5
- [25] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. *IEEE*, 2019. 13