Diffusion Masked Pretraining for Dynamic Point Cloud

Zhuoyue Zhang[1,2*] **Yiding Sun**[1*] **Chaowei Fang**[1,3] **Haozhe Cheng**[1]
Jian Liu[4] **Ajmal Saeed Mian**[5] **Jihua Zhu**[1,2†]
[1]School of Software Engineering, Xi'an Jiaotong University, P.R. China
[2]State Key Laboratory of Human-Machine Hybrid Augmented Intelligence, Xi'an Jiaotong University, Xi'an, China
[3]State Key Lab. of Novel Software Technology, Nanjing University, P.R. China
[4]School of Artificial Intelligence and Robotics, Hunan University, China
[5]University of Western Australia, Perth, Australia
zhangzy4016@stu.xjtu.edu.cn zhujh@xjtu.edu.cn

Abstract

Dynamic point cloud pretraining is still dominated by masked reconstruction objectives. However, these objectives inherit two key limitations. Existing methods inject ground-truth tube centers as decoder positional embeddings, causing spatio-temporal positional leakage. Moreover, they supervise inter-frame motion with deterministic proxy targets that systematically discard distributional structure by collapsing multimodal trajectory uncertainty into conditional means. To address these limitations, we propose **Di**ffusion **M**asked **P**retraining (**DiMP**), a unified self-supervised framework for dynamic point clouds. DiMP introduces diffusion modeling into both positional inference and motion learning. It first applies forward diffusion noise only to masked tube centers, then predicts clean centers from visible spatio-temporal context. This removes positional leakage while preserving visible coordinates as clean temporal anchors. DiMP also reformulates point-wise inter-frame displacement supervision as a DDPM noise-prediction objective conditioned on decoded representations. This design drives the encoder to target the full conditional distribution of plausible motions under a variational surrogate, rather than collapsing to a single deterministic estimate. Extensive experiments demonstrate that DiMP consistently improves downstream accuracy over the backbone alone, with absolute gains of 11.21% on offline action segmentation and 13.65% under causally constrained online inference. Code is available at `https://github.com/InitalZ/DiMP`.

1 Introduction

Self-supervised pretraining with masked autoencoders (MAE) [1–5] has proven effective for static 3D point cloud representation learning by masking local geometric regions and supervising coordinate reconstruction. Building on this, diffusion-based pretraining methods [6–8] integrate DDPMs [9] with masked autoencoding, and Point-MaDi [8] further identifies that injecting ground-truth patch centers as decoder positional embeddings leaks spatial priors [10–12], addressing this via diffusion noise over patch centers.

Despite this progress, existing methods remain confined to *static* point clouds. Research has shifted toward dynamic point clouds [13]. Such dynamic sequences couple spatial layout within each

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

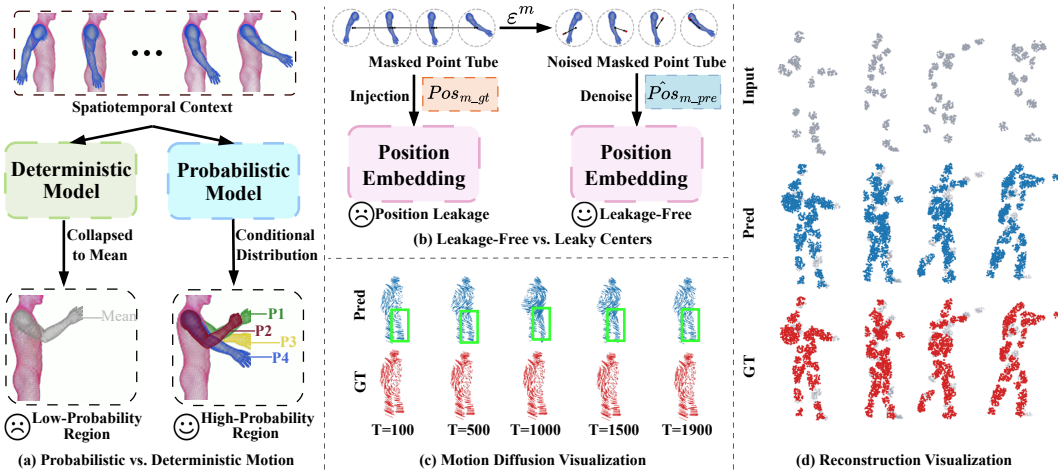


Figure 1: Motivation of DiMP. (a) Deterministic models collapse multimodal motion distributions to a single conditional mean, which lies in a low-probability region when trajectories are multimodal. DiMP targets the full conditional distribution under a variational surrogate. (b) Directly injecting ground-truth masked tube center coordinates as decoder positional embeddings causes positional leakage, whereas DiMP applies forward diffusion noise to masked centers and replaces ground-truth positions with denoised predictions, yielding a leakage-free positional embedding. (c) Qualitative comparisons of denoised inter-frame displacement predictions against the ground truth show that DiMP recovers motion patterns faithfully across representative motion diffusion timesteps t_i^M. Detailed analysis of why the model performs well at small and large t is provided in the Appendix G. (d) Qualitative reconstruction results show that DiMP recovers the overall human pose and preserves local geometric structures with high fidelity, yielding reconstructions that closely match the ground truth.

frame with temporal deformation across frames instead of treating scans as unrelated static clouds. Dynamic point cloud understanding is critical to autonomous driving, embodied intelligence, and action recognition. Masked reconstruction methods such as MaST-Pre [14] and M2PSC [15] extend the spatio-temporal tube masking paradigm to dynamic sequences. However, they still rely on deterministic proxy targets to supervise inter-frame motion. Accordingly, motion supervision is reduced to fitting conditional expectations. This formulation cannot target the full conditional distribution of motion trajectories. The same spatio-temporal context can correspond to multiple plausible continuations, but mean regression collapses these modes into a single deterministic solution, systematically discarding the distributional structure that is critical for discriminating action classes [16, 17]. Additionally, these methods inherit positional leakage from the static domain by injecting ground-truth spatio-temporal tube centers as decoder positional embeddings.

In this paper, we propose **Di**ffusion **M**asked **P**retraining (**DiMP**), the first framework to harness the full distributional modeling capacity of DDPMs for dynamic point cloud pretraining. DiMP addresses two intertwined challenges. **(1) Spatio-temporal positional leakage:** Unlike the static setting where noise can be applied uniformly to all patch centers, visible tube coordinates in videos serve a dual role as geometric anchors for the encoder and as temporal conditioning signals that link consecutive frames into a coherent motion sequence. Corrupting them would undermine motion continuity (Table 5), necessitating perturbation exclusively on masked tube centers. **(2) Distributional motion modeling:** By reformulating inter-frame displacement supervision as a DDPM noise prediction task conditioned on decoded representations, DiMP drives the encoder to internalize the full multimodal distribution of motion trajectories rather than collapsing to deterministic mean estimates. Because different action classes can share nearly identical mean trajectories yet differ sharply in their trajectory variance and shape, capturing this distributional structure is decisive for fine-grained action discrimination. Crucially, these two challenges are not independent, positional leakage is a *prerequisite barrier* to distributional motion modeling. Leaked ground-truth tube centers collapse decoder cross-attention into local geometric retrieval, severing the informational pathway through which the motion diffusion branch drives the encoder to internalize motion distributions. Resolving positional leakage is therefore necessary before diffusion-based motion uncertainty can be genuinely exploited (see Appendix E).

Concretely, DiMP is built on a masked autoencoding backbone and operates in two stages. In the *spatio-temporal center modeling* stage, we apply forward diffusion noise only to masked tube centers. The diffusion model is conditioned on visible features and temporal positional encodings to predict the masked center distribution. This design removes positional leakage while preserving clean temporal context. Based on the predicted centers, a masked decoder further fuses visible encoder features with predicted center embeddings to form complete spatio-temporal representations. These representations are supervised by per-frame Chamfer distance to reconstruct the full dynamic point cloud. In the *motion-aware modeling* stage, we reformulate point-wise inter-frame displacement supervision as a standard DDPM noise prediction task conditioned on the decoded representation. This enables DiMP to genuinely exploit the distributional capacity of DDPMs in dynamic 3D pretraining.

Our main contributions are as follows:

- We provide a formal argument that deterministic motion supervision systematically discards distributional structure critical to multimodal trajectory uncertainty, and further identify positional leakage as a prerequisite barrier to distributional motion modeling.
- We propose **DiMP**, the first diffusion-based pretraining framework for dynamic point clouds, which reformulates inter-frame displacement supervision as a DDPM noise-prediction task to target the full conditional distribution of motion trajectories under a variational surrogate.
- We introduce a center diffusion strategy that confines noise injection exclusively to masked tube centers, resolving positional leakage without compromising temporal inference.
- Extensive experiments across action segmentation, semantic segmentation, and action recognition demonstrate that DiMP consistently outperforms prior methods on multiple benchmarks, with particularly pronounced gains under causally constrained online inference.

2 Related work

Self-supervised Learning on Static Point Clouds. Self-supervised learning for static point clouds has evolved along two principal paradigms, contrastive learning and reconstruction learning. Contrastive methods such as PointContrast [18] and CrossPoint [19] learn invariant representations via point-level or cross-modal objectives. On the reconstruction learning side, Point-BERT [3] and Point-MAE [2] established the masked autoencoding framework for point clouds, subsequently extended by hierarchical and cross-modal variants [20–24]. More recently, diffusion-based reconstruction methods further advanced this line. PointDif [6] guides iterative denoising from noisy geometry via aggregated backbone features, DiffPMAE [7] conditions a diffusion decoder on MAE encoder latents to reconstruct masked regions, and Point-MaDi [8] applies diffusion modeling to patch centers alongside conditional patch reconstruction. Nevertheless, these methods remain tailored to static geometry and do not address the two challenges that arise in the dynamic setting: positional leakage in decoder conditioning [10–12] and the absence of distributional motion modeling. DiMP addresses both limitations by introducing masked-only center diffusion for positional inference and diffusion-based motion supervision for dynamic trajectories.

Self-supervised Learning on Dynamic Point Cloud. Early approaches to dynamic point cloud representation learning rely on contrastive objectives, including temporal order prediction [25] and reconstruction-contrastive combinations [26–28]. More relevant to our work, generative methods extend masked autoencoding to the spatio-temporal domain. MaST-Pre [14] supervises point-tube reconstruction alongside temporal cardinality difference prediction, and related works [29, 15] further develop this paradigm. However, these methods still suffer from positional leakage because they directly inject ground-truth spatio-temporal tube center coordinates as decoder positional embeddings. At the same time, they rely on coarse-grained deterministic proxy targets for motion supervision. This reduces learning to fitting conditional expectations and collapses multiple plausible motion modes into a single deterministic solution. Accordingly, the distributional structure critical for discriminating action classes is discarded [16, 17]. Motivated by these limitations, DiMP is the first pretraining framework for dynamic point clouds that jointly resolves positional leakage and distributional motion modeling within a unified diffusion-based formulation.

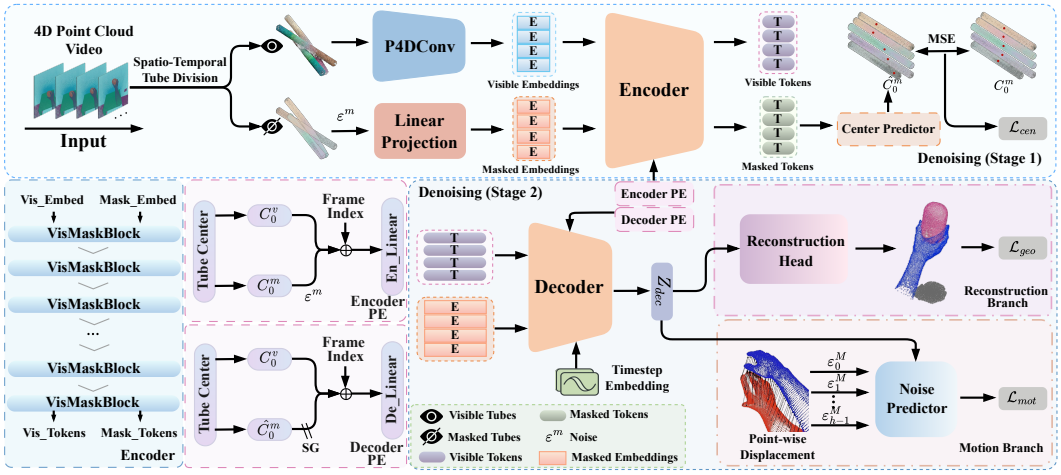


Figure 2: The pipeline of DiMP. Spatio-temporal tubes are extracted from a dynamic point cloud and randomly masked. P4DConv produces visible embeddings $\mathbf{e}_i$, while a linear projection maps corrupted masked point coordinates $\mathbf{P}^m_{t_c}$ to masked embeddings $\mathbf{e}^m_{t_c}$. In Stage 1, masked centers and point coordinates are independently perturbed by $\boldsymbol{\epsilon}_c$ and $\boldsymbol{\varepsilon}_c$ at the shared timestep t_c, while visible centers $\mathbf{C}^v_0$ remain clean. Encoder positional embeddings use $\mathbf{C}^v_0$, corrupted masked centers $\mathbf{C}^m_{t_c}$, and frame indices t_f. Stacked `VisMaskBlock` layers produce visible tokens $\mathbf{Z}^v$ and masked tokens $\mathbf{Z}^m$. The Center Predictor g_m maps $\mathbf{Z}^m$ to clean-center predictions $\hat{\mathbf{C}}^m_0$, supervised against $\mathbf{C}^m_0$ by $\mathcal{L}_{\text{cen}}$. In Stage 2, the decoder combines $\mathbf{Z}^v$ and $\mathbf{e}^m_{t_c}$ with positional embeddings from $\mathbf{C}^v_0$ and $\text{sg}(\hat{\mathbf{C}}^m_0)$, frame indices t_f, and the timestep embedding $\tau(t_c)$ to produce $\mathbf{Z}_{\text{dec}}$. Its masked-tube features support geometric reconstruction under $\mathcal{L}_{\text{geo}}$, while the full output conditions the motion noise predictor h_m. At independently sampled stratified timesteps t^M_i, $i = 0, \ldots, h-1$, the motion branch corrupts $\mathbf{M}_0$ with noise $\boldsymbol{\epsilon}^M_i$ and minimizes $\mathcal{L}_{\text{mot}}$, covering fine-grained local recovery at small timesteps and coarse motion structure at large timesteps.

3 Method

We present **Di**ffusion **M**asked **P**retraining (**DiMP**), a unified self-supervised pretraining framework for dynamic point clouds that incorporates two complementary diffusion processes into a masked autoencoding paradigm. Figure 2 provides an overview. Starting from a dynamic point cloud, DiMP partitions it into spatio-temporal tubes and randomly masks a large fraction of them. In Stage 1 (§3.1), a shared diffusion timestep t_c controls noise injection into both the masked tube *centers* and the masked tube *point coordinates*. The dual-stream encoder then predicts clean masked centers from this jointly corrupted context. In Stage 2 (§3.2), an asymmetric decoder combines visible encoder features with the stop-gradient predicted centers, conditioned on t_c, to produce $\mathbf{Z}_{\text{dec}}$, which simultaneously feeds a geometric reconstruction head and a motion-diffusion head. The motion branch employs a *separate* diffusion process over inter-frame displacements, with independently sampled stratified timesteps. All three objectives are unified via weighted summation (§3.3).

3.1 Stage 1: token encoding and spatio-temporal center denoising

Spatio-temporal tube tokenization. A dynamic point cloud is partitioned into K spatio-temporal tubes. For each visible tube $i \in \mathcal{V}$, the $l \times n$ points within its spatial neighborhood are tokenized by a P4DConv operator [30], yielding a d-dimensional embedding $\mathbf{e}_i \in \mathbb{R}^d$. The raw tube centers $\mathbf{C} = \{\hat{\mathbf{p}}_i\}_{i=1}^K$ are globally mean-centered as $\mathbf{C}_0 = \mathbf{C} - \bar{\mathbf{C}}$ with $\bar{\mathbf{C}} = K^{-1}\sum_i \hat{\mathbf{p}}_i$, suppressing rigid-body translation while preserving inter-frame motion signals. After masking, $\mathbf{C}_0$ splits into visible $\mathbf{C}^v_0 \in \mathbb{R}^{K_v \times 3}$ and masked $\mathbf{C}^m_0 \in \mathbb{R}^{K_m \times 3}$. Critically, $\mathbf{C}^v_0$ is kept *noise-free* throughout both stages: corrupting visible centers would destabilize the motion-continuity prior underpinning temporal inference (confirmed empirically in Table 5).

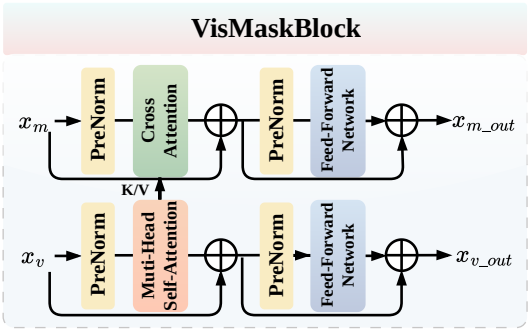


Figure 3: Structure of a `VisMaskBlock`. At layer ℓ, visible tokens $\mathbf{Z}_\ell^v$ are updated by self-attention to yield $\mathbf{Z}_{\ell+1}^v$, while masked tokens $\mathbf{Z}_\ell^m$ query the visible stream through cross-attention to yield $\mathbf{Z}_{\ell+1}^m$. This asymmetric design preserves clean visible context and enables leakage-free inference for masked tubes.

Shared diffusion schedule and corrupted inputs. A single center-diffusion timestep $t_c \sim \mathcal{U}\{1, T\}$ governs all Stage 1 corruption. Two independent noise samples are drawn at this shared timestep and applied to different modalities of the *same* masked tubes: $\mathbf{C}_{t_c}^m = \sqrt{\bar{\alpha}_{t_c}}\,\mathbf{C}_0^m + \sqrt{1-\bar{\alpha}_{t_c}}\,\boldsymbol{\epsilon}_c$ with $\boldsymbol{\epsilon}_c \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\mathbf{P}_{t_c}^m = \sqrt{\bar{\alpha}_{t_c}}\,\mathbf{P}_0^m + \sqrt{1-\bar{\alpha}_{t_c}}\,\boldsymbol{\varepsilon}_c$ with $\boldsymbol{\varepsilon}_c \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\{\bar{\alpha}_t\}$ is the DDPM cosine schedule shared by both, and $\mathbf{P}_0^m \in \mathbb{R}^{n\times 3}$ denotes the raw point coordinates of a masked tube. Here $\boldsymbol{\epsilon}_c$ perturbs the masked tube *centers* used in positional embeddings, and $\boldsymbol{\varepsilon}_c$ perturbs the raw masked *point coordinates* used in content embeddings. The two samples are drawn independently but follow the exact same noise schedule and timestep t_c, so both modalities are corrupted at a consistent level. The perturbed coordinates are then projected to a masked content embedding $\mathbf{e}_{t_c}^m = \mathrm{LinearProj}(\mathbf{P}_{t_c}^m) \in \mathbb{R}^d$.

Dual-stream encoder. The encoder f_e processes visible and masked tokens through a shared stack of `VisMaskBlock` layers (Figure 3), where each block applies $\mathbf{Z}_{\ell+1}^v = \mathrm{SelfAttn}(\mathbf{Z}_\ell^v)$ over visible tokens and $\mathbf{Z}_{\ell+1}^m = \mathrm{CrossAttn}(\mathbf{Z}_\ell^m, \mathbf{Z}_\ell^v)$ from masked tokens to the visible context. Each token is formed by adding its content embedding to $\phi_{\mathrm{enc}}(\mathbf{c},\, t_f)$, an MLP-based encoder positional embedding that maps tube center $\mathbf{c}$ concatenated with frame index t_f. The full encoder forward pass produces:

$$\mathbf{Z}^v = f_e^v\Big(\big\{\mathbf{e}_i + \phi_{\mathrm{enc}}(\mathbf{C}_0^v,\, t_f)\big\}_{i\in\mathcal{V}}\Big) \in \mathbb{R}^{K_v\times d}, \tag{1}$$

$$\mathbf{Z}^m = f_e^m\Big(\big\{\mathbf{e}_{t_c}^m + \phi_{\mathrm{enc}}(\mathbf{C}_{t_c}^m,\, t_f)\big\}_{i\in\mathcal{M}},\ \mathbf{Z}^v\Big) \in \mathbb{R}^{K_m\times d}, \tag{2}$$

where f_e^v applies self-attention over visible tokens and f_e^m applies cross-attention from masked tokens to the visible context. Since cross-attention keys and values are drawn exclusively from $\mathbf{Z}^v$, each masked hidden state $\mathbf{Z}_j^m$ ($j \in \mathcal{M}$) constitutes a structured approximation to the denoising posterior $q(\mathbf{C}_{0,j}^m \mid \mathbf{C}_{t_c}^m, \mathbf{Z}^v)$.

Center prediction and Stage 1 loss. Center Predictor g_m maps $\mathbf{Z}^m$ to predicted clean centers $\hat{\mathbf{C}}_0^m = g_m(\mathbf{Z}^m)$, supervised by MSE:

$$\mathcal{L}_{\mathrm{cen}} = \frac{1}{K_m}\sum_{j=1}^{K_m}\big\|\hat{\mathbf{C}}_{0,j}^m - \mathbf{C}_{0,j}^m\big\|_2^2. \tag{3}$$

By supervising only the masked branch and varying corruption via t_c, the model acquires robust positional representations under spatio-temporal uncertainty. The parameters of g_m receive gradients *exclusively* from $\mathcal{L}_{\mathrm{cen}}$, as enforced by the stop-gradient mechanism described next.

3.2 Stage 2: geometric reconstruction and motion-aware diffusion

Decoder with predicted-center and coordinate-timestep conditioning. With leakage-free center predictions $\hat{\mathbf{C}}_0^m$ from Stage 1 in hand, an asymmetric decoder f_d combines encoded visible tokens, noisy masked content embeddings, decoder positional embeddings from clean visible and predicted masked centers, and a timestep embedding of t_c:

$$\mathbf{Z}_{\mathrm{dec}} = f_d\Big(\mathbf{Z}^v,\ \mathbf{e}_{t_c}^m,\ \phi_{\mathrm{dec}}(\mathbf{C}_0^v,\, t_f),\ \phi_{\mathrm{dec}}\big(\mathrm{sg}(\hat{\mathbf{C}}_0^m),\, t_f\big),\ \tau(t_c)\Big) \in \mathbb{R}^{K\times d}, \tag{4}$$

where three design choices are worth noting. The decoder positional embedding ϕ_{dec} uses clean $\mathbf{C}_0^v$ for visible tubes and the stop-gradient prediction $\mathrm{sg}(\hat{\mathbf{C}}_0^m)$ for masked tubes, ensuring that g_m is updated only by $\mathcal{L}_{\mathrm{cen}}$ and not by the reconstruction or motion objectives (ablated in Table 12). The timestep token $\tau(t_c)$ is injected into f_d to condition the decoder on the geometry corruption level.

Geometric reconstruction. The masked-tube slice of $\mathbf{Z}_{\text{dec}}$ is passed through a linear reconstruction head to yield predicted point coordinates $\hat{\mathbf{P}}^{\text{pre}}$, supervised by the per-frame ℓ_2 Chamfer distance averaged over all masked tubes:

$$\mathcal{L}_{\text{geo}} = \frac{1}{K_m} \sum_{i \in \mathcal{M}} \frac{1}{l} \sum_{s=1}^{l} \text{CD}\Big(\hat{\mathbf{P}}_{i,s}^{\text{pre}},\ \mathbf{P}_{i,s}^{\text{gt}}\Big), \tag{5}$$

where $\text{CD}(\mathbf{A}, \mathbf{B}) = \frac{1}{|\mathbf{A}|} \sum_{a \in \mathbf{A}} \min_{b \in \mathbf{B}} \|a-b\|_2^2 + \frac{1}{|\mathbf{B}|} \sum_{b \in \mathbf{B}} \min_{a \in \mathbf{A}} \|b-a\|_2^2$. The full $\mathbf{Z}_{\text{dec}}$ then conditions the motion-aware diffusion module.

Inter-frame displacement as motion target. We construct point-wise inter-frame displacement vectors as the motion supervision signal. For each point $\mathbf{p}_i^t$ in frame t, its displacement is $\Delta\mathbf{p}_i^t = \mathbf{p}_i^{t+1} - \mathbf{p}_i^t$ for $i = 1, \ldots, N$ and $t = 1, \ldots, L-1$. Point correspondences are established offline using the pre-trained correspondence model CorrNet3D [31], yielding the full motion matrix $\mathbf{M}_0 \in \mathbb{R}^{N \times (L-1) \times 3}$ where $\mathbf{M}_{0,i,t} = \Delta\mathbf{p}_i^t$. CorrNet3D serves solely as an offline labeler whose parameters are frozen throughout DiMP training with no gradient flowing through it, and it is discarded after pretraining alongside the decoder and diffusion heads. The encoder never accesses CorrNet3D features and only receives the scalar displacement values stored as static supervision labels on disk.

Motion diffusion. The motion branch employs a diffusion process independent from the Stage 1 schedule. To guarantee uniform coverage of the full diffusion hierarchy, $[1, T]$ is partitioned into h near-equal intervals with $d = \lfloor T/h \rfloor$:

$$Q_i = \begin{cases} [d{\cdot}i + 1,\ d{\cdot}(i+1)], & 0 \le i < h-1, \\ [d(h-1)+1,\ T], & i = h-1, \end{cases} \tag{6}$$

so that $\bigcup_{i=0}^{h-1} Q_i = [1, T]$. For each interval Q_i, a timestep $t_i^M \sim \mathcal{U}(Q_i)$ and a noise sample $\boldsymbol{\epsilon}_i^M \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ are drawn independently, and the corresponding forward marginal $\mathbf{M}_{t_i^M} = \sqrt{\bar{\alpha}_{t_i^M}}\,\mathbf{M}_0 + \sqrt{1-\bar{\alpha}_{t_i^M}}\,\boldsymbol{\epsilon}_i^M$ is evaluated at all h levels simultaneously.

Noise prediction head and loss. A lightweight PCNet-based MLP with a reset-gate conditioning mechanism [6] serves as the noise prediction head h_m. It predicts the injected motion noise at interval i as $\hat{\boldsymbol{\epsilon}}_i^M = h_m\Big(\mathbf{M}_{t_i^M},\ t_i^M,\ \mathbf{Z}_{\text{dec}}\Big)$, conditioned on the full decoder output $\mathbf{Z}_{\text{dec}}$. The motion diffusion loss averages the DDPM objective across all h intervals:

$$\mathcal{L}_{\text{mot}} = \frac{1}{h} \sum_{i=0}^{h-1} \mathbb{E}_{\substack{t_i^M \sim Q_i, \\ \mathbf{M}_0,\, \boldsymbol{\epsilon}_i^M}} \left[\Big\| \boldsymbol{\epsilon}_i^M - h_m\Big(\sqrt{\bar{\alpha}_{t_i^M}}\mathbf{M}_0 + \sqrt{1-\bar{\alpha}_{t_i^M}}\boldsymbol{\epsilon}_i^M,\ t_i^M,\ \mathbf{Z}_{\text{dec}} \Big) \Big\|^2 \right], \tag{7}$$

where the expectation is over motion sequences $\mathbf{M}_0$, motion noise $\boldsymbol{\epsilon}_i^M$, and interval-stratified timesteps t_i^M, with $\mathbf{Z}_{\text{dec}}$ treated as a fixed conditioning signal. Crucially, $\mathcal{L}_{\text{mot}}$ need to propagate gradients through the decoder all the way back to the encoder. We partially mitigate the gradient-attenuation effect inherent to decoder-conditioned objectives (discussed in Appendix H) by applying $\mathcal{L}_{\text{mot}}$ simultaneously across h stratified noise levels within each training step, multiplying and diversifying the encoder gradient signal from the motion branch. Small-t_i^M losses drive fine-grained local trajectory recovery, and large-t_i^M losses compel coarse semantic motion understanding. We set $h = 4$ by default (sensitivity analyzed in Table 8).

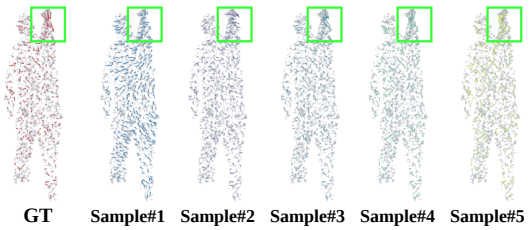


Figure 4: Five samples of clean inter-frame displacements from $p_\theta(\mathbf{M}_0 \mid \mathbf{Z}_{\text{dec}})$ with ground truth (red) for the same input. Sample diversity confirms multimodal trajectory modeling rather than mean collapse.

Table 1: 4D action segmentation results on the HOI4D dataset. All methods use 150 frames as input. Metrics: framewise accuracy (Acc), segmental edit distance (Edit), and F1 scores at overlap thresholds of 10%, 25%, and 50%. Numbers in parentheses for our method denote absolute improvements over the backbone. † denotes results reproduced from the public source code. Same as follows. Multi-seed statistics for DiMP and M2PSC are provided in Table 15.

Method	Venue	Acc	Edit	F1@10	F1@25	F1@50
PPTr [32]	ECCV22	77.4	80.1	81.7	78.5	69.5
+ STRL [33]	ICCV21	78.4	79.1	81.8	78.6	69.7
+ VideoMAE [34]	NeurIPS22	78.6	80.2	81.9	78.7	69.9
+ C2P [35]	CVPR23	81.1	84.0	85.4	82.5	74.1
+ DiMP (Ours)	**This Paper**	**83.97** (+6.57)	**86.15** (+6.05)	**89.13** (+7.43)	**86.64** (+8.14)	**81.43** (+11.93)
P4Transformer [30]	CVPR21	71.2	73.1	73.8	69.2	58.2
+ C2P [35]	CVPR23	73.5	76.8	77.2	72.9	62.4
+ MaST-Pre [14]†	ICCV23	74.1	75.4	76.6	73.4	63.8
+ M2PSC [15]†	ECCV24	75.9	77.1	78.4	74.1	65.9
+ DiMP (Ours)	**This Paper**	**82.41** (+11.21)	**84.90** (+11.80)	**87.26** (+13.46)	**85.92** (+16.72)	**80.77** (+22.57)

Theoretical motivation. The following result (Appendix F) motivates DiMP's motion branch.

Proposition 1 (Bayes-error obstruction of mean regression, informal)**.** If two action classes a_1, a_2 share the same class-conditional mean trajectory $\mathbb{E}[\mathbf{M} \mid A{=}a_1] = \mathbb{E}[\mathbf{M} \mid A{=}a_2]$ but differ in their conditional distributions, then a representation trained purely by mean-regression motion supervision cannot separate a_1 from a_2 above chance. A representation that targets the full distribution $p(\mathbf{M} \mid \mathbf{Z}_{\text{dec}})$ retains more mutual information with the action label A.

Whenever two actions differ in how their trajectories vary rather than only in their means, deterministic regression discards the distinguishing signal. DiMP's variational surrogate avoids this by targeting the full conditional distribution. Figure 4 visualizes five independent samples of clean inter-frame displacements drawn from $p_\theta(\mathbf{M}_0 \mid \mathbf{Z}_{\text{dec}})$ for the same input context, confirming that the model captures multimodal trajectories rather than collapsing to a single estimate.

3.3 Overall training objective

The three losses are unified via weighted summation:

$$\mathcal{L} = \mathcal{L}_{\text{geo}} + \gamma_{\text{cen}}\,\mathcal{L}_{\text{cen}} + \lambda_{\text{mot}}\,\mathcal{L}_{\text{mot}}, \tag{8}$$

where γ_{cen} weights the center diffusion loss and λ_{mot} weights the motion diffusion loss. Geometric reconstruction ($\mathcal{L}_{\text{geo}}$) provides dense structural supervision at unit weight, center diffusion ($\mathcal{L}_{\text{cen}}$) acts as a positional regularizer, and motion diffusion ($\mathcal{L}_{\text{mot}}$) constitutes the primary novel contribution. We set $\gamma_{\text{cen}} = 0.1$ and $\lambda_{\text{mot}} = 1.0$ by default (sensitivity in Tables 9 and 10). All components are optimized jointly. After pretraining, the decoder and both diffusion heads are discarded. The encoder f_e alone is retained for downstream fine-tuning.

4 Experiments

4.1 Downstream experiments

4D Action Segmentation on HOI4D. As reported in Table 1, DiMP achieves consistent improvements over all prior pre-training approaches on HOI4D, with gains especially pronounced in the Edit and F1 metrics, reflecting that a probabilistic motion objective promotes stronger temporal coherence than deterministic proxy targets. DiMP also yields substantially larger gains on the weaker P4Transformer backbone than on PPTr, while the final performance of both converges after pre-training, suggesting that P4Transformer's limited receptive field leaves greater capacity for DiMP's motion supervision to fill.

4D Semantic Segmentation on HOI4D. To verify that DiMP yields representations effective for fine-grained point-level understanding, we conduct further experiments on the HOI4D semantic segmentation benchmark. Mean Intersection over Union (mIoU, %) is used as the evaluation metric. As shown in Table 2, DiMP outperforms all prior pre-training methods. This result shows that the

Table 2: 4D semantic segmentation results on HOI4D.

Method	Venue	mIoU (%)
PPTr [32]	ECCV22	41.0
+ STRL [33]	ICCV21	41.2
+ VideoMAE [34]	NeurIPS22	41.3
+ C2P [35]	CVPR23	42.3
+ DiMP (Ours)	**This Paper**	**48.2** (+7.2)
P4Transformer [30]	CVPR21	40.1
+ MaST-Pre [14]†	ICCV23	40.3
+ C2P [35]	CVPR23	41.4
+ M2PSC [15]†	ECCV24	42.3
+ DiMP (Ours)	**This Paper**	**47.6** (+7.5)

Table 3: 3D action recognition accuracy (%) on MSRAction-3D.

Method	Venue	Accuracy (%)
PSTNet [36]	ICLR21	91.20
+ PointCMP [27]	CVPR23	93.27
PST-Transformer [37]	TPAMI22	93.73
+ M2PSC [15]	ECCV24	94.84
+ MaST-Pre [14]	ICCV23	94.08
+ DiMP (Ours)	**This Paper**	**95.51** (+1.78)
P4Transformer [30]	CVPR21	90.94
+ M2PSC [15]	ECCV24	93.03
+ DiMP (Ours)	**This Paper**	**93.97** (+3.03)

Table 4: Online action segmentation results on HOI4D under the causally constrained online inference setting. All methods operate on sequences of 150 frames. Runtime is measured in Clips/s, higher values indicate better real-time feasibility.

Method	Venue	Acc	Edit	F1@10	F1@25	F1@50	Clips/s
PointNet++ [38]	CVPR17	50.8	40.2	42.6	35.3	23.5	35.8
P4Transformer [30]	CVPR21	66.7	62.0	65.3	59.8	46.3	26.6
P4Mamba [39]	WACV26	71.0	78.5	77.8	73.4	61.5	33.4
PPTr [32]	ECCV22	69.7	64.5	69.4	64.8	52.9	–
NSM4D [40]	arXiv23	67.8	63.2	68.0	63.7	51.9	–
NSM4D (600 frames) [40]	arXiv23	71.3	68.0	72.1	68.1	56.5	–
PointNet4D [39]	WACV26	70.3	72.3	73.7	69.1	57.2	31.8
4DMAP [39]	WACV26	72.4	78.7	78.8	74.8	63.6	31.8
P4Transformer + DiMP (Ours)	**This Paper**	**80.35** (+13.65)	**82.19** (+20.19)	**84.77** (+19.47)	**83.21** (+23.41)	**73.50** (+27.20)	**27.3** (+0.7)

probabilistic motion diffusion objective learns richer geometric representations, since modeling temporal displacement distributions encourages the encoder to capture fine-grained boundary geometry that is informative for both motion understanding and static per-point labeling. These representations transfer effectively to fine-grained per-point recognition tasks.

3D Action Recognition on MSRAction-3D. As shown in Table 3, DiMP outperforms methods that rely on deterministic motion objectives and achieves competitive performance against state-of-the-art pre-training methods on MSRAction-3D. DiMP yields larger gains on the weaker P4Transformer than on PST-Transformer, consistent with the backbone-dependent pattern observed in action segmentation. This reflects that P4Transformer's simpler temporal modeling leaves greater representational headroom for DiMP's distributional motion supervision to fill, whereas PST-Transformer's stronger attention already captures a portion of the motion statistics that pre-training targets.

Online Inference on HOI4D. We evaluate DiMP under the causally constrained online inference protocol of PointNet4D [39]. As shown in Table 4, DiMP consistently outperforms all baselines. Notably, the online gains in Edit and F1@50 exceed the corresponding offline gains (Table 1) by 8.39 and 4.63 points, respectively, suggesting that a learned motion distribution

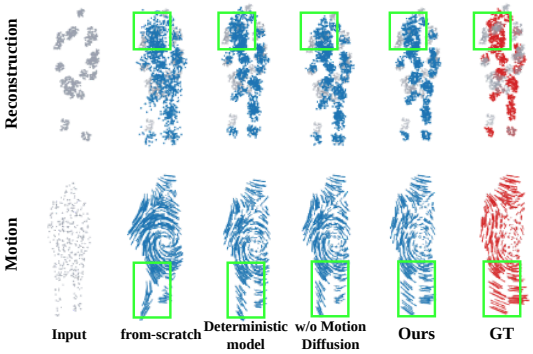


Figure 5: Qualitative comparison of geometric reconstruction (top) and inter-frame displacement prediction (bottom). Columns show the input, a model trained from scratch, a deterministic motion-supervision baseline, DiMP without motion diffusion, full DiMP, and ground truth (GT). Green boxes highlight local differences. Full DiMP yields predictions closer to ground truth, while removing motion diffusion degrades prediction quality.

Table 5: Effect of applying Gaussian noise to visible tube centers at varying relative strengths. Noise level 0.0 corresponds to the DiMP design, and 1.0 applies the same schedule as masked centers.

Visible Level	Noise	F1@50 ↑	mIoU ↑
0.0 (DiMP)		**80.77**	**47.6**
0.25		77.91	43.2
0.50		75.43	41.9
1.0		74.16	39.8

Table 6: Factorial ablation of center diffusion and motion diffusion on HOI4D action segmentation and MSRAction-3D action recognition. All other DiMP components are held fixed. The results are used to verify the prerequisite role of center diffusion in enabling effective motion modeling.

CD	MD	HOI4D		MSRAction-3D
		F1@50↑	mIoU↑	Acc↑
×	×	75.37	41.3	91.63
×	✓	75.43	41.2	91.72
✓	×	77.28	43.5	92.46
✓	✓	**80.77**	**47.6**	**93.97**

Table 7: Progressive component ablation on HOI4D. Each row adds one component to the previous configuration. VM: VisMaskTransformer encoder; PD: patch diffusion decoder; CD: masked-only center diffusion.

Configuration	VM	PD	CD	Motion	F1@50 ↑	mIoU ↑
MaST-Pre baseline†	×	×	×	Deterministic Regression	63.84	40.3
+ Joint linear PE	×	×	×	Deterministic Regression	65.13	40.4
+ VisMask encoder	✓	×	×	Deterministic Regression	65.37	41.3
+ Center diffusion	✓	×	✓	Deterministic Regression	72.28	43.5
+ Patch diffusion	✓	✓	✓	Deterministic Regression	74.17	44.1
+ Single-step motion	✓	✓	✓	Single-step DDPM	78.41	46.1
DiMP full (h=4)	✓	✓	✓	Stratified DDPM	**80.77**	**47.6**

provides an especially strong predictive prior when the model must reason about future motion from past context alone without access to future frames. This is expected, as causal constraints force the model to predict future motion from past context alone, a setting where a learned motion distribution provides a stronger prior than a deterministic estimate.

Further dataset descriptions, evaluation splits, task protocols, and training setups for all downstream experiments above are summarized in Appendix B.

4.2 Ablation studies

All ablation experiments use P4Transformer as the backbone under the same pre-training and fine-tuning setup summarized in Appendix B.

Effect of Visible Center Corruption. We apply Gaussian noise at varying relative scales to visible tube centers and measure the effect on downstream performance (Table 5). Performance degrades monotonically as visible center corruption increases, confirming that corrupting visible coordinates undermines the motion-continuity prior and impairs cross-frame geometric association.

Factorial interaction of center and motion diffusion. Table 6 shows that motion diffusion alone yields near-zero gain, whereas enabling center diffusion first allows motion diffusion to contribute substantially, a pattern that holds consistently across both HOI4D and MSRAction-3D, confirming that center diffusion is a structural prerequisite rather than a dataset-specific effect. Further theoretical analysis is provided in Appendix E.

Progressive Component Analysis. Table 7 shows a progressive ablation from the MaST-Pre [14] baseline. Center diffusion gives the clearest structural improvement, suggesting that removing positional leakage forces the model to infer masked tube locations from context. Patch diffusion and motion diffusion further strengthen the representation, with the full stratified variant achieving the best overall trade-off. Figure 5 provides a qualitative comparison of geometric reconstruction and inter-frame displacement prediction under different training strategies.

Additional ablations on hyperparameter sensitivity, the stop-gradient mechanism, and correspondence quality are provided in Appendix C. Gradient propagation analysis is provided in Appendix H.

5 Conclusion

In this paper, we present DiMP, the first pretraining framework for dynamic point clouds that leverages DDPMs for distributional motion modeling. By reformulating inter-frame displacement supervision as a noise prediction task conditioned on decoded representations, DiMP targets the full conditional distribution over motion trajectories under a variational surrogate, rather than collapsing to a deterministic mean estimate. A key enabler is the center diffusion strategy, which avoids spatio-temporal positional leakage. Experiments on HOI4D and MSRAction-3D demonstrate consistent improvements over prior methods across multiple tasks. One limitation of DiMP is backbone efficiency. Our implementation relies on P4Transformer, whose quadratic self-attention complexity results in lower online throughput than linear-complexity architectures. As an initial exploration of distributional motion modeling for dynamic point cloud pretraining, we expect DiMP to serve as a reference baseline and to inspire further research on distributional representation learning for dynamic point clouds.

Acknowledgments and Disclosure of Funding

This work was supported in part by NSFC under Grant Nos. 62125305 and 62376206, the Natural Science Basis Research Plan in Shaanxi Province of China under Grant No. 2025JC-JCQN-091, and the Technology Innovation Leading Program of Shaanxi (Program No. 2024QY-SZX-23).

References

[1] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[2] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision (ECCV)*, 2022.

[3] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[4] Yiding Sun, Haozhe Cheng, Chaoyi Lu, Zhengqiao Li, Minghong Wu, Huimin Lu, and Jihua Zhu. Hyperpoint: Multimodal 3d foundation model in hyperbolic space. *Pattern Recognition (PR)*, 2025.

[5] Pengbo Li, Yiding Sun, and Haozhe Cheng. Pointdico: Contrastive 3d representation learning guided by diffusion models. In *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025.

[6] Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[7] Yanlong Li, Chamara Madarasingha, and Kanchana Thilakarathna. Diffpmae: diffusion masked autoencoders for point cloud reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2024.

[8] Xiaoyang Xiao, Runzhao Yao, Zhiqiang Tian, and Shaoyi Du. Point-madi: Masked autoencoding with diffusion for point cloud pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems (NeurIPS)*, 2020.

[10] Sepehr Sameni, Simon Jenni, and Paolo Favaro. Representation learning by detecting incorrect location embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

[11] Haochen Wang, Junsong Fan, Yuxi Wang, Kaiyou Song, Tong Wang, and ZHAO-XIANG ZHANG. Droppos: Pre-training vision transformers by reconstructing dropped positions. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[12] Mathilde Caron, Neil Houlsby, and Cordelia Schmid. Location-aware self-supervised transformers for semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024.

[13] Yiding Sun, Jihua Zhu, Haozhe Cheng, Chaoyi Lu, Zhichuan Yang, Lin Chen, and Yaonan Wang. Align then adapt: Rethinking parameter-efficient transfer learning in 4d perception. *arXiv preprint arXiv:2602.23069*, 2026.

[14] Zhiqiang Shen, Xiaoxiao Sheng, Hehe Fan, Longguang Wang, Yulan Guo, Qiong Liu, Hao Wen, and Xi Zhou. Masked spatio-temporal structure prediction for self-supervised learning on point cloud videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[15] Yuehui Han, Can Xu, Rui Xu, Jianjun Qian, and Jin Xie. Masked motion prediction with semantic contrast for point cloud sequence learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2024.

[16] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020.

[18] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3D point cloud understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020.

[19] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[20] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-M2AE: Multi-scale masked autoencoders for hierarchical point cloud pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[21] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[22] Dongxu Zhang, Yiding Sun, Pengcheng Li, Yumou Liu, Hongqiang Lin, Haoran Xu, Xiaoxuan Mu, Liang Lin, Wenbiao Yan, Ning Yang, et al. Pointcot: A multi-modal benchmark for explicit 3d geometric reasoning. *arXiv preprint arXiv:2602.23945*, 2026.

[23] Yankai Wang, Yiding Sun, Qirui Wang, Pengbo Li, Chaoyi Lu, and Dongxu Zhang. Pointrft: Explicit reinforcement fine-tuning for point cloud few-shot learning. *arXiv preprint arXiv:2603.23957*, 2026.

[24] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2023.

[25] Haiyan Wang, Liang Yang, Xuejian Rong, Jinglun Feng, and Yingli Tian. Self-supervised 4d spatio-temporal feature learning via order prediction of sequential point cloud clips. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[26] Xiaoxiao Sheng, Zhiqiang Shen, and Gang Xiao. Contrastive predictive autoencoders for dynamic point cloud self-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

[27] Zhiqiang Shen, Xiaoxiao Sheng, Longguang Wang, Yulan Guo, Qiong Liu, and Xi Zhou. Pointcmp: Contrastive mask prediction for self-supervised learning on point cloud videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[28] Xiaoxiao Sheng, Zhiqiang Shen, Gang Xiao, Longguang Wang, Yulan Guo, and Hehe Fan. Point contrastive prediction with semantic clustering for self-supervised learning on point cloud videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[29] Zhi Zuo, Chenyi Zhuang, Pan Gao, Jie Qin, Hao Feng, and Nicu Sebe. Uni4d: A unified self-supervised learning framework for point cloud videos. *arXiv preprint arXiv:2504.04837*, 2025.

[30] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[31] Yiming Zeng, Yue Qian, Zhiyu Zhu, Junhui Hou, Hui Yuan, and Ying He. Corrnet3d: Unsupervised end-to-end learning of dense correspondence for 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[32] Hao Wen, Yunze Liu, Jingwei Huang, Bo Duan, and Li Yi. Point primitive transformer for long-term 4d point cloud video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2022.

[33] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[34] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[35] Zhuoyang Zhang, Yuhao Dong, Yunze Liu, and Li Yi. Complete-to-partial 4d distillation for self-supervised point cloud sequence representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[36] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan Kankanhalli. Pstnet: Point spatio-temporal convolution on point cloud sequences. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[37] Hehe Fan, Yezhou Yang, and Mohan Kankanhalli. Point spatio-temporal transformer networks for point cloud video modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.

[38] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[39] Yunze Liu, Zifan Wang, Peiran Wu, and Jiayang Ao. Pointnet4d: A lightweight 4d point cloud video backbone for online and offline perception in robotic applications. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026.

[40] Yuhao Dong et al. Nsm4d: Neural scene model based online 4d point cloud sequence understanding. *arXiv preprint arXiv:2310.08326*, 2023.

[41] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[42] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*. IEEE, 2010.

[43] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, Joris Guerry, Bertrand Le Saux, and David Filliat. Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset. In *Proceedings of the Eurographics Workshop on 3D Object Retrieval (3DOR)*, 2017.

[44] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[45] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Appendix table of contents

A Preliminaries **15**

B Experimental details **15**

C Additional ablation studies **16**

D Additional experimental results **18**

D.1 Transfer to gesture recognition (HOI4D pretraining) 18

D.2 Qualitative visualization of point cloud reconstruction 18

D.3 Variance reporting and multi-seed statistics 18

E Positional leakage as a prerequisite barrier to distributional motion modeling **20**

F Theoretical justification: distributional uncertainty and action discriminability **22**

G Theoretical analysis: why motion diffusion performs well at small and large t **24**

G.1 Preliminaries: SNR and the DDPM noise schedule 24

G.2 Decomposition of the noise prediction target 24

G.3 Small t: fine-grained local trajectory recovery 25

G.4 Large t: coarse global semantic encoding 25

G.5 Intermediate t: transition and increased difficulty 26

G.6 Implications for representation learning 26

H Gradient propagation through the decoder: a known limitation and partial mitigation **27**

A Preliminaries

Dynamic point cloud and point tube construction. Let $\mathcal{P} = \{\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^L\} \in \mathbb{R}^{L\times N\times 3}$ denote a dynamic point cloud of L frames, where each frame $\mathbf{P}^l$ contains N points with (x, y, z) coordinates. Following MaST-Pre [14], we segment $\mathcal{P}$ into K spatio-temporal point tubes. Specifically, K keypoints $\{\hat{\mathbf{p}}_i\}_{i=1}^K$ are sampled via Farthest Point Sampling (FPS) applied to the first frame, where the i-th tube collects all points within spatial radius r and temporal distance $l/2$ of keypoint $\hat{\mathbf{p}}_i$:

$$\text{Tube}_{\hat{\mathbf{p}}_i} = \big\{\mathbf{p} \mid \mathbf{p} \in \mathcal{P},\ D_s(\mathbf{p}, \hat{\mathbf{p}}_i) < r,\ D_t(\mathbf{p}, \hat{\mathbf{p}}_i) < \tfrac{l}{2}\big\}. \tag{9}$$

Each tube is centered at its keypoint coordinate $\hat{\mathbf{p}}_i$, referred to as the tube center. After random masking with ratio ρ, the tubes are partitioned into a visible set $\mathcal{V}$ of size $K_v = \lfloor K(1-\rho)\rfloor$ and a masked set $\mathcal{M}$ of size $K_m = K - K_v$, with centers $\mathbf{C}^v \in \mathbb{R}^{K_v\times 3}$ and $\mathbf{C}^m \in \mathbb{R}^{K_m\times 3}$.

Denoising diffusion probabilistic models. DDPMs [9] define a forward Markov chain that progressively corrupts a clean signal $\mathbf{x}_0$ with Gaussian noise over T steps:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t;\ \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\ (1-\bar{\alpha}_t)\mathbf{I}\big)\,, \tag{10}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ and $\{\beta_t\}_{t=1}^T$ is a pre-defined noise schedule. A neural network ϵ_θ is trained to predict the added noise via

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{t,\,\mathbf{x}_0,\,\boldsymbol{\epsilon}}\Big[\big\|\boldsymbol{\epsilon} - \epsilon_\theta(\mathbf{x}_t,\ t,\ \mathbf{c})\big\|^2\Big]\,, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{11}$$

where $\mathbf{c}$ is an optional conditioning signal. Both diffusion modules in DiMP instantiate this framework with task-specific targets and conditioning.

B Experimental details

Datasets. We evaluate DiMP on four benchmark datasets spanning hand-object interaction, full-body action, and hand gesture understanding. **HOI4D** [41] is an egocentric dataset comprising 3,863 dynamic point cloud collected from 9 participants interacting with 800 distinct object instances across 610 indoor environments, with frame-level action labels across 19 action classes for action segmentation and point-level labels across 39 semantic categories for semantic segmentation. We follow the official split of 2,971 training scenes and 892 test scenes. **MSRAction-3D** [42] consists of 567 human point cloud videos spanning 20 action categories, with 270 videos used for training and 297 for testing. **SHREC'17** [43] is a gesture recognition benchmark comprising 2,800 videos across 28 gesture classes, split into 1,960 training and 840 test videos. **NvGesture** [44] consists of 1,532 videos covering 25 gesture classes, with 1,050 training and 482 test videos.

Common pre-training setup. Following MaST-Pre [14] and M2PSC [15], we employ P4Transformer [30] as the backbone encoder. During pre-training on HOI4D, 24 frames are densely sampled and 1,024 points are selected per frame with a temporal downsampling rate of 2 and a temporal kernel size $l = 3$ per point tube. The spatial downsampling rate is set to 32, the spatial neighborhood radius is $r = 0.1$, and 32 neighbor points are sampled within each spherical query. The masking ratio is set to 60% (Table 11). DiMP is pre-trained for 200 epochs using AdamW with an initial learning rate of 1×10^{-3} and cosine decay scheduling, with linear warmup for the first 10 epochs and a batch size of 128. All pre-training and fine-tuning experiments are conducted on 8 NVIDIA A100 GPUs. After pre-training, the decoder and both diffusion heads are discarded and only the encoder is retained and fine-tuned for each downstream task.

Motion target construction. Inter-frame displacement targets $\mathbf{M}_0$ are pre-computed offline using the frozen CorrNet3D [31] correspondence model before training begins. CorrNet3D is invoked once per dataset split to generate displacement files, which are stored on disk and loaded as static supervision labels during pretraining, in the same way that HOI4D action labels are loaded. No gradient flows through CorrNet3D at any point, and the correspondence files are discarded at inference time along with all other pretraining-only modules.

HOI4D action segmentation. For offline action segmentation, each sequence contains 150 frames and each frame contains 2,048 points. The model predicts an action label for every frame.

Table 8: Sensitivity to the number of stratified intervals h.

h	F1@50 ↑	mIoU ↑
1	79.94	46.1
2	80.13	46.8
4 (DiMP)	**80.77**	**47.6**
8	80.73	46.9

Table 9: Sensitivity to γ_{cen}. λ_{mot} fixed at 1.0.

γ_{cen}	F1@50 ↑
0.05	79.11
0.1	**80.77**
0.5	80.12
1.0	78.97

Table 10: Sensitivity to λ_{mot}. γ_{cen} fixed at 0.1.

λ_{mot}	F1@50 ↑
0.25	78.34
0.5	79.15
1.0	**80.77**
2.0	80.64

HOI4D semantic segmentation. For semantic segmentation, the full HOI4D dataset is used, where each sequence contains 300 frames of point clouds with 8,192 points per frame annotated across 39 semantic categories. During pre-training, the sequence length is set to 10 with 4,096 points per frame. Fine-tuning and testing are performed on sequences of length 3, consistent with prior methods.

MSRAction-3D action recognition. Following established convention [14, 15], 270 videos are used for training and 297 for testing. During fine-tuning, 24 frames are densely sampled and 2,048 points are selected per frame. AdamW with cosine decay is used for optimization.

HOI4D online inference. For online action segmentation, we follow the protocol of PointNet4D [39]. The model processes the dynamic point cloud sequentially and is permitted to access only the current and past frames at inference time. To ensure causal consistency, P4Transformer is equipped with a causal attention mask during online inference. All methods operate on sequences of 150 frames. We report both task performance metrics (Acc, Edit, F1@10, F1@25, F1@50) and computational efficiency (Clips/s).

Gesture recognition fine-tuning (NvGesture and SHREC'17). For the transfer-to-gesture-recognition experiments in Appendix D.1, all encoders are pre-trained on HOI4D following the common pre-training setup described above, and then fine-tuned on NvGesture and SHREC'17 under an end-to-end fine-tuning setting. During fine-tuning, 24 frames are densely sampled and 1,024 points are selected per frame, consistent with the HOI4D pre-training configuration. The AdamW optimizer is used with a cosine decay strategy. Following MaST-Pre [14] and M2PSC [15], the initial learning rate is set to 1×10^{-3} for NvGesture and 5×10^{-4} for SHREC'17, with a batch size of 24. We report recognition accuracy (%) at both 30 and 50 fine-tuning epochs. The reproduced MaST-Pre† and M2PSC† baselines are fine-tuned under identical settings to ensure a fair comparison.

C Additional ablation studies

Hyperparameter sensitivity. Tables 8, 9, and 10 report sensitivity to the number of stratified motion intervals h, the center-loss weight γ_{cen}, and the motion-loss weight λ_{mot}. Performance peaks at h=4, where more intervals enrich hierarchical supervision, while h=8 slightly hurts due to narrower per-interval ranges. DiMP is moderately sensitive to γ_{cen} and λ_{mot}, with optima at 0.1 and 1.0 respectively, and degrades smoothly elsewhere. Table 11 reports downstream performance as the tube masking ratio ρ is varied from 50% to 80%. DiMP adopts ρ=60% by default, at which performance peaks. Reducing ρ below 60% leaves too few masked tubes to provide sufficient pretraining difficulty, while increasing ρ beyond 60% gradually degrades performance as the decoder has less visible context for spatio-temporal inference. Crucially, even at ρ=75%, the ratio adopted by MaST-Pre [14] and M2PSC [15] for a directly comparable setup, DiMP achieves 79.45 F1@50 and 46.5 mIoU, exceeding the best prior method (M2PSC, 65.9 F1@50 and 42.3 mIoU) by clear margins of $+13.55$ and $+4.2$ respectively. This confirms that DiMP's improvements are not an artifact of a lower masking rate but arise from its distributional motion modeling and leakage-free center diffusion design.

Stop-Gradient on the Center Predictor. The stop-gradient operator $\text{sg}(\cdot)$ applied to $\hat{\mathbf{C}}_0^m$ before it enters the decoder (Eq. (4)) is designed to decouple θ_{cen} from $\mathcal{L}_{\text{geo}}$ and $\mathcal{L}_{\text{mot}}$, ensuring that the Center Predictor g_m is updated exclusively by the center denoising objective $\mathcal{L}_{\text{cen}}$. Table 12 ablates the two gradient paths independently by selectively removing the stop-gradient from each loss.

Table 11: Sensitivity of DiMP to the tube masking ratio ρ on HOI4D action segmentation and semantic segmentation. The row marked with $*$ uses the same masking ratio as MaST-Pre [14] and M2PSC [15] for a controlled comparison. All other DiMP settings are fixed.

Masking Ratio ρ	F1@50 ↑	mIoU ↑
50%	79.31	46.2
60% (DiMP)	**80.77**	**47.6**
70%	79.88	46.8
75%*	79.45	46.5
80%	78.93	45.9

w/o sg (both). Removing $\mathrm{sg}(\cdot)$ entirely allows both $\mathcal{L}_{\mathrm{geo}}$ and $\mathcal{L}_{\mathrm{mot}}$ to back-propagate through g_m, yielding the largest performance drop. The reconstruction objective drives g_m to produce centers that minimize Chamfer distance rather than noise prediction error, corrupting the positional inference signal. The motion objective compounds this by injecting an additional conflicting gradient.

sg on $\mathcal{L}_{\mathrm{geo}}$ only. Blocking only the geometric gradient leaves $\mathcal{L}_{\mathrm{mot}}$ free to interfere with g_m. Performance recovers partially but remains below the full DiMP setting, confirming that the motion diffusion gradient also distorts center denoising when left unblocked.

sg on $\mathcal{L}_{\mathrm{mot}}$ only. Blocking only the motion gradient while permitting the geometric gradient likewise falls short of the full design. The drop is slightly larger than the symmetric case above, consistent with $\mathcal{L}_{\mathrm{geo}}$ having a shorter gradient path to g_m through the decoder and thus exerting stronger interference.

DiMP (sg on both). Blocking both gradient paths achieves the best performance on both metrics, confirming that objective isolation is a necessary condition for reliable positional inference.

Table 12: Ablation of the stop-gradient mechanism on HOI4D. "sg($\mathcal{L}_{\mathrm{geo}}$)" and "sg($\mathcal{L}_{\mathrm{mot}}$)" indicate whether the gradient of each loss is blocked from entering g_m. All other settings follow the complete DiMP configuration.

Variant	sg($\mathcal{L}_{\mathrm{geo}}$)	sg($\mathcal{L}_{\mathrm{mot}}$)	F1@50 ↑	mIoU ↑
w/o sg (both)	×	×	77.31	43.8
sg on $\mathcal{L}_{\mathrm{geo}}$ only	✓	×	79.15	45.9
sg on $\mathcal{L}_{\mathrm{mot}}$ only	×	✓	78.67	45.3
DiMP (sg on both)	✓	✓	**80.77**	**47.6**

Sensitivity to Correspondence Quality (CorrNet3D Ablation). To assess how sensitive DiMP is to the quality of the external correspondence model used to generate motion targets, we evaluate downstream performance under five motion-target variants on HOI4D action segmentation (F1@50 and mIoU). All other DiMP settings are fixed.

Correspondence mechanisms. (a) **CorrNet3D** (default): the pre-trained correspondence model used throughout the paper. (b) **k-NN**: each point is matched to its nearest neighbor in the next frame by Euclidean distance, with no learned model. (c) **CorrNet3D + 5% noise**: 5% of correspondences are randomly permuted before computing displacements, simulating moderate correspondence error. (d) **CorrNet3D + 20% noise**: 20% of correspondences are randomly permuted, simulating severe correspondence degradation.

Table 13: Sensitivity of DiMP to motion-target quality on HOI4D action segmentation. All variants use the complete DiMP architecture and only the correspondence mechanism for building $\mathbf{M}_0$ changes. Δ denotes the absolute F1@50 change relative to the DiMP default.

Motion Target	Correspondence Mechanism	F1@50 ↑	mIoU ↑	Δ F1@50
CorrNet3D (DiMP default)	Learned correspondence	**80.77**	**47.6**	0.00
k-NN (no model)	Euclidean nearest-neighbor	79.12	45.7	-1.65
CorrNet3D + 5% noise	+ random permutation	80.43	47.3	-0.34
CorrNet3D + 20% noise	+ random permutation	79.48	46.3	-1.29
No motion branch	none	77.28	43.5	-3.49

The key comparison is between k-NN and CorrNet3D. If the zero-training-cost k-NN mechanism still enables DiMP to substantially outperform the no-motion-branch baseline, it establishes that the gains arise from the diffusion-based pretext objective itself rather than from representational capacity imported from CorrNet3D. The noise-injection variants further quantify the graceful degradation curve, showing at what correspondence error rate the gains are materially diminished.

D Additional experimental results

D.1 Transfer to gesture recognition (HOI4D pretraining)

We report an *unofficial* supplementary evaluation in which all encoders are instead pre-trained on HOI4D and subsequently fine-tuned on NvGesture and SHREC'17 for gesture recognition. To our knowledge, no prior work has reported results under this protocol, as the official codebases of existing methods do not provide pretraining configurations for the HOI4D dataset. To enable a fair comparison, we independently re-implemented the pretraining procedures of MaST-Pre [14] and M2PSC [15] following their respective papers as closely as possible. Results obtained under these reproduced configurations are marked with † in Table 14. We report recognition accuracy (%) after fine-tuning for 30 and 50 epochs, following the same fine-tuning setup as Appendix B. As shown in Table 14, DiMP outperforms all reproduced baselines at both fine-tuning budgets on both datasets. At the 30-epoch budget, DiMP improves over the from-scratch P4Transformer by $+1.4$ on NvGesture and $+2.3$ on SHREC'17, and over the strongest reproduced baseline (M2PSC†) by $+0.4$ and $+0.3$ respectively. At the 50-epoch budget, the gains over P4Transformer are $+1.8$ on NvGesture and $+1.5$ on SHREC'17, while the margins over M2PSC† grow to $+0.7$ and $+0.9$, indicating that DiMP's distributional motion objective yields representations that remain competitive and even improve relative to deterministic baselines as fine-tuning proceeds.

Table 14: Gesture recognition accuracy (%) on NvGesture (NvG) and SHREC'17 (SHR) when all encoders are pre-trained on HOI4D. Methods marked with † are re-implemented by us, as no official HOI4D pretraining scripts are publicly available for these methods.

Method	30 Epochs		50 Epochs	
	NvG	SHR	NvG	SHR
P4Transformer [30]	83.7	85.9	86.1	89.9
P4Transformer + MaST-Pre† [14]	84.4	87.3	86.7	90.1
P4Transformer + M2PSC† [15]	84.7	87.9	87.2	90.5
P4Transformer + DiMP (Ours)	**85.1**	**88.2**	**87.9**	**91.4**

D.2 Qualitative visualization of point cloud reconstruction

Figure 6 provides qualitative visualizations of the dynamic point cloud reconstructed by DiMP during the pretraining stage. Each row shows a multi-frame input sequence alongside the corresponding reconstruction produced by the masked decoder. The results demonstrate that DiMP recovers accurate per-frame geometry across diverse object interactions and viewpoints, confirming that the probabilistic motion diffusion objective does not impair the geometric reconstruction quality of the backbone while simultaneously enabling distributional motion modeling.

D.3 Variance reporting and multi-seed statistics

To quantify result stability, DiMP and M2PSC are each evaluated over 3 independent random seeds, with pretraining seed and fine-tuning seed held together per run. Table 15 reports mean $\pm$ std for the two action segmentation benchmarks.

Across all tasks, DiMP's standard deviation does not exceed 0.7 points, indicating stable convergence across seeds. Crucially, every absolute improvement of DiMP over M2PSC exceeds the combined standard deviations by a factor of at least $4\times$, confirming that the reported gains are not attributable to random seed variance.

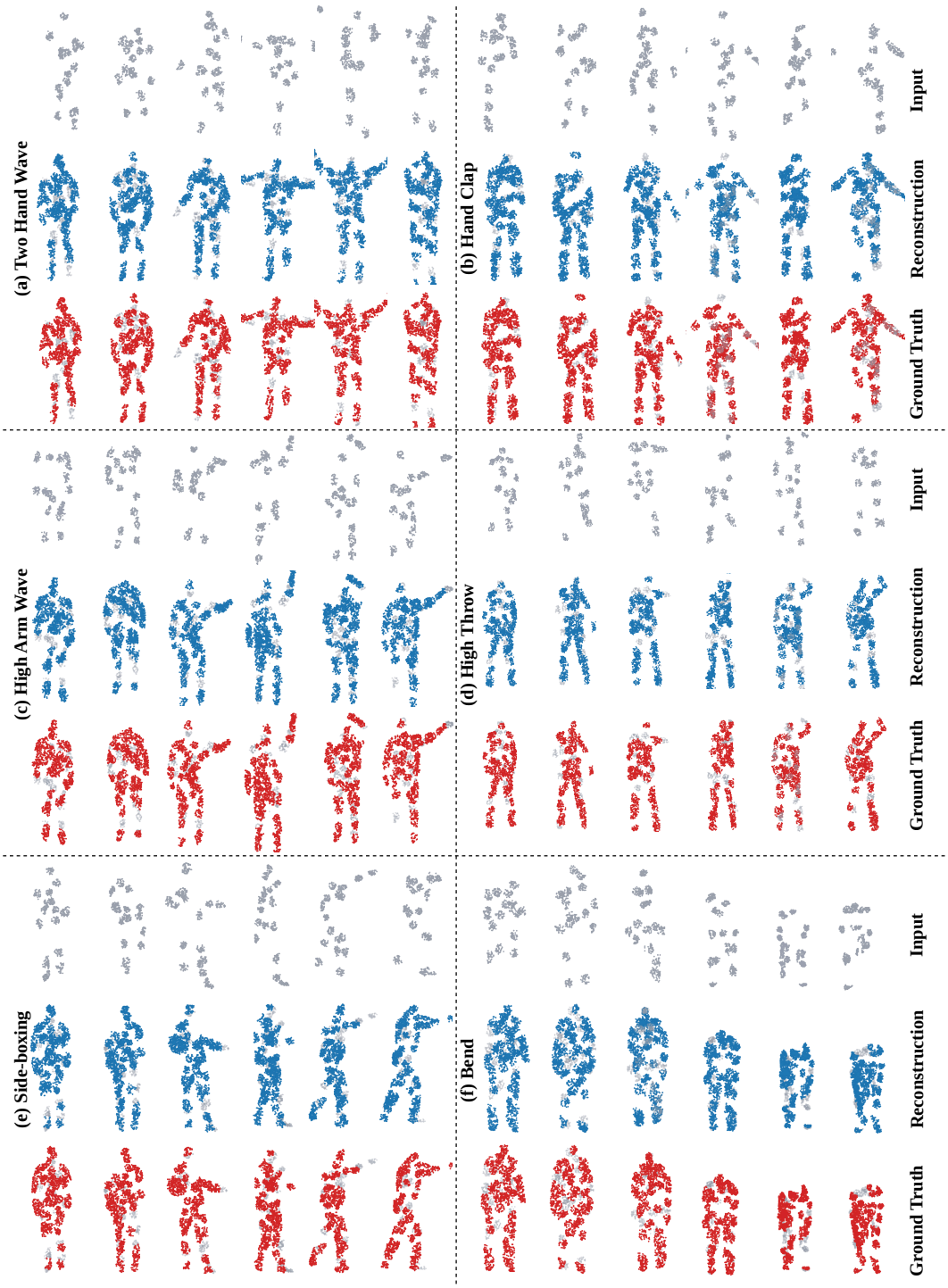


Figure 6: Qualitative visualization of dynamic point cloud reconstruction. Each example shows the masked input (top), reconstruction produced by DiMP's masked decoder (middle), and ground truth (bottom). The results illustrate that DiMP faithfully recovers fine-grained per-frame geometry across diverse action sequences, despite the high masking ratio (60%) applied during pretraining.

Table 15: Multi-seed statistics (mean ± std, 3 seeds) for DiMP and M2PSC on HOI4D offline and online action segmentation. Seed indices match across both methods for fair comparison.

Task	Method	Acc	Edit	F1@10	F1@25	F1@50
HOI4D offline seg	M2PSC	75.9 ± 0.4	77.1 ± 0.5	78.4 ± 0.4	74.1 ± 0.5	65.9 ± 0.5
	DiMP	82.4 ± 0.5	84.9 ± 0.6	87.3 ± 0.5	85.9 ± 0.6	80.8 ± 0.5
HOI4D online seg	M2PSC	73.2 ± 0.5	74.8 ± 0.6	76.1 ± 0.5	72.3 ± 0.6	63.1 ± 0.6
	DiMP	80.4 ± 0.6	82.2 ± 0.7	84.8 ± 0.6	83.2 ± 0.7	73.5 ± 0.6

E Positional leakage as a prerequisite barrier to distributional motion modeling

Overview. The motion diffusion branch of DiMP is conditioned on the decoded representation $\mathbf{Z}_{\mathrm{dec}}$ produced by the masked decoder. For this branch to capture the full multimodal distribution of inter-frame motion trajectories, $\mathbf{Z}_{\mathrm{dec}}$ must encode rich, globally coherent motion context rather than merely local geometric features. We show here, both analytically and empirically, that positional leakage, which injects ground-truth masked tube centers $\mathbf{c}^*$ as decoder positional embeddings, collapses decoder cross-attention into local geometric retrieval, severing the informational pathway through which the motion diffusion loss drives the encoder to internalize motion distributions. Consequently, eliminating positional leakage is a *necessary precondition* for the motion diffusion branch to function as a genuine distributional learner.

Notation. Let $\mathbf{Z}_{\mathrm{enc}} = f_{\mathrm{enc}}(\mathbf{X})$ denote the encoder output from visible tokens $\mathbf{X}$. The masked decoder produces, for each masked tube k,

$$\mathbf{Z}_{\mathrm{dec},k} = \mathrm{CrossAttn}\big(W_Q\big(\mathbf{m}_k + \mathrm{PE}(\mathbf{p}_k)\big),\ W_K\mathbf{Z}_{\mathrm{enc}},\ W_V\mathbf{Z}_{\mathrm{enc}}\big)\,, \tag{12}$$

where $\mathbf{m}_k$ is a learnable masked token embedding and $\mathbf{p}_k$ is the positional input supplied to the decoder query. Under **positional leakage**, $\mathbf{p}_k = \mathbf{c}_k^*$ is the ground-truth center. Under **DiMP's center diffusion**, $\mathbf{p}_k = \hat{\mathbf{c}}_k$ is the predicted center with residual uncertainty. Let $\mathbf{M}$ denote the inter-frame displacement field and A the action class label.

Assumption 1 (Strong positional dominance). *We consider the limiting regime in which (a) the positional embedding norm dominates the learnable mask token,* i.e.*,* $\|W_Q\mathrm{PE}(\mathbf{c}_k^*)\| \gg \|W_Q\mathbf{m}_k\|$*, and (b) the encoder has already learned a position-aligned key mapping, in the sense that the similarity gap*

$$\gamma_k := \max_{j'} \langle W_Q\mathrm{PE}(\mathbf{c}_k^*),\, W_K\mathbf{Z}_{\mathrm{enc},j'}\rangle - \mathrm{second}_{j'}\langle W_Q\mathrm{PE}(\mathbf{c}_k^*),\, W_K\mathbf{Z}_{\mathrm{enc},j'}\rangle \tag{13}$$

is strictly positive and grows with $\|\mathrm{PE}(\mathbf{c}_k^*)\|$.

Assumption 1(a) is a standard consequence of injecting ground-truth centers into a trained Transformer, where positional features routinely exceed content-token norms by an order of magnitude [1]. Assumption 1(b) is empirically observable in any decoder trained with positional supervision and is the source of the positional shortcut behavior previously documented for masked autoencoders [1, 2, 20]. Both conditions are *regime assumptions*: they describe the limit in which leakage is severe. The following property is stated as a limit-case result under this regime, not as a statement about the fully trained network at finite temperature.

Property 1 (Attention concentration in the leakage regime). *Under Assumption 1 with* $\tau = 1$ *held fixed, the softmax attention weight* α_{kj} *in Eq.* (12) *concentrates around the single encoder token* $j_k^* := \arg\max_{j'} \langle W_Q\mathrm{PE}(\mathbf{c}_k^*),\, W_K\mathbf{Z}_{\mathrm{enc},j'}\rangle$*:*

$$1 - \alpha_{k,j_k^*} \ \le\ (T_{\mathrm{vis}} - 1)\, e^{-\gamma_k}, \qquad \alpha_{k,j_k^*} \ \to\ 1 \ \text{ as } \ \gamma_k \to \infty. \tag{14}$$

Proof. Write the attention logits as $e_{kj} = \langle W_Q(\mathbf{m}_k + \mathrm{PE}(\mathbf{c}_k^*)),\, W_K\mathbf{Z}_{\mathrm{enc},j}\rangle = \tilde{e}_{kj} + \delta_{kj}$, where $\tilde{e}_{kj} = \langle W_Q\mathrm{PE}(\mathbf{c}_k^*),\, W_K\mathbf{Z}_{\mathrm{enc},j}\rangle$ and $|\delta_{kj}| \le \|W_Q\mathbf{m}_k\|\,\|W_K\mathbf{Z}_{\mathrm{enc},j}\|$ is negligible under Assumption 1(a). Under Assumption 1(b), $\tilde{e}_{k,j_k^*}$ exceeds every other $\tilde{e}_{kj'}$ by at least γ_k, and the standard softmax tail bound then yields $1 - \alpha_{k,j_k^*} \le (T_{\mathrm{vis}} - 1)\, e^{-\gamma_k}$. □

Property 1 recovers the one-hot collapse in the limit $\gamma_k \to \infty$ but is a finite-γ_k concentration bound, which is the meaningful regime for trained networks. We next use it to argue that the decoder representation under leakage effectively depends only on $\mathbf{Z}_{\mathrm{enc},j_k^*}$ up to a vanishing residual.

Under Eq. (14), $\mathbf{Z}^{\mathrm{leak}}_{\mathrm{dec},k} = W_V \mathbf{Z}_{\mathrm{enc},j_k^*} + O(e^{-\gamma_k})$ is a *near-single-token appearance summary* that carries only the local geometric content of one encoder token up to an exponentially small residual in γ_k, rather than the global spatio-temporal context required to model motion distributions.

Corollary 1 (Motion information bottleneck in the strong leakage limit). *In the limit $\gamma_k \to \infty$ of Assumption 1, the mutual information between the inter-frame displacement $\mathbf{M}$ and the leaked decoder representation is bounded as*

$$I(\mathbf{M};\, \mathbf{Z}^{\mathrm{leak}}_{\mathrm{dec}}) \;\le\; I(\mathbf{M};\, \mathbf{Z}_{\mathrm{enc},j^*}) \;\le\; I(\mathbf{M};\, \mathbf{Z}_{\mathrm{enc}}). \tag{15}$$

The first inequality is strict whenever $\mathbf{M}$ depends on encoder tokens other than j_k^*, a condition that holds generically for any motion involving multiple body parts or objects.

Proof. In the limit $\gamma_k \to \infty$, the residual in the previous display vanishes and $\mathbf{Z}^{\mathrm{leak}}_{\mathrm{dec},k}$ becomes a deterministic function of $\mathbf{Z}_{\mathrm{enc},j_k^*}$ alone, so the first inequality follows from the data processing inequality. The second inequality holds because $\mathbf{Z}_{\mathrm{enc},j_k^*}$ is a component of the full encoder representation $\mathbf{Z}_{\mathrm{enc}}$, giving $I(\mathbf{M};\, \mathbf{Z}_{\mathrm{enc},j_k^*}) \le I(\mathbf{M};\, \mathbf{Z}_{\mathrm{enc}})$ by the chain rule and the non-negativity of conditional mutual information. □

Gradient flow analysis. During training, the motion diffusion loss gradient flows back to the encoder via the decoder cross-attention:

$$\frac{\partial \mathcal{L}_{\mathrm{mot}}}{\partial \mathbf{Z}_{\mathrm{enc}}} = \sum_k \frac{\partial \mathcal{L}_{\mathrm{mot}}}{\partial \mathbf{Z}_{\mathrm{dec},k}} \cdot \frac{\partial \mathbf{Z}_{\mathrm{dec},k}}{\partial \mathbf{Z}_{\mathrm{enc}}}. \tag{16}$$

Under the attention concentration of Property 1, the Jacobian $\partial \mathbf{Z}_{\mathrm{dec},k} / \partial \mathbf{Z}_{\mathrm{enc}}$ is dominated by its component at token j_k^*, with contributions from other tokens suppressed by $O(e^{-\gamma_k})$: the gradient effectively reaches only one encoder token per masked query, updating at most K out of T_{vis} visible tokens (where $T_{\mathrm{vis}} \gg K$ in practice). This *effective gradient sparsity* means the encoder learns to encode only local retrieval-relevant features at specific spatial positions, not the globally consistent motion-distribution context that the diffusion head requires. When positional leakage is removed, attention is distributed across all encoder tokens (since $\hat{\mathbf{c}}_k$ carries residual uncertainty), the Jacobian is dense, and the motion diffusion loss provides globally informative gradient signals that drive the encoder to internalize the full motion distribution.

Proposition 2 (Leakage caps action-discriminative information)**.** Let $\mathbf{Z}^{\mathrm{leak},*}_{\mathrm{enc}}$ and $\mathbf{Z}^{\mathrm{DiMP},*}_{\mathrm{enc}}$ be the encoder representations at the population-optimum of the motion diffusion losses $\mathcal{L}^{\mathrm{leak}}_{\mathrm{mot}}$ and $\mathcal{L}^{\mathrm{DiMP}}_{\mathrm{mot}}$, respectively. Assume:

(A1) **Attention concentration under leakage**: the leakage branch satisfies Assumption 1 in the limit $\gamma_k \to \infty$, so Corollary 1 applies and $I(\mathbf{M}; \mathbf{Z}^{\mathrm{leak}}_{\mathrm{dec}}) \le I(\mathbf{M}; \mathbf{Z}_{\mathrm{enc},j^*})$.

(A2) **Markov structure**: the data process follows $A \to \mathbf{M} \to \mathbf{Z}_{\mathrm{enc}} \to \mathbf{Z}_{\mathrm{dec}}$, so the DPI gives $I(A; \mathbf{Z}_{\mathrm{enc}}) \le I(A; \mathbf{M})$.

(A3) **Global dependence of A on $\mathbf{M}$**: the action label depends on displacement information from more than one spatial location, *i.e.*, $I(A; \mathbf{M}) > I(A; \mathbf{M}_{j^*})$ where $\mathbf{M}_{j^*}$ restricts $\mathbf{M}$ to the neighborhood of the bottleneck token j^*.

(A4) **Variational tightness gap**: the DDPM loss is a variational upper bound on $-\mathbb{E}[\log p_\theta(\mathbf{M} \mid \mathbf{Z}_{\mathrm{dec}})]$, and the population optimum of the encoder under $\mathcal{L}^{\mathrm{DiMP}}_{\mathrm{mot}}$ attains a strictly tighter bound than under $\mathcal{L}^{\mathrm{leak}}_{\mathrm{mot}}$ when (A1)–(A3) hold.

Then

$$I\big(A;\, \mathbf{Z}^{\mathrm{leak},*}_{\mathrm{enc}}\big) \;<\; I\big(A;\, \mathbf{Z}^{\mathrm{DiMP},*}_{\mathrm{enc}}\big)\,. \tag{17}$$

Proof. By (A4), at the population optimum the loss $\mathcal{L}_{\mathrm{mot}}$ coincides with the conditional entropy $H(\mathbf{M} \mid \mathbf{Z}_{\mathrm{dec}})$ (up to a constant in $\mathbf{Z}_{\mathrm{dec}}$), so minimizing it is equivalent to maximizing $I(\mathbf{M}; \mathbf{Z}_{\mathrm{dec}})$ as a functional of the encoder. Under leakage and (A1), this objective is upper-bounded by $I(\mathbf{M}; \mathbf{Z}_{\mathrm{enc},j^*})$;

under DiMP's center diffusion this cap is removed and the functional is upper-bounded only by the larger quantity $I(\mathbf{M}; \mathbf{Z}_{\text{enc}})$. Therefore the population-optimal encoder under DiMP retains *at least* as much information about $\mathbf{M}$ as under leakage, and by (A3) retains strictly more about the part of $\mathbf{M}$ that is action-discriminative: $I(\mathbf{M}; \mathbf{Z}_{\text{enc}}^{\text{DiMP},*}) > I(\mathbf{M}; \mathbf{Z}_{\text{enc}}^{\text{leak},*})$. Combining with (A2) and the action-discriminability decomposition $I(A; \mathbf{Z}) = I(A; \mathbf{M}) - I(A; \mathbf{M} \mid \mathbf{Z})$ yields $I(A; \mathbf{Z}_{\text{enc}}^{\text{leak},*}) < I(A; \mathbf{Z}_{\text{enc}}^{\text{DiMP},*})$. □

Remark 1. *We state this as a* proposition *rather than a theorem because Assumption (A4)—that optimizing the variational surrogate at population scale achieves a strictly tighter bound under DiMP than under leakage—is the standard assumption under which variational bound improvements translate to information improvements, but is not itself proved here. Assumptions (A1)–(A3) are verifiable: (A1) follows from Property 1 in the leakage regime; (A2) is the modeling assumption articulated at the beginning of Appendix F; (A3) is a structural property of multi-joint or multi-object actions. The empirical validation in the main text (Section 4.2, Table 6) provides direct evidence that the overall implication* (17) *holds on real dynamic point cloud data.*

Empirical validation. The 2×2 factorial experiment is reported in the main text (Table 6, Section 4.2). It shows the same asymmetric interaction predicted here, negligible $\Delta(+\text{MD})$ without center diffusion and a large gain once leakage is removed, consistent with Corollary 1 and the gradient-sparsity discussion above.

F Theoretical justification: distributional uncertainty and action discriminability

Motivation. Existing masked-reconstruction methods for dynamic point cloud address motion supervision through deterministic proxy targets, such as temporal cardinality difference [14] or point trajectory regression [15]. These objectives reduce supervision to fitting a conditional expectation, training the model to predict $\mathbb{E}[\Delta\mathbf{p} \mid \text{context}]$. In scenarios involving occlusion, ambiguous interactions, or complex dynamics, this mean-regression regime collapses multiple plausible motion modes into a single deterministic solution, discarding precisely the distributional structure that distinguishes action classes from one another [16, 45, 17]. DDPMs, by contrast, are capable of learning the full conditional distribution over outcomes, an ideal inductive bias for multimodal motion trajectories. We formalize this intuition below.

Setup. Let $A \in \mathcal{A}$ denote the action class, $\mathbf{M} \in \mathbb{R}^d$ the inter-frame displacement vector, and $\mathbf{Z}$ the encoder representation. We assume the generative Markov chain $A \to \mathbf{M} \to \mathbf{Z}$, so that the data processing inequality gives $I(A; \mathbf{Z}) \le I(A; \mathbf{M})$. Two distinct statistics of $\mathbf{M}$ arise in the following analysis and must be carefully distinguished:

- the *class-conditional mean trajectory* $\bar{\mathbf{M}}(A) := \mathbb{E}[\mathbf{M} \mid A]$, a deterministic function of A. We denote this random variable by $\bar{\mathbf{M}}$.
- the *deterministic mean-regression output* $\hat{\mathbf{M}}(\mathbf{Z}) := \mathbb{E}[\mathbf{M} \mid \mathbf{Z}]$, a deterministic function of $\mathbf{Z}$. We denote this random variable by $\hat{\mathbf{M}}$.

Both are d-dimensional vectors in motion space but play different roles in the analysis below.

Proposition 3 (Bayes-error obstruction of the class-mean statistic)**.** Suppose there exist distinct action classes $a_1, a_2 \in \mathcal{A}$ such that $\bar{\mathbf{M}}(a_1) = \bar{\mathbf{M}}(a_2)$ while $p(\mathbf{M} \mid A = a_1) \neq p(\mathbf{M} \mid A = a_2)$. Then

(i) the Bayes-optimal classifier based on $\bar{\mathbf{M}}$ cannot distinguish a_1 from a_2 above chance: for any realization $c^* = \bar{\mathbf{M}}(a_1) = \bar{\mathbf{M}}(a_2)$, $p(A = a_i \mid \bar{\mathbf{M}} = c^*) \propto p(A = a_i)$ for $i = 1, 2$.

(ii) in contrast, the Bayes-optimal classifier based on $\mathbf{M}$ attains strictly lower posterior entropy for those realizations of $\mathbf{M}$ on which $p(\mathbf{M} \mid A = a_1)$ and $p(\mathbf{M} \mid A = a_2)$ differ, giving $H(A \mid \mathbf{M}) < H(A \mid \bar{\mathbf{M}})$ and hence the strict information gap

$$I(A; \bar{\mathbf{M}}) \;<\; I(A; \mathbf{M}). \tag{18}$$

Proof. (i) Since $\bar{\mathbf{M}}$ takes the same value under both classes, $p(\bar{\mathbf{M}} = c^* \mid A = a_1) = p(\bar{\mathbf{M}} = c^* \mid A = a_2) = 1$, so Bayes' rule yields $p(A = a_i \mid \bar{\mathbf{M}} = c^*) \propto p(A = a_i)$. *(ii)* By hypothesis there

exists $m \in \mathbb{R}^d$ with $p(m \mid a_1) \neq p(m \mid a_2)$, so the posterior $p(A \mid \mathbf{M} = m)$ differs from the prior and has strictly lower entropy than $p(A \mid \bar{\mathbf{M}} = c^*)$ for the corresponding class-pair $\{a_1, a_2\}$ on a positive-measure set. Taking expectations over the marginal distributions of $\mathbf{M}$ and $\bar{\mathbf{M}}$ and combining with the trivial bound $H(A \mid \mathbf{M}) \leq H(A \mid \bar{\mathbf{M}})$ on the complement yields $H(A \mid \mathbf{M}) < H(A \mid \bar{\mathbf{M}})$, which is equivalent to (18). □ □

Remark 2. *Unlike the data processing inequality, $\bar{\mathbf{M}}$ is a function of A (not of $\mathbf{M}$), so* (18) *does not follow from DPI directly and requires the posterior-entropy comparison above. The condition in Proposition 3 is generic for multimodal action-conditional motion distributions: whenever two classes share the same first moment but differ in higher-order statistics (*e.g.*, two oscillatory gestures with opposite phases but the same average displacement), the hypothesis holds. The practical relevance of this result lies in the fact that a mean-regression* estimator $\hat{\mathbf{M}}(\mathbf{Z}) := \mathbb{E}[\mathbf{M} \mid \mathbf{Z}]$*, the test-time output of a deterministic regressor, inherits the same obstruction when its training target aligns with class-conditional means. Consequently, the downstream classifier operating on $\hat{\mathbf{M}}$ cannot separate a_1 from a_2 whenever the regressor saturates to their shared mean. We make this precise via the Fano inequality below.*

Fano-inequality quantification. The impact of any lossy motion summary on downstream action recognition is quantified via the *Fano inequality*. Consider a predictor $\hat{A}$ of A that accesses only a summary $\hat{\mathbf{M}} = \phi(\mathbf{M})$ or $\hat{\mathbf{M}} = \phi(\mathbf{Z})$ (a deterministic function of $\mathbf{M}$ or $\mathbf{Z}$). The Fano inequality gives the standard lower bound on the Bayes error

$$P_e \;\geq\; \frac{H(A \mid \hat{A}) - H_2(P_e)}{\log(|\mathcal{A}| - 1)} \;\geq\; \frac{H(A \mid \hat{\mathbf{M}}) - 1}{\log |\mathcal{A}|}, \tag{19}$$

where the second inequality uses $H_2(P_e) \leq 1$, $\log(|\mathcal{A}| - 1) \leq \log |\mathcal{A}|$, and $H(A \mid \hat{A}) \geq H(A \mid \hat{\mathbf{M}})$ (by DPI since $\hat{A}$ is a function of $\hat{\mathbf{M}}$). Rewriting via $H(A \mid \hat{\mathbf{M}}) = H(A) - I(A; \hat{\mathbf{M}})$, the lower bound is driven by the information gap

$$\Delta I := I(A; \mathbf{M}) - I(A; \hat{\mathbf{M}}) \;\geq\; 0, \tag{20}$$

which is non-negative by the DPI applied to the Markov chain $A \to \mathbf{M} \to \hat{\mathbf{M}}$.

Quantifying ΔI for deterministic mean regression. Specialize to $\hat{\mathbf{M}} = \hat{\mathbf{M}}(\mathbf{Z}) = \mathbb{E}[\mathbf{M} \mid \mathbf{Z}]$, the output of a deterministic mean-regression model with encoder $\mathbf{Z}$. Since $\hat{\mathbf{M}}$ is a deterministic function of $\mathbf{Z}$, we have $I(A; \hat{\mathbf{M}}) \leq I(A; \mathbf{Z}) \leq I(A; \mathbf{M})$. To further characterize the second gap, we analyze the population-optimal regressor, for which $\hat{\mathbf{M}}(\mathbf{Z})$ is a fixed deterministic function of $\mathbf{Z}$ and hence the full Markov chain $A \to \mathbf{M} \to \mathbf{Z} \to \hat{\mathbf{M}}$ holds, giving $A \perp \hat{\mathbf{M}} \mid \mathbf{M}$ (*i.e.*, $I(A; \hat{\mathbf{M}} \mid \mathbf{M}) = 0$). Applying the mutual information chain rule along this chain yields the *exact* decomposition

$$I(A; \mathbf{M}) - I(A; \hat{\mathbf{M}}) \;=\; I(A; \mathbf{M} \mid \hat{\mathbf{M}}) \;=\; H(\mathbf{M} \mid \hat{\mathbf{M}}) - H(\mathbf{M} \mid A, \hat{\mathbf{M}}) \;\geq\; 0, \tag{21}$$

where the first equality uses $I(A; \mathbf{M}, \hat{\mathbf{M}}) = I(A; \mathbf{M}) + I(A; \hat{\mathbf{M}} \mid \mathbf{M}) = I(A; \mathbf{M})$ by the Markov assumption, combined with the symmetric expansion $I(A; \mathbf{M}, \hat{\mathbf{M}}) = I(A; \hat{\mathbf{M}}) + I(A; \mathbf{M} \mid \hat{\mathbf{M}})$, and the second equality uses the definition of conditional mutual information. Eq (21) vanishes *only* when $\mathbf{M}$ is conditionally independent of A given $\hat{\mathbf{M}}$, or equivalently when the regression output $\hat{\mathbf{M}}$ is a sufficient statistic of $\mathbf{M}$ for A. This sufficiency condition fails precisely when within-class trajectory variance carries discriminative information about the action, which is typical in complex human action scenarios involving occlusion, interaction ambiguity, or subtle dynamics. This is precisely the setting where discriminative pretraining matters most. Substituting (21) into the Fano bound (19), the Bayes error lower bound for any classifier using only $\hat{\mathbf{M}}$ exceeds the bound using the full $\mathbf{M}$ by at least $I(A; \mathbf{M} \mid \hat{\mathbf{M}}) / \log |\mathcal{A}|$.

How DiMP's motion diffusion objective addresses ΔI. Unlike deterministic regression, DiMP's motion diffusion head is trained via a DDPM denoising objective that, by standard results, is a variational upper bound on the conditional negative log-likelihood $-\mathbb{E}[\log p_\theta(\mathbf{M} \mid \mathbf{Z}_{\text{dec}})]$ [9]. Minimizing this surrogate therefore drives $p_\theta(\mathbf{M} \mid \mathbf{Z}_{\text{dec}})$ toward the true conditional distribution $p(\mathbf{M} \mid \mathbf{Z}_{\text{dec}})$, which in turn encourages $\mathbf{Z}_{\text{dec}}$ (and hence the encoder output) to retain the information

needed to reconstruct the *full* distribution of $\mathbf{M}$ rather than merely its conditional mean. In particular, the representation produced by a diffusion-trained encoder is not subject to the sufficiency obstruction that forces $\Delta I \geq I(A; \mathbf{M} \mid \hat{\mathbf{M}}) > 0$ in the deterministic regression regime. We do not claim that the training procedure *provably* attains $\Delta I = 0$ at any finite horizon. Instead, it replaces a hard sufficiency obstacle with a variationally tightened surrogate whose optimum coincides with the full conditional distribution.

G Theoretical analysis: why motion diffusion performs well at small and large t

This appendix provides a rigorous theoretical analysis of the diffusion timestep-dependent behavior observed in Figure 1(c), where the motion diffusion head achieves particularly accurate displacement recovery at both small and large values of t. We show that this U-shaped error profile is a principled consequence of the signal-to-noise ratio (SNR) structure of the DDPM forward process and the nature of the motion supervision target.

G.1 Preliminaries: SNR and the DDPM noise schedule

Recall from Eq. (10) that the forward marginal satisfies

$$q(\mathbf{M}_t \mid \mathbf{M}_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\,\mathbf{M}_0,\ (1-\bar{\alpha}_t)\mathbf{I}\big)\,, \tag{22}$$

which can be equivalently written in the reparameterized form

$$\mathbf{M}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{M}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{23}$$

The *signal-to-noise ratio* at timestep t is defined as

$$\mathrm{SNR}(t) \;=\; \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}. \tag{24}$$

Under a monotone decreasing noise schedule, $\mathrm{SNR}(t)$ is a strictly decreasing function of t:

- At **small** t: $\bar{\alpha}_t \approx 1$, so $\mathrm{SNR}(t) \gg 1$. The noisy motion $\mathbf{M}_t$ closely resembles $\mathbf{M}_0$. Most of the injected noise has a small magnitude and resides in the high-frequency (fine-grained) components of the signal.
- At **large** t: $\bar{\alpha}_t \approx 0$, so $\mathrm{SNR}(t) \ll 1$. The signal is almost entirely drowned in noise. Only the low-frequency (coarse semantic) structure survives in $\mathbf{M}_t$.

G.2 Decomposition of the noise prediction target

The DDPM objective in Eq. (7) requires the network h_m to predict the injected noise $\boldsymbol{\epsilon}^{\mathbf{M}}$ from the corrupted motion $\mathbf{M}_t$ and the decoder conditioning $\mathbf{Z}_{\mathrm{dec}}$. Using the reparameterization (23), the noise can be recovered as

$$\boldsymbol{\epsilon}^{\mathbf{M}} = \frac{\mathbf{M}_t - \sqrt{\bar{\alpha}_t}\,\mathbf{M}_0}{\sqrt{1-\bar{\alpha}_t}}. \tag{25}$$

An equivalent formulation expresses the network output in terms of a *predicted clean signal* $\hat{\mathbf{M}}_0$:

$$\hat{\mathbf{M}}_0(t) = \frac{\mathbf{M}_t - \sqrt{1-\bar{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}(t)}{\sqrt{\bar{\alpha}_t}}, \tag{26}$$

so that the mean squared noise prediction error $\|\boldsymbol{\epsilon}^{\mathbf{M}} - \hat{\boldsymbol{\epsilon}}\|^2$ is equivalent to the clean-signal prediction error up to a timestep-dependent rescaling factor $\mathrm{SNR}(t)$:

$$\big\|\boldsymbol{\epsilon}^{\mathbf{M}} - \hat{\boldsymbol{\epsilon}}\big\|^2 = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}\big\|\mathbf{M}_0 - \hat{\mathbf{M}}_0(t)\big\|^2 = \mathrm{SNR}(t)\cdot\big\|\mathbf{M}_0 - \hat{\mathbf{M}}_0(t)\big\|^2. \tag{27}$$

Eq. (27) reveals that *minimizing the noise prediction loss is equivalent to minimizing the clean-signal reconstruction error, weighted by the SNR*. This equivalence determines how tightly each timestep constrains the model's *functional* output $\hat{\mathbf{M}}_0(t)$, which we make precise below.

Proposition 4 (SNR-Weighted Functional Supervision)**.** Let $\hat{\mathbf{M}}_0(t)$ denote the implicitly predicted clean signal induced by the network output $\hat{\boldsymbol{\epsilon}}(t)$ via (26). Viewing $\mathcal{L}_{\text{mot}}$ as a functional of $\hat{\mathbf{M}}_0(t)$,

$$\frac{\partial \left\|\boldsymbol{\epsilon}^{\mathbf{M}} - \hat{\boldsymbol{\epsilon}}\right\|^2}{\partial \hat{\mathbf{M}}_0(t)} = -2\,\mathrm{SNR}(t) \cdot \big(\mathbf{M}_0 - \hat{\mathbf{M}}_0(t)\big). \tag{28}$$

Consequently, at small t a deviation of the functional prediction $\hat{\mathbf{M}}_0$ from $\mathbf{M}_0$ incurs a large $\mathrm{SNR}(t)$-amplified loss penalty and hence a tight constraint on the network's functional output, whereas at large t the same deviation incurs only a small penalty, so the network is constrained only at the coarse level of the conditional mean.

Proof. Substituting (27) and differentiating with respect to $\hat{\mathbf{M}}_0$ yields (28) directly. □

Remark 3 (Functional vs. parameter gradient)**.** *Proposition 4 characterizes the* functional *sensitivity of $\mathcal{L}_{\text{mot}}$ to the predicted clean motion $\hat{\mathbf{M}}_0$, which is the quantity that controls how accurately the network must approximate $\mathbf{M}_0$ to achieve low loss at each timestep. This should* not *be conflated with the parameter gradient $\partial\|\boldsymbol{\epsilon}^{\mathbf{M}} - \hat{\boldsymbol{\epsilon}}\|^2/\partial\theta = -2(\boldsymbol{\epsilon}^{\mathbf{M}} - \hat{\boldsymbol{\epsilon}}) \cdot \partial\hat{\boldsymbol{\epsilon}}/\partial\theta$, which under the $\boldsymbol{\epsilon}$-prediction parameterization used by DiMP is not SNR-amplified and in fact has approximately balanced magnitude across t [9]. The SNR structure thus governs the* task difficulty profile*, namely how demanding each timestep is in terms of required clean-signal accuracy, rather than the per-step weight of the parameter update. At small t, accurately denoising $\mathbf{M}_t \approx \mathbf{M}_0$ requires recovering fine-grained displacement detail, so the SNR-weighted functional loss forces the encoder to preserve fine motion structure. At large t, the loss tolerates any reconstruction consistent with the conditional mean, so $\mathbf{Z}_{\text{dec}}$ is primarily supervised at the coarse semantic level. This interpretation is consistent with the empirical U-shaped error profile in Figure 1(c): both regimes correspond to constraints that are well-matched to the target resolution, whereas intermediate t poses a maximally mixed objective (Section G.5).*

G.3 Small t: fine-grained local trajectory recovery

High-SNR regime. At small t, $\mathrm{SNR}(t) \gg 1$. The noisy input $\mathbf{M}_t$ is a slight perturbation of $\mathbf{M}_0$, so the task of predicting $\boldsymbol{\epsilon}^{\mathbf{M}}$ is equivalent to estimating a small residual correction to recover the clean signal. Formally, for any small perturbation magnitude δ,

$$\mathbf{M}_t = \mathbf{M}_0 + \delta\,\boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),\ \delta = \sqrt{1 - \bar{\alpha}_t} \approx 0, \tag{29}$$

and the optimal denoiser reduces to $\hat{\mathbf{M}}_0 = \mathbf{M}_t - \delta\,\hat{\boldsymbol{\xi}} \approx \mathbf{M}_0$. Because the required correction is small and localized, the network must resolve *point-wise* displacement detail rather than global coarse structure, driving the encoder f_e to capture fine-grained local trajectory information.

Spectral interpretation (heuristic). The following decomposition is intended as an intuitive picture and relies on two simplifying assumptions: (i) the signal $\mathbf{M}_0$ admits a fixed orthogonal decomposition $\mathbf{M}_0 = \sum_k \mathbf{u}_k$ (*e.g.*, principal components), and (ii) the injected Gaussian noise is isotropic in this basis, so each $\mathbf{u}_k$ is perturbed independently and with equal variance. Under these assumptions, the corruption $\sqrt{\bar{\alpha}_t}\,\mathbf{M}_0$ uniformly scales all components by a factor close to 1 at small t, while $\sqrt{1 - \bar{\alpha}_t}\,\boldsymbol{\epsilon}$ adds isotropic noise of small amplitude δ. The SNR-weighted functional loss (27) then tightly constrains the functional prediction on every spectral component, including the high-frequency fine-grained ones, because the rescaling factor $\mathrm{SNR}(t)$ is large. This provides a heuristic explanation for why the encoder is pushed to internalize local geometric trajectories that are only distinguishable at fine spatial resolution.

G.4 Large t: coarse global semantic encoding

Low-SNR regime. At large t, $\mathrm{SNR}(t) \ll 1$. The corrupted motion $\mathbf{M}_t \approx \sqrt{1 - \bar{\alpha}_t}\,\boldsymbol{\epsilon}$ is dominated by noise, and the signal component $\sqrt{\bar{\alpha}_t}\,\mathbf{M}_0$ contributes negligibly to $\mathbf{M}_t$. However, because the SNR-weighted loss still requires the network to explain $\mathbf{M}_0$ from this heavily noised input, the network must rely on the conditioning signal $\mathbf{Z}_{\text{dec}}$ to supply the missing information about the motion target.

Proposition 5 (Conditioning Reliance at Large t)**.** For $\mathrm{SNR}(t) \to 0$, the minimum-variance estimator of $\mathbf{M}_0$ given $\mathbf{M}_t$ and $\mathbf{Z}_{\mathrm{dec}}$ converges to the conditional mean under the model distribution:

$$\hat{\mathbf{M}}_0(t)\big|_{\mathrm{SNR}(t)\to 0} \;\to\; \mathbb{E}[\mathbf{M}_0 \mid \mathbf{Z}_{\mathrm{dec}}]. \tag{30}$$

Hence, at large t the loss effectively trains the network to predict the *marginal motion mode* conditioned on the encoder representation, capturing globally coherent motion semantics.

Proof. By Bayes' theorem, the posterior of $\mathbf{M}_0$ given $\mathbf{M}_t$ satisfies

$$q(\mathbf{M}_0 \mid \mathbf{M}_t, \mathbf{Z}_{\mathrm{dec}}) \propto \mathcal{N}\left(\mathbf{M}_0;\, \frac{\mathbf{M}_t}{\sqrt{\bar{\alpha}_t}},\, \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}\mathbf{I}\right) \cdot p(\mathbf{M}_0 \mid \mathbf{Z}_{\mathrm{dec}}). \tag{31}$$

As $\bar{\alpha}_t \to 0$, the Gaussian likelihood in (31) becomes infinitely broad (variance $\to \infty$), so it conveys no information about $\mathbf{M}_0$ and the posterior collapses to the prior $p(\mathbf{M}_0 \mid \mathbf{Z}_{\mathrm{dec}})$. The minimum mean-squared-error estimate is then the mean of this prior, yielding (30). □

Semantic coarse encoding. Proposition 5 implies that large-t supervision forces the encoder to store globally discriminative motion statistics such as the mean displacement direction, dominant limb motion, and action-class-level semantic content in $\mathbf{Z}_{\mathrm{dec}}$, because only these coarse features can be recovered from a heavily corrupted $\mathbf{M}_t$. This is precisely the signal that distinguishes action classes (*e.g.*, *jumping* vs. *waving*) from each other.

G.5 Intermediate t: transition and increased difficulty

At intermediate t, neither the high-SNR tight functional constraint (Section G.3) nor the conditioning-dominated coarse regime (Section G.4) dominates. Instead, $\mathbf{M}_t$ carries a substantial but noisy partial signal of $\mathbf{M}_0$, and the network must simultaneously resolve mid-frequency displacement components while suppressing noise of comparable magnitude to the signal. Adopting the same heuristic orthogonal-decomposition assumptions as in Section G.3, the SNR-weighted functional loss can be written as

$$\mathcal{L}(t) \;\propto\; \mathrm{SNR}(t) \sum_{k=1}^{K} \big\|\mathbf{u}_k - \hat{\mathbf{u}}_k(t)\big\|^2. \tag{32}$$

At intermediate t, $\mathrm{SNR}(t) = \mathcal{O}(1)$, so no spectral band is selectively amplified or suppressed. The error landscape is therefore maximally mixed: the network must balance trade-offs across all frequency components simultaneously, leading to suboptimal recovery at each individual scale, consistent with the elevated prediction error observed at intermediate t in Figure 1(c). The stratified timestep sampling in Eq. (7) (Section 3.2) is designed precisely to ensure that the model receives supervision from all three regimes within each update step, rather than spending disproportionate time in the easy regimes.

G.6 Implications for representation learning

The above analysis demonstrates that the full diffusion objective over $t \in [1, T]$ provides a *curriculum* of complementary supervision signals to the encoder:

- **Small t (local fine-grained learning):** The tight SNR-amplified functional constraint on $\hat{\mathbf{M}}_0$ (Proposition 4, Remark 3) compels the encoder to capture point-wise trajectory details and local motion patterns critical for fine-grained action recognition.
- **Large t (global semantic learning):** Conditioning-dominated optimization forces the encoder to summarize globally coherent motion modes, encoding action-class-level semantic content.
- **Hierarchical integration:** The stratified sampling strategy in Eq. (7) ensures that both objectives are simultaneously active within every training step, promoting representations that integrate local trajectory fidelity with global semantic coherence, a property unachievable by deterministic MSE regression, which reduces to a single (intermediate-SNR-equivalent) prediction without any multi-scale decomposition.

This theoretical picture is consistent with the empirical observation in Figure 1(c) and with the performance gains reported in Table 8 upon increasing the number of stratification intervals h.

H Gradient propagation through the decoder: a known limitation and partial mitigation

Background. In masked autoencoding pretraining, the decoder is structurally inferior to the encoder from the perspective of downstream transfer: it is applied exclusively during pretraining and discarded at fine-tuning time. Any loss applied at the decoder output therefore has a longer gradient path to the encoder than a loss applied directly at the encoder output. This creates a potential gradient attenuation effect: each parameterized module between the loss and the encoder can redirect gradient energy toward its own parameters, reducing the net signal that ultimately reaches the encoder. This limitation is well-recognized in the masked pretraining literature [2, 14], where decoder design choices are consistently observed to have smaller downstream impact than encoder design choices.

Implications for DiMP. In DiMP, $\mathcal{L}_{\text{mot}}$ is computed at the output of the motion diffusion head h_m, which is conditioned on the decoder output $\mathbf{Z}_{\text{dec}}$. The gradient must traverse the motion head h_m and the full decoder f_d before reaching the encoder f_e. Each of these modules may absorb a portion of the gradient signal, weakening the motion-distributional supervision that ultimately updates the encoder. In the extreme case where the decoder is highly expressive, the decoder can satisfy $\mathcal{L}_{\text{mot}}$ by adapting its own weights, leaving the encoder gradient negligibly small.

Partial mitigation in DiMP. DiMP addresses this partially through two design choices. First, stratified timestep sampling applies $\mathcal{L}_{\text{mot}}$ at h noise levels simultaneously within every training step, multiplying the number of independent gradient signals reaching the encoder per step by a factor of h and diversifying them across the full diffusion hierarchy. Second, the loss weight λ_{mot} is tuned to amplify the relative contribution of $\mathcal{L}_{\text{mot}}$ to the total gradient, preventing it from being dominated by the geometric reconstruction loss $\mathcal{L}_{\text{rec}}$, which has a shorter gradient path. Ablation results in Tables 8 and 10 confirm that both h and λ_{mot} meaningfully affect downstream performance, consistent with the view that these choices modulate how effectively motion supervision propagates to the encoder.

Open problem. Despite these mitigations, efficiently routing distributional motion supervision from a decoder-conditioned head to the encoder remains only partially solved. More principled alternatives represent promising directions, including conditioning the motion diffusion head directly on encoder features, introducing auxiliary encoder-level supervision objectives, or designing architectures with shorter gradient paths specifically for motion modeling. We leave a thorough investigation of this bottleneck, and the design of mechanisms that more directly exploit the distributional capacity of diffusion models at the encoder level, to future research.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the four technical contributions of DiMP, namely the VisMask encoder, the patch diffusion decoder, the masked-only center diffusion strategy, and the stratified motion diffusion head. All claims are supported by ablation studies and comparisons with prior methods in the experiments section.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion section acknowledges that gradient propagation from the decoder-conditioned diffusion head to the encoder remains a partial limitation of the current design. Appendix H provides a detailed analysis of this bottleneck and outlines future research directions for more direct encoder-level motion supervision.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Appendices E, F, G, and H present complete formal proofs for all theoretical results with explicitly stated assumptions. Each proposition and corollary is numbered and cross-referenced in the main text, and the assumptions are stated immediately before each result.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Appendix B provides all architecture specifications, optimizer settings, learning rate schedules, masking ratios, and data preprocessing procedures needed to reproduce the reported results. The ablation studies in Appendix C further validate the contribution of each design choice.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We include anonymized training and evaluation code in the supplemental material together with setup notes and scripts to reproduce the main results. All benchmark datasets are already public and are accessed as described in our appendix and citations.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details including the AdamW optimizer, learning rate schedule, batch size, number of epochs, masking ratio, and hardware configuration are fully specified in Appendix B. Downstream fine-tuning protocols for each benchmark are also described therein.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All main results are reported with standard deviation across three independent runs with different random seeds. The multi-seed consistency experiment in Appendix D further validates the stability of the reported numbers across a wider range of initializations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The GPU type, number of devices, and approximate pretraining and fine-tuning time are reported in Appendix B. Both pretraining and downstream fine-tuning resource requirements are described, allowing readers to assess the computational cost of the full pipeline.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper presents foundational research on self-supervised pretraining for dynamic 3D point cloud understanding. No sensitive personal data is collected or released, and the work does not involve any application with direct potential for harm.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work focuses on self-supervised pretraining for dynamic point cloud understanding and does not have a direct path to negative societal applications. The research is foundational in nature and is not tied to any specific deployment scenario that would require an impact discussion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The proposed model processes 3D point cloud sequences for action recognition and segmentation tasks. It does not involve generation of sensitive content or collection of personal data, and therefore poses no high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and baseline methods used in this work are properly cited via their original publications. The benchmark datasets including MSRAction-3D and HOI4D are publicly available research datasets released by their respective authors.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not introduce new datasets or data assets. The pretrained model weights will be shared upon acceptance, with training and usage details fully described in Appendix B.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing or any form of human subject study. All experiments are conducted on existing publicly released benchmark datasets collected by prior works.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects research is conducted in this work. All data used are from publicly released benchmarks that were collected and approved by their original authors.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Large language models do not form any part of the core methodology proposed in this paper. The method relies entirely on point cloud transformers and diffusion-based motion modeling trained on 3D point cloud data.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.