A NON-PARAMETRIC FRAMEWORK FOR DYNAMIC POINT CLOUD CLASSIFICATION USING OPTIMAL TRANSPORT-BASED EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamic point clouds, time-varying collections of unordered 3D points, exhibit complex variability arising from two distinct sources: spatial deformation of the permutation-invariant point configuration at each instant, and temporal variation of trajectories across time. Existing approaches to dynamic point cloud recognition predominantly rely on deep learning architectures that capture these variations through large numbers of learnable parameters and GPU-dependent optimization, posing practical challenges in resource-constrained settings. In this work, we propose a completely non-parametric framework for dynamic point cloud classification grounded in optimal transport theory. Linear Optimal Transport (LOT) resolves spatial permutation ambiguity by mapping each unordered point cloud to a canonical Euclidean representation, while the Cumulative Distribution Transform (CDT) encodes per-joint temporal signals into a representation where temporal variability becomes more tractable. Together, these embeddings capture the full dynamic structure in a space suitable to linear analysis, where each class is modeled as a low-dimensional subspace and classification is performed by nearest subspace assignment, requiring no gradient-based optimization, no learnable parameters, and no GPU at either training or inference time. Evaluated on KIMORE, UI-PRMD, MSR-Action3D, and NTU RGB+D against eight competitive parametric baselines, the proposed framework achieves competitive classification performance while requiring no learnable parameters or GPU-based optimization. It also maintains substantially lower floating point operations (FLOPs) and very low CPU inference latency, making it a strong candidate for resource-constrained deployment. Code is available at `https://anonymous.4open.science/r/drab_white_fly-7685/`

1 INTRODUCTION

Dynamic motion classification is important for rehabilitation monitoring, skeleton-based action recognition, human-computer interaction, and robotic perception. Although lightweight dynamic point-cloud and skeleton-based models have been developed for efficient real-time recognition, they largely remain deep learning-based, requiring trainable parameters, backpropagation, and hardware-supported optimization Cheng et al. (2020); Li et al. (2022). Dynamic 3D point clouds differ from static point clouds because each sample is a sequence of spatial configurations, not a single shape. In skeleton-based motion data, each frame is a 3D joint set, and the full sequence captures its temporal evolution Vakanski et al. (2018); Shahroudy et al. (2016). This creates two coupled sources of variability: spatial point-set variability within frames, caused by pose, subject morphology, and point ordering, and temporal trajectory variability across frames, caused by differences in speed, duration, and trajectory shape. Effective classification must therefore compare motions under both spatial and temporal variation.

Existing works relates to this problem through two complementary directions. Recent dynamic motion recognition research has increasingly focused on making deep models smaller, faster, and more suitable for deployment, using graph-based skeleton networks, point-cloud video models, and lightweight temporal architectures Yan et al. (2018); Fan et al. (2021); Li et al. (2022); Huang et al. (2024). Although these methods improve efficiency, they remain deep-learning based: their

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

representations depend on learned neural weights, backpropagation-based training, and model-specific optimization, often making training dependent on GPU-supported computation. In parallel, recent non-parametric point-cloud methods use fixed geometric operations for efficient 3D analysis Zhang et al. (2023); Mohammadi & Salarpour (2025); Mohammadi et al. (2025), but they are primarily designed for static object-level point clouds rather than time-varying motion sequences. Thus, a CPU-oriented, backpropagation-free, non-parametric framework for dynamic 3D motion recognition remains underexplored.

We address this gap by introducing a different representation paradigm for dynamic 3D motion classification. Rather than learning spatiotemporal features with neural parameters, the framework explicitly separates spatial and temporal variability. Linear Optimal Transport (LOT) maps unordered point configurations into a consistent Euclidean representation, while the Cumulative Distribution Transform (CDT) encodes per-joint coordinate trajectories into a representation where temporal variability becomes more tractable. SVD-based nearest subspace assignment then performs classification without neural parameter learning, backpropagation, or GPU-based optimization. Across KIMORE, UI-PRMD, MSR-Action3D, and NTU RGB+D Capecci et al. (2019); Vakanski et al. (2018); Li et al. (2010); Shahroudy et al. (2016), the framework achieves competitive, and in several settings superior, performance against state-of-the-art parametric dynamic motion classification models, while requiring only CPU-based inference with very low per-sample latency.

We highlight our main contributions as follows:

- To the best of our knowledge, we introduce the first fully non-parametric frameworks for dynamic 3D motion recognition from time-varying 3D point-cloud sequences.
- The proposed framework is entirely backpropagation-free, involves no parameter learning, and operates on CPU for both training and inference, eliminating the need for GPU-based optimization.
- The proposed framework achieves competitive classification performance against state-of-the-art parametric dynamic 3D motion recognition models while maintaining very low CPU inference latency, supporting edge and resource-constrained deployment.

2 Literature Review

Dynamic point-cloud and skeleton-based motion analysis have largely been studied through parametric neural architectures that learn spatiotemporal representations from sequential 3D data. Early recurrent approaches such as PointRNN Fan & Yang (2019) and PointLSTM Min et al. (2020) extend recurrent modeling to unordered point-cloud sequences by updating point states over time and aggregating information from local spatial neighborhoods across frames. Transformer- and convolution-based methods later introduced explicit spatiotemporal operators for point-cloud sequences. P4Transformer Fan et al. (2021) combines point 4D convolution with Transformer self-attention to model point-cloud videos without explicit tracking, while PSTNet Fan et al. (2022) disentangles spatial and temporal convolution to capture local geometry and temporal evolution.

Recent work has focused more directly on lightweight and real-time dynamic perception. SequentialPointNet Li et al. (2022) simplifies point-cloud sequence modeling by converting point-cloud sequences into Hyperpoint sequences and using frame-level parallelism for real-time 3D human action recognition. MotionPointNet Huang et al. (2024) combines a lightweight PointNet-style encoder with PDE-inspired temporal modeling to capture spatiotemporal correlations in point-cloud videos. EventMamba Ren et al. (2025) treats event streams as point-based spatiotemporal data and uses temporal aggregation with Mamba-based state-space modeling for efficient long-range temporal extraction. PointNet4D Liu et al. (2026) introduces a lightweight 4D point-cloud video backbone with Hybrid Mamba-Transformer temporal fusion for online and offline dynamic perception. These methods are explicitly designed for dynamic or sequential data and improve efficiency through neural architectural design. However, they remain parametric models that require learned weights, backpropagation-based optimization, and model-specific training.

In terms of non-parametric point-cloud methods, including Point-NN Zhang et al. (2023), Point-GN Mohammadi & Salarpour (2025), and Point-LN Mohammadi et al. (2025) have been explored for 3D point-cloud classification. However, their evaluation mainly focuses on static object-level datasets such as ModelNet40 Wu et al. (2015) and ScanObjectNN Uy et al. (2019), rather than dynamic,

time-varying point-cloud data. Together, these observations reveal a key methodological gap: a backpropagation-free, non-parametric formulation for dynamic 3D motion classification that jointly handles spatial point-set variability and temporal motion variability remains largely unexplored. We address this gap through a spatial–temporal optimal-transport based formulation.

3 METHOD

3.1 OPTIMAL TRANSPORT

Optimal transport (OT) compares probability distributions by quantifying the minimal effort (mass times distance) required to transform one distribution into another, via a cost function that determines the minimum-cost transformation Kolouri et al. (2017).

This makes OT well suited to point set analysis, where sets are treated as probability distributions and compared using the Wasserstein metric. In geometric settings such as skeleton-based human pose representations, each pose is a collection of N joint positions in $\mathbb{R}^3$, observed as an *unordered* point set with no canonical labeling or ordering. We therefore randomly permute the joint order before processing, explicitly removing dependence on predefined joint identities or ordering. Importantly, our goal is not to recover anatomical identity through OT, but to evaluate whether useful dynamic representations can be obtained when anatomical labels and topology are unavailable (Details in Appendix D). A pose instance is consequently represented as an empirical probability distribution with uniform mass over its points. OT compares such point sets by quantifying how one configuration must be "rearranged" to match another, capturing geometric differences in shape and structure without requiring point-wise correspondences or predefined ordering.

Formally, a point set of N points in $\mathbb{R}^3$ can be modeled as a discrete probability measure:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\delta_{x_i}, \quad x_i \in \mathbb{R}^3,$$

where δ_{x_i} is a Dirac measure at x_i. For a measurable map $T : \mathbb{R}^3 \to \mathbb{R}^3$ and a probability measure σ on $\mathbb{R}^3$, the *push-forward* of σ under T, denoted $T_\sharp\sigma$, is defined by $(T_\sharp\sigma)(B) := \sigma\big(T^{-1}(B)\big)$ for every measurable set $B \subseteq \mathbb{R}^3$.

Within the framework of Optimal Transport (OT), the cost of transporting one distribution to another is characterized through two complementary formulations Kolouri et al. (2017). The **Monge problem** asks for the most efficient pointwise map T rearranging σ into μ, minimising total squared transport cost:

$$W_2^2(\mu,\sigma) = \min_{T:\, T_\sharp\sigma = \mu} \int_{\mathbb{R}^3} \|x - T(x)\|^2 \, d\sigma(x).$$

When no such deterministic map exists, the **Kantorovich relaxation** instead searches over *transport plans* γ – probabilistic couplings that allows for mass splitting fractionally – leading to:

$$W_2^2(\mu,\sigma) = \min_{\gamma \,\in\, \Pi(\mu,\sigma)} \sum_{i,j} \|x_i - y_j\|^2 \, \gamma_{ij},$$

where $\Pi(\mu,\sigma)$ denotes all joint distributions over $\mathbb{R}^3 \times \mathbb{R}^3$ whose marginals are μ and σ, respectively.

3.2 LINEAR OPTIMAL TRANSPORT

In this work, we employ a linearized version of this metric, constructed through a tangent space approximation of the underlying manifold, as outlined in Wang et al. (2013). Linear optimal transport (LOT) of a distribution is defined relative to a pre-established reference distribution σ. In this study, we define σ as the Lebesgue measure over a compact, convex domain in $\mathbb{R}^3$. This operates within the space $\mathcal{P}_2(\mathbb{R}^3)$, which comprises absolutely continuous probability measures characterized by finite second moments and bounded densities. By Brenier's theorem Brenier (1991), any $\mu \in \mathcal{P}_2(\mathbb{R}^3)$ can be uniquely identified with a map T_σ^μ, which can be obtained by solving the Monge optimization problem:

$$\min_{T_\sharp\sigma=\mu} \int_{\mathbb{R}^3} \|x - T(x)\|^2 \, d\sigma(x).$$

The LOT embedding of μ is then given by the following correspondence:

$$\widehat{\mu} := T_\sigma^\mu,$$

with associated norm $\|T\|_\sigma$. The LOT distance between two distributions $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^3)$ is

$$d_{\mathrm{LOT}}(\mu, \nu) := \|T_\sigma^\mu - T_\sigma^\nu\|_\sigma.$$

Remark. Brenier's theorem requires $\sigma, \mu \in \mathcal{P}_2(\mathbb{R}^3)$ to be absolutely continuous; skeleton data are finite point sets, so in practice σ, μ are replaced by empirical measures and T by a discrete assignment (Sec. 3.2.1), the continuous case above is the theoretical motivation, not a description of the implementation. Following Wang et al. (2013), we call d_{LOT} the ***LOT distance***, but it is generally a tractable proxy for $W_2(\mu, \nu)$ obtained by embedding measures into the tangent space at σ, not an identity with it; equality holds only in special cases (e.g. $\mu = \sigma$ or $\nu = \sigma$).

3.2.1 Implementation for Discrete Point Sets

When working with discrete point set data, both the reference distribution σ and the target distribution μ are treated as discrete probability measures, each supported on a finite collection of points in $\mathbb{R}^3$. A point set Ω_s consisting of N elements can be described through an injective mapping $s : \{1, \ldots, N\} \to \mathbb{R}^3$, where each index corresponds to a unique point in the set. Given the point set Ω_s with N elements we define a discrete probability distribution associated with the point set as

$$P_s := \frac{1}{N} \sum_{i=1}^{N} \delta_{s(i)}.$$

Let $\mathcal{F}_{N,3}$ denote the collection of injective maps from $\{1, \ldots, N\}$ to $\mathbb{R}^3$. Let $s, r \in \mathcal{F}_{N,3}$, where r is a fixed reference injective map corresponding to a reference point set distribution P_r. The squared Wasserstein-2 distance between P_s and P_r can be computed via the Kantorovich relaxation by solving the linear program:

$$W_2^2(P_s, P_r) = \min_{\pi \in \mathbb{R}^{N \times N}} \sum_{i=1}^{N} \sum_{j=1}^{N} \pi_{ij} \, \|s(i) - r(j)\|^2,$$

where $\pi_{ij} \geq 0$ and $\sum_{j=1}^{N} \pi_{ij} = \sum_{i=1}^{N} \pi_{ij} = 1/N, \ \forall \, i, j \in \{1, \ldots, N\}$.

For discrete sets with equal cardinality N and uniform weight $1/N$, the linear program's feasible region is the Birkhoff polytope, so an optimal solution π^* can be taken to be a permutation matrix Villani (2021), recovered via the Hungarian algorithm; cost-matrix ties can admit multiple optimal permutations, so σ_s^* denotes *an* optimum (the deterministic solver's output), not necessarily a unique one.The discrete LOT embedding of the distribution P_s can then be defined as

$$\widehat{P}_s := s \circ \sigma_s^* = \left[s(\sigma_s^*(1)), \ldots, s(\sigma_s^*(N)) \right]^\top \in \mathbb{R}^{N \times 3},$$

and we define the *reference-aligned distance* $d_{LOT}(P_s, P_q) := \|\hat{P}_s - \hat{P}_q\|$. Since $\hat{P}_s, \hat{P}_q$ are each aligned to the common reference r rather than to each other, d_{LOT} is generally not the pairwise $W_2(P_s, P_q)$, but a reference-induced surrogate that embeds all samples into one coordinate system, enabling the linear analysis of Section 3.4.

An optimal assignment σ_s^* imposes an ordering over the N points relative to the fixed reference r: point i in $\hat{P}_s$ corresponds to point i in any other embedded configuration $\hat{P}_q$ computed against the same r, resolving the label ambiguity of unordered point sets.

We therefore index points by $i \in \{1, \ldots, N\}$ following this LOT-induced ordering, and denote the d-th spatial coordinate ($d \in \{x, y, z\}$) of the i-th point in the embedded configuration as $\widehat{P}_s^{(i,d)}$. This linearization enables geometric analysis of structured point configurations using standard tools such as visualization, classification, and regression, while naturally accounting for spatial deformations and geometric invariances. Applied per-frame to a dynamic sequence (Sec 3.4), this alignment's temporal consistency is quantified in Appendix C.

3.3 CUMULATIVE DISTRIBUTION TRANSFORM

The cumulative distribution transform (CDT) Park et al. (2018) is a nonlinear invertible transform mapping smooth probability density functions to the space of one-dimensional monotonic diffeomorphisms, defined relative to a reference density and capturing the transformation needed to morph one density into another. This encoding accounts for nonlinearities that would otherwise limit class separability.

Let $s(t),\ t \in \Omega_s$ be a one-dimensional probability density function, and let $r(t),\ t \in \Omega_r$ denote a fixed reference probability density function satisfying $s(t) > 0$, $r(t) > 0$, and $\int_{\Omega_s} s(t)\,dt = \int_{\Omega_r} r(t)\,dt = 1$. The cumulative distribution transform of $s(t)$ with respect to $r(t)$ is defined as the mapping $\widehat{s}(t)$ such that

$$\int_{-\infty}^{\widehat{s}(t)} s(u)\,du = \int_{-\infty}^{t} r(u)\,du.$$

Geometrically, the CDT corresponds to the optimal transport map from the reference density r to s in one dimension. The transform preserves all information in the original density, with losses only due to numerical approximation. The CDT is also invertible, with the inverse given by

$$s(t) = \big(\widehat{s}^{-1}\big)'(t) \cdot r\big(\widehat{s}^{-1}(t)\big),\ \widehat{s}^{-1}(\widehat{s}(t)) = t.$$

The CDT satisfies key properties useful for analyzing 1D densities Rubaiyat et al. (2024). A diffeomorphism h acting on a density $s(t)$ as $s_h = h' \cdot (s \circ h)$ transforms its CDT simply as $\widehat{s}_h = h^{-1} \circ \widehat{s}$, so deformations in the density domain become compositions in CDT space. The CDT also induces an isometric embedding into a weighted Euclidean space, with $W_2^2(s_1, s_2) = \|(\widehat{s}_1 - \widehat{s}_2)\sqrt{r}\|^2_{L^2(\Omega_r)}$ for any densities s_1, s_2. Together, these properties simplify otherwise nonlinear deformations in CDT space, facilitating standard tools for analysis, comparison, and classification.

3.4 GENERATIVE MODEL: TEMPLATE UNDER SPATIAL AND TEMPORAL DEFORMATION

Once the LOT embedding establishes a consistent point ordering, each point $i \in \{1, \ldots, N\}$ and coordinate $d \in \{x, y, z\}$ yields a time-varying function $\widehat{P}^{(i,d)}(t), t \in [1, T]$, which we normalize into a 1D probability density $s^{(i,d)}(t)$ and transform individually via the CDT, using a shared reference density $r(t)$ across all points and coordinates. To formalize structured variability in time-varying point clouds, we model each class k as arising from a canonical template representing a prototypical motion sequence, subject to both spatial and temporal deformations.

3.4.1 SPATIAL DEFORMATION IN LOT SPACE

At the level of spatial structure, each class k is characterized by a canonical template point-set distribution $P_\varphi^{(k)}$ on $\mathbb{R}^3$.

Individual pose instances can be modeled as realizations of this template under the action of spatial maps g_j drawn from a deformation group $\mathcal{G}$ restricted to the classes for which LOT convexity is established (Shifat-E-Rabbi et al., 2025) (translations, isotropic scalings, or their near-affine perturbations):

$$\mathcal{S}^{(k)} = \Big\{ P_{s,j}^{(k)} \;\big|\; P_{s,j}^{(k)} = g_{j\#}\, P_\varphi^{(k)},\ \ g_j \in \mathcal{G} \Big\}.$$

where $\mathcal{G}$ is restricted to the deformation classes for which LOT convexity is established (Shifat-E-Rabbi et al., 2025) (translations, isotropic scalings, or their near-affine perturbations). After applying LOT with respect to the shared reference distribution $P_\varphi^{(k)}$, the embeddings of the elements of $\mathcal{S}^{(k)}$ in the LOT space are obtained as $\widehat{P}_{s,j}^{(k)} = g_j \circ \widehat{P}_\varphi^{(k)},\ g_j \in \mathcal{G}$ Shifat-E-Rabbi et al. (2025). The LOT embedding maps the set of spatially deformed versions of a template to a convex set in the embedding space, making the class structure more simpler for linear analysis.

3.4.2 TEMPORAL DEFORMATION IN CDT SPACE

At the level of temporal dynamics, each per-point coordinate signal $s^{(i,d;k)}$ from class k, for point $i \in \{1, \ldots, N\}$ and coordinate $d \in \{x, y, z\}$, can be modeled as a temporally deformed version of a template signal $\rho^{(i,d;k)}$. The generative model for class k under a temporal diffeomorphism $h_j \in \mathcal{H}$ is given by:

$$\mathcal{T}^{(k)} = \left\{ s_j^{(i,d;k)}(t) = h_j'(t) \cdot \rho^{(i,d;k)}(h_j(t)),\ h_j \in \mathcal{H} \right\}.$$

where $\mathcal{H}^{-1}$ is convex under composition, as required for CDT convexity (Rubaiyat et al., 2024). After applying CDT with respect to the shared reference signal $\rho^{(i,d;k)}$, the embeddings of the elements of $\mathcal{T}^{(k)}$ in the CDT space are obtained as $\widehat{s}_j^{(i,d;k)} = h_j^{-1} \circ \widehat{\rho}^{(i,d;k)},\ h_j \in \mathcal{H}$. Similarly to LOT, it is established that the CDT maps temporally deformed versions of a template signal into a convex set in the CDT embedding space, supporting subspace-based modeling of temporal variability Rubaiyat et al. (2024).

3.4.3 JOINT GENERATIVE MODEL

Combining both levels, a time-varying point cloud for subject j of class k is fully characterized by the pair (g_j, h_j), where $g_j \in \mathcal{G}$ governs spatial deformation of the joint configuration and $h_j \in \mathcal{H}$ governs temporal deformation of the motion signals. Specifically, the spatial structure of the point cloud is captured by the LOT embedding $\widehat{P}_{s,j}^{(k)} = g_j \circ \widehat{P}_{\varphi}^{(k)}$, while the temporal dynamics of each per-point coordinate signal are captured by the CDT embedding $\widehat{s}_j^{(i,d;k)} = h_j^{-1} \circ \widehat{\rho}^{(i,d;k)}$. The full embedding space generative model is:

$$\widehat{\mathcal{M}}^{(k)} = \left\{ \left(\widehat{P}_{s,j}^{(k)},\ \widehat{s}_j^{(i,d;k)} \right) \;\middle|\; g_j \in \mathcal{G},\ h_j \in \mathcal{H} \right\}.$$

We establish the convexity of the joint LOT–CDT representation $\widehat{\mathcal{M}}^{(k)}$; the full proof is provided in Appendix A.

4 EXPERIMENT

4.1 DATASET DETAILS

We evaluate our framework on four standard 3D skeleton-based dynamic motion classification benchmark datasets: KIMORE Capecci et al. (2019), UI-PRMD Vakanski et al. (2018), NTU RGB+D Shahroudy et al. (2016), and MSR-Action3D Li et al. (2010), spanning rehabilitation exercises and general action recognition tasks at varying scales. More details are provided in Appendix B.

4.2 BASELINES

We evaluate our proposed method against a comprehensive set of well-established and recent lightweight, computationally efficient, and edge device–friendly models designed for 3D dynamic data processing. Specifically, we include eight representative baselines: PointRNN Fan & Yang (2019), PointLSTM Min et al. (2020), PSTNet Fan et al. (2022), P4Transformer Fan et al. (2021), SequentialPointNet Li et al. (2022), MotionPointNet Huang et al. (2024), EventMamba Ren et al. (2025), and PointNet4D Liu et al. (2026).

4.3 IMPLEMENTATION DETAILS

We evaluate two settings: **(i) data-scarcity**, using 10% training and 90% held-out test data, and **(ii) conventional**, using 90% training and 10% held-out test data. All splits are independent, with no samples appearing in both training and test sets. This setup allows us to evaluate the proposed method under both data-limited and conventional training conditions.Our method runs entirely on CPU, with no GPU support required. Sequences are resampled to a fixed $T = 100$ frames, LOT-aligned via the Hungarian algorithm against a single global reference frame, and transformed per-channel via the

CDT. Class subspaces are formed via thin SVD, retaining the top singular vectors capturing 99% of cumulative spectral energy per class (the threshold varied in Table 6). Full preprocessing, alignment, calculation details, and hyperparameter details are given in Appendix E.

4.4 RESULTS

4.4.1 MAIN RESULTS

The most striking advantages lie in computational efficiency alongside competitive accuracy. Our model reduces FLOPs by a minimum of approximately 2.3K$\times$ (vs. PointRNN on NTU RGB+D) and up to approximately 2.1M$\times$ (vs. PointRNN on MSR-Action3D), with similar-magnitude reductions across all datasets and training regimes, keeping our footprint in the single-digit to low-hundreds of GFLOPs range against tens to tens-of-thousands of TFLOPs for competing methods (Tables 1–4). Inference latency improves by approximately 1.92$\times$ to over 202$\times$ on a single-core CPU across all four datasets and both split configurations, the sole exception being NTU RGB+D (90:10), where our 7.30 ms marginally exceeds P4Transformer's 5.25 ms while still substantially beating all other baselines. EventMamba and PointNet4D are excluded from direct efficiency comparison, as both require multi-core GPU execution, making a head-to-head comparison with our CPU-resident pipeline fundamentally unfair; our latency is nonetheless competitive with or superior to EventMamba in several settings. Complete results are in Appendix F.

4.4.2 LINEAR SEPARABILITY

Our formulation establishes intra-class convexity of the joint LOT-CDT embedding but does not guarantee inter-class linear separability, which depends on class-template distinctness. We therefore empirically evaluated linear separability by applying the The Fisher Discriminant Ratio (FDR) and Linear Discriminant Analysis (LDA) classifier to raw point-cloud coordinates and the LOT-CDT embedding. Both FDR and LDA increase with the LOT-CDT embedding, providing empirical evidence that the LOT-CDT embedding is more linearly separable than the raw representation. LDA accuracy differs from our full pipeline because LDA learn global linear boundaries, whereas our nearest-subspace classifier models each class with its own low-dimensional affine subspace. Thus, LDA is used only to assess linear separability of the representation, not to reproduce the classification performance of our full pipeline.

Table 5: Linear separability checking using Fisher Discriminant Ratio (FDR) and Linear Discriminant Analysis (LDA). Higher values indicate better linear separability.

Data Representation	**KIMORE**		**UI-PRMD**		**NTU RGB+D**		**MSR-Action3D**	
	FDR $\uparrow$	**LDA** $\uparrow$	**FDR** $\uparrow$	**LDA** $\uparrow$	**FDR** $\uparrow$	**LDA** $\uparrow$	**FDR** $\uparrow$	**LDA** $\uparrow$
raw point-cloud	0.0036	25.00	0.011	10.0	0.025	15.87	0.094	19.23
LOT-CDT embedding	**0.0045**	**44.11**	**0.891**	**96.0**	**0.137**	**42.51**	**0.122**	**80.76**

4.4.3 TIME BREAKDOWN OF THE END-TO-END INFERENCE PIPELINE

As shown in Figure 1, the total end to end inference time is broken down into OT alignment (Hungarian matching), CDT transform, and subspace projection (classification) across the four evaluated datasets. While OT alignment constitutes the largest fraction for KIMORE, UI-PRMD and MSR-Action3D, the projection step dominates in the NTU-RGB+D setting, a shift driven by the larger number of action classes, which significantly increases the cost of evaluating distances to all class subspaces.

4.4.4 ABLATION STUDY

We conduct an ablation study on the SVD energy threshold to examine the sensitivity of our representation to the retained subspace dimension. As shown in Table 6, the best performance is consistently obtained at the 99% threshold across all four datasets, indicating that preserving more singular value energy is important for capturing the fine-grained motion structure in 3D dynamic data. Reducing the threshold to 90% leads to a noticeable performance drop on KIMORE, NTU RGB+D, and

Table 1: Results on the KIMORE dataset using a 10:90 train–test split, where Acc., Prec., Rec., F1, and Params. denote accuracy, precision, recall, F1-score, and learnable parameters, respectively.

Ratio	Model	Acc	Prec	Rec	F1	FLOPs	Params	Inference Latency
	PointRNN	40.02	16.15	33.37	21.76	155.03 T	1.35M	172.63 ms
	PointLSTM	68.90	70.02	65.77	66.58	53.48 T	87.301K	46.90 ms
	PSTNet	82.49	82.68	82.14	82.36	121.09 T	8.408M	26.86 ms
	P4Transformer	64.24	63.21	63.87	63.48	5.73 T	0.11M	4.59 ms
10:90	EventMamba	88.46	88.92	87.80	88.26	20.41 T	1.59 M	2.19 ms (GPU)
	SequentialPointNet	89.28	90.85	88.24	89.08	32.19 T	2.986M	14.91 ms
	MotionPointNet	84.53	86.43	82.69	83.89	23.30 T	0.08M	59.30 ms
	PointNet4D	89.61	89.28	89.40	89.29	4.75 T	1.238M	0.04 ms (GPU)
	Ours	86.33	88.98	84.50	85.94	0.33 G	0	0.8527 ms

Table 2: Results on the **UI-PRMD** using a 10:90 train-test split ratio.

Ratio	Model	Acc	Prec	Rec	F1	FLOPs	Params	Inference Latency
	PointRNN	10.00	1.00	10.00	1.82	56.67 T	1.35M	50.96 ms
	PointLSTM	66.89	68.22	66.89	65.58	25.41 T	87.946K	30.59 ms
	PSTNet	50.44	51.47	50.44	49.92	51.29 T	8.419M	16.34 ms
	P4Transformer	41.56	42.34	41.56	41.70	0.97 T	0.11M	2.00 ms
10:90	EventMamba	69.78	71.93	69.78	69.41	30.53 T	1.59 M	2.15 ms (GPU)
	SequentialPointNet	19.22	19.85	19.22	19.03	17.39 T	2.987M	10.99 ms
	MotionPointNet	58.78	59.18	58.78	58.68	6.54 T	0.08M	49.58 ms
	PointNet4D	10.89	11.40	10.89	10.99	2.56 T	1.237M	0.03 ms (GPU)
	Ours	76.22	77.83	76.22	76.03	0.12 G	0	0.5053 ms

Table 3: Results on the **NTU RGB+D** using a 10:90 train-test split ratio.

Ratio	Model	Acc	Prec	Rec	F1	FLOPs	Params	Inference Latency
	PointRNN	18.4	17.9	18.1	17.8	365.5 T	1.35M	256.5 ms
	PointLSTM	45.53	45.89	45.52	45.55	2353.05 T	94.396K	52.42 ms
	PSTNet	28.77	29.72	28.76	26.96	144.79 T	8.521M	28.21 ms
	P4Transformer	35.62	35.77	35.62	35.63	252.07 T	0.12M	5.32 ms
10:90	EventMamba	62.96	63.48	62.97	63.06	1796.07 T	1.61 M	2.17 ms (GPU)
	SequentialPointNet	47.10	46.73	47.10	46.71	1416.40 T	3.000M	17.30 ms
	MotionPointNet	46.7	45.9	45.1	45.4	518.64 T	0.10M	0.00
	PointNet4D	13.34	13.08	13.35	13.13	208.81 T	1.252M	0.03 ms (GPU)
	Ours	39.17	42.54	39.17	38.71	16.32 G	0	2.7714 ms

Table 4: Results on the **MSR-Action3D** using a 10:90 train-test split ratio.

Ratio	Model	Acc	Prec	Rec	F1	FLOPs	Params	Inference Latency
	PointRNN	5.91	0.61	5.56	0.96	91.56 T	1.35M	64.28 ms
	PointLSTM	9.06	2.80	9.26	3.67	9.52 T	89.236K	45.87 ms
	PSTNet	46.26	51.60	46.66	45.93	46.93 T	8.439M	22.46 ms
	P4Transformer	7.68	5.10	7.42	4.62	2.15 T	0.11M	3.39 ms
10:90	EventMamba	53.74	58.21	53.56	53.90	20.38 T	1.60 M	2.18 ms (GPU)
	SequentialPointNet	17.91	17.89	17.40	16.55	16.10 T	2.99M	21.24 ms
	MotionPointNet	63.78	64.56	63.68	63.31	8.42 T	0.09M	50.58 ms
	PointNet4D	8.86	9.99	8.62	8.75	7.10 T	1.238M	0.03 ms(GPU)
	Ours	57.68	59.83	57.18	56.98	0.07 G	0	1.0632 ms

MSR-Action3D, while the 80% setting further degrades accuracy, especially on KIMORE and NTU RGB+D. These results suggest that overly aggressive truncation removes discriminative motion information, whereas a higher-energy subspace offers a better performance.

5 Discussion

Comparing performance across the two training settings, the proposed model remains competitive with, and in some cases outperforms, the baselines under both data-scarce (10%) and full (90%) training, using the same training subsets for every method. Performance holds up particularly well under data scarcity: at 10% training data, the model outperforms the full comparison field on

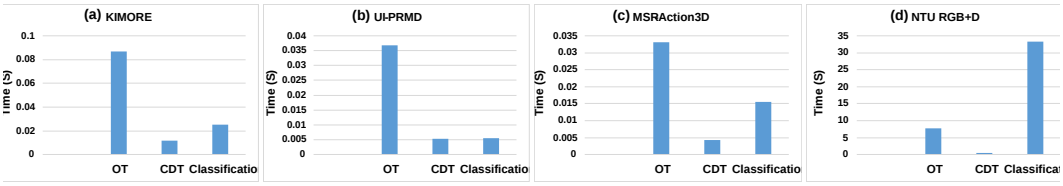

Figure 1: Breakdown of each component involved in the end-to-end implementation of our proposed pipeline.

Table 6: Accuracy comparison using a 90:10 train-test split under different SVD energy thresholds across datasets.

SVD Energy Threshold (%)	KIMORE	UI-PRMD	NTU RGB+D	MSR-Action3D
99	90.44	86.00	45.25	92.31
90	82.35	70.00	24.86	76.92
80	32.35	70.00	19.23	69.23

UI-PRMD and remains competitive on KIMORE and MSR-Action3D. Under full training (90%), performance remains competitive on three of four datasets without consistently leading the field, a gap we attribute to the inherently limited representational capacity of a parameter-free design relative to deeply supervised models trained on abundant labelled data; improving the discriminative capacity of non-parametric models under large-scale supervised settings remains an open direction for future work. More consequentially, even where accuracy trails the best baseline by a few points, the accompanying efficiency profile, FLOPs reductions of up to 2.1M$\times$, 1.92$\times$ to over 202$\times$ inference-latency improvement, and no learned-parameter overhead, reframes the comparison entirely: for edge devices, embedded rehabilitation systems, and clinical wearables where GPU access is unavailable and real-time response is non-negotiable, a modest accuracy trade-off for orders-of-magnitude lower compute is a rational engineering decision, not a compromise. The NTU RGB+D results warrant a separate reading, as the performance drop there appears to be an isolated case driven by the dataset's scale and class diversity (60 classes), which genuinely stress the capacity ceiling of our design.

Limitations and Future Work

The model's representational capacity remains limited relative to deeply supervised methods, occasionally trailing the field under full training (Sec. 5); temporal consistency of the per-frame LOT alignment also varies across datasets, with UI-PRMD showing the lowest frame-to-frame agreement (Appendix C). Future work will extend evaluation to dense dynamic point clouds and LiDAR sequences with repeated-run analysis, and deployment in human motion applications should consider privacy, consent, and responsible use.

Conclusion

This work establishes the joint LOT–CDT convexity formulation and empirically validates the resulting linearization via improved linear separability, providing a theoretical and empirical basis for the subspace-based classification. Building on this foundation, we presented a parameter-free, CPU-trainable framework for dynamic point cloud sequence recognition that operates without learnable weights, GPU dependency, or heavy computational overhead. Across four benchmark datasets, our method demonstrates strong generalisation and dramatic reductions in FLOPs and inference latency, offering a favorable position on the accuracy-efficiency trade-off: no baseline matches both the accuracy and efficiency profile achieved here simultaneously. While our model does not outperform all baselines in every setting, this combination of competitive performance and practical usability, rather than state-of-the-art accuracy alone, is the paper's principal contribution: a fully hardware-agnostic, GPU-free model with fast inference and no learned parameters. We believe this represents a valuable and underexplored research axis, and offer it as a foundation for future work on efficient, deployable spatiotemporal recognition at the edge.

AI USE STATEMENT

AI tools were used solely to polish the writing at the grammar and phrasing level. No results, experiments, analyses, or claims in this work were generated, derived, or verified using AI; all technical content was produced and validated entirely by the authors. We take full responsibility for the final content of this work.

ETHICS STATEMENT

This work uses publicly available, already-anonymized human motion benchmark datasets (KIMORE, UI-PRMD, NTU RGB+D, MSR-Action3D) for skeleton-based classification; no new human-subject data was collected. As noted in Sec. 5, deployment of this or similar methods in human motion applications (e.g., rehabilitation monitoring) should consider privacy, consent, and responsible use. We are not aware of any conflicts of interest or additional ethical concerns raised by this work.

REFERENCES

Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

Marianna Capecci, Maria Gabriella Ceravolo, Francesco Ferracuti, Sabrina Iarlori, Andrea Monteriù, Luca Romeo, and Federica Verdini. The kimore dataset: Kinematic assessment of movement and clinical scores for remote monitoring of physical rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(7):1436–1448, 2019. doi: 10.1109/TNSRE.2019.2923060.

Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 180–189, 2020. doi: 10.1109/CVPR42600.2020.00026.

Lin Chi-Jen and Lin Wan-Ting. A systematic weighted-hungarian-algorithm for optimization and postoptimal analysis of transportation problem. *J Stat Manage Syst*, 26:843–866, 2023.

Hehe Fan and Yi Yang. Pointrnn: Point recurrent neural network for moving point cloud processing. *arXiv preprint arXiv:1910.08287*, 2019.

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*, pp. 14204–14213, 2021.

Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan Kankanhalli. Pstnet: Point spatio-temporal convolution on point cloud sequences. *arXiv preprint arXiv:2205.13713*, 2022.

Zhuoxu Huang, Zhenkun Fan, Tao Xu, and Jungong Han. On exploring pde modeling for point cloud video representation learning. *arXiv preprint arXiv:2404.04720*, 2024.

Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K Rohde. Optimal mass transport: Signal processing and machine-learning applications. *IEEE signal processing magazine*, 34(4):43–59, 2017.

W. Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pp. 9–14, 2010. URL https://api.semanticscholar.org/CorpusID:4333408.

Xing Li, Qian Huang, Zhijian Wang, Tianjin Yang, Zhenjie Hou, and Zhuang Miao. Real-time 3-d human action recognition based on hyperpoint sequence. *IEEE Transactions on Industrial Informatics*, 19(8):8933–8942, 2022.

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*, pp. 3203–3212, 2026.

Yuecong Min, Yanxiao Zhang, Xiujuan Chai, and Xilin Chen. An efficient pointlstm for point clouds based gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5761–5770, 2020.

Marzieh Mohammadi and Amir Salarpour. Point-gn: A non-parametric network using gaussian positional encoding for point cloud classification. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3487–3496, 2025. doi: 10.1109/WACV61041.2025.00344.

Marzieh Mohammadi, Amir Salarpour, and Pedram MohajerAnsari. Point-ln: A lightweight framework for efficient point cloud classification using non-parametric positional encoding. In *2025 29th International Computer Conference, Computer Society of Iran (CSICC)*, pp. 1–7, 2025. doi: 10.1109/CSICC65765.2025.10967399.

Caroline Moosmüller and Alexander Cloninger. Linear optimal transport embedding: provable wasserstein classification for certain rigid transformations and perturbations. *Information and Inference: A Journal of the IMA*, 12(1):363–389, 2023.

Se Rim Park, Soheil Kolouri, Shinjini Kundu, and Gustavo K Rohde. The cumulative distribution transform and linear pattern classification. *Applied and computational harmonic analysis*, 45(3):616–641, 2018.

Hongwei Ren, Yue Zhou, Jiadong Zhu, Xiaopeng Lin, Haotian Fu, Yulong Huang, Yuetong Fang, Fei Ma, Hao Yu, and Bojun Cheng. Rethinking efficient and effective point-based networks for event camera classification and regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Abu Hasnat Mohammad Rubaiyat, Shiying Li, Xuwang Yin, Mohammad Shifat-E-Rabbi, Yan Zhuang, and Gustavo K Rohde. End-to-end signal classification in signed cumulative distribution transform space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(9):5936–5950, 2024.

Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1010–1019, 2016. doi: 10.1109/CVPR.2016.115.

Mohammad Shifat-E-Rabbi, Naqib Sad Pathan, Shiying Li, Yan Zhuang, Abu Hasnat Mohammad Rubaiyat, and Gustavo K Rohde. Linear optimal transport subspaces for point set classification. *Journal of Mathematical Imaging and Vision*, 67(4):43, 2025.

Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1588–1597, 2019. doi: 10.1109/ICCV.2019.00167.

Aleksandar Vakanski, Hyung-pil Jun, David Paul, and Russell Baker. A data set of human body movements for physical rehabilitation exercises. *Data*, 3(1):2, 2018. doi: 10.3390/data3010002.

Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

Wei Wang, Dejan Slepčev, Saurav Basu, John A Ozolek, and Gustavo K Rohde. A linear optimal transportation framework for quantifying and visualizing variations in sets of images. *International journal of computer vision*, 101(2):254–269, 2013.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, 2015. doi: 10.1109/CVPR.2015.7298801.

Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press, 2018.

Renrui Zhang, Liuhui Wang, Yali Wang, Peng Gao, Hongsheng Li, and Jianbo Shi. Starting from non-parametric networks for 3d point cloud analysis. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5344–5353, 2023. doi: 10.1109/CVPR52729.2023.00517.

A Proof of Joint Convexity of the LOT–CDT Representation

Proposition 1. *Let $\mathcal{G}$ denote the spatial deformation group acting on unordered point sets and $\mathcal{H}$ denote the temporal deformation group acting on aligned joint trajectories (Section 3), with joint deformation group*

$$\Gamma = \mathcal{G} \times \mathcal{H},$$

so that spatial and temporal deformations act independently. For class k, define the LOT embedding of the spatial deformation orbit as

$$C_{\mathrm{LOT}}^{(k)} = \left\{ g \circ \widehat{P}_{\varphi}^{(k)} \;\middle|\; g \in \mathcal{G} \right\} \subset \mathbb{R}^{3N},$$

which is convex under the composition and convexity conditions of the LOT formulation Shifat-E-Rabbi et al. (2025).

Define the stacked CDT embedding of the temporal deformation orbit as

$$C_{\mathrm{CDT}}^{(k)} = \left\{ \left(h^{-1} \circ \widehat{\rho}^{(i,d;k)} \right)_{i,d} \;\middle|\; h \in \mathcal{H} \right\} \subset \mathbb{R}^{T \times 3N}.$$

(Here $\mathcal{H}^{-1}$ is the nonnegative-coefficient combination set $\{\sum_i \alpha_i f_i : \alpha_i \geq 0\}$ over basis deformation functions f_i, as in Rubaiyat et al. (2024); evaluation at a fixed point is linear in the α_i, which is why the stacking map below is linear.) Since the same temporal deformation h acts on all $3N$ channels simultaneously, let $q = h^{-1}$. Under the convex-group condition of the CDT formulation, $\mathcal{H}^{-1}$ is convex Rubaiyat et al. (2024). Moreover, for fixed $\widehat{\rho}^{(i,d;k)}$, the stacking map

$$q \;\mapsto\; \left(q \circ \widehat{\rho}^{(i,d;k)} \right)_{i,d}$$

is linear in q, since

$$[\lambda q_1 + (1-\lambda) q_2] \circ \widehat{\rho}^{(i,d;k)} = \lambda \left(q_1 \circ \widehat{\rho}^{(i,d;k)} \right) + (1-\lambda) \left(q_2 \circ \widehat{\rho}^{(i,d;k)} \right)$$

for any $\lambda \in [0,1]$. As the linear image of the convex set $\mathcal{H}^{-1}$, $C_{\mathrm{CDT}}^{(k)}$ is therefore convex.

Because $\mathcal{G}$ and $\mathcal{H}$ act independently under $\Gamma = \mathcal{G} \times \mathcal{H}$, the image of the joint deformation orbit is

$$\widehat{\mathcal{M}}^{(k)} = C_{\mathrm{LOT}}^{(k)} \times C_{\mathrm{CDT}}^{(k)} \;\subset\; \mathbb{R}^{3N} \times \mathbb{R}^{T \times 3N}.$$

Therefore, $\widehat{\mathcal{M}}^{(k)}$ is convex.

Proof. Take any two points

$$(P_1, S_1),\ (P_2, S_2) \;\in\; C_{\mathrm{LOT}}^{(k)} \times C_{\mathrm{CDT}}^{(k)},$$

where $P_1, P_2 \in C_{\mathrm{LOT}}^{(k)}$ and $S_1, S_2 \in C_{\mathrm{CDT}}^{(k)}$. For any $\lambda \in [0,1]$, convexity of $C_{\mathrm{LOT}}^{(k)}$ and $C_{\mathrm{CDT}}^{(k)}$ gives

$$\lambda P_1 + (1-\lambda) P_2 \in C_{\mathrm{LOT}}^{(k)}, \qquad \lambda S_1 + (1-\lambda) S_2 \in C_{\mathrm{CDT}}^{(k)}.$$

Therefore,

$$\begin{aligned} &\lambda (P_1, S_1) + (1-\lambda)(P_2, S_2) \\ &\quad = (\lambda P_1 + (1-\lambda) P_2,\ \lambda S_1 + (1-\lambda) S_2) \\ &\quad \in\; C_{\mathrm{LOT}}^{(k)} \times C_{\mathrm{CDT}}^{(k)} \;=\; \widehat{\mathcal{M}}^{(k)}. \end{aligned}$$

Hence $\widehat{\mathcal{M}}^{(k)}$ is convex. □

A.1 Discrete Implementation and Approximate Convexity

Proposition 1 establishes exact convexity of the joint LOT–CDT embedding $\widehat{\mathcal{M}}^{(k)}$ under the continuous generative model of Sec: 3.4. In practice, spatial alignment is carried out via the discrete, Hungarian-based procedure of Sec: 3.2.1, which is a finite-sample computational realization of the continuous LOT transport map T_{σ}^{μ}.

This discrete-continuous relationship mirrors one already established in the foundational LOT literature: exact convexity of the LOT embedding holds only for deformations satisfying the exact composition property $T_\sigma^{g\#\mu} = g \circ T_\sigma^\mu$ (basic transformations such as translations and isotropic scalings), while deformations within an ϵ-neighborhood of this class satisfy an *approximate*, 2δ-convexity property (Moosmüller & Cloninger, 2023; Shifat-E-Rabbi et al., 2025), with $\delta \to 0$ as the deformation approaches the exact composition regime. The discrete embeddings produced by our pipeline inherit this same guarantee: for deformations near the basic transformation classes, the discrete joint embedding is 2δ-convex, with the additional discretization error vanishing as the point set's empirical measure converges to its continuous limit. This is the same sense in which a discrete implementation is understood to approximate a continuous transport-based embedding throughout the LOT and CDT literature (Shifat-E-Rabbi et al., 2025; Rubaiyat et al., 2024); it does not weaken Proposition 1 itself, which characterizes the exact continuous generative model that the discrete pipeline is designed to estimate.

B Dataset Details

KIMORE

KIMORE Capecci et al. (2019) is a rehabilitation motion dataset comprising five exercise classes performed by 78 subjects, including healthy and motor-impaired participants, captured using a Microsoft Kinect v2 sensor. We use the 3D skeleton joint-position sequences, where each frame is represented by 25 body joints in 3D space, and treat the dataset as a 5-class exercise classification task.

UI-PRMD

UI-PRMD Vakanski et al. (2018) contains 10 rehabilitation exercise classes performed by 10 healthy subjects and recorded using both a Vicon system and a Microsoft Kinect sensor. We use the Kinect-derived 3D skeleton joint-position sequences, where each frame is represented by 22 body joints in 3D space, to evaluate rehabilitation motion classification under correct and non-optimal execution variations.

MSR-Action3D

MSR-Action3D Li et al. (2010) is a small-scale action recognition benchmark containing 20 action classes performed by 10 subjects, with several kinematically similar categories. We use the 3D skeleton joint-position sequences to assess performance under limited-scale, fine-grained action recognition conditions.

NTU RGB+D

NTU RGB+D Shahroudy et al. (2016) is a large-scale action recognition benchmark containing 60 action classes and 56,880 samples performed by 40 subjects. Each skeleton frame is represented by 25 body joints in 3D space. We use the 3D skeleton joint-position sequences to evaluate scalability to large-scale multi-class action recognition.

Table 7: Summary of dynamic 3D skeleton motion benchmarks used in this study.

Dataset	Domain	Classes	Subjects	Samples Used	Joints
KIMORE Capecci et al. (2019)	Rehabilitation	5	78	2516	25
UI-PRMD Vakanski et al. (2018)	Rehabilitation	10	10	2000	22
MSR-Action3D Li et al. (2010)	Action recognition	20	10	567	20
NTU RGB+D Shahroudy et al. (2016)	Action recognition	60	40	56880	25

C Temporal Stability of the LOT Alignment

Since the proposed framework aligns each frame independently to a single global reference, it is natural to ask how consistent the resulting canonical point ordering is across time, and how this

interacts with downstream classification performance. We quantify this using the *frame-to-frame assignment agreement rate*: for each sample, the fraction of consecutive frame pairs $(t, t+1)$ whose Hungarian assignment (relative to the fixed reference frame) is identical, averaged over all samples in a split. Table 8 reports this rate across all four datasets.

Agreement rates range from 60–87% across datasets, with KIMORE, NTU RGB+D, and MSR-Action3D each exceeding 77% and UI-PRMD lower at around 60%. Notably, this variation does not translate into a corresponding drop in classification accuracy: UI-PRMD, despite having the lowest agreement rate, still achieves 86.00% accuracy at a 90:10 split (Table 10), among the strongest results across all four datasets and comparable methods. This indicates that the per-channel CDT representation, combined with subspace-based classification, is robust to variation in frame-to-frame assignment consistency: the method does not require a perfectly stable point correspondence over time to extract a discriminative temporal signal, since class-relevant structure is captured collectively across the trajectory rather than depending on any single frame's exact correspondence. We view explicit enforcement of sequence-level consistency (e.g., regularizing frame t's assignment toward frame $t-1$'s) as a promising avenue for recovering further gains on datasets such as UI-PRMD where agreement is lower, and report these measurements to support reproducibility and transparency around this component of the pipeline.

Table 8: Frame-to-frame assignment agreement rate (%) across datasets. Higher values indicate a more temporally stable canonical point ordering under per-frame LOT alignment against the fixed reference.

Dataset	Train Agreement (%)	Test Agreement (%)
KIMORE	86.75	86.22
UI-PRMD	61.64	59.94
NTU RGB+D	82.34	81.59
MSR-Action3D	79.64	76.98

D Justification for Joint-Order Scrambling

We randomly permutes joint indices per sample (train and test, applied identically to all methods), holding the permutation fixed across all T frames of that sample so temporal structure within a sequence is preserved.

Motivation. All methods compared in this work, including the eight baselines, are designed for dynamic point cloud classification, where the input is fundamentally a set of 3D coordinates rather than a labeled, topologically structured skeleton. In practical sensing scenarios such as LiDAR, depth cameras, and markerless motion capture, raw point cloud output typically carries no persistent joint identity, label, or guaranteed consistent ordering across frames or sensors; the sensor provides geometry alone. We scramble joint order to reflect this realistic setting directly, rather than the more favorable case where a fixed, known index-to-anatomy correspondence is assumed to be available at input time.

Evaluating classification performance under this scrambled, identity-free input is therefore the setting we consider most representative of real-world deployment for point cloud based dynamic classification systems, and we evaluate all methods, including our own, under this same realistic condition rather than a more idealized one.

Consistency of the protocol. The same permutation is applied identically to every method for a given split; the permutation is fixed per sample across all frames of that sequence; train and test use the same scrambling procedure; and all baselines are trained from scratch on scrambled data using their original published architectures and hyperparameters, with no modification made to either side.

E Complete Implementation Details

Our method runs entirely on CPU, with no GPU support required. Variable-length skeleton sequences are first resampled to a fixed temporal resolution of T=100 frames via per-channel linear interpolation,

and coordinates are normalized by subtracting the per-axis minimum over the training set. Joint indices are then randomly permuted sample-by-sample using a fixed seed, and this same permuted data is used as input to all baseline models to ensure a fair comparison. LOT alignment is applied to every sequence in both splits: a single global reference frame (the first frame of the first training sample) is fixed for the entire experiment, and each frame is independently aligned to it by solving a linear assignment problem on the squared-Euclidean joint-to-joint cost matrix via the Hungarian algorithm Chi-Jen & Wan-Ting (2023), yielding a permutation that best matches the reference configuration. Each aligned sequence of shape $(T, J, 3)$ is then reshaped to (T, D), $D{=}J{\times}3$, and the CDT is applied column-wise along the temporal axis, with each column treated as a discrete probability mass function after non-negativity projection and ℓ_1 normalization (a 2% clipping threshold is applied for numerical stability, with a fallback to full support when fewer than two unmasked points remain). For subspace learning, CDT-transformed training samples of each class c are flattened to $T{\times}D$ vectors and stacked into a class data matrix; thin SVD is applied to each, and the class subspace $\mathbf{B}_c$ retains the top-k_c right singular vectors capturing 99% of the total spectral energy, so k_c adapts automatically to within-class variation rather than being manually tuned.

At inference, each test sequence undergoes the same preprocessing pipeline — resampling, normalisation, LOT alignment, and CDT — after which it is flattened and classified by the subspace yielding the minimum ℓ_2 projection residual:

$$\hat{y} = \arg\min_c \left\| \hat{\mathbf{x}} - \mathbf{B}_c^\top \left(\mathbf{B}_c \hat{\mathbf{x}} \right) \right\|_2 , \tag{1}$$

Evaluation Protocol. We report overall accuracy, macro-averaged precision, recall, and F1-score, alongside FLOPs, parameter count, and average per-sample inference latency, measured on a single CPU core for all methods except EventMamba and PointNet4D, which require multi-core GPU inference for their Mamba-based modules. All baselines are trained on a single NVIDIA RTX 5090 GPU using their original architectures, without any component of our pipeline.

FLOPs computation. FLOPs for the proposed method are analytically derived from each stage's operation count, not measured via a runtime profiler, since there is no learned forward pass to instrument. Alignment FLOPs combine the cost-matrix construction (N^2 pairwise squared-Euclidean distances) and the Hungarian algorithm's $O(N^3)$ cost, summed over all T frames; CDT FLOPs count the cumulative-sum, normalization, and interpolation operations per channel; projection FLOPs count the matrix-vector multiplications and norms in the classification rule. Baseline FLOPs reflect full training-time forward and backward passes across all epochs, whereas ours reflect a single forward-only, non-iterative computation; the resulting gap in magnitude is a consequence of this difference in computational paradigm, not a measurement inconsistency. Inference latency, measured directly as wall-clock time, is the primary efficiency claim of this work; FLOPs are a secondary, coarser indicator.

Parameter count. The proposed method involves no gradient-trained, backpropagation-learned parameters at any stage; class subspace bases B_c are computed directly from training data via a single closed-form SVD, with no iterative optimization, loss function, or gradient update involved. We report "0" in the Params columns specifically to denote the absence of such learned parameters, consistent with the paper's core claim of a backpropagation-free, non-parametric framework.

Reproducibility statement

Full source code is provided anonymously in Abstract, with no author-identifying information, enabling complete reproduction of all reported results. Implementation details, including preprocessing, alignment, and hyperparameter settings, are provided in Appendix E; and dataset details and processing steps are provided in Appendix B.

F Complete Results

Table 9: Results on **KIMORE** across train/test splits.

Ratio	Model	Acc	Prec	Rec	F1	FLOPs	Params	Inference Latency
90:10	PointRNN	88.97	91.66	87.61	88.96	1209.23 T	1.35M	172.97 ms
	PointLSTM	88.24	88.75	88.55	88.62	508.04 T	87.301K	47.89 ms
	PSTNet	94.12	95.09	93.98	94.41	1250.35 T	8.408M	30.68 ms
	P4Transformer	88.97	89.85	89.02	89.23	54.42 T	0.11M	4.60 ms
	EventMamba	92.65	93.59	92.65	93.03	387.79 T	1.59 M	2.49 ms ms (GPU)
	SequentialPointNet	94.85	95.90	94.98	95.20	305.81 T	2.986M	17.65 ms
	MotionPointNet	93.38	94.36	93.42	93.66	221.33 T	0.08M	63.85 ms
	PointNet4D	91.18	91.20	91.29	91.24	45.08 T	1.238M	0.09 ms (GPU)
	Ours	90.44	92.66	89.40	90.65	5.03 G	0	0.9086 ms
10:90	PointRNN	40.02	16.15	33.37	21.76	155.03 T	1.35M	172.63 ms
	PointLSTM	68.90	70.02	65.77	66.58	53.48 T	87.301K	46.90 ms
	PSTNet	82.49	82.68	82.14	82.36	121.09 T	8.408M	26.86 ms
	P4Transformer	64.24	63.21	63.87	63.48	5.73 T	0.11M	4.59 ms
	EventMamba	88.46	88.92	87.80	88.26	20.41 T	1.59 M	2.19 ms (GPU)
	SequentialPointNet	89.28	90.85	88.24	89.08	32.19 T	2.986M	14.91 ms
	MotionPointNet	84.53	86.43	82.69	83.89	23.30 T	0.08M	59.30 ms
	PointNet4D	89.61	89.28	89.40	89.29	4.75 T	1.238M	0.04 ms (GPU)
	Ours	86.33	88.98	84.50	85.94	0.33 G	0	0.8527 ms

Table 10: Results on **UI-PRMD** across train/test splits.

Ratio	Model	Acc	Prec	Rec	F1	FLOPs	Params	Inference Latency
90:10	PointRNN	10.00	1.00	10.00	1.82	296.75 T	1.35M	51.13 ms
	PointLSTM	99.00	99.09	99.00	99.00	237.19 T	87.946K	31.63 ms
	PSTNet	92.00	93.14	92.00	91.84	478.67 T	8.419M	19.83 ms
	P4Transformer	100	100	100	100	9.07 T	0.11M	2.20 ms
	EventMamba	98.00	98.18	98.00	97.99	284.94 T	1.59 M	2.57 ms (GPU)
	SequentialPointNet	56.00	56.66	56.00	55.39	162.33 T	2.987M	12.25 ms
	MotionPointNet	87.00	88.40	87.00	87.07	61.00 T	0.08M	52.19 ms
	PointNet4D	10.00	11.12	10.00	10.28	23.87 T	1.237M	0.10 ms (GPU)
	Ours	86.00	91.92	86.00	84.95	1.39 G	0	0.4739 ms
10:90	PointRNN	10.00	1.00	10.00	1.82	56.67 T	1.35M	50.96 ms
	PointLSTM	66.89	68.22	66.89	65.58	25.41 T	87.946K	30.59 ms
	PSTNet	50.44	51.47	50.44	49.92	51.29 T	8.419M	16.34 ms
	P4Transformer	41.56	42.34	41.56	41.70	0.97 T	0.11M	2.00 ms
	EventMamba	69.78	71.93	69.78	69.41	30.53 T	1.59 M	2.15 ms (GPU)
	SequentialPointNet	19.22	19.85	19.22	19.03	17.39 T	2.987M	10.99 ms
	MotionPointNet	58.78	59.18	58.78	58.68	6.54 T	0.08M	49.58 ms
	PointNet4D	10.89	11.40	10.89	10.99	2.56 T	1.237M	0.03 ms (GPU)
	Ours	76.22	77.83	76.22	76.03	0.12 G	0	0.5053 ms

Table 11: Results on **NTU RGB+D** across train/test splits.

Ratio	Model	Acc	Prec	Rec	F1	FLOPs	Params	Inference Latency
90:10	PointRNN	36.8	35.2	34.7	34.9	986.6 T	1.35M	261.2 ms
	PointLSTM	68.96	69.17	68.97	68.94	21270.99 T	94.396K	48.70 ms
	PSTNet	66.90	67.63	66.91	66.90	13088.38 T	8.521M	27.49 ms
	P4Transformer	66.31	66.35	66.31	66.24	2278.63 T	0.12M	5.25 ms
	EventMamba	80.54	80.85	80.55	80.60	16236.06 T	1.61 M	2.15 ms (GPU)
	SequentialPointNet	69.46	69.35	69.47	69.26	12803.93 T	3.000M	16.67 ms
	MotionPointNet	63.4	62.1	61.8	61.9	13625.4 T	0.10M	78.16 ms
	PointNet4D	50.97	51.15	50.97	50.97	1887.63 T	1.252M	0.03 ms (GPU)
	Ours	45.25	55.13	45.28	44.62	434.78 G	0	7.3003 ms
10:90	PointRNN	18.4	17.9	18.1	17.8	365.5 T	1.35M	256.5 ms
	PointLSTM	45.53	45.89	45.52	45.55	2353.05 T	94.396K	52.42 ms
	PSTNet	28.77	29.72	28.76	26.96	144.79 T	8.521M	28.21 ms
	P4Transformer	35.62	35.77	35.62	35.63	252.07 T	0.12M	5.32 ms
	EventMamba	62.96	63.48	62.97	63.06	1796.07 T	1.61 M	2.17 ms (GPU)
	SequentialPointNet	47.10	46.73	47.10	46.71	1416.40 T	3.000M	17.30 ms
	MotionPointNet	46.7	45.9	45.1	45.4	518.64 T	0.10M	0.00
	PointNet4D	13.34	13.08	13.35	13.13	208.81 T	1.252M	0.03 ms (GPU)
	Ours	39.17	42.54	39.17	38.71	16.32 G	0	2.7714 ms

Table 12: Results on **MSR-Action3D** across train/test splits.

Ratio	Model	Acc	Prec	Rec	F1	FLOPs	Params	Inference Latency
90:10	PointRNN	41.44	39.24	40.58	38.88	1492.84 T	1.35M	68.06 ms
	PointLSTM	88.46	87.17	87.50	85.65	171.13 T	89.236K	47.21 ms
	PSTNet	84.68	88.31	84.83	85.25	259.26 T	8.439M	30.73 ms
	P4Transformer	75.00	79.58	73.33	73.11	18.28 T	0.11M	3.93 ms
	EventMamba	96.15	97.08	95.83	95.62	173.22 T	1.60 M	2.53 ms(GPU)
	SequentialPointNet	55.77	45.83	51.67	46.54	136.81 T	2.99M	15.26 ms
	MotionPointNet	93.69	95.45	93.67	93.75	63.17 T	0.09M	46.40 ms
	PointNet4D	25.00	26.25	23.33	23.86	60.32 T	1.238M	0.09 ms (GPU)
	Ours	92.31	90.83	90.83	89.90	0.70 G	0	1.0213 ms
10:90	PointRNN	5.91	0.61	5.56	0.96	91.56 T	1.35M	64.28 ms
	PointLSTM	9.06	2.80	9.26	3.67	9.52 T	89.236K	45.87 ms
	PSTNet	46.26	51.60	46.66	45.93	46.93 T	8.439M	22.46 ms
	P4Transformer	7.68	5.10	7.42	4.62	2.15 T	0.11M	3.39 ms
	EventMamba	53.74	58.21	53.56	53.90	20.38 T	1.60 M	2.18 ms (GPU)
	SequentialPointNet	17.91	17.89	17.40	16.55	16.10 T	2.99M	21.24 ms
	MotionPointNet	63.78	64.56	63.68	63.31	8.42 T	0.09M	50.58 ms
	PointNet4D	8.86	9.99	8.62	8.75	7.10 T	1.238M	0.03 ms(GPU)
	Ours	57.68	59.83	57.18	56.98	0.07 G	0	1.0632 ms