Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Explainable Point-Cloud Deep Learning for 3D Conformation Recognition of Temperature-Sensitive Bottlebrush Polymers

Amirul Islam Saimon
Virginia Tech
Soumil Joshi
Virginia Tech
Jihoon Chung
Hanyang University
Blake Johnson
Virginia Tech
Sanket Deshmukh
Virginia Tech
Zhenyu Kong
zkong@vt.edu

Virginia Tech

Article

Keywords: Bottlebrush polymers, PointNet, explainable artificial intelligence, molecular dynamics

Posted Date: September 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11134256/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Explainable Point-Cloud Deep Learning for 3D Conformation Recognition of Temperature-Sensitive Bottlebrush Polymers

Amirul Islam Saimon[1], Soumil Y. Joshi[2], Jihoon Chung[3], Blake N. Johnson[1], Sanket A. Deshmukh[2*], Zhenyu (James) Kong[1*]

[1]Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA, USA.
[2]Department of Chemical Engineering, Virginia Tech, Blacksburg, VA, USA.
[3]Department of Industrial Engineering, Hanyang University, Seoul, South Korea.

*Corresponding author(s). E-mail(s): sanketad@vt.edu; zkong@vt.edu;

Abstract

Bottlebrush polymers (BBPs) exhibit highly tunable physicochemical properties arising from their unique 3D architectures of densely grafted side chains. Accurately recognizing and classifying their conformations is critical for understanding structure-property relationships, yet remains challenging due to the complex geometries. Coarse-grained molecular dynamics (MD) simulations can generate 3D BBP conformations as bead-level coordinates that provide point cloud data for quantitative shape analysis. Conventional approaches either compress these data into hand-crafted geometric descriptors or discretize the structures onto 3D grids (voxelization) before applying data-driven models. However, low-resolution voxelization may discard fine-grained conformational information, while high-resolution voxelization can be computationally expensive for large BBP datasets. In this paper, we apply an enhanced PointNet-based deep learning model that directly operates on raw 3D point clouds of BBP conformations. The proposed model integrates multi-scale feature aggregation and a dual pooling mechanism to enhance geometric representation learning. Applied to a 12-class dataset of 5-tiered temperature-sensitive BBPs (TS-BBPs) spanning three architectures, two temperatures across the LCST of poly(***N***-isopropylacrylamide), and two degrees of dextran functionalization, our model achieves a mean test accuracy of 94.81%,

exceeding both traditional machine learning and voxel-based 3D convolutional neural network (3D-CNN) baselines while eliminating voxelization entirely. For explainability, we adapt Grad-CAM to point-cloud networks to compute bead-level importance scores, and validate them with a deletion-based perturbation test in which removing highly attributed beads degrades accuracy substantially faster than random removal. Proposed point-cloud learning thus offers an efficient, accurate, and physically explainable route to conformation recognition in TS-BBPs, paving the way for scalable trajectory analysis and inverse design of soft-matter architectures.

Keywords: Bottlebrush polymers, PointNet, explainable artificial intelligence, molecular dynamics

1 INTRODUCTION

The topological structure of a polymer is a primary factor that governs its physicochemical properties and strongly influences both bulk behaviors (e.g., rheology, mechanical strength) and solution properties (e.g., shear viscosity, self-assembly) [1–3]. Among the various polymer architectures, bottlebrush polymers (BBPs) [3–5] represent a distinctive class of branched macromolecules that have received substantial attention in recent years [6]. These BBPs feature a central backbone densely grafted with flexible side chains, producing a brush-like morphology that enables precise control over molecular dimensions, packing, and functionality [6]. The resulting 3D architecture imparts BBPs with unique, as well as tunable, properties pertaining to their softness, entanglement, and responsiveness, which have been harnessed in diverse applications ranging from drug delivery [7–9] and surfactant design [10, 11] to organic electronics [12] and stimuli-responsive coatings [13]. Because the macroscopic performance of these materials is intrinsically linked to their 3D conformations, accurate structural recognition and classification of BBP conformations are critical for understanding, predicting, and ultimately guiding their function in advanced material systems [6]. Within this broader family of BBPs, the design and study of temperature-sensitive bottlebrush polymers (TS-BBPs) have recently gained significant attention due to their tunable conformational behaviors resulting from a simplistic administration of stimulus, and their diverse applications in smart materials [4, 6, 14–19].

Despite progress in synthesis and experimental characterization, accurately probing the 3D conformations of TS-BBPs remains challenging [20]. The intrinsic architectural complexity of these BBPs, arising from variations in grafting density, side chain length, and backbone rigidity, combined with experimental resolution limits, makes it difficult to directly correlate microscopic structural parameters with macroscopic conformational states [21, 22]. Consequently, computational modeling approaches, particularly coarse-grained (CG) molecular dynamics (MD) simulations, have emerged as useful tools to explore the conformational behavior of BBPs at the molecular level [23, 24]. Snapshots of BBPs obtained from a CG MD simulation trajectory, where each CG

bead is represented by its x, y and z coordinates, also provide geometric representations of their structures in the form of 3D point clouds. This enables a systematic and efficient investigation of how the architecture, temperature, and functionalization influence global shape transitions, providing a controlled framework to study the molecular origins of responsiveness in TS-BBPs [6]. These trajectories are conventionally analyzed through hand-crafted structural descriptors such as the radius of gyration, solvent-accessible surface area (SASA), asphericity, free volume, and intra-polymer contacts, which have long been the standard means of quantifying polymer conformations and remain the standard descriptors used to characterize conformational changes [6]. We report this conventional analysis for all BBP systems studied in this paper in Sections S1 and S2 of the Supporting Information (SI). Each descriptor, however, compresses a full 3D structure into a single scalar, so conformations that differ in shape but have similar values for the chosen descriptor are not distinguished. The descriptors must also be selected in advance, which presumes that the structural features distinguishing the architectures are already known.

These limitations motivate approaches that learn representations directly from the 3D point clouds of BBP conformations generated by CG MD simulations, rather than relying on a predetermined set of descriptors. Machine learning (ML) [25–29] and deep learning (DL) [6, 30–32] have emerged as powerful tools for extracting structural patterns in soft-matter systems, particularly given the volume of high-dimensional data generated by CG MD simulations. These methods provide a framework for learning hidden relationships between molecular conformations and macroscopic properties directly from simulation data [19].

Among the DL-based approaches, our earlier work (Joshi et al.[6]) introduced a 3D convolutional neural network (3D-CNN) framework to quantify shape transformations in uniquely structured TS-BBPs of poly(N-isopropylacrylamide) (PNIPAM). Because standard 3D-CNNs operate on regular grids rather than unordered bead coordinates, raw CG MD coordinates first had to be mapped onto their volumetric representations [33–35]. To do so, each BBP conformation, originally represented as a 3D point cloud of beads, was converted into voxel grids by discretizing the continuous 3D space into a fixed-size grid (resolution 50^3) [6], where each voxel represented a small cubic region analogous to a pixel in 2D images. Through the study, we broadly demonstrated how side chain length distribution, whether gradually or abruptly varied, significantly influenced the compactness, bending, and overall conformations of these TS-BBPs above and below the lower critical solution temperature (LCST) of PNIPAM.

However, while voxelization enables the use of CNN architectures to capture spatial correlations, it introduces several drawbacks [36, 37]. The discretization process can cause quantization artifacts and loss of fine geometric details, especially at lower resolutions [36] (Figure 1). In contrast, increasing the voxel-grid resolution generally improves geometric fidelity but substantially increases memory usage and computational cost, since voxel-based representations scale volumetrically with grid resolution. In addition, for voxel-based inference, each raw MD-generated BBP frame must first be converted into a voxel grid before being passed to the trained 3D-CNN, adding an extra preprocessing step to the prediction pipeline. In the context of large BBP conformation datasets, this resolution-efficiency trade-off can limit scalability and motivates

approaches that operate directly on raw 3D molecular coordinates. Point cloud-based learning offers a natural alternative, as it avoids discretization and preserves geometric fidelity. In this formulation, each BBP conformation is represented as a set of 3D bead coordinates rather than as a grid or ordered sequence, so the learning model should be invariant to permutations of the input beads while still capturing their spatial relationships [36, 37].

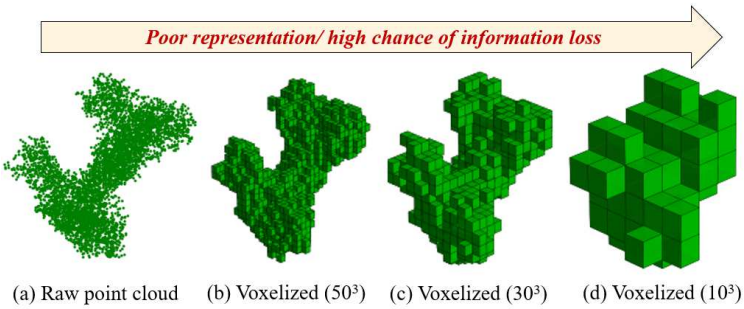


Fig. 1: A representative BBP conformation is shown in (a) raw point cloud format and in (b-d) voxelized formats at progressively coarser resolutions: (b) 50^3, (c) 30^3, and (d) 10^3. All representations are constructed within the same cubic voxelization domain (300^3); only the voxel-grid resolution varies.

Architectures such as PointNet [36] are particularly designed for point cloud data, as they respect this permutation-invariance and learn spatial features directly from point sets. Recent studies have leveraged PointNet [36] for molecular and materials modeling tasks [38], including local structural motif detection in atomistic simulations [39], prediction of liquid structural properties [40], and cocrystal formation analysis [41]. Most of these studies used the vanilla PointNet network proposed by Qi et al. [36] for their respective tasks.

In this paper, we formulate the TS-BBP recognition problem as a point cloud classification task, where each conformation is represented as an unordered set of 3D bead coordinates obtained from CG MD simulations. We apply an improved PointNet architecture that integrates multi-scale feature aggregation and a dual pooling mechanism to enhance recognition performance. This design enables the model to more effectively capture the geometric complexity of polymer conformations, particularly in TS-BBPs that exhibit conformational diversity driven by side chain distributions and functionalization.

While the PointNet-based point-cloud network provides a flexible framework for recognizing polymer conformations, its black-box nature necessitates model explanation to identify the structural features driving its predictions. This need is especially critical for BBPs, where the input consists solely of 3D bead coordinates, abstracting away explicit chemical or bonding information. Such model explanation is commonly framed as explainable artificial intelligence (XAI), which aims to expose the rationale of a trained network by highlighting which parts of the input most influence the output.

For image data, XAI methods such as Gradient-weighted Class Activation Mapping (Grad-CAM) [42], Integrated Gradients [43], and Local Interpretable Model-Agnostic Explanations (LIME) [44] are widely used to interpret CNN predictions. In contrast, XAI for 3D point-cloud models is still relatively less mature. Existing XAI approaches for point-cloud neural networks can generally be grouped into three categories: (*i*) gradient-based attribution methods [45, 46], which use gradients to assign importance scores to individual points; (*ii*) perturbation-based methods [47, 48], which assess how predictions change when selected points or local regions are modified or removed; and (*iii*) surrogate-model approaches [49, 50], which locally approximate the behavior of the black-box model using a simpler interpretable model. In this paper, we employ a gradient-based attribution strategy to explain the PointNet-based BBP classification network, specifically by adapting Grad-CAM [42] to point-cloud networks. Unlike images, our inputs are unordered 3D bead coordinates, so we compute the explanation on the network's bead-wise feature representation. We visualize the resulting bead-level importance scores as 3D saliency maps over the input BBP conformation. We further quantitatively validate this explanation using a deletion-based perturbation test, in which we show that removing the most highly attributed beads degrades classification accuracy much faster than removing random beads.

In summary, the main contributions of this paper are:

- We apply an enhanced PointNet-based deep learning model that recognizes complex 3D BBP conformations directly from raw MD-generated bead coordinates, and we show through comparison with traditional ML methods and voxel-based 3D CNNs [6] across multiple grid resolutions that the proposed approach achieves superior accuracy.
- We apply a Grad-CAM-based attribution method to the bead-wise learned features to explain the model and generate bead-level saliency maps, and we validate these explanations using a deletion-based perturbation test.

The remainder of this paper is structured as follows: Section 2 describes the generation of the BBP dataset using CG MD simulations and the formulation of the BBP conformation recognition problem. Section 3 introduces the PointNet-based BBP recognition framework used in this paper, detailing its architecture, hyperparameter tuning, training procedure, and explainability method. Section 4 presents the results, including comparative model performance, ablation studies, and Grad-CAM-based explainability analysis. Finally, Section 5 concludes the paper with key findings and future research directions.

2 GENERATION OF THE BBP DATASET AND PROBLEM SETUP

This section describes the generation of the BBP dataset using CG MD simulations and the mathematical formulation of the conformation recognition problem. The SI provides further details on the simulated systems and the conventional analyses of their trajectories.

2.1 Molecular models and simulation details

We first outline the BBP systems considered in this study and the CG models used to represent them. We then describe the simulation protocol and explain how the resulting trajectories are processed to construct the dataset used for BBP conformation recognition.

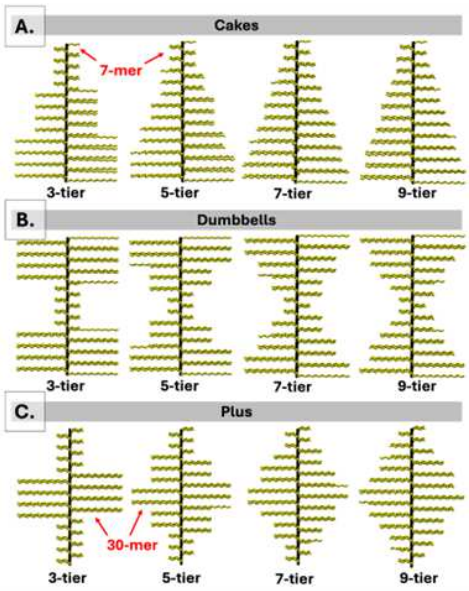


Fig. 2: Schematic representations of the tiered TS-BBP architectures considered in this study. The BBP backbone is shown in black, and the PNIPAM side chains are shown as yellow spheres. Side-chain lengths vary from 7 to 30 monomer units, forming 3-, 5-, 7-, and 9-tier Cake-like, Dumbbell-like, and Plus-like architectures.

BBP architectures

The dataset used in this paper is generated from over 20 μs of combined CG MD simulations of TS-BBPs. These TS-BBPs consist of a hydrophobic backbone of 72 beads with PNIPAM side chains grafted symmetrically at 100% grafting density, similar to those considered in our previous work [6]. By varying the PNIPAM side chain lengths from 7-mers to 30-mers along the backbone, different 3D BBP structures were generated. Figure 2 illustrates our broader tiered TS-BBP design space, in which discrete side chain arrangements produce three architecture families: Cake-like, Dumbbell-like, and Plus-like, with possible 3-tier, 5-tier, 7-tier, and 9-tier designs.

In addition to these unfunctionalized PNIPAM bottlebrushes, we also studied dextran-functionalized variants of our TS-BBPs [51], in which selected PNIPAM side chains were functionalized with dextran 10-mers at their free ends. Dextran, an

oligosaccharide known for its inherent hydrophilicity and biocompatibility, was specifically chosen as the functional group due to its strong aggregation potential [51–53]. This aggregation behavior of dextran chains may significantly influence the conformations of TS-BBP architectures, potentially even overriding the thermosensitive behavior inherent to PNIPAM side chains. The dextran-functionalized TS-BBPs were constructed for all three BBP families but only for the 5-tiered systems with three degrees of functionalization - 33% (where every third PNIPAM side chain was functionalized), 50% (every second side chain was functionalized), and 100% (every side chain was functionalized). This allowed us to focus on the effects of functionalization density and temperature on TS-BBP conformations without introducing additional variations from tier numbers. The 5-tiered systems were chosen because they provide an intermediate configuration between the abruptly varying 3-tiered and the gradually varying 9-tiered systems [51, 54]. Representative functionalized TS-BBP systems are presented in Figures S10–S12 of the SI. For the PointNet-based BBP recognition framework, however, the dataset used to train and evaluate the proposed classifier includes only the 5-tier systems at 300 and 320 K, with either no dextran functionalization or 100% dextran functionalization. The remaining 3-, 7-, and 9-tier systems, together with the 33% and 50% dextran-functionalized 5-tier systems, are excluded from model training and are subsequently considered in Section 4.4 to examine how the trained classifier projects these additional conformations onto the learned 12-class label space.

CG models used

Because the simulated BBP systems consist of PNIPAM-based bottlebrush polymers in explicit water, with some systems additionally carrying dextran terminal groups, the CG MD simulations require force-fields for all three components: water, PNIPAM, and dextran. These force-fields govern the intermolecular and intramolecular interactions that ultimately determine the equilibrium 3D conformations of the BBPs, which form the point-cloud inputs for our classification task. Our simulations employ CG force-fields that have been specifically developed and validated for these components.

- *Water* is represented using a 1-site CG model with a 2:1 mapping scheme, where each bead corresponds to two water molecules. This chargeless bead has a mass of 36.03 amu and interacts exclusively through van der Waals interactions modeled with a 12-6 Lennard-Jones (LJ) potential. The force-field parameters were optimized via particle swarm optimization (PSO) and gradient descent (GD) algorithms against experimental benchmarks such as density, diffusion coefficient, and dielectric constant at 300 K. Despite these limited optimization targets, the final CG model reproduces additional properties like surface tension, isothermal compressibility, and heat of vaporization, and has been validated across varying system sizes, time steps, and temperatures, confirming its robustness and transferability [55].
- For *PNIPAM*, we adopt the transferable CG model of Bejagam et al. [56], which maps each N-isopropylacrylamide (NIPAM) monomer into three beads corresponding to the backbone C_2H_4 group, the NH−CO amide group, and the terminal

isopropyl group. Bonded interactions among these beads are described using harmonic bond and angle potentials supplemented by dihedral terms to preserve tacticity, while nonbonded interactions follow a 12-6 LJ potential optimized via PSO. This model successfully reproduces PNIPAM's lower critical solution temperature (LCST) behavior, yielding coil-like conformations at 290 K and globule-like conformations at 320 K when simulated in the CG water model [56].

- *Dextran* functionalization is modeled using a carbohydrate CG model that is designed to provide a transferable representation for monosaccharides, employing a 2:1 mapping scheme. Specifically, β-D-glucose is mapped from atomistic simulations using the CHARMM, GLYCAM06j, and OPLS-AA force-fields, ensuring that the CG model closely replicates atomistic structural properties. Bond lengths and angles obtained from CG simulations match closely with atomistic MD results, highlighting the accuracy of the structural representation. Nonbonded interactions are optimized by simulations conducted under periodic boundary conditions at 300 K and 1 atm, effectively reproducing bulk properties such as density and heat of vaporization observed in atomistic models. Moreover, interactions with water are accurately captured by employing adaptive biasing force (ABF) [57, 58] methods to calculate solvation free energies. The robustness of this carbohydrate CG model is further confirmed through evaluations of surface tension and thermodynamic properties in both vacuum and aqueous environments, demonstrating its reliability in large-scale CG simulations. This model has also been successfully applied in previous studies to investigate precision glycopolymers, exploring how carbohydrate density and multivalency influence lectin recognition [53], as well as to investigate the impact of dextran grafting on the LCST behavior of PNIPAM [51].

These validated CG models enable reliable generation of diverse TS-BBP conformations across temperatures, structures, and functionalization states. We next describe the simulation protocol used to generate these conformations.

Simulation protocol

Individual BBPs are solvated in a cubic simulation box with dimensions of 250 Å along the x, y, and z directions, employing explicit 1-site CG water beads [6]. The box dimensions are carefully chosen to prevent any unwanted interactions between the BBP and its periodic images, and the total number of water beads ($\sim$200,000) is set to closely match the experimental density of water. Solvation is performed using the Packmol package developed by Martinez et al. [59]. CG MD simulations are conducted to examine polymer system behaviors at two distinct temperatures: 300 K (below LCST) and 320 K (above LCST). Each simulation is run for a total duration of 500 ns within the NPT ensemble, maintaining a constant pressure of 1 bar. All CG MD simulations are performed using the NAMD software package with a time step of 10 fs[60, 61]. Nonbonded interactions are evaluated using a cutoff radius of 12 Å. Temperature and pressure are maintained using the Langevin thermostat and barostat, respectively, ensuring precise thermodynamic control throughout the simulations. BBP coordinates are recorded every 50 ps, resulting in simulation trajectories containing 10,000 frames each. The first 200 ns (4,000 frames) are designated as an equilibration

phase, and the subsequent 300 ns (last 6,000 frames, indices 4,000-9,999) are used as the production run for subsequent analyses [6]. Details regarding the conventional analyses of BBP trajectories are presented in Section S1 in the SI.

Dataset construction

We construct a balanced 12-class dataset spanning three BBP architectures (Cake-like, Dumbbell-like, and Plus-like), two temperatures (300 K and 320 K), and two functionalization states (unfunctionalized and 100% dextran functionalized 5-tiered BBPs), as summarized in Table 1. The resulting classes are `Cake5-300`, `Cake5-320`, `Cake-D100-300`, `Cake-D100-320`, `Dumbbell5-300`, `Dumbbell5-320`, `Dumbbell-D100-300`, `Dumbbell-D100-320`, `Plus5-300`, `Plus5-320`, `Plus-D100-300`, and `Plus-D100-320`. Here, "`5`" indicates the five-tier distribution of PNIPAM side chain lengths along the backbone. "`D100`" indicates 100% dextran functionalization, and classes without "`D100`" correspond to unfunctionalized systems (0% dextran). The numeric suffix denotes the simulation temperature in Kelvin (e.g., 300 or 320). Each class is constructed by selecting 300 frames from the production phase of the simulations, specifically frames 4,000 to 9,999 at intervals of 20. Each frame is treated as one sample for the corresponding class, resulting in 300 samples per class and a total of 3,600 samples across the 12 classes, with strict balance maintained.

Table 1: Composition of the 12-class BBP conformation dataset. Each class contains 300 samples (3,600 samples in total).

Architecture	Dextran functionalization (%)	T (K)	Class label
Cake-like	0	300	`Cake5-300`
		320	`Cake5-320`
	100	300	`Cake-D100-300`
		320	`Cake-D100-320`
Dumbbell-like	0	300	`Dumbbell5-300`
		320	`Dumbbell5-320`
	100	300	`Dumbbell-D100-300`
		320	`Dumbbell-D100-320`
Plus-like	0	300	`Plus5-300`
		320	`Plus5-320`
	100	300	`Plus-D100-300`
		320	`Plus-D100-320`

2.2 Problem setup for BBP conformation recognition

Each BBP conformation obtained from the MD trajectories is a set of 3D beads, where each bead is represented by its Cartesian coordinates $p_j = (x_j, y_j, z_j)$ for $j = 1, 2, \ldots, N$. Collectively, the full BBP conformation at a given frame can be represented as a 3D point cloud $\mathcal{X} = \{p_1, p_2, \ldots, p_N\} \in \mathbb{R}^{N\times 3}$. We consider an equal number of beads (N) for all samples across classes. In practice, BBP systems may contain different numbers of beads due to variations in side chain length and functionalization state. While PointNet-based networks can, in principle, handle variable N through symmetric aggregation (see discussion in the next section), efficient mini-batch training on GPUs benefits from a fixed bead count. To achieve this, a common approach is to randomly upsample BBPs with fewer beads [62] or randomly downsample those with more beads [63]. In this paper, we adopt farthest point sampling (FPS) [64] for downsampling. Unlike *random* downsampling, which may discard beads arbitrarily and leave geometric gaps, FPS iteratively selects beads that are maximally spread across the 3D conformation, thereby better preserving the overall spatial distribution of the BBP structure. In our experiments, FPS also yielded better recognition performance compared to random downsampling, consistent with its stronger preservation of global shape geometry.

With bead counts equalized across all classes via FPS, each BBP conformation is thus represented as a fixed-size 3D point cloud that solely encodes the geometric shape of the BBP in space. This allows the problem of 3D BBP conformation recognition to be treated as a point cloud classification task, where the spatial structure of the beads is the primary feature used for recognizing the BBP class. Formally, the problem is defined by a function $f_{\boldsymbol{\Theta}} : \mathbb{R}^{N\times 3} \to \{1, 2, \ldots, 12\}$, where $\boldsymbol{\Theta}$ represents the learnable model parameters. Given a BBP conformation $\mathcal{X}$, the model predicts its class label as $\hat{c} = f_{\boldsymbol{\Theta}}(\mathcal{X})$. The goal is to learn $\boldsymbol{\Theta}$ such that $f_{\boldsymbol{\Theta}}$ minimizes the recognition error and generalizes well to unseen BBP shapes based on their 3D geometry.

3 METHODS

In this section, we first introduce the PointNet-based deep learning network used for 3D BBP conformation recognition. We then outline the evaluation metrics, training and hyperparameter tuning procedures, and the baseline methods used for performance comparison with traditional ML and voxel-based 3D CNN models. Finally, we describe the Grad-CAM-based explainability approach used to interpret the model's predictions.

3.1 PointNet-based learning for BBP conformation recognition

This paper proposes to use the PointNet-based network [36] to model the mapping function $f_{\boldsymbol{\Theta}}$. PointNet [36] is specifically designed to handle unordered point cloud data, such as 3D BBP structures from MD simulations. Our PointNet-based BBP recognition network, depicted in Figure 3, processes each BBP point cloud $\mathcal{X}$ through three major steps: (a) a transformation is applied to each bead in the cloud to learn bead-wise local features (Section 3.1.1), (b) these local features are aggregated across

all beads to obtain BBP-wise global shape features (Section 3.1.2), and (c) a classification head uses these global features to assign the BBP to one of the 12 classes (Section 3.1.3). The following subsections briefly detail each of these three steps.

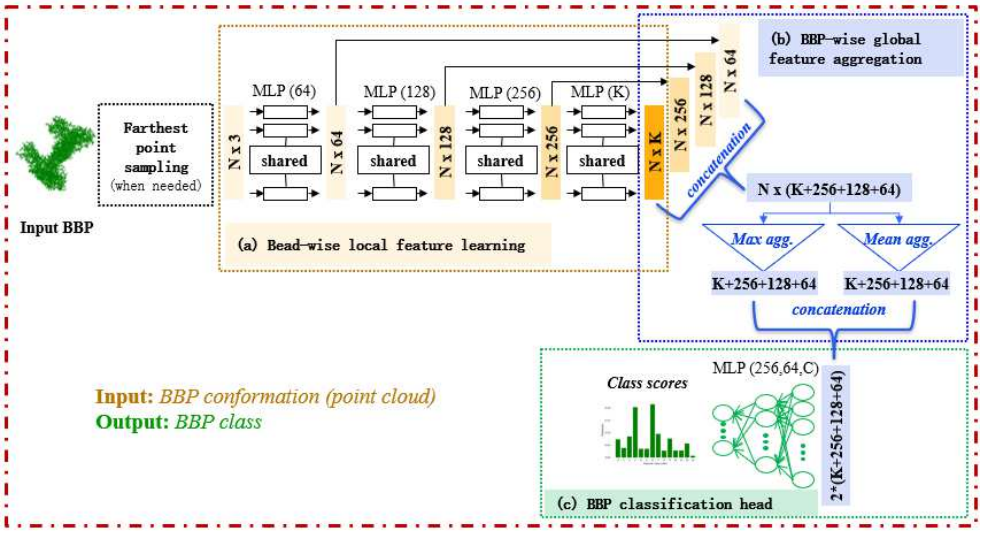


Fig. 3: **PointNet-based BBP conformation recognition network.** The input is a 3D point cloud of BBP bead coordinates, and the output is the predicted BBP conformation class among the 12 classes considered in this paper.

3.1.1 Mapping 3D bead coordinates to high-dimensional features

The first step of the PointNet-based BBP conformation recognition network (orange dashed box in Figure 3) involves the learning of bead-level local features from raw 3D coordinates. For each bead $\mathbf{p}_j$ in the BBP point cloud $\mathcal{X}$, a multilayer perceptron (MLP) transforms the 3D input into a K-dimensional feature vector $\mathbf{f}_j \in \mathbb{R}^K$. The MLP consists of L layers in total, where $l = 1, 2, ..., L$ indexes each individual layer, each followed by a nonlinear activation function σ (specified in Section 3.3). It operates as a sequence of transformations:

$$\mathbf{f}_j = \sigma_L(\mathbf{W}_L\sigma_{L-1}(\cdots\sigma_1(\mathbf{W}_1\mathbf{p}_j + b_1)\cdots) + b_L) \tag{1}$$

where $\mathbf{W}_l$ and $\mathbf{b}_l$ denote the weights and biases at layer l, respectively. The depth L and the final output dimension K control the richness of the per-bead structural representation and are treated as tunable hyperparameters in this paper. We use $L = 4$ layers with an output feature size of $K = 512$, selected based on validation performance. After applying the *shared* MLP to all beads, we obtain a set of per-bead feature vectors $\{\mathbf{f}_1, \ldots, \mathbf{f}_N\}$, which serve as the foundation for the subsequent BBP-level global feature aggregation step in Section 3.1.2. It is important to note that the

model does not assume any order among the feature vectors, which aligns with the unordered nature of the beads in the BBP point cloud.

3.1.2 Aggregating local bead features into global BBP descriptors

The second step of the network (blue dashed box in Figure 3) aggregates the bead-wise local features into a single global descriptor that characterizes the overall 3D BBP conformation. In the vanilla PointNet network[36], feature aggregation is performed solely on the final bead-level features after the last MLP layer. In this work, we improve the representation of BBP conformations by concatenating the outputs of all intermediate MLP layers, which allows the network to capture multi-scale geometric information at different levels of abstraction. Such multi-scale feature concatenation has proven effective in point cloud networks such as DGCNN [65]. For each BBP point-cloud sample $\mathcal{X}$, this process results in a final set of bead-wise feature vectors $\left\{\mathbf{f}'_j \,\middle|\, j = 1, \ldots, N\right\}$, where each $\mathbf{f}'_j \in \mathbb{R}^D$ and the total feature dimension D is the sum of feature dimensions from each contributing layer. In our optimized BBP recognition network (Figure 3), we use three intermediate layers with 64, 128, and 256 output channels, and a final layer with an output dimension of 512, resulting in $D = 64+128+256+512 = 960$. The resulting concatenated bead-wise feature map, of size $N \times 960$, serves as the pre-bottleneck representation before global pooling and is later used for the Grad-CAM-based bead-saliency analysis described in Section 3.5.1. To obtain a global descriptor of the BBP conformation that is invariant to bead order, we apply symmetric aggregation functions across all N beads. We apply both max pooling and mean pooling, following similar strategies employed in prior point cloud studies [66–68]. The final global conformation descriptor $\mathbf{g}$ is then obtained by concatenating the results of these two pooling operations.

$$\begin{aligned} \mathbf{g}_{\text{max}} &= \max_{j=1,\ldots,N} \mathbf{f}'_j, \\ \mathbf{g}_{\text{mean}} &= \frac{1}{N} \sum_{j=1}^{N} \mathbf{f}'_j, \\ \mathbf{g} &= [\mathbf{g}_{\text{max}} \| \mathbf{g}_{\text{mean}}] \in \mathbb{R}^{2D}. \end{aligned} \tag{2}$$

This combination of multi-scale feature concatenation and dual pooling operations produces a more informative representation of BBP conformations. Our ablation study (Section 4.2; Table 4) demonstrates that both design elements lead to performance gains compared to the vanilla PointNet [36].

3.1.3 Classification head for BBP conformation recognition

In the final step of the network (green dashed box in Figure 3), the global BBP descriptor $\mathbf{g} \in \mathbb{R}^{2D}$ is passed through a classification head consisting of another MLP. This MLP transforms the global feature vector $\mathbf{g}$ into a set of unnormalized scores, or logits, corresponding to each of the $C = 12$ BBP classes. The output is a logit vector $\mathbf{z} = (z_1, z_2, \ldots, z_C)$, which is converted into class probabilities using the softmax

function:

$$p_k = \frac{\exp(z_k)}{\sum_{k'=1}^{C} \exp(z_{k'})}, \quad \text{for} \quad k = 1, 2, \ldots, C \tag{3}$$

where p_k denotes the probability that the input BBP conformation belongs to class k. The predicted class $\hat{c}$ is then determined by selecting the class with the highest probability:

$$\hat{c} = \arg\max_k p_k \tag{4}$$

3.2 Performance metrics

We select *accuracy* and *F1 score* as performance metrics to evaluate our classification model. Accuracy is the ratio of correctly classified BBP conformations to the total number of BBP conformations in the test set:

$$Acc = \frac{1}{T} \sum_{i=1}^{T} \mathbb{I}(\hat{c}_i = c_i) \tag{5}$$

where T is the total number of samples in the test dataset, $\hat{c}_i$ is the predicted class label, c_i is the true class label, and $\mathbb{I}(\hat{c}_i = c_i)$ is the indicator function. Indicator function $\mathbb{I}(\hat{c}_i = c_i)$ returns 1 if the predicted class $\hat{c}_i$ matches the true class c_i (i.e., correct prediction), and 0 otherwise. Higher accuracy indicates better classification performance. In this paper, we compute accuracy using `accuracy_score` from the `sklearn.metrics` module. We also report the F1 score to provide a balanced measure of the model's precision and recall:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

where precision is the ratio of correctly predicted positive observations to all predicted positives, and recall is the ratio of correctly predicted positives to all actual positives. A higher F1 score indicates a better balance between precision and recall. In this paper, F1 score is computed using `f1_score` from the `sklearn.metrics` module, with a *weighted* average.

3.3 Training, hyperparameter tuning, and testing details

The dataset contains 3,600 BBP conformations evenly distributed across 12 classes (300 samples per class). We apply *stratified partitioning* to maintain class balance, allocating 80% of the data for training and validation, and reserving the remaining 20% as a held-out test set. We implement this partition using the `StratifiedShuffleSplit` utility from `scikit-learn`. We train all models to minimize the categorical cross-entropy loss, $\mathcal{L}_{CE}(\mathbf{p}, \mathbf{y}) = -\sum_{k=1}^{C} y_k \log(p_k)$, where $\mathbf{p}$ is the predicted class probability vector obtained after softmax, and $\mathbf{y}$ is the one-hot encoded ground truth label. Optimization is performed using the Adam optimizer [69] with a learning rate $\eta = 0.0005$, first- and second-moment decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$, respectively, and a numerical stability $\epsilon = 10^{-8}$. We use a mini-batch size

of 32 for all training runs. We set the number of training epochs to $E = 200$ unless otherwise specified.

Algorithm 1 BBP conformation recognition: training, hyperparameter tuning, and evaluation pipeline

Require: Dataset $\mathcal{D} = \{(\mathcal{X}^{(i)}, c^{(i)})\}_{i=1}^{3,600}$, number of classes $C = 12$.

Stratify-split $\mathcal{D}$ into training set $\mathcal{D}_{\text{train}}$ (80%) and test set $\mathcal{D}_{\text{test}}$ (20%).
Define candidate hyperparameters $\mathcal{H} = \{(L, K)\}$ using grid search[70].
for each $(L, K) \in \mathcal{H}$ **do**
    for $r = 1$ to 5 independent random seeds **do**
        Perform 5-fold stratified cross-validation on $\mathcal{D}_{\text{train}}$:
        for each fold $m = 1, \ldots, 5$ **do**
            Split $\mathcal{D}_{\text{train}}$ into training subset $\mathcal{D}_{\text{train}}^{(m)}$ and validation subset $\mathcal{D}_{\text{val}}^{(m)}$.
            Initialize PointNet-based model with (L, K).
            Train the model on $\mathcal{D}_{\text{train}}^{(m)}$.
            Evaluate on $\mathcal{D}_{\text{val}}^{(m)}$; record $\text{Acc}^{(m,r)}$ (Eq. 5) and $\text{F1}^{(m,r)}$ (Eq. 6).
        end for
    end for
    Compute average cross-validation metrics:

$$\overline{\text{Acc}} = \frac{1}{25} \sum_{r=1}^{5} \sum_{m=1}^{5} \text{Acc}^{(m,r)},$$

$$\overline{\text{F1}} = \frac{1}{25} \sum_{r=1}^{5} \sum_{m=1}^{5} \text{F1}^{(m,r)}.$$

end for
Select (L^*, K^*) that maximizes $\overline{\text{Acc}}$ and $\overline{\text{F1}}$.
for $s = 1$ to 10 independent runs **do**
    Initialize final PointNet-based proposed model with (L^*, K^*) (Figure 3).
    Train on full $\mathcal{D}_{\text{train}}$.
    Evaluate model on $\mathcal{D}_{\text{test}}$.
    Record $\text{Acc}_{\text{test}}^{(s)}$ and $\text{F1}_{\text{test}}^{(s)}$.
end for
return: mean $\pm$ standard deviation for Acc_{test}, F1_{test}

We tune two previously introduced model hyperparameters - the number of MLP layers L used for bead-wise feature learning, and the output feature dimension K of the final MLP layer prior to feature concatenation. Hyperparameter tuning is performed via a stratified 5-fold cross-validation procedure, repeated across 5 independent random seeds to ensure robustness. We select the optimal (L^*, K^*) pair based on the highest mean validation accuracy (Eq. 5) and F1 score (Eq. 6) across cross-validation

folds and seeds. We use ReLU activations with Batch Normalization [71] across hidden layers, and apply a dropout rate of 0.6 after each fully connected block in the classification head.

Once the best hyperparameters are selected, we retrain the final model from scratch using the entire training set (2,880 samples) and evaluate its performance on the held-out test set (720 samples). We repeat this final training and evaluation 10 times with different random initializations to assess the generalization performance of the model. The test set performance obtained from these runs is summarized in Section 4.1. All models are implemented in PyTorch[72] and trained on a workstation equipped with an NVIDIA GeForce RTX GPU and 128 GB of system memory. The full pipeline for training, hyperparameter tuning, and evaluation is summarized in Algorithm 1.

3.4 Baseline methods

To evaluate the performance of the PointNet-based BBP recognition model, we compare it against both traditional ML classifiers and a voxel-based 3D CNN. For traditional ML benchmarks, we consider Random Forest (RF)[73], Light Gradient Boosting Machine (LightGBM)[74], and Support Vector Machines (SVM) with both radial basis function (RBF) and polynomial (poly) kernels[75]. These widely used classifiers provide reference comparisons across tree-based and kernel-based approaches. As these methods require structured feature vectors as input, we voxelize each BBP point cloud at a resolution of 50^3. The resulting voxel grids, each containing 125,000 binary occupancy values, are then flattened into one-dimensional feature vectors and used to train and evaluate these traditional ML models.

For the 3D CNN baseline, we start with a 50^3 voxel resolution, consistent with Joshi et al. [6], and additionally evaluate reduced resolutions of 30^3 and 10^3 to investigate the trade-off between voxel resolution and classification performance. To maintain valid spatial representations after convolution and pooling, we adapt the network depth to the input resolution: four convolution-pooling blocks at 50^3, three at 30^3, and two at 10^3. In each case, we include as many blocks as the resolution allows before the voxel grid becomes too small for further convolutions. This ensures that lower-resolution models remain well-posed while still following the same design principles as the higher-resolution network, allowing a fair comparison across voxel sizes. An overview of the architectures, input formats, hyperparameters, and optimization settings for the baseline methods is given in Table 2. To maintain consistency, we apply the same final training and testing pipeline formalized in Algorithm 1 to all baseline models.

3.5 Bead-level attribution of BBP classification decisions

This subsection presents the explainability approach used to explain BBP-level predictions back to individual beads. We apply a Grad-CAM-based [42] attribution method to estimate bead-level contributions and a deletion-based perturbation test to evaluate the resulting explanations.

Table 2: Architectures, key hyperparameters, and optimization settings of the baseline models.

Model	Input shape	Key architecture/ Parameters	Optimization and schedule	Package/Library
RF [73]	(B, 125,000)	n_estimators = 100	Package default	scikit-learn [76]
LightGBM [74]	(B, 125,000)	n_estimators = 100 learning_rate = 0.1	Package default	LightGBM
SVM (`rbf`) [75]	(B, 125,000)	kernel = 'rbf', C = 1.0, gamma = 'scale', probability = False	Package default	scikit-learn [76]
SVM (`poly`) [75]	(B, 125,000)	kernel = 'poly', degree = 3, coef0 = 1, C = 1.0, gamma = 'scale', probability = False	Package default	scikit-learn [76]
3D-CNN [6] @ 10^3	(B, 1, 10, 10, 10)	Conv blocks: (1 → 32 → 512), valid 3^3 kernels, GroupNorm, ReLU, MaxPool3d(k=2); AdaptiveMaxPool3d(1); MLP head 512 → 128 → 64 → 12 with Dropout(0.2)	Adam [69], lr=0.0005, batch size 32, optimized at 50 epochs	PyTorch [72]
3D-CNN [6] @ 30^3	(B, 1, 30, 30, 30)	Conv blocks: (1 → 32 → 64 → 512), valid 3^3 kernels, GroupNorm, ReLU, MaxPool3d(k=2); AdaptiveMaxPool3d(1); MLP head 512 → 128 → 64 → 12 with Dropout(0.2)	Adam [69], lr=0.0005, batch size 32, optimized at 100 epochs	PyTorch [72]
3D-CNN [6] @ 50^3	(B, 1, 50, 50, 50)	Conv blocks: (1 → 32 → 64 → 128 → 512), valid 3^3 kernels, GroupNorm, ReLU, MaxPool3d(k=2); AdaptiveMaxPool3d(1); MLP head 512 → 128 → 64 → 12 with Dropout(0.2)	Adam [69], lr=0.0005, batch size 32, optimized at 150 epochs	PyTorch [72]

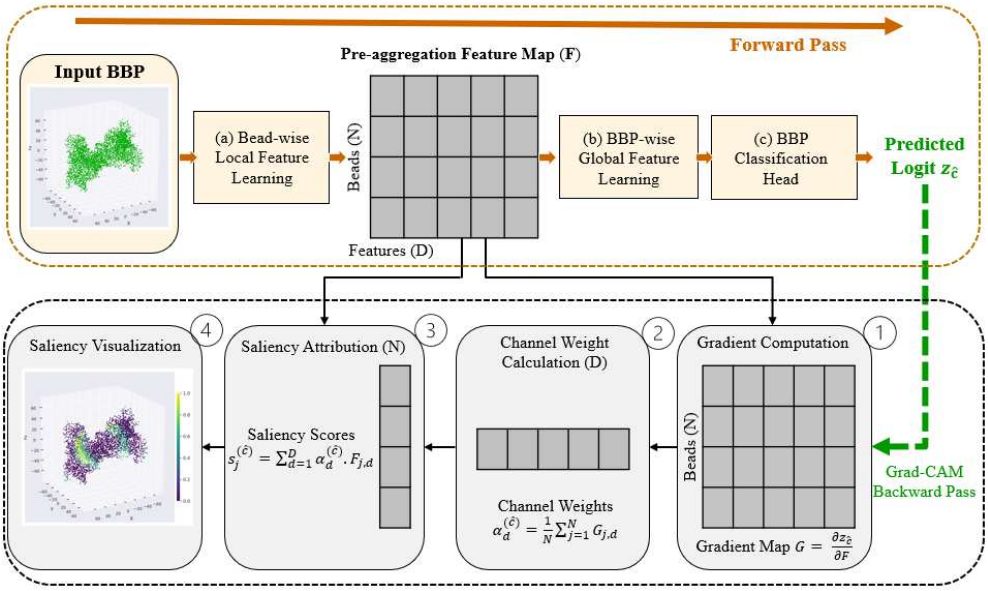


Fig. 4: Schematic of the Grad-CAM-based bead-saliency analysis for the PointNet-based BBP recognition model. The top panel recaps the forward classification pipeline from Figure 3, and the bottom panel shows the backward attribution procedure used to generate bead-level saliency maps.

3.5.1 Utilization of Grad-CAM for point-cloud bead saliency

For each input BBP conformation, we explain the model's predicted class $\hat{c}$ by computing bead-level saliency scores that quantify how strongly each bead contributes to the corresponding predicted logit $z_{\hat{c}}$. As introduced earlier, after the shared MLP stack and the multi-level feature concatenation, each bead j is represented by a learned feature vector $\mathbf{f}_j^{'} \in \mathbb{R}^D$. These bead-wise feature vectors across all N beads form the pre-aggregation feature map $\mathbf{F} \in \mathbb{R}^{N \times D}$, where the j-th row of $\mathbf{F}$ is $\mathbf{f}_j^{'}$. Thus, $\mathbf{F}$ is simply the pre-bottleneck bead-wise feature representation as described in Section 3.1.2. The bottom panel of Figure 4 summarizes the overall Grad-CAM workflow used in this paper. It first computes the gradient of the predicted logit with respect to this feature map as $\mathbf{G} = \frac{\partial z_{\hat{c}}}{\partial \mathbf{F}} \in \mathbb{R}^{N \times D}$.

In the original Grad-CAM formulation for images [42], the feature map is defined over a 2D spatial grid, and one importance weight is computed for each channel by averaging gradients over all spatial locations. In our setting, the *spatial locations* are the beads themselves. Therefore, we compute importance weight per feature channel by averaging the gradients over the bead dimension:

$$\alpha_d^{(\hat{c})} = \frac{1}{N} \sum_{j=1}^{N} G_{j,d} = \frac{1}{N} \sum_{j=1}^{N} \frac{\partial z_{\hat{c}}}{\partial F_{j,d}}, \quad d = 1, \ldots, D. \tag{7}$$

Using these channel weights, a raw attribution score is computed for each bead as the weighted sum of its features:

$$s_j^{(\hat{c})} = \sum_{d=1}^{D} \alpha_d^{(\hat{c})} F_{j,d}, \quad j = 1, \ldots, N. \tag{8}$$

Following the standard Grad-CAM principle [42], we retain only positive evidence for the predicted class by applying a rectified linear operation $\hat{s}_j^{(\hat{c})} = \max\left(0, s_j^{(\hat{c})}\right)$. To visualize the bead-saliency heatmap for a given BBP sample, we normalize the raw scores across all beads to the interval $[0, 1]$:

$$\tilde{s}_j^{(\hat{c})} = \frac{\hat{s}_j^{(\hat{c})} - \min_k \hat{s}_k^{(\hat{c})}}{\max_k \hat{s}_k^{(\hat{c})} - \min_k \hat{s}_k^{(\hat{c})} + \delta}. \tag{9}$$

Here, k indexes all beads in the selected sample and the small constant δ is added to the denominator to avoid division by zero in cases where all bead scores are equal. The resulting normalized bead-level saliency scores $\tilde{s}_j^{(\hat{c})}$ are then used to visualize 3D attribution heatmaps and to support the quantitative faithfulness analysis described next.

3.5.2 Deletion-based perturbation test

To quantitatively assess the faithfulness of model explanations, we perform a deletion-based perturbation test in a manner similar to the perturbation-based evaluation of Levi and Gilboa [77]. For each test BBP conformation $\mathcal{X}$, we first compute Grad-CAM bead-level saliency scores $\tilde{s}_j^{(\hat{c})}$ for the predicted class $\hat{c}$, rank the beads in descending order based on these scores, and remove the top $p\%$ of beads with the highest saliency. This results in a reduced point cloud $\mathcal{X}_{\text{del}}(p)$ that is then evaluated by the unchanged model. For each of the 12 BBP classes, we compute the classification accuracy over its test samples as a function of the removal percentage p. As a baseline, we compare this saliency-guided deletion procedure against random bead removal at the same deletion ratios. A larger reduction in accuracy following bead removal suggests that the removed beads contribute more strongly to the model's prediction.

4 RESULTS AND DISCUSSION

In this section, we present the results and associated findings. We first compare the proposed PointNet-based classifier with traditional ML baselines and voxel-based 3D CNNs on the held-out BBP test set. We then evaluate the architectural design choices of the PointNet variant used in this paper via an ablation study and examine model explainability using Grad-CAM-based bead-saliency maps with a deletion-based perturbation test. Finally, we apply the trained 12-class model, without further training, to additional BBP systems outside the training label set and examine how these systems are assigned within the learned class space. In addition to the ML-based results

presented in this section, we performed conventional analyses of the BBP trajectories following our previous study [6]; the detailed methodology and results of these conventional analyses are provided in Sections S1 and S2 of the SI.

4.1 Performance evaluation on unseen BBP conformations

We evaluate the final PointNet-based model on the held-out BBP test set and compare its performance against traditional ML and 3D CNN baselines described in Section 3.4. Table 3 summarizes test performance, averaged over 10 independent runs for each method (*mean* ± *std*). Among the traditional ML methods, SVM [75] with a polynomial kernel achieves the best performance with an accuracy of 80.14% and an F1 score of 0.8005, outperforming the RBF kernel and tree-based models (RF [73] and LightGBM [74]). Notably, all traditional ML approaches depend on converting 3D BBP conformation geometries into structured, fixed-dimensional representations, through voxelization or manual feature engineering, which incurs preprocessing overhead.

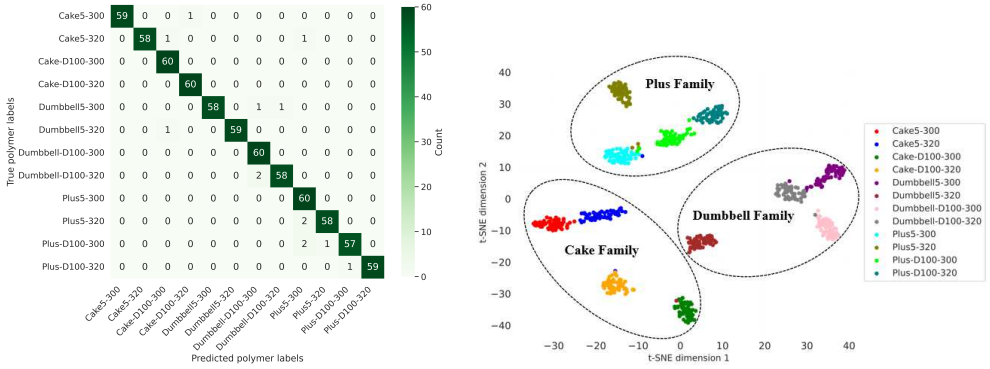
Table 3: Comparison of classification performance, voxelization preprocessing time, and model training time across baseline methods and the proposed PointNet-based approach. Results are based on 10 independent runs and reported as *mean* ± *standard deviation*.

Models	Voxelization time (s)	Mean model training time (s)	Accuracy (%) ↑	F1 score ↑
Voxelized (50^3) + RF[73]	340	2	46.93 ± 0.99	0.4688 ± 0.0095
Voxelized (50^3) + LightGBM[74]	339	9	59.86 ± 0.00	0.5997 ± 0.0000
Voxelized (50^3) + SVM (`rbf`)[75]	341	299	68.06 ± 0.00	0.6796 ± 0.0000
Voxelized (50^3) + SVM (`poly`)[75]	341	295	80.14 ± 0.00	0.8005 ± 0.0000
Voxelized (10^3) + 3D-CNN[6]	247	15	48.06 ± 1.94	0.4804 ± 0.0202
Voxelized (30^3) + 3D-CNN[6]	261	60	81.29 ± 3.56	0.8096 ± 0.0410
Voxelized (50^3) + 3D-CNN[6]	339	442	91.44 ± 2.96	0.9134 ± 0.0326
PointNet-based (*this paper*)	0*	288	94.81 ± 3.33	0.9471 ± 0.0360

* Voxelization is not required for point cloud-based methods, so preprocessing time is zero. In this work, FPS is used to equalize bead counts across classes, which takes around 100 s for the entire BBP dataset.

The 3D CNN approach improves performance with increasing voxel resolution. At 50^3 resolution, the 3D CNN model of Joshi et al. [6] achieves an accuracy of 91.44% and an F1 score of 0.9134. Lower voxel resolutions (30^3 and 10^3) result in substantially worse recognition accuracy, indicating the importance of fine spatial representation in preserving the 3D shape information. In contrast, the proposed PointNet-based model achieves a higher accuracy of 94.81% ± 3.33% and an F1 score of 0.9471 ± 0.0360. Moreover, it completely eliminates the voxelization overhead and, as shown in Table 3, attains a shorter model training time than the 50^3 3D CNN.

To further investigate the model behavior on the unseen BBP test dataset, Figure 5(a) presents the confusion matrix from the best-performing run. In the illustrated run, the model correctly classifies most samples across all 12 classes, with only a handful of misclassifications. Importantly, these errors occur predominantly between



(a) (b)

Fig. 5: Results on the BBP test set obtained with the best-performing run of the proposed PointNet-based model: *(a)* confusion matrix, with color intensity indicating the number of BBP conformations, and *(b)* t-SNE visualization of the learned intermediate feature space.

closely related variants within the same architectural family rather than across fundamentally different shapes. For instance, two samples from `Plus{D100{300` were predicted as `Plus5{300`, and one sample from `Dumbbell5{300` was misclassified as `Dumbbell{D100{320`. Such cases highlight that misclassifications are confined within local neighborhoods of BBP variants that share the same global morphology but differ in subtle thermoresponsive or functionalization features. This error structure is consistent with the conventional CG-MD analysis of the same systems reported in the SI (Tables S1 and S2). Across every system in those tables, SASA separates the three architectures without overlap, ordering them as Plus ($1.15 - 1.35 \times 10^5$ Å^2) < Cake ($1.37 - 1.46 \times 10^5$ Å^2) < Dumbbell ($1.56 - 1.79 \times 10^5$ Å^2). The within-family variations associated with temperature and dextran functionalization are smaller and, unlike the family separations, are not consistent in sign across architectures, since raising the temperature increases the asphericity of the 5-tiered Cake and Plus systems ($0.37 \rightarrow 0.41$ and $0.34 \rightarrow 0.38$) but decreases that of the Dumbbell ($0.60 \rightarrow 0.58$). Because the conventional descriptors and the proposed classifier read the same underlying geometry, the families should be separated clearly while the finer within-family distinctions are the harder ones, and this is where the majority of the misclassifications in Figure 5(a) fall. In support of this observation, the t-SNE visualization of the learned feature space in Figure 5(b) shows well-separated clusters for all classes, with related variants within the same global shape grouped closer together, consistent with both the confusion matrix and the descriptor-based family separation.

Alongside the baseline comparison in Table 3, these results show that the proposed PointNet-based classifier attains the highest BBP conformation recognition accuracy among all methods compared, and that it recognizes the conformations in a manner consistent with their known structural differences. Because it operates directly on raw

MD point clouds and avoids the voxelization that the baseline methods require, it is scalable and flexible for analyzing large or continuously generated BBP datasets.

4.2 Ablation study of the PointNet variant used in this work

To justify the design choices of the PointNet-based variant applied in this work, we perform an ablation study only on PointNet-based models, using the same optimized setting and evaluating them on the test set without further model tuning. As shown in Table 4, the vanilla PointNet (Variant 1), which uses only the final layer features with max pooling, reaches 92.3% accuracy, while replacing max pooling with mean (Variant 2) or sum (Variant 3) degrades performance considerably. Introducing multi-level feature concatenation improves accuracy across all aggregation choices, with the best results obtained when combining max and mean pooling, which yields 94.8% accuracy and 0.9471 F1. These results indicate that both multi-scale feature concatenation and the dual pooling strategy are beneficial for enhancing the recognition of BBP conformations.

Table 4: Ablation study of multi-level feature concatenation and symmetric aggregation (max, mean, sum) on classification performance.

Models	Multi-level features	Symmetric feature aggregation			Accuracy (%) ↑	F1 score ↑
		Max pool	Mean pool	Sum pool		
This paper	✓	✓	✓	×	94.81	0.9471
Variant 1*	×	✓	×	×	92.33	0.9234
Variant 2	×	×	✓	×	86.15	0.8457
Variant 3	×	×	×	✓	85.29	0.8367
Variant 4	✓	✓	×	×	93.10	0.9307
Variant 5	✓	×	✓	×	88.35	0.8689
Variant 6	✓	×	×	✓	87.90	0.8627

* Variant 1 corresponds to the vanilla PointNet [36] architecture.

4.3 Grad-CAM visualization and faithfulness analysis

We achieve high BBP conformation recognition accuracy using our PointNet-based network (Section 4.1), but the reasoning behind its predictions is not directly observable from the predicted labels alone. We now use the Grad-CAM procedure described in Section 3.5 to visualize bead-level attributions.

Representative Grad-CAM visualizations for correctly classified test samples are shown in Figure 6, where we randomly include 12 samples, one from each of the 12 classes. The first row (a-d) corresponds to the Cake family, the second row (e-h) to the Dumbbell family, and the third row (i-l) to the Plus family. In each case, we visualize the Grad-CAM heatmap by plotting the original bead coordinates $\{p_j\}$ in 3D and color-coding each bead by $\tilde{s}_j^{(\hat{c})}$, where warmer colors indicate beads that provide stronger positive evidence for the predicted class.

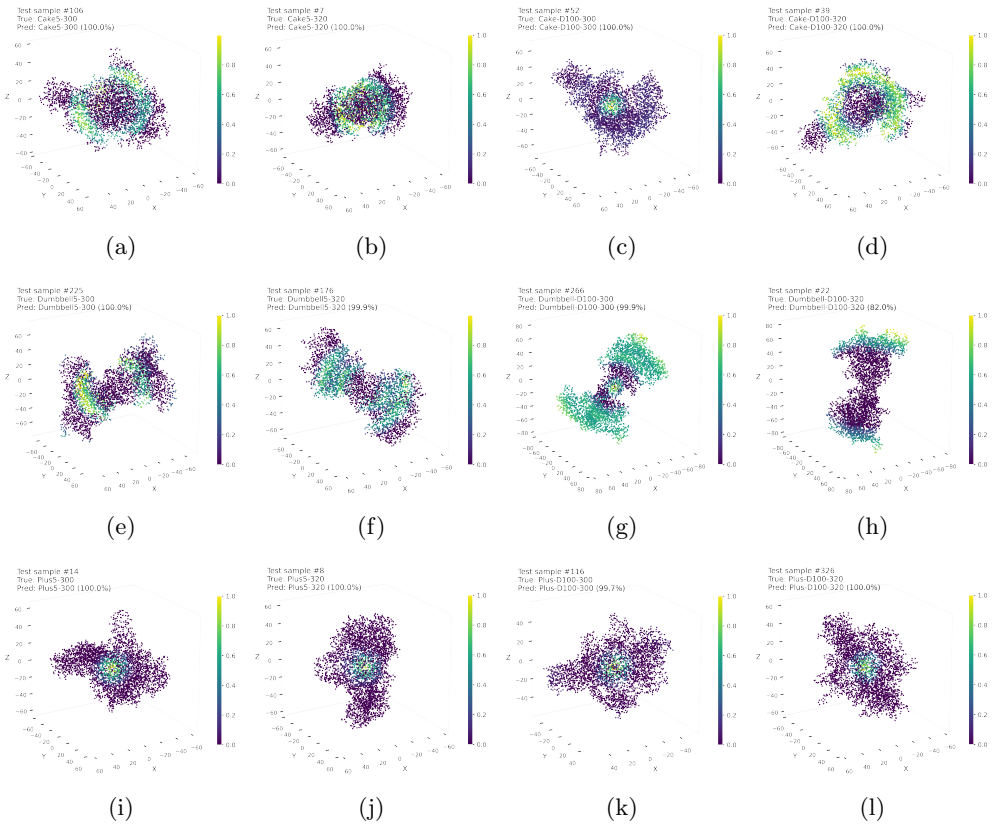


Fig. 6: Grad-CAM bead-saliency heatmaps for 12 correctly classified BBP test samples (one per class). Samples are arranged by architectural family: Cake (top row), Dumbbell (middle row), and Plus (bottom row). Warmer colors indicate stronger positive evidence for the predicted class.

Across the correctly classified examples of Figure 6, the Grad-CAM attributions concentrate on subsets of beads that capture the most shape-defining regions of each conformation. (*i*) For the Cake family, the highest-saliency beads typically align with the broad disc-like mass distribution that characterizes this topology. (*ii*) For the Dumbbell family, the strongest attributions are commonly located on the two terminal lobes and, in many cases, the narrowed bridge between them, indicating that lobe separation and neck geometry are important discriminative features. (*iii*) For the Plus family, saliency is often concentrated near the central junction where multiple arms meet, with attribution extending partially along the arms when those features are more pronounced. These attribution patterns can be interpreted in the context of the conventional CG-MD descriptors. The Dumbbell saliency, concentrated on the two terminal lobes and the connecting neck, falls on the geometry responsible for the family's high asphericity ($\approx$ 0.58–0.66 across the four trained Dumbbell classes, the

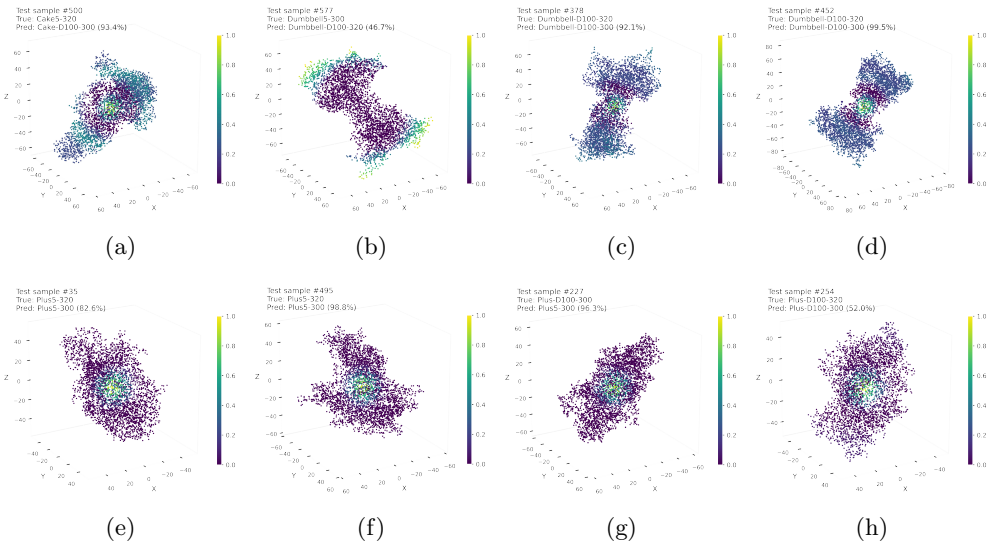


Fig. 7: Grad-CAM bead-saliency heatmaps for representative misclassified BBP test samples. Warmer colors indicate stronger positive evidence for the predicted class.

largest of the three families in Tables S1 and S2), since lobe separation and neck constriction are the primary geometric features that determine the asphericity. In contrast, the Cake saliency, spread over the broad disc-like mass, and the Plus saliency, localized at the central junction where the arms meet, highlight the compact, more isotropic mass distributions of these families, consistent with their lower asphericity ($0.37 - 0.45$ and $0.34 - 0.43$, respectively). This more compact, isotropic organization also results in the smaller SASA that separates both families from the Dumbbell across all systems in Tables S1 and S2. These attribution patterns suggest that the PointNet-based network assigns high attribution to geometrically discriminative regions that are qualitatively consistent with the defining morphology of each architecture.

Figure 7 presents example misclassified samples. Notably, these misclassifications are almost always assigned to closely related classes within the same architectural family, typically differing only in temperature or dextran functionalization rather than in the overall topology. For example, the sample in Figure 7(a), originally from `Cake5-320`, is predicted as `Cake-D100-300`, while the sample in Figure 7(e), originally from `Plus5-320`, is predicted as `Plus5-300`. This pattern indicates that the dominant failure mode of the model is not confusion between the major BBP shape families (Cake, Dumbbell, and Plus), but rather difficulty distinguishing finer geometric variations associated with thermally induced structural changes or functionalization.

Next, we apply the perturbation procedure described in Section 3.5.2 to assess whether these Grad-CAM-based explanations are faithful to the model's decision process. Figure 8 shows the accuracy-deletion curves obtained by progressively removing beads either in descending order of Grad-CAM saliency (blue) or at random (red).

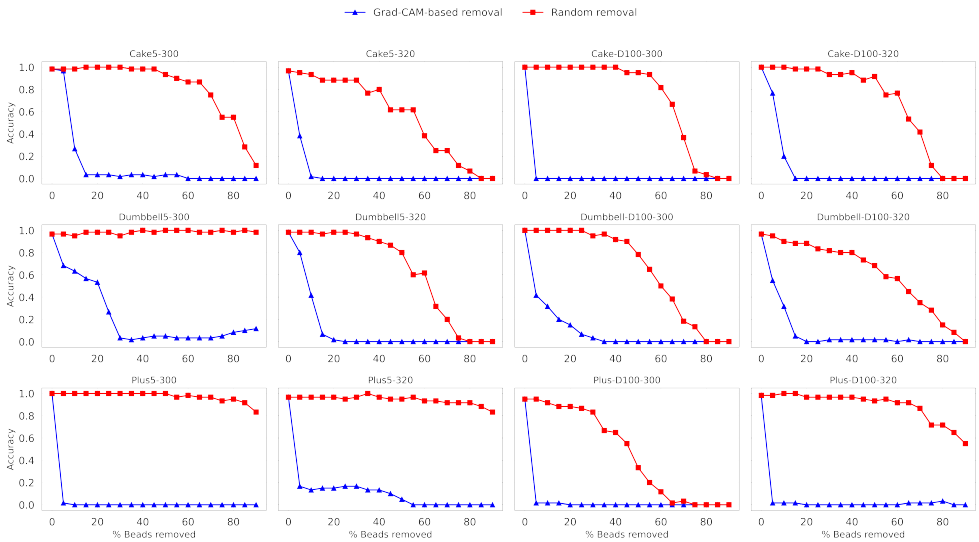


Fig. 8: Deletion-based perturbation test for Grad-CAM faithfulness across all 12 BBP classes. Accuracy is reported after removing an increasing fraction of beads ranked by Grad-CAM saliency (blue) or by random order (red); faster accuracy drop under saliency-based removal indicates more faithful attributions.

Across all 12 classes, removing the most highly attributed beads results in a substantially faster decline in accuracy than random bead removal at the same deletion levels. This consistent gap indicates that the Grad-CAM method is identifying beads that contribute disproportionately to correct recognition. In other words, the attributed regions are functionally important to the model's predictions.

4.4 Projection of additional BBP systems onto the learned class space

To examine how the trained 12-class model responds to BBP systems outside its training label set, we perform inference on polymer conformations drawn from additional systems that were not included in model development or testing. These 28 additional systems extend beyond the training distribution along two axes: tier count (3-, 7-, and 9-tier, whereas the model was trained only on 5-tier systems; the 3-tier Plus systems were unavailable and are not included) and functionalization degree (33% and 50%, whereas training used only 0% and 100%). Because the classifier is restricted to the 12 learned output classes, each additional conformation is necessarily assigned to one of these classes. We therefore record the frequency with which the conformations from each additional system are assigned to the learned classes, as summarized in Table 5. The following observations emerge from these projections.

First, the architectural family of these additional BBP systems is largely preserved. For most of these additional systems, the majority of conformations (66–100%) are

Table 5: Predicted distributions over the 12 trained classes when the model is applied to BBP conformations from additional classes not included in training or testing.

Additional class	Cake5-300	Cake5-320	Cake-D100-300	Cake-D100-320	Dumbbell5-300	Dumbbell5-320	Dumbbell-D100-300	Dumbbell-D100-320	Plus5-300	Plus5-320	Plus-D100-300	Plus-D100-320
Cake3-300	0	243	44	7	0	2	0	0	3	0	0	1
Cake3-320	21	144	8	124	0	0	0	0	0	0	0	3
Cake7-300	47	11	128	47	0	0	0	0	0	0	10	57
Cake7-320	0	191	14	0	3	39	0	0	42	0	10	1
Cake9-300	0	3	1	296	0	0	0	0	0	0	0	0
Cake9-320	0	204	68	1	0	1	0	0	21	0	3	2
Cake-D33-300	55	136	18	86	3	2	0	0	0	0	0	0
Cake-D33-320	0	0	0	197	103	0	0	0	0	0	0	0
Cake-D50-300	0	157	101	41	0	1	0	0	0	0	0	0
Cake-D50-320	18	148	0	134	0	0	0	0	0	0	0	0
Dumbbell3-300	0	0	0	0	172	0	128	0	0	0	0	0
Dumbbell3-320	41	0	0	0	109	55	1	94	0	0	0	0
Dumbbell7-300	0	0	0	0	57	240	1	2	0	0	0	0
Dumbbell7-320	0	94	0	2	0	201	0	3	0	0	0	0
Dumbbell9-300	0	0	4	35	119	140	1	1	0	0	0	0
Dumbbell9-320	256	6	0	26	0	9	0	0	0	0	0	0
Dumbbell-D33-300	0	0	0	0	269	0	28	3	0	0	0	0
Dumbbell-D33-320	0	0	0	0	295	0	5	0	0	0	0	0
Dumbbell-D50-300	0	0	0	0	60	0	120	120	0	0	0	0
Dumbbell-D50-320	0	0	0	0	86	11	56	147	0	0	0	0
Plus7-300	0	23	27	0	0	0	0	0	178	0	21	51
Plus7-320	0	0	0	0	0	0	0	0	170	1	0	129
Plus9-300	1	49	2	0	0	0	0	0	178	24	2	44
Plus9-320	27	17	0	0	0	0	0	0	3	246	2	5
Plus-D33-300	0	0	0	0	0	0	0	0	0	48	0	252
Plus-D33-320	0	0	0	0	0	0	0	0	1	96	2	201
Plus-D50-300	0	0	0	0	0	0	0	0	0	1	0	299
Plus-D50-320	0	0	0	0	0	0	0	0	293	0	7	0

assigned to trained classes of the same family (cake systems to cake classes, dumbbell systems to dumbbell classes, and plus systems to plus classes). This holds even though the additional systems carry tier counts absent from training, indicating that the learned representation relies on family-level shape. This behavior is consistent with the conventional descriptors of Table S1 and S2. Across every tier, temperature, and functionalization state in these tables, SASA separates the three families without overlap, and tier count shifts it only within a family. For the unfunctionalized dumbbells at 300K, SASA falls from 1.79×10^5 to 1.55×10^5 Å^2 between the 3- and 9-tier systems, still well above the cake and plus ranges. The classifier therefore generalizes by relying on the similar global-shape signatures that distinguish the families in the conventional analysis.

Second, partially dextran-functionalized systems, including the D33 and D50 variants, are often distributed across both unfunctionalized and fully functionalized trained classes, although the assignment patterns vary with architecture and temperature. This intermediate placement is consistent with the conventional descriptors in

Table S2, where the partially functionalized systems take values between the unfunctionalized (Dex-0) and fully functionalized (Dex-100) endpoints. The cakes illustrate this clearly, as their SASA decreases monotonically with dextran content (at 300 K, $1.440 \times 10^5 \rightarrow 1.415 \times 10^5 \rightarrow 1.395 \times 10^5 \rightarrow 1.377 \times 10^5$ Å^2 for Dex-0/33/50/100), so a geometry lying between the two endpoints in descriptor space is, reasonably, projected onto both endpoint classes. More broadly, because the within-family axes of temperature and functionalization correspond to smaller descriptor changes than those separating the families, it is along these finer axes, rather than across families, that the model's assignments spread.

Third, for the Plus family the model treats partially and fully grafted systems as nearly the same shape. Three of the four partially functionalized Plus systems (Plus-D33-300, Plus-D33-320, and Plus-D50-300) are assigned predominantly to a fully functionalized class, Plus-D100-320 (252, 201, and 299 of 300 conformations, respectively). This is consistent with the descriptor trends in Table S2, where the Plus asphericity at 300 K rises sharply from 0.34 (Dex-0) to 0.44–0.45 (Dex-33/Dex-50) and then no longer increases at Dex-100 (0.43); most of the functionalization-induced shape change is therefore already present by 33% grafting, so a partially grafted Plus BBP is geometrically close to the fully grafted one. The model, comparing whole conformations rather than a single scalar, reaches a similar conclusion by mapping these systems onto the fully functionalized class.

Across all these observations, the proposed PointNet-based classifier organizes BBP systems outside its training label set in a physically coherent way. It preserves architectural family across tier and functionalization variations, and its finer within-family assignments spread along the axes where the descriptors change least.

5 CONCLUSION

In this work, we showed that an enhanced PointNet-based point-cloud model can classify temperature-sensitive BBP conformations directly from raw MD point clouds with high accuracy, without the need for voxelization or hand-crafted geometric features. Against a suite of baselines, including tree-based ensemble methods, kernel-based support vector machines, and a high-resolution (50^3) 3D CNN, the model achieved the best overall performance on the 12-class BBP dataset, with a test accuracy of 94.81% and an F1 score of 0.9471. These results underscore the advantage of learning directly from point-cloud representations of BBP conformations. The ablation study showed that multi-level feature concatenation combined with dual pooling improves performance by providing a more informative representation of BBP conformations. The confusion matrix and feature-space visualization further revealed strong separation among the 12 classes, with only a few misclassifications between closely related BBP variants within the same shape family. Beyond classification, we demonstrated how Grad-CAM-based explainability provides bead-level attributions that localize geometrically discriminative regions of the input conformations. The deletion-based perturbation test provided quantitative evidence that highly attributed beads are more influential to model performance than randomly selected beads. Overall, the proposed

pipeline achieves the highest accuracy among the compared methods while also offering explainable insights into BBP conformation recognition. Because it requires no geometry-specific feature engineering, it is readily transferable to other BBP families, solvent conditions, and simulation protocols. We envision this approach as a useful tool for high-throughput virtual screening of polymer systems whose macroscopic function depends on 3D structure. In future work, this framework can be extended from conformation recognition to downstream structure-property prediction and upstream inverse design of BBP systems.

Data and Software Availability. The data and software supporting the findings of this study are available on Zenodo at Zenodo link.

Supplementary information. The following supporting information file is available free of charge at [SI link here].

- Additional CG MD analysis, representative BBP conformations, and supporting figures and tables for unfunctionalized and dextran-functionalized TS-BBPs (PDF).

Acknowledgements. The authors thank **GlycoMIP**, a National Science Foundation Materials Innovation Platform funded through Cooperative Agreement DMR-1933525 and Virginia Innovation Partnership Corporation under Award CCF23-0245-ERR.

Author contributions. A.I.S., S.Y.J., S.A.D., and Z.J.K. conceived and designed the study. S.Y.J. and S.A.D. performed the molecular dynamics simulations and generated the data. A.I.S. and Z.J.K. developed and implemented the machine-learning and data-analysis methods. A.I.S. and S.Y.J. wrote the original manuscript. J.C., B.N.J., S.A.D., and Z.J.K. reviewed and edited the manuscript. All authors discussed the results, contributed to the interpretation of the findings, and approved the final manuscript.

References

[1] Xie, G., Martinez, M. R., Olszewski, M., Sheiko, S. S. & Matyjaszewski, K. Molecular bottlebrushes as novel materials. *Biomacromolecules* **20**, 27–54 (2018).

[2] Nian, S. & Cai, L.-H. Dynamic mechanical properties of self-assembled bottlebrush polymer networks. *Macromolecules* **55**, 8058–8066 (2022).

[3] Hu, K., Mu, R., Ma, Z. & Li, B. Bottlebrush polymers: From synthesis to properties and applications. *Macromolecular Rapid Communications* 2500215 (2025).

[4] Lahasky, S. H. *et al.* Synthesis and characterization of thermo-responsive polypeptoid bottlebrushes. *Polymer Chemistry* **5**, 1418–1426 (2014).

[5] Li, Z. *et al.* Bottlebrush polymers: From controlled synthesis, self-assembly, properties to applications. *Progress in Polymer Science* **116**, 101387 (2021).

[6] Joshi, S. Y., Singh, S. & Deshmukh, S. A. Coarse-grained molecular dynamics integrated with convolutional neural network for comparing shapes of temperature sensitive bottlebrushes. *npj Computational Materials* **8**, 45 (2022).

[7] Burts, A. O. *et al.* Using epr to compare peg-branch-nitroxide "bivalent-brush polymers" and traditional peg bottle–brush polymers: branching makes a difference. *Macromolecules* **45**, 8310–8318 (2012).

[8] Li, Z., Ma, J., Cheng, C., Zhang, K. & Wooley, K. L. Synthesis of hetero-grafted amphiphilic diblock molecular brushes and their self-assembly in aqueous medium. *Macromolecules* **43**, 1182–1184 (2010).

[9] Ohnsorg, M. L. *et al.* Bottlebrush polymer excipients enhance drug solubility: influence of end-group hydrophilicity and thermoresponsiveness. *ACS macro letters* **10**, 375–381 (2021).

[10] Li, Y., Zou, J., Das, B. P., Tsianou, M. & Cheng, C. Well-defined amphiphilic double-brush copolymers and their performance as emulsion surfactants. *Macromolecules* **45**, 4623–4629 (2012).

[11] Seong, H.-G., Fink, Z., Chen, Z., Emrick, T. & Russell, T. P. Bottlebrush polymers at liquid interfaces: Assembly dynamics, mechanical properties, and all-liquid printed constructs. *ACS nano* **17**, 14731–14741 (2023).

[12] Ahn, S.-k. *et al.* Poly (3-hexylthiophene) molecular bottlebrushes via ring-opening metathesis polymerization: Macromolecular architecture enhanced aggregation. *ACS Macro Letters* **2**, 761–765 (2013).

[13] Li, X., Prukop, S. L., Biswal, S. L. & Verduzco, R. Surface properties of bottlebrush polymer thin films. *Macromolecules* **45**, 7118–7127 (2012).

[14] Azzaroni, O., Brown, A. A. & Huck, W. T. Ucst wetting transitions of polyzwitterionic brushes driven by self-association. *Angewandte Chemie International Edition* **45**, 1770–1774 (2006).

[15] Henn, D. M., Holmes, J. A., Kent, E. W. & Zhao, B. Worm-to-sphere shape transition of thermoresponsive linear molecular bottlebrushes in moderately concentrated aqueous solution. *The Journal of Physical Chemistry B* **122**, 7015–7025 (2018).

[16] Basak, S., Chatterjee, R. & Bandyopadhyay, A. Beyond traditional stimuli: Exploring salt-responsive bottlebrush polymers trends, applications, and perspectives. *ACS omega* **9**, 33365–33385 (2024).

[17] Wang, S., Liu, C., Zhou, H., Gao, C. & Zhang, W. An efficient route to synthesize thermoresponsive molecular bottlebrushes of poly [o-aminobenzyl alcohol-graft-poly (n-isopropylacrylamide)]. *Polymer Chemistry* **8**, 1932–1942 (2017).

[18] Li, X. *et al.* Thermoresponsive pnipaam bottlebrush polymers with tailored side-chain length and end-group structure. *Soft matter* **10**, 2008–2015 (2014).

[19] Bejagam, K. K., Singh, S. K., Ahn, R. & Deshmukh, S. A. Unraveling the conformations of backbone and side chains in thermosensitive bottlebrush polymers. *Macromolecules* **52**, 9398–9408 (2019).

[20] Radzinski, S. C., Foster, J. C., Lewis, S. E., French, E. V. & Matson, J. B. Factors affecting bottlebrush polymer synthesis by the transfer-to method using reversible addition–fragmentation chain transfer (raft) polymerization. *Polymer Chemistry* **8**, 1636–1643 (2017).

[21] Dalsin, S. J. *et al.* Bottlebrush block polymers: Quantitative theory and experiments. *ACS nano* **9**, 12233–12245 (2015).

[22] Chatterjee, D. & Vilgis, T. A. Scaling laws of bottle-brush polymers in dilute solutions. *Macromolecular Theory and Simulations* **25**, 518–523 (2016).

[23] Joshi, S. Y. & Deshmukh, S. A. A review of advancements in coarse-grained molecular dynamics simulations. *Molecular Simulation* **47**, 786–803 (2021).

[24] Mohammadi, E., Joshi, S. Y. & Deshmukh, S. A. A review of computational studies of bottlebrush polymers. *Computational Materials Science* **199**, 110720 (2021).

[25] Decherchi, S., Berteotti, A., Bottegoni, G., Rocchia, W. & Cavalli, A. The ligand binding mechanism to purine nucleoside phosphorylase elucidated via molecular dynamics and machine learning. *Nature communications* **6**, 6155 (2015).

[26] Riniker, S. Molecular dynamics fingerprints (mdfp): machine learning from md data to predict free-energy differences. *Journal of chemical information and modeling* **57**, 726–741 (2017).

[27] Karamzadeh, R. *et al.* Machine learning and network analysis of molecular dynamics trajectories reveal two chains of red/ox-specific residue interactions in human protein disulfide isomerase. *Scientific reports* **7**, 3666 (2017).

[28] Kadupitiya, J., Sun, F., Fox, G. & Jadhao, V. Machine learning surrogates for molecular dynamics simulations of soft materials. *Journal of Computational Science* **42**, 101107 (2020).

[29] Ferguson, A. L. Machine learning and data science in soft materials engineering. *Journal of Physics: Condensed Matter* **30**, 043002 (2017).

[30] So, S., Mun, J. & Rho, J. Simultaneous inverse design of materials and structures via deep learning: demonstration of dipole resonance engineering using core–shell nanoparticles. *ACS applied materials & interfaces* **11**, 24264–24268 (2019).

[31] Coli, G. M., Boattini, E., Filion, L. & Dijkstra, M. Inverse design of soft materials via a deep learning–based evolutionary strategy. *Science advances* **8**, eabj6731 (2022).

[32] Swanson, K., Trivedi, S., Lequieu, J., Swanson, K. & Kondor, R. Deep learning for automated classification and characterization of amorphous materials. *Soft matter* **16**, 435–446 (2020).

[33] Maturana, D. & Scherer, S. *VoxNet: A 3d convolutional neural network for real-time object recognition*, 922–928 (Ieee, 2015).

[34] Qi, C. R. *et al. Volumetric and multi-view cnns for object classification on 3d data*, 5648–5656 (2016).

[35] Wu, Z. *et al. 3d shapenets: A deep representation for volumetric shapes*, 1912–1920 (2015).

[36] Qi, C. R., Su, H., Mo, K. & Guibas, L. J. *PointNet: Deep learning on point sets for 3d classification and segmentation*, 652–660 (2017).

[37] Qi, C. R., Yi, L., Su, H. & Guibas, L. J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* **30** (2017).

[38] Shen, Z., Sun, Y., Lodge, T. P. & Siepmann, J. I. Development of a pointnet for detecting morphologies of self-assembled block oligomers in atomistic simulations. *The Journal of Physical Chemistry B* **125**, 5275–5284 (2021).

[39] DeFever, R. S., Targonski, C., Hall, S. W., Smith, M. C. & Sarupria, S. A generalized deep learning approach for local structure identification in molecular simulations. *Chemical science* **10**, 7503–7515 (2019).

[40] Li, C., Gilbert, B., Farrell, S. & Zarzycki, P. Rapid prediction of a liquid structure from a single molecular configuration using deep learning. *Journal of Chemical Information and Modeling* **63**, 3742–3750 (2023).

[41] Ahmadi, S., Ghanavati, M. A. & Rohani, S. Machine learning-guided prediction of cocrystals using point cloud-based molecular representation. *Chemistry of Materials* **36**, 1153–1161 (2024).

[42] Selvaraju, R. R. *et al. Grad-cam: Visual explanations from deep networks via gradient-based localization*, 618–626 (2017).

[43] Sundararajan, M., Taly, A. & Yan, Q. *Axiomatic attribution for deep networks*, 3319–3328 (PMLR, 2017).

[44] Ribeiro, M. T., Singh, S. & Guestrin, C. *" why should i trust you?" explaining the predictions of any classifier*, 1135–1144 (2016).

[45] Gupta, A., Watson, S. & Yin, H. *3d point cloud feature explanations using gradient-based methods*, 1–8 (IEEE, 2020).

[46] Matrone, F., Paolanti, M., Felicetti, A., Martini, M. & Pierdicca, R. Bubblex: An explainable deep learning framework for point-cloud classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **15**, 6571–6587 (2022).

[47] Zheng, T., Chen, C., Yuan, J., Li, B. & Ren, K. *Pointcloud saliency maps*, 1598–1606 (2019).

[48] Tan, H. *Visualizing global explanations of point cloud dnns*, 4741–4750 (2023).

[49] Zhang, M., You, H., Kadam, P., Liu, S. & Kuo, C.-C. J. Pointhop: An explainable machine learning method for point cloud classification. *IEEE Transactions on Multimedia* **22**, 1744–1755 (2020).

[50] Tan, H. & Kotthaus, H. *Surrogate model-based explainability methods for point cloud nns*, 2239–2248 (2022).

[51] Joshi, S. Y. & Deshmukh, S. A. Aggregation of functional groups leads to unexpected lowering of pnipam lcst. *Computational Materials Science* **258**, 114108 (2025).

[52] Farzeen, P. *et al.* Atomistic insights into conformations and solvation dynamics of amylose, dextran, and pullulan using three force fields. *The Journal of Physical Chemistry B* (2026).

[53] Williams, C. A. *et al.* Systematic evaluation of macromolecular carbohydrate-lectin recognition using precision glycopolymers. *Biomacromolecules* **25**, 7985–7994 (2024).

[54] Chinn, A. F. *et al.* Dextran-block-poly (benzyl glutamate) block copolymers via aqueous polymerization-induced self-assembly. *Carbohydrate Polymers* **352**, 123186 (2025).

[55] Bejagam, K. K., Singh, S., An, Y., Berry, C. & Deshmukh, S. A. Pso-assisted development of new transferable coarse-grained water models. *The Journal of Physical Chemistry B* **122**, 1958–1971 (2018).

[56] Bejagam, K. K., An, Y., Singh, S. & Deshmukh, S. A. Machine-learning enabled new insights into the coil-to-globule transition of thermosensitive polymers using a coarse-grained model. *The Journal of Physical Chemistry Letters* **9**, 6480–6488 (2018).

[57] Mohammadi, E., Joshi, S. Y. & Deshmukh, S. A. Development, validation, and applications of nonbonded interaction parameters between coarse-grained amino

acid and water models. *Biomacromolecules* **24**, 4078–4092 (2023).

[58] An, Y., Bejagam, K. K. & Deshmukh, S. A. Development of transferable non-bonded interactions between coarse-grained hydrocarbon and water models. *The Journal of Physical Chemistry B* **123**, 909–921 (2019).

[59] Martínez, L., Andrade, R., Birgin, E. G. & Martínez, J. M. Packmol: A package for building initial configurations for molecular dynamics simulations. *Journal of computational chemistry* **30**, 2157–2164 (2009).

[60] Phillips, J. C. *et al.* Scalable molecular dynamics on cpu and gpu architectures with namd. *The Journal of chemical physics* **153** (2020).

[61] Phillips, J. C. *et al.* Scalable molecular dynamics with namd. *Journal of computational chemistry* **26**, 1781–1802 (2005).

[62] Zhang, Y., Zhao, W., Sun, B., Zhang, Y. & Wen, W. Point cloud upsampling algorithm: A systematic review. *Algorithms* **15**, 124 (2022).

[63] Dovrat, O., Lang, I. & Avidan, S. *Learning to sample*, 2760–2769 (2019).

[64] Eldar, Y., Lindenbaum, M., Porat, M. & Zeevi, Y. Y. The farthest point strategy for progressive image sampling. *IEEE transactions on image processing* **6**, 1305–1315 (1997).

[65] Wang, Y. *et al.* Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)* **38**, 1–12 (2019).

[66] He, Y., Xia, G., Feng, H. & Wang, Z. Pctp: point cloud transformer pooling block for points set abstraction structure. *The Visual Computer* **39**, 5669–5681 (2023).

[67] Li, Z., Liu, S. & Li, G. *Pointelm: Fast point cloud classification using deep random mapping based extreme learning machines*, 1–6 (IEEE, 2024).

[68] Saimon, A. I., Gnanasambandam, R. & Kong, Z. Geometry-aware physics informed pointnet (geopipn) for fast thermal distribution prediction in additive manufacturing of unseen part geometries. *IEEE Transactions on Automation Science and Engineering* (2026).

[69] Kingma, D. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[70] Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *Journal of machine learning research* **13** (2012).

[71] Ioffe, S. & Szegedy, C. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, 448–456 (pmlr, 2015).

[72] Paszke, A. *et al.* Automatic differentiation in pytorch (2017).

[73] Breiman, L. Random forests. *Machine learning* **45**, 5–32 (2001).

[74] Ke, G. *et al.* Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* **30** (2017).

[75] Chang, C.-C. & Lin, C.-J. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* **2**, 1–27 (2011).

[76] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

[77] Levi, M. Y. & Gilboa, G. Fast and simple explainability for point cloud networks. *arXiv preprint arXiv:2403.07706* (2024).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SIdexBBPPointNetGradCAMSciRepformat.pdf