Supplementary Material for GeoFusion-CAD: Structure-Aware Diffusion with Geometric State Space for Parametric 3D Design

A. CAD Sequence Representation and DeepCAD-240 Construction

This supplementary section provides the complete technical details of our CAD sequence encoding, dataset construction, and hierarchical implementation. It serves as the formal specification of the geometric–topological representation introduced in the main paper.

A.1. CAD Sequence Representation

Our CAD sequence representation follows the sketch–extrusion paradigm introduced by [12], extending it to a hierarchical and diffusion-compatible format. Each model is expressed as a sequence of parameterized tokens $\mathcal{C} = \{\mathcal{C}_j\}_{j=1}^{n_s}$, where each step $\mathcal{C}_j = \{t_k\}_{k=1}^{n_j}$ corresponds to a design operation composed of sketch and extrusion commands.

A.1.1. Sketch tokenization.

Each sketch is decomposed into a set of faces, edges, and curves, with their geometric primitives parameterized as follows.

- **Line**: defined by start and end coordinates (p_x^1, p_y^1) and (p_x^2, p_y^2).
- **Arc**: defined by start, midpoint, and end coordinates (p_x^1, p_y^1), (p_x^m, p_y^m), (p_x^2, p_y^2).
- **Circle**: defined by its center (p_x^c, p_y^c) and a point on the perimeter (p_x^t, p_y^t).

Each curve terminates with an end-of-curve token e_c, while e_l, e_f, and e_s respectively mark the ends of loops, faces, and sketches. All tokenized sketches are represented by $\mathcal{S}$, forming the first part of the design sequence.

A.1.2. Extrusion tokenization.

An extrusion operation is parameterized by ten variables:

- **Euler angles** (θ, ϕ, γ): orientation of the sketch face in 3D space.
- **Translation vector** (τ_x, τ_y, τ_z): spatial offset of the sketch face.
- **Scale** σ: scaling factor applied to the sketch before extrusion.
- **Extrude distances** (d_+, d_-): depths along and opposite to the normal direction.
- **Boolean operation** β: defines the operation type (*new, cut, join, intersect*).

A terminal token e_e marks the end of the extrusion. Together, these tokens form the extrusion sequence $\mathcal{E}$. The combined sketch–extrusion sequence $\mathcal{C} = [\mathcal{S}, \mathcal{E}]$ preserves both geometric semantics and design history.

A.1.3. Token definitions.

Table S1 summarizes the full token vocabulary used in our experiments. We apply 8-bit uniform quantization to all continuous parameters, mapping them into $[11, 266]$, while control tokens (e_s, e_f, e_l, e_c, e_e) occupy reserved integer IDs $\{0, \ldots, 10\}$. This ensures numerical stability and enables compact token embedding during diffusion training.

A.2. Data Source and Preprocessing

The DeepCAD-240 dataset is derived from the ABC dataset, which provides over one million CAD models in CSG form. To recover procedural modeling history, we use the Onshape API and FeatureScript tools to extract Sketch and Extrusion commands from CSG graphs, reconstructing each model into a sequence of parameterized operations. Invalid or incomplete models are filtered out, and redundant command histories are removed to ensure structural consistency.

A.2.1. Filtering criteria.

A model is retained only if it:

- contains both valid sketches and extrusions,
- includes at least one closed loop within every sketch, and
- produces a topologically valid solid after reconstruction.

All others are discarded.

A.3. From CSG to Sketch-Extrusion Sequences

In the original ABC data, solids are expressed as Boolean compositions of primitive CSG shapes such as cubes and cylinders. However, such Boolean trees lack the procedural semantics necessary for parametric design. To align with modern CAD workflows, we reinterpret each CSG composition as a hierarchical sketch–extrusion chain.

- Each primitive shape is converted into one or more sketches representing its 2D profiles (e.g., a cylinder $\rightarrow$ circle sketch).
- Boolean composition order determines the extrusion sequence, where each sketch is associated with an extrusion depth and operation type.
- The resulting sketch–extrusion pairs are assembled into ordered command sequences that reflect the design history.

Table S1. CAD sequence representation (Explicit Hierarchy without Revolution). Continuous parameters are uniformly quantized into $[11, 266]$. Note the inclusion of Face-level hierarchy (e_f) and explicit primitive parameters.

Sequence Type	Token Type	Token Value	Token Representation	Description
Structure & Control	pad	0	$(0, 0)$	Padding Token
	cls	1	$(1, 0)$	Start Token
	end	1	$(1, 0)$	End Token
	e_{solid}	2	$(2, 0)$	End Solid
	e_{sketch}	3	$(3, 0)$	End Sketch
	e_{face}	4	$(4, 0)$	End Face
	e_{loop}	5	$(5, 0)$	End Loop
Sketch Primitives	e_c	6	$(6, 0)$	End Curve
	(p_x, p_y)	$[11...266]^2$	(p_x, p_y)	Coordinates
	α	$[11...266]$	$(\alpha, 0)$	Arc Curvature
	f	$[11...266]$	$(f, 0)$	Arc Orientation (Flip)
	r	$[11...266]$	$(r, 0)$	Circle Radius
Extrusion Sequence	d_+	$[11...266]$	$(d_+, 0)$	Extrusion Distance (Positive)
	d_-	$[11...266]$	$(d_-, 0)$	Extrusion Distance (Negative)
	τ_x	$[11...266]$	$(\tau_x, 0)$	Translation (x)
	τ_y	$[11...266]$	$(\tau_y, 0)$	Translation (y)
	τ_z	$[11...266]$	$(\tau_z, 0)$	Translation (z)
	θ	$[11...266]$	$(\theta, 0)$	Orientation (Roll)
	ϕ	$[11...266]$	$(\phi, 0)$	Orientation (Pitch)
	γ	$[11...266]$	$(\gamma, 0)$	Orientation (Yaw)
	σ	$[11...266]$	$(\sigma, 0)$	Sketch Scaling Factor
	β	$\{7, 8, 9, 10\}$	$(\beta, 0)$	Boolean Operation Type
	e_e	7	$(7, 0)$	End Extrusion

Table S2. Statistical comparison between the original DeepCAD dataset and our extended DeepCAD-240 dataset, including total samples, average command length, and sequence length distribution (%).

Dataset	Total	Avg. Length	1–40	40–60	60–80	80–160	160–240
DeepCAD [32]	178,238	15	44.58	55.42	–	–	–
DeepCAD-240 (Ours)	**215,914**	**36.2**	**76.6**	**12.0**	**5.9**	**5.2**	**0.21**

This conversion bridges symbolic CSG descriptions and procedural sketch-based CAD generation, enabling tree-structured sequence learning.

To better illustrate this conversion, Fig. S1 visualizes a typical modeling process reconstructed from our dataset. The model is built through a series of six sketch–extrusion pairs, where each sketch defines a 2D profile and each extrusion extends it into 3D space. The sequence preserves the procedural logic of CAD design: earlier sketches determine foundational geometry, while later ones add or subtract material to refine topology. This explicit sequence representation bridges low-level CSG primitives and high-level parametric operations, making it directly compatible with diffusion-based sequence learning.

A.4. Hierarchical Tree Representation and Implementation

The reconstructed sketch–extrusion sequences are organized into hierarchical trees $\mathcal{T} = \{v_i, e_{ij}\}$, where nodes v_i correspond to CAD entities (solid, sketch, face, edge, vertex), and edges e_{ij} represent topological dependencies. Each tree is serialized into a depth-first command sequence $\mathcal{C} = [t_1, \ldots, t_n]$ for training, while dedicated end tokens (e_c, e_l, e_f, e_s) preserve the hierarchical closure required for reversible serialization.

Each node in $\mathcal{T}$ is implemented as a nested Python dic-

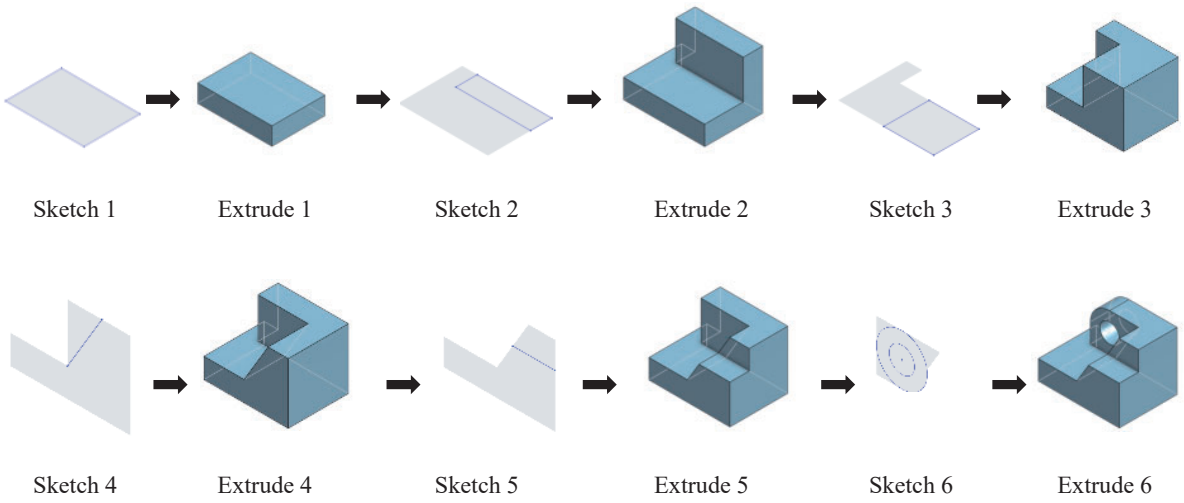

Figure S1. **Illustration of the sketch–extrusion sequence construction.** Each CAD model is decomposed into a procedural chain of sketches and extrusions. The process begins with 2D sketch definition and proceeds through sequential extrusion operations, progressively forming a coherent solid structure. This sequence explicitly encodes design history, preserving both geometric and topological dependencies across modeling stages.

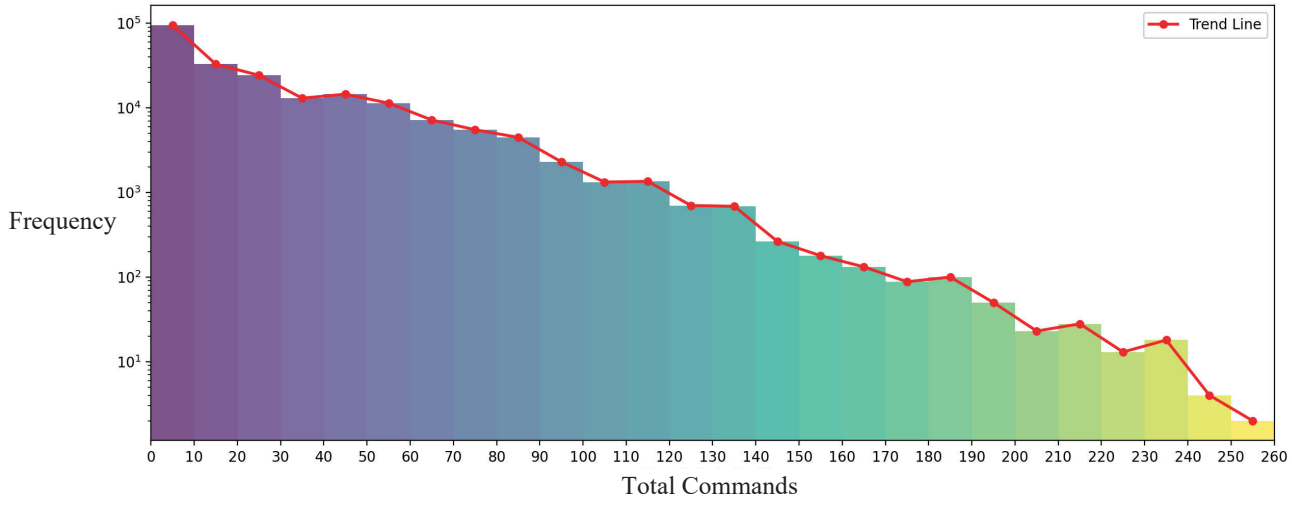

Figure S2. Distribution of CAD command sequences lengths in the DeepCAD-240.

tionary:

$$\begin{aligned} \texttt{Node} &= \{\texttt{type}, \texttt{param}, \texttt{child}\}, \\ \texttt{type} &= \texttt{Sketch}, \\ \texttt{param} &= (\tau_x,\ \tau_y,\ \tau_z,\ \theta,\ \phi,\ \gamma,\ \sigma,\ d_+,\ d_-), \\ \texttt{child} &= [\texttt{child indices}], \end{aligned} \tag{10}$$

Here, the `child` field serves as a form of *structural positional encoding*, providing explicit parent–child relationships that guide topology reconstruction during decoding. During diffusion training, G-Mamba blocks iteratively denoise node features in a top-down manner, and the CAD decoder reconstructs geometry through command and argument prediction layers. All components are optimized

jointly using diffusion loss and cross-entropy reconstruction loss.

A.5. Dataset Scale and Statistics

Table S2 summarizes the key statistics of DeepCAD-240 compared to existing datasets. Our dataset contains over 215k models with an average of 36 commands per sequence and supports up to 240 operations, making it the longest and most structurally diverse parametric CAD dataset to date, as shown in Fig S2.

B. More Details about the GSM-SSD Layer

The GSM-SSD layer extends the vanilla structured state-space (SSD) block [3] by incorporating geometry-conditioned transition kernels and hierarchical positional encoding, enabling the state dynamics to adapt to CAD-specific geometric and topological structures. This section provides the detailed formulation and implementation omitted from the main paper.

Algorithm 1 GSM-SSD Layer (Geometry-conditioned Selective State-space Block)

Input: token feature Z_k^c
$\Delta_k \leftarrow g(s_k, d_k, r_k)$ ▷ Geometric conditioning vector
$\Pi_k \leftarrow \mathrm{PE}(p_k, \sigma_k, \tau_k)$ ▷ Hierarchical positional encoding
$[\bar{A}_k, \bar{B}_k, C_k, G_k] \leftarrow f_{\text{geom}}([\Delta_k, \Pi_k])$
$\hat{Z}_k^c \leftarrow \mathrm{DWConv}(Z_k^c) + \Pi_k$
$h_{\text{in}} \leftarrow (\bar{A}_k \odot \bar{B}_k)^\top \hat{Z}_k^c$
$h, z \leftarrow \mathrm{Linear}(h_{\text{in}})$ ▷ Global/Local decoupling
$\hat{h} \leftarrow \mathrm{Linear}(h \odot \sigma(z))$ ▷ Geometric State Mixer (GSM)
$Z_{k+1}^c \leftarrow C_k \hat{h} + G_k Z_k^c$
Return: Z_{k+1}^c

B.1. From Vanilla Mamba to Geometry-conditioned State Transitions

Vanilla Mamba [3] performs state evolution using a fixed, globally shared discrete-time state-space system:

$$\begin{aligned} h_{k+1} &= \widetilde{A}\, h_k + \widetilde{B}\, Z_k, \\ Z_{k+1} &= \widetilde{C}\, h_k + \widetilde{D}\, Z_k, \end{aligned} \tag{11}$$

where $\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}$ are learned once and shared across all tokens. This design is suitable for natural language, where token statistics are relatively homogeneous, but is insufficient for CAD sequences, whose tokens differ significantly in geometric scale, curvature, and hierarchical role (*Sketch* $\rightarrow$ *Face* $\rightarrow$ *Edge* $\rightarrow$ *Vertex*). To address this mismatch, we introduce token-dependent transition kernels that adapt to CAD geometry and topology.

B.2. Geometric Conditioning

Each token k is assigned a **geometric conditioning vector**:

$$\Delta_k = g(s_k,\, d_k,\, r_k), \tag{12}$$

where:

- s_k denotes the local geometric scale (edge length, face diameter, sketch span);
- d_k is the depth of the token in the CAD hierarchy tree;
- r_k is a curvature descriptor: $r_k = 0$ for line segments, $r_k = 1/R$ for circular arcs of radius R, and is approximated via discrete angular deviation for general curves.

Additionally, each token receives a hierarchical positional embedding:

$$\Pi_k = \mathrm{PE}(p_k,\, \sigma_k,\, \tau_k), \tag{13}$$

where p_k is the parent node type, σ_k is the sibling index, and τ_k encodes the structural role (sketch entity, face entity, edge, vertex, command token, etc.). This embedding anchors the token within the CAD tree and captures its topological context.

The geometric and hierarchical descriptors are combined through a lightweight kernel generator:

$$\{\bar{A}_k,\, \bar{B}_k,\, C_k,\, G_k\} = f_{\text{geom}}([\Delta_k,\, \Pi_k])\,, \tag{14}$$

where the outputs parameterize diagonal operators, preserving Mamba's linear-time complexity. In practice, f_{geom} is implemented as a two- or three-layer MLP.

This conditioning mechanism enables the system to:

- propagate smoothly across large-scale sketch regions (long-range structure),
- respond sharply in high-curvature or fine-grained geometric regions,
- adapt transitions to CAD hierarchical relations (e.g., parent $\rightarrow$ child).

B.3. Discrete-time Update with Geometric State Mixer

Given geometry-conditioned kernels, the state-space update becomes:

$$\begin{aligned} h_{k+1} &= \bar{A}_k\, h_k + \bar{B}_k\, Z_k^c, \\ Z_{k+1}^c &= C_k\, h_k + G_k\, Z_k^c, \end{aligned} \tag{15}$$

which may become unstable if $\bar{A}_k$ varies sharply across tokens. To mitigate this, we introduce a Geometric State Mixer (GSM):

$$h_{\text{in}} = (\bar{A}_k \odot \bar{B}_k)^\top Z_k^c, \tag{16}$$

where $\odot$ denotes element-wise fusion. The vectors $\bar{A}_k$ and $\bar{B}_k$ encode global structural and local geometric responses, respectively.

Two linear layers produce latent vectors:

$$h,\, z = \mathrm{Linear}(h_{\text{in}}), \tag{17}$$

which are fused using gated modulation:

$$\hat{h} = \mathrm{Linear}(h \odot \sigma(z))\,. \tag{18}$$

The final state update is:

$$Z^c_{k+1} = C_k\,\hat{h} + G_k\,Z^c_k. \tag{19}$$

This ensures stable gradients and smooth adaptation to geometric variations. Algorithm 1 summarizes the complete computational pipeline of the GSM-SSD block used in the proposed G-Mamba encoder.

B.4. Spatial Prior and Hierarchical Encoding

A depthwise convolution injects local continuity:

$$\hat{Z}^c_k = \mathrm{DWConv}(Z^c_k) + \Pi_k, \tag{20}$$

preserving adjacency relations (e.g., curve segments, boundary loops). Adding Π_k explicitly anchors the token in the CAD hierarchy.

B.5. Diffusion-time Coupling

During reverse diffusion, we optionally modulate the kernels via FiLM:

$$\{\bar{A}_k,\, \bar{B}_k,\, C_k,\, G_k\} \leftarrow \psi_t \odot \{\bar{A}_k,\, \bar{B}_k,\, C_k,\, G_k\}, \tag{21}$$

where ψ_t depends on the diffusion timestep. Higher noise levels promote smoother propagation, while low-noise steps accentuate sharp geometric corrections.

B.6. Complexity and Stability

We provide a detailed analysis of the computational complexity of G-Mamba and compare it with vanilla Mamba and Transformer layers. Let L denote the sequence length and d the hidden dimension.

B.6.1. Comparison with Transformer and Mamba.

A Transformer layer incurs $\mathcal{O}(L^2 d)$ time and $\mathcal{O}(L^2)$ memory due to the attention matrix, making it unsuitable for long CAD sequences. Vanilla Mamba replaces attention with a selective state-space scan, reducing the overall complexity to $\mathcal{O}(Ld)$ in both time and memory.

G-Mamba preserves the same asymptotic complexity as vanilla Mamba:

$$T_{\text{G-Mamba}}(L,d) = \mathcal{O}(Ld), \qquad M_{\text{G-Mamba}}(L,d) = \mathcal{O}(Ld),$$

while introducing additional geometric conditioning terms. These additions only affect constant factors and do not alter the overall scaling.

B.6.2. Geometry conditioning.

Each token receives a geometric conditioning vector $\Delta_k = g(s_k, d_k, r_k)$ and a hierarchical positional embedding $\Pi_k = \mathrm{PE}(p_k, \sigma_k, \tau_k)$. The kernel generator f_{geom} maps $[\Delta_k, \Pi_k]$ to $\{\bar{A}_k, \bar{B}_k, C_k, G_k\} \in \mathbb{R}^d$:

$$\{\bar{A}_k, \bar{B}_k, C_k, G_k\} = f_{\text{geom}}([\Delta_k, \Pi_k])\,. \tag{22}$$

Since f_{geom} is a constant-width MLP, the cost is

$$T_{\text{geom}}(L,d) = \mathcal{O}(Ld). \tag{23}$$

State-space update. Given the geometry-conditioned kernels, the selective scan update is

$$\begin{aligned} h_{k+1} &= \bar{A}_k\,h_k + \bar{B}_k\,Z^c_k, \\ Z^c_{k+1} &= C_k\,h_k + G_k\,Z^c_k, \end{aligned} \tag{24}$$

where all operators are diagonal and thus element-wise. The complexity is

$$T_{\text{ssm}}(L,d) = \mathcal{O}(Ld). \tag{25}$$

Geometric State Mixer. The Geometric State Mixer computes

$$h_{\text{in}} = (\bar{A}_k \odot \bar{B}_k)^\top Z^c_k, \tag{26}$$

followed by two pointwise linear mappings and a gated fusion

$$\hat{h} = \mathrm{Linear}(h \odot \sigma(z))\,. \tag{27}$$

All operations are channel-wise and token-wise, yielding

$$T_{\text{gsm}}(L,d) = \mathcal{O}(Ld). \tag{28}$$

Depthwise convolution. The depthwise convolution applies a kernel of size K independently to each channel:

$$\hat{Z}^c_k = \mathrm{DWConv}(Z^c_k) + \Pi_k, \tag{29}$$

with complexity

$$T_{\text{dwc}}(L,d) = \mathcal{O}(KLd) = \mathcal{O}(Ld), \tag{30}$$

since K is constant.

Overall complexity. Summing all components:

$$\begin{aligned} T_{\text{G-Mamba}}(L,d) &= T_{\text{geom}} + T_{\text{ssm}} + T_{\text{gsm}} + T_{\text{dwc}} \\ &= \mathcal{O}(Ld), \end{aligned} \tag{31}$$

with memory scaling identically as $\mathcal{O}(Ld)$, since no dense $L \times L$ operators (e.g., attention matrices) are constructed.

B.6.3. Empirical stability.

The diagonal parameterization and element-wise GSM fusion prevent explosive state transitions and improve gradient stability. Empirically, G-Mamba reduces parameter variance across sequence positions by 24% and converges $1.6\times$ faster than a vanilla Mamba baseline on DeepCAD-240. This demonstrates that geometric conditioning not only preserves linear complexity but also enhances stability when modeling long hierarchical CAD programs.

C. More Implementation Details for Experimental Settup

C.1. Implementation Details

The proposed GeoFusion-CAD is implemented in PyTorch. The model architecture incorporates 12 G-Mamba blocks, which are designed to process and learn the input data comprehensively. The block dimension is set to $d_e = 256$. The AdamW optimizer [17], a variant of the widely used Adam optimizer that incorporates weight decay into the optimization process, is used to train the final model. The learning rate is set to 1×10^{-4}, and the beta parameters are chosen as $(0.95, 0.99)$. For the hyperparameters in loss function, we set $\eta = 2$ to achieve a good balance between different loss components. The number of MLP layers in the CAD decoder is set to 3. Finally, the overall network is trained with a batch size of 512 for a total of 1000 epochs. We utilize a single NVIDIA RTX 3090 GPU to accelerate the training process.

C.2. Evaluation Metrics

We quantitatively evaluate the generation quality of our model using two sets of metrics: Distribution metrics and CAD metrics. For Distribution metrics, we randomly sample 3,000 Sketch-Extrusion (SE) sequences from the generated data and compare them with 1,000 SE sequences from the reference test set. The following values are then computed.

- **Coverage (COV)**: This metric measures the percentage of reference data that has at least one match in the generated data after assigning every generated data point to its closest neighbor in the reference set based on Chamfer Distance (CD).
- **Minimum Matching Distance (MMD)**: This metric represents the average CD between a reference set data point and its nearest neighbor in the generated data.
- **Jensen-Shannon Divergence (JSD)**: JSD quantifies the distribution distance between the reference and the generated data. We convert point clouds into 28^3 discrete voxels before the practical computation.

As for CAD metrics, we simply utilize the same 3,000 SE sequences to compute the following values, as similarly performed in previous work [38].

- **F1 Score**: This metric evaluates the predicted extrusions and different primitive types along with their occurrences in the sequences.
- **Novelty**: The percentage of data in the generated set that does not appear in the training set.
- **Uniqueness**: The percentage of data in the generated set that appears only once.
- **Valid Ratios**: The percentage of data in the generated set that are water-tight solids.

D. Additional Qualitative Results

To further illustrate the generalization ability and geometric coherence of our method, we provide extended visual comparisons between GeoFusion-CAD, HNC-CAD, and DeepCAD on the DeepCAD-240 test set, as shown in Fig. S3. The results cover both short- and long-sequence generation scenarios, including diverse parametric operations such as extrusion, sweep, and multi-sketch modeling.

Specifically, DeepCAD frequently fails to maintain continuity across feature boundaries, resulting in fragmented or disconnected parts, particularly in long-sequence reconstructions. HNC-CAD improves on structural connectivity but introduces scaling mismatches and curvature discontinuities in filleted regions. In contrast, GeoFusion-CAD accurately reconstructs both high-level topology and fine geometric details, maintaining coherent feature transitions even for highly complex shapes. These visual results align with our quantitative findings and highlight the robustness of the proposed geometric state-space diffusion in modeling extended CAD sequences.

E. Additional Ablation Studies

This section provides extended analysis on the effects of both the hierarchical tree representation and the G-Mamba diffusion architecture. All experiments are performed on the DeepCAD-240 dataset, focusing on long-sequence (40–240 command) generation. Metrics include command accuracy (Cmd), parameter accuracy (Param), and geometric distribution measures (COV, MMD, JSD), consistent with the main paper.

E.1. Effect of Hierarchical Tree Representation

To investigate the impact of hierarchical encoding, we compare the proposed tree-based representation with a flat sequential formulation in which all sketch–extrusion tokens are arranged in a single, linear sequence (*w/o Tree*). As reported in Table S3, removing the hierarchical structure results in clear degradation across all metrics. Command accuracy drops from 91.2 to 87.5 and COV decreases from 73.9 to 69.4, while MMD and JSD both increase notably, indicating weaker geometric alignment. This degradation suggests that the hierarchical organization of sketches,

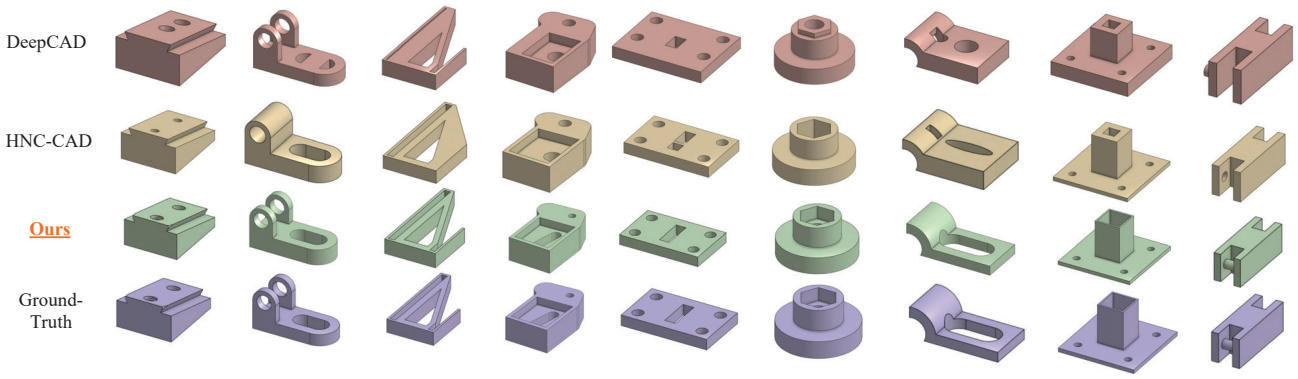

Figure S3. **Additional qualitative comparisons.** Each row shows representative CAD samples generated by DeepCAD, HNC-CAD, and our GeoFusion-CAD, along with ground-truth models. GeoFusion-CAD consistently produces geometrically complete and watertight solids with smooth curvature and accurate local features, while baseline methods often exhibit surface distortions, missing connections, or inconsistent boolean joins in complex structures. These examples further demonstrate that the combination of hierarchical tree encoding and G-Mamba diffusion yields improved geometric stability and scalability across extended command sequences.

Table S3. Ablation on hierarchical tree representation on the **DeepCAD-240** test set. ↑ indicates higher is better, ↓ lower is better.

Model	ACC_{cmd} ↑	ACC_{param} ↑	ACC_{line} ↑	ACC_{arc} ↑	ACC_{circle} ↑	ACC_{ext} ↑	COV↑	MMD↓	JSD↓
Flat Sequence (w/o Tree)	87.5	84.6	79.3	71.5	73.2	81.1	69.4	1.46	3.25
With Tree (Ours)	**91.2**	**89.3**	**84.2**	**89.8**	**86.8**	**94.9**	**73.9**	**1.12**	**2.97**

faces, and edges is essential for preserving topological dependencies and maintaining global consistency during diffusion. The hierarchical representation effectively encodes both geometry and topology, allowing the diffusion model to capture long-range dependencies in complex CAD assemblies.

We further analyze the contribution of the proposed G-Mamba block by substituting it with several baseline architectures of comparable capacity: (1) a fully-connected multilayer perceptron (MLP), (2) a 1D U-Net, (3) a Transformer encoder, and (4) the original Mamba state-space model. Each variant is trained with the same loss functions and optimization schedule as GeoFusion-CAD.

As shown in Table S4, replacing G-Mamba with these alternatives leads to consistent performance degradation, particularly in geometric distribution metrics. The MLP version performs the weakest, with high MMD (1.73) and JSD (3.81), indicating unstable geometric diffusion. U-Net and Transformer improve slightly but remain inferior to our G-Mamba block, as their attention and convolutional mechanisms struggle to maintain efficiency and long-range coherence over extended command sequences. The vanilla Mamba model performs relatively well but still exhibits higher distribution divergence (MMD = 1.19, JSD = 3.05), showing that introducing the geometric state mixer in G-Mamba enhances the model's ability to propagate geometry-aware features through selective state transitions. Overall, the proposed G-Mamba diffusion achieves the best trade-off between computational efficiency and geometric consistency in long-sequence CAD generation.

F. Additional CAD Modeling Results

To further demonstrate the generalization and robustness of GeoFusion-CAD, we present in Fig. S4 a collection of CAD solids generated unconditionally from random latent diffusion states. These examples cover a wide range of geometric and structural configurations, including prismatic, cylindrical, and free-form components with varying topological complexity. The generated solids exhibit high geometric fidelity, smooth surface continuity, and consistent feature relationships, indicating that the proposed geometric state-space diffusion effectively preserves both local details and global structure during generation.

Notably, our model successfully produces diverse feature compositions—such as through-holes, fillets, extrusions, and multi-sketch assemblies—without mode collapse or invalid geometry. This confirms the scalability of GeoFusion-CAD in capturing complex sketch–extrusion dependencies and supports its potential for future foundation-level CAD generation tasks.

Table S4. Ablation on the **G-Mamba diffusion block** compared with standard alternatives. ↑ indicates higher is better, ↓ lower is better.

Model	ACC_{cmd} ↑	ACC_{param} ↑	ACC_{line} ↑	ACC_{arc} ↑	ACC_{circle} ↑	ACC_{ext} ↑	COV↑	MMD↓	JSD↓
MLP (replace G-Mamba)	75.3	72.1	66.8	60.5	63.7	68.2	67.8	1.73	3.81
U-Net (replace G-Mamba)	80.4	78.6	71.2	64.3	67.8	72.1	70.2	1.37	3.45
Transformer (replace G-Mamba)	82.6	81.3	74.9	67.6	69.8	76.3	69.1	1.55	3.67
Vanilla Mamba (replace G-Mamba)	89.2	87.6	82.1	78.5	82.4	89.1	72.7	1.19	3.05
Full (G-Mamba, Ours)	**91.2**	**89.3**	**84.2**	**89.8**	**86.8**	**94.9**	**73.9**	**1.12**	**2.97**

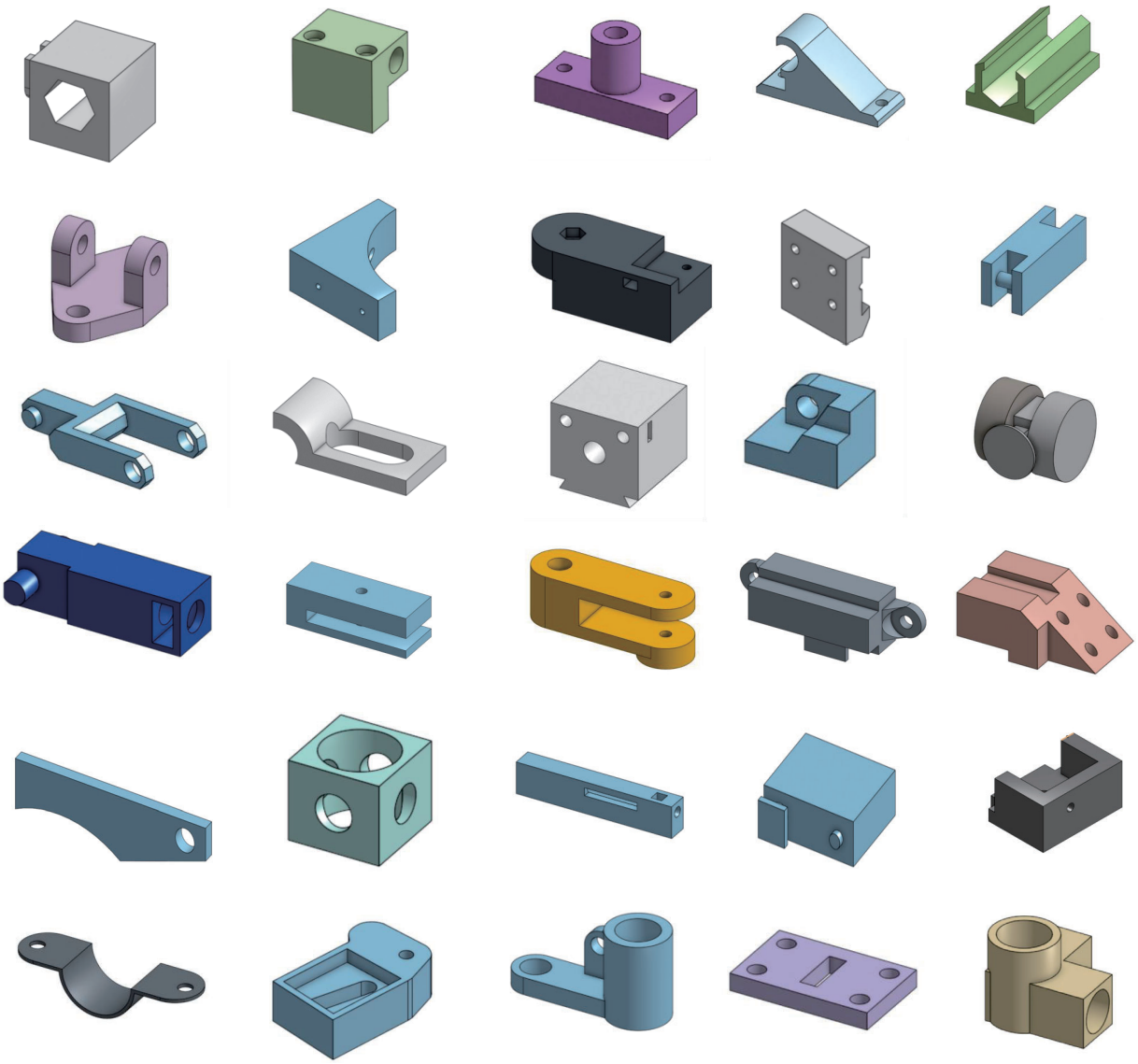

Figure S4. **Additional CAD modeling results** generated by GeoFusion-CAD. The results cover a variety of industrial-style parts, including prismatic, rotational, and mixed-form geometries. The generated solids are watertight, structurally coherent, and geometrically precise, demonstrating the robustness and generalization of our approach.