ProST: A Procedural Synthesis Transformer for Precision 3D CAD Modeling

Rohith Garapati

August 2025

Abstract

This paper introduces the Procedural Synthesis Transformer (ProST), a novel deep learning architecture designed for high-precision, text-driven 3D Computer-Aided Design (CAD) modeling. The primary objective of ProST is to interpret nuanced natural language descriptions and transform them into manufacturable, geometrically precise 3D objects, rivaling the quality of expert human engineers. The architecture uniquely combines the semantic reasoning of transformers with the mathematical exactitude of procedural CAD kernels and the subtle refinement capabilities of graph-based neural networks. This document details the conceptual framework, the design choices behind its four core components—the Semantic Decomposer, the Procedural Synthesis Kernel, the Geometric Refinement Network, and the Self-Correction & Validation Loop—along with comprehensive model specifications and training configurations. The paper provides a complete blueprint for implementing a system capable of mastering both the art and science of automated 3D modeling.

1 Introduction

The field of generative AI has made significant strides in creating 3D content from text prompts. However, a persistent gap remains between generating visually plausible shapes and producing dimensionally accurate, mathematically defined models suitable for engineering and manufacturing. Existing methods often rely on representations like meshes, voxels, or neural radiance fields, which excel at organic forms but lack the precision required for CAD applications. To address this, the Procedural Synthesis Transformer (ProST) architecture is proposed.

ProST is engineered to bridge this gap by creating a system that thinks and builds like an expert engineer. It translates high-level creative or technical intent into a structured, logical sequence of construction steps, executes them with deterministic precision, and refines the output with learned knowledge of aesthetics and functionality. This approach moves beyond mere shape generation, aiming to create a tool capable of understanding and executing complex design tasks with a level of rigor that meets professional standards.

2 The ProST Architecture

The ProST architecture is a multi-stage, iterative system designed for clarity, precision, and continuous self-improvement. It is composed of four distinct yet deeply interconnected components that work in sequence to transform a text embedding into a final, validated 3D model. The design philosophy is to separate high-level semantic interpretation from low-level geometric construction, allowing each component to specialize in its task.

2.1 The Semantic Decomposer

The first stage of the ProST pipeline is the Semantic Decomposer. This component is responsible for translating the abstract, feature-rich text embedding into a structured, machine-executable plan.

- **Design Choice:** A domain-specific Transformer decoder was selected for this role. Unlike a general-purpose language model, this decoder is trained to output a sequence in a custom Domain-Specific Language (DSL) designed for 3D modeling.
- **Rationale:** A DSL provides a structured and unambiguous format that is ideal for defining geometric operations. It decomposes a complex prompt (e.g., "a stainless steel ergonomic handle") into a logical script of commands, parameters, and constraints, such as `CREATE_CYLINDER`, `APPLY_CURVATURE`, and `SET_MATERIAL`. This ensures that the user's intent is captured in a format that the subsequent components can execute without ambiguity.

2.2 The Procedural Synthesis Kernel

At the heart of ProST lies the Procedural Synthesis Kernel. This component is not a neural network but a deterministic engine that provides the ground truth of perfect geometry.

- **Design Choice:** The kernel utilizes a standard CAD engine that operates on Boundary Representation (B-Rep). B-Rep is the industry standard for professional CAD, defining objects as a set of mathematically precise surfaces, edges, and vertices.
- **Rationale:** To achieve manufacturable precision, it is essential to build upon a foundation of mathematical exactitude. By executing the DSL script from the Decomposer, the Kernel creates a 3D model that is geometrically flawless and dimensionally accurate. This deterministic core ensures that all specified constraints (e.g., a hole diameter of exactly 5mm) are met perfectly.

2.3 The Geometric Refinement Network

While the Kernel produces a precise model, it may lack the subtle, non-parametric details that define a high-quality finished product. The Geometric Refinement Network is designed to add these artistic and functional nuances.

- **Design Choice:** A Graph Neural Network (GNN) was chosen for this task. The B-Rep model from the Kernel is converted into a graph structure, where faces, edges, and vertices serve as nodes.
- **Rationale:** A GNN is uniquely suited to operate on the topological structure of a 3D model. It can learn relationships between different parts of the object and make context-aware adjustments. For instance, it can refine a simple handle into an ergonomic shape by subtly shifting vertex positions based on the global prompt ("ergonomic") and local geometry. This allows the model to learn and apply details that are difficult to specify procedurally.

2.4 The Self-Correction & Validation Loop

The final component elevates ProST from a generative tool to a reliable design partner. It ensures the output is not just plausible but correct, functional, and manufacturable.

- **Design Choice:** A multi-faceted validation suite, including a discriminator network, a lightweight physics simulator, and a constraint checker.
- **Rationale:** A robust validation process is critical for building trust in an automated system. The discriminator checks for aesthetic coherence, the physics simulator identifies potential structural flaws, and the constraint checker verifies that all original parameters have been maintained. If any validator flags an issue, a corrective feedback signal is sent to the earlier components, initiating an iterative refinement loop. This process continues until the model passes all checks, achieving a level of quality assurance that emulates, and can eventually surpass, a human's review process.

3 Model Specifications and Architecture Details

This section provides comprehensive technical specifications for the three core neural network components of ProST. Each model is designed for production-scale deployment with mixed-precision training capabilities and optimized for maximum performance.

3.1 Semantic Decomposer Configuration

The Semantic Decomposer employs a decoder-only Transformer architecture, specifically designed for conditional sequence generation. The model translates natural language prompts into structured DSL commands with high precision and consistency.

```
model:
  type: "Decoder-only Transformer (GPT-style)"
  vocab_size: 4096
  embedding_dim: 1024
  num_layers: 24
  num_attention_heads: 16
  feed_forward_dim: 4096
  activation_function: "GeLU"
  dropout_rate: 0.1
  max_sequence_length: 1024

training:
  optimizer: "AdamW"
  learning_rate_schedule: "CosineDecayWithWarmup"
  initial_learning_rate: 5e-5
  warmup_steps: 4000
  batch_size: 64
  loss_function: "CrossEntropyLoss"
  precision: ['fp16', 'bf16']
  dataset: "Curated (Text Prompt, DSL Script) pairs with data
      augmentation."
```

Listing 1: Semantic Decomposer Model Configuration

3.2 Geometric Refinement Network Configuration

The Geometric Refinement Network utilizes a Graph Attention Network architecture to perform context-aware geometric refinements on the 3D models generated by the Procedural Synthesis Kernel.

```
model:
  type: "Graph Attention Network (GAT)"
  num_layers: 12
  num_attention_heads: 8
  input_node_features: 6
  hidden_feature_dim: 256
  output_node_features: 3
  activation_function: "LeakyReLU"

training:
  optimizer: "Adam"
  learning_rate: 5e-4
  batch_size: 32
  loss_function: "CombinedLoss"
  loss_components:
    ChamferDistanceLoss: 1.0
```

```
    LaplacianSmoothingLoss: 0.2
  precision: ['fp16', 'bf16']
  dataset: "Dataset of (Procedural Model, Expert-Refined Model) pairs with
     synthetic augmentations."
```

Listing 2: Geometric Refinement Network Configuration

3.3 Discriminator Network Configuration

The discriminator component employs a PointNet++ architecture to evaluate the quality and authenticity of generated 3D models, providing critical feedback for the iterative refinement process.

```
model:
  type: "PointNet++"
  num_input_points: 4096

  set_abstraction_layers:
    - npoint: 1024
      radius: 0.2
      nsample: 32
      mlp_channels: [128, 128, 256]
    - npoint: 256
      radius: 0.4
      nsample: 64
      mlp_channels: [256, 256, 512]
    - mlp_channels: [512, 1024, 2048]

  classifier_mlp:
    layers: [1024, 512, 1]
    dropout_rate: 0.5
    output_activation: "Sigmoid"

training:
  optimizer: "Adam"
  learning_rate: 1e-4
  batch_size: 48
  loss_function: "BinaryCrossEntropyLoss"
  precision: ['fp16', 'bf16']
  dataset: "A mixed dataset of expert and ProST-generated CAD models with
     data balancing."
```

Listing 3: Discriminator Network Configuration

4 Implementation Considerations

4.1 Mixed-Precision Training

All neural network components are configured to support both FP16 and BF16 mixed-precision training. This approach significantly reduces memory consumption and accelerates training on modern hardware while maintaining numerical stability and model performance.

4.2 Data Requirements

The successful implementation of ProST requires three distinct datasets:

- **Text-to-DSL Dataset:** Pairs of natural language descriptions and corresponding DSL scripts for training the Semantic Decomposer.

- **Geometric Refinement Dataset:** Pairs of procedurally generated models and expert-refined versions for training the GNN.
- **Quality Assessment Dataset:** A balanced collection of expert-created and AI-generated models for discriminator training.

4.3 Hardware Requirements

The proposed configurations are designed for high-performance computing environments with substantial GPU memory and computational capacity. The large parameter counts and batch sizes optimize for distributed training across multiple GPUs or TPUs.

5 Training Pipeline and Workflow

5.1 Sequential Training Strategy

The ProST system employs a multi-stage training approach where each component is trained progressively:

1. **Phase 1:** Train the Semantic Decomposer independently on text-to-DSL pairs
2. **Phase 2:** Train the Geometric Refinement Network using procedural models as input
3. **Phase 3:** Train the Discriminator on a mixed dataset of real and generated models
4. **Phase 4:** Joint fine-tuning of all components with adversarial feedback

5.2 Quality Metrics

The system performance is evaluated using multiple metrics:

- **Geometric Accuracy:** Chamfer distance between generated and target models
- **Dimensional Precision:** Percentage of constraints satisfied within tolerance
- **Manufacturing Viability:** Physics simulation pass rate
- **Aesthetic Quality:** Discriminator confidence scores

6 Conclusion

The Procedural Synthesis Transformer (ProST) presents a new architectural paradigm for text-to-3D generation, specifically tailored for the demands of precision CAD and engineering. By integrating a reasoning-based Transformer, a deterministic CAD kernel, a refining Graph Neural Network, and a rigorous validation loop, ProST is designed to produce models that are not only visually compelling but also mathematically sound and functionally reliable. The detailed model specifications provided in this paper offer a complete blueprint for implementing a production-ready system capable of automated, high-precision 3D modeling that meets professional engineering standards.

The system's modular design allows for independent optimization of each component while maintaining seamless integration across the entire pipeline. With proper implementation of the proposed configurations and training procedures, ProST represents a significant advancement toward truly intelligent computer-aided design systems that can understand, interpret, and execute complex engineering tasks with human-level precision and creativity.