Language-Augmented Semantic Priors for B-Spline Surface Fitting (Supplementary Materials)

Yunzhong Lou   Yusheng Luo   Jiahao Li   Yu Song   Xiangdong Zhou*
College of Computer Science and Artificial Intelligence, Fudan University
{yzlou20, xdzhou}@fudan.edu.cn   {lijh23, ycluo24, songy23}@m.fudan.edu.cn

1. Rich-text Semantic Descriptions

Parametric modeling histories are typically represented as symbolic operation sequences (e.g., Sketch, Extrude, Fillet), which encode geometric procedures but lack explicit information about design intent, functional semantics, continuity requirements, or engineering constraints. These high-level concepts are essential for inferring B-spline priors but cannot be captured directly from symbolic code alone. To address this gap, we introduce LLM-generated rich-text semantic descriptions that augment the modeling history with contextual geometric and engineering semantics, providing a meaningful inductive bias for B-spline parameter prediction. This section describes how such rich-text is generated and presents representative examples.

1.1. Generating Rich-text via LLM Prompting

To enrich the symbolic modeling history with high-level geometric semantics, we generate a structured rich-text description for each parametric modeling script using a large language model (DeepSeek). These descriptions are not intended to restate the code; instead, they provide the design intent, geometric rationale, and contextual engineering semantics that are often implicit in CAD workflows but essential for predicting B-spline surface priors. To make this process systematic and reproducible, we construct a unified prompting template that guides the LLM to interpret the modeling code from multiple complementary semantic perspectives.

Prompt Template. We adopt the following structured prompt to generate rich-text descriptions. It instructs the LLM to analyze the modeling code as a CAD expert, focusing on semantics relevant to B-spline surface generation.

```
As a professional CAD modeling expert,
    analyze the following parametric
modeling code to provide design context
    for B-spline surface generation.

[Code Content]
<parametric modeling code here>

[Analysis Requirements]
Please analyze the code from the following
    aspects:

1. Overall Design Overview
- Infer the part type, functional intent,
    and primary usage.
- Evaluate global geometric
    characteristics and overall complexity.
- Explain the modeling strategy and
    construction logic.

2. Step-by-step Modeling Logic
For each modeling operation:
- Engineering purpose: its role in the
    overall design.
- Geometric features: the geometry created
    (lines / arcs / B-splines).
- Design intent behind the operation.
- Dependencies on previous and subsequent
    steps.

3. Key Geometric Feature Analysis
- Curve-type composition in each sketch
    and main geometric parameters.
- Contour complexity, curvature behavior,
    and dimensional properties.
- Impact on subsequent surface and
    B-spline generation.

4. Surface Quality and Continuity Focus
- Regions that require high-quality or
    high-continuity surfaces.
- Complex transition and blending zones
    between features.
- Differences between aesthetic and
    functional surfaces.

5. Engineering Constraints Consideration
- Possible manufacturing process
    implications.
```

```
- Fillet / shell / wall-thickness related
    constraints.
- Quality control considerations and
    potential design optimizations.

[Output Requirements]
- Emphasize geometric semantics relevant
    to B-spline construction.
- Provide clear engineering reasoning for
    each step.
- Use concise and professional technical
    language.
- Do NOT output concrete B-spline
    parameter values.
```

Design Rationale. The prompt is structured around five semantic dimensions that are highly relevant to B-spline prior prediction.

Overall Design Overview encourages the LLM to infer global shape intent—such as whether the part functions as a structural element, a transition region, or a functional interface. These global semantics influence the expected continuity level, curvature scale, and surface role.

Process-level Semantics encourage the LLM to reinterpret the symbolic operation sequence as a causally meaningful construction process. Although operations such as `Extrude`, `Fillet`, or `Cut` are explicit in the code, their engineering purposes are not. Understanding whether an extrusion defines the primary load-bearing volume or an intermediate boolean feature directly affects how the model interprets the geometric complexity and surface behaviors associated with each step.

Geometric-feature Semantics help the LLM identify curve composition, curvature distribution, and contour complexity—factors that strongly affect the expected degree, pole count, and smoothness of the fitted B-spline surface. Symbolic modeling code alone does not reveal curvature characteristics, so this semantic extraction adds crucial geometric context.

Surface Continuity and Quality Semantics capture whether specific regions require G0, G1, or G2 transitions, or involve complex curvature blending. Many CAD operations create surfaces with implicit continuity requirements that do not directly appear in the code. Making these requirements explicit helps the model form more accurate priors for continuity class and curvature behavior.

Engineering-constraint Semantics such as manufacturability, tool-radius limitations, wall-thickness considerations, or assembly-related cuts provide valuable signals for determining surface roles and curvature scales. These constraints reflect real-world design decisions that shape the underlying surface geometry but are not encoded in the symbolic operation history.

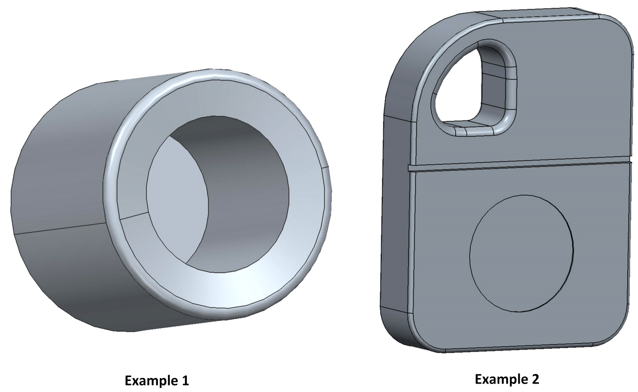

Figure 1. **Rich-text Examples.**

Together, these semantic dimensions transform a purely symbolic modeling script into a language-based representation carrying explicit geometric, functional, and engineering intent. This enriched representation introduces a meaningful inductive bias for B-spline parameter prediction.

1.2. Rich-text Examples

To make the effect of our prompting scheme more concrete, we provide representative examples of LLM-generated rich-text descriptions conditioned on parametric modeling code. Each example pairs a short code snippet with the corresponding semantic analysis produced by the LLM. The goal is not to exhaustively list all generated text, but to illustrate the multi-paragraph structure and the type of geometric and engineering semantics that augment the symbolic modeling history.

Example 1. Code snippet.

```
sk0 = Sketch(plane={"normal": [0.0,-1.0,0.0],
                    "origin":[0,0,0],
                    "x":[1,0,0]})
p0_0 = Profile()
p0_0.addLoop(Loop()
    .moveTo(-0.04,0.0)
    .lineTo(0.0,0.0)
    .lineTo(0.0,0.00956001)
    .lineTo(-0.025,0.00956001)
    .lineTo(-0.025,0.07196412)
    .lineTo(-0.04,0.075).pointTag("JGI")
    .lineTo(-0.04,0.0)
)
sk0.addProfile(p0_0)
shape0 = Revolve(sk0,
                 axis={"point":[0,0,0.00956],
                       "direction":[0,0,-1]},
                 angle=360)
shape0.Fillet(radius=0.003,
              entities=[{"referenceId":"JGI",
                         "capType":"SWEPT"}])
```

LLM-generated rich-text description-Example 1.

Overall Design Overview. *Part Type & Function:* Small rotational mechanical component, likely a precision shaft shoulder or bearing seat with filleted transitions. The 0.04-0.075m dimensional range suggests industrial machinery scale.

Geometric Characteristics: Axisymmetric revolution with one critical fillet transition. Moderate complexity due to the stepped profile and specific fillet requirement.

Modeling Strategy: Classic revolve-first approach with post-fillet modification. Conservative parametric methodology suitable for manufacturing-driven design.

Modeling Logic Analysis. *Step 1 – Sketch Definition:* -Engineering purpose: Establish 2D cross-section for rotational generation - Geometric features: XY-plane profile with normal [-0,1,0] orientation, containing 6 linear segments forming closed polygonal contour - Design intent: Create manufacturable cross-section with clear dimensional references - Dependencies: Foundation for all subsequent operations

Step 2 – Profile Construction: - Engineering purpose: Define precise boundary path for revolution - Geometric features: Rectilinear profile with one tagged vertex (J̈GÏ) at coordinate (-0.04, 0.075) - Design intent: Establish reference point for subsequent fillet operation while maintaining dimensional control - Dependencies: Directly builds on sketch coordinate system

Step 3 – Revolve Operation: - Engineering purpose: Generate 3D solid from 2D profile - Geometric features: Full 360° revolution about Z-axis through point (0,0,0.00956) - Design intent: Create axisymmetric part with consistent cross-section - Dependencies: Transforms 2D profile into manufacturable 3D geometry

Step 4 – Fillet Application: - Engineering purpose: Stress concentration reduction and manufacturing feasibility - Geometric features: 3mm radius applied to swept edge referenced by J̈GIẗag - Design intent: Transition smoothing between vertical and horizontal surfaces - Dependencies: Requires stable edge geometry from revolve operation

Key Geometric Feature Analysis. *Sketch Geometric Features:* - 100% linear segments (6 connected lines) - Contour complexity: Low (simple polygonal shape) - Critical dimensions: 40mm base width, 9.56mm initial height, 75mm maximum height - Profile impact: Creates sharp transitions requiring post-processing

Modification Operation Features: - Fillet location: Transition between vertical (-0.04,0.0→-0.04,0.075) and horizontal (-0.025,0.07196→-0.04,0.075) surfaces - Radius: 3mm - moderate for the scale, suitable for casting or machining - Engineering purpose: Stress flow optimization and tool access consideration - Continuity requirements: G1 continuity sufficient for functional surfaces

Surface Quality Focus Areas. *High Continuity Regions:* Fillet transition zone requires smooth G2 continuity for stress distribution

Complex Transition Areas: Junction at point (-0.04,0.075) where fillet meets two planar surfaces

Aesthetic vs Functional: Primarily functional surfaces - fillet quality critical for mechanical performance

Surface Generation Challenges: Sharp corner at (-0.025,0.07196) before fillet may cause uneven parameterization

Engineering Constraints Consideration. *Manufacturing Process:* Likely CNC machining or precision casting based on 3mm fillet radius and dimensional tolerances.

Quality Control: Fillet blend quality, revolution symmetry, dimensional accuracy at stepped transitions

Design Optimization: Consider adding draft angles for casting or adjusting fillet size based on load analysis

B-spline Context: The revolved surfaces provide excellent foundation for B-spline conversion with inherent continuity at the axis of revolution

Example 2. Code snippet.

```
sk0 = Sketch(plane={"normal": [0.0, 0.0, 1.0],
                    "origin": [0.0, 0.0, 0.0],
                    "x": [1.0, 0.0, 0.0]})
p0_0 = Profile()
p0_0.addLoop(Loop()
    .moveTo(0.0,-0.04)
    .lineTo(0.03,-0.04).pointTag("JGM")
    .lineTo(0.03,0.0).pointTag("JGE")
                    .curveTag("JGN")
    .lineTo(0.0,0.0).pointTag("IG")
    .lineTo(0.0,-0.04).pointTag("JGI")
)
sk0.addProfile(p0_0)
shape0 = Extrude(sk0, distance=0.008)
shape0.Fillet(radius=0.012,
              entities=[{"referenceId": "IG",
                         "capType": "SWEPT"}])
shape0.Fillet(radius=0.005,
              entities=[{"referenceId": "JGE",
                        "capType": "SWEPT"},
                        {"referenceId": "JGM",
                        "capType": "SWEPT"},
                        {"referenceId": "JGI",
                        "capType": "SWEPT"}])
shape0.Fillet(radius=0.0005,
              entities=[{"referenceId": "JGN",
                        "capType": "END"},
                        {"referenceId": "JGN",
                        "capType": "START"}])

sk1 = Sketch(plane={"normal": [0.0, 0.0, 1.0],
                    "origin": [0.0, 0.0, 0.008],
                    "x": [1.0, 0.0, 0.0]})
p1_0 = Profile()
p1_0.addLoop(Loop()
    .moveTo(0.003,-0.013)
    .lineTo(0.013,-0.013).pointTag("JLs")
    .lineTo(0.013,-0.003)
    .lineTo(0.012,-0.003).pointTag("JLk")
    .threePointArc((0.00563604,-0.00563604),
                   (0.003,-0.012)).pointTag("JLg")
                    .curveTag("JLZ")
```

```
    .lineTo(0.003,-0.013)
)
sk1.addProfile(p1_0)
shape1 = Extrude(sk1, distance=-0.025)
shape1.Fillet(radius=0.003,
              entities=[{"referenceId": "JLk",
                         "capType": "SWEPT"},
                         {"referenceId": "JLg",
                         "capType": "SWEPT"}])
shape1.Fillet(radius=0.002,
              entities=[{"referenceId": "JLs",
                         "capType": "SWEPT"}])
shape1.Fillet(radius=0.001,
              entities=[{"referenceId": "JLZ",
                         "capType": "END"}])
shape0 = shape0.cut(shape1)

sk2 = Sketch(plane={"normal": [0.0, 0.0, 1.0],
                    "origin": [0.0, 0.0, 0.008],
                    "x": [1.0, 0.0, 0.0]})
p2_0 = Profile()
p2_0.addLoop(Loop()
    .moveTo(0.0,-0.018)
    .lineTo(0.03,-0.018)
    .lineTo(0.03,-0.0172)
    .lineTo(0.0,-0.0172)
    .lineTo(0.0,-0.018)
)
sk2.addProfile(p2_0)
shape2 = Extrude(sk2, distance=-0.001)
shape0 = shape0.cut(shape2)

sk3 = Sketch(plane={"normal": [0.0, 0.0, 1.0],
                    "origin": [0.0, 0.0, 0.008],
                    "x": [1.0, 0.0, 0.0]})
p3_0 = Profile()
p3_0.addLoop(Loop()
    .moveTo(0.01506018,-0.02837698)
    .circle(0.00705001)
)
sk3.addProfile(p3_0)
shape3 = Extrude(sk3, distance=-0.0002)
shape0 = shape0.cut(shape3)
# End of code
```

LLM-generated rich-text description-Example 2.

Overall Design Overview. *Part Type & Function:* This is a precision mechanical component with layered features, likely a micro-mechanical bracket, electrical contact, or small-scale structural interface. The design combines thin-wall structures with precision cutouts and fillets, suggesting potential applications in connectors, sensor mounts, or miniature mechanism interfaces.

Geometric Characteristics: Multi-level extrusion architecture with base plate (0.008 thickness) and subtractive features. The design exhibits mixed 2D/3D complexity with rectangular profiles, circular cutouts, and complex corner transitions.

Modeling Strategy: Bottom-up constructive approach using sequential Boolean operations. The methodology employs: Primary base creation followed by precision feature subtraction. Strategic fillet application for stress reduction and manufacturability. Layered sketching on parallel planes for controlled feature depth.

Modeling Logic Analysis. *Step 1 – Base Plate Creation:* Engineering Purpose: Establish primary structural foundation and mounting surface. Geometric Features: 30×40 mm rectangular profile (0.03×0.04 units) with significant corner radii (R12, R5, R0.5). Design Intent: Provide robust base geometry for subsequent feature integration while ensuring stress distribution through generous fillets. Dependencies: Foundation for all subsequent Boolean operations.

Step 2 – Primary Cutout Feature: Engineering Purpose: Create functional clearance or mechanical interface. Geometric Features: Complex L-shaped profile with arc transition ($R\approx7$ mm via three-point arc), depth penetration of 0.025. Design Intent: Generate precise internal geometry with smooth transitions for component clearance or motion path. Dependencies: References base plate Z-height (0.008), creates main internal cavity.

Step 3 – Secondary Slot Feature: Engineering Purpose: Add thin slot for alignment, ventilation, or wiring passage. Geometric Features: Simple rectangular slot (30×0.8 mm, depth 0.001). Design Intent: Create minimal material removal for specific functional requirement. Dependencies: Positioned at same Z-level as primary cutout start plane.

Step 4 – Circular Cutout: Engineering Purpose: Precision hole for fastener, bearing, or electrical contact. Geometric Features: 14.1 mm diameter circle (radius 7.05), shallow depth (0.0002). Design Intent: Create precise circular interface with minimal material engagement. Dependencies: Final subtraction operation affecting overall structural integrity.

Key Geometric Feature Analysis. *Sketch Geometric Composition:* Sketch 0 uses 100% linear segments and defines the simple rectangular foundation. Sketch 1 mixes three lines with a three-point arc, creating a complex transition profile. Sketch 2 again uses four linear segments for the slot geometry, while Sketch 3 consists of a single circular element that defines the radial feature.

Modification Operation Characteristics: Primary fillets (R12, R5) act as major stress concentration mitigation at base plate corners. Secondary fillets (R3, R2) treat internal corners for manufacturability. Micro fillets (R0.5, R0.001) represent edge break requirements for precision components. The resulting surface network exhibits mixed G0/G1 transitions that require careful surface parameterization.

Surface Quality Focus Areas. *High-continuity* regions include the three-point arc transition in the primary cutout (tag JLZ) and the large-radius fillets on the base plate corners. Complex transition zones occur at the intersection between the primary cutout and the base plate fillets, around convergence points of multiple fillets at tags JGE, JGM, and JGI, and near the shallow circular cutout edges.

Table 1. Semantic differences across the three rich-text descriptions (Samples A/B/C).

Semantic Category	**Sample A**	**Sample B**	**Sample C**
Engineering Positioning	Precision guiding / fluid–structure transition	High-precision mechanical component, possibly optical or aerodynamics continuity	Moderate-complexity sliding / structural component
Curvature Description	Smooth transition, controlled curvature	High curvature complexity, emphasis on G2/G3	Moderate curvature, visually regulated
Manufacturing Implication	Precision machining or small molds	High-precision machining, strict tolerance	Standard machining / regular molds
Continuity Requirement	**G1** mainly	**G2+** or higher-order continuity	**G1** sufficient
Surface Functionality	General guiding / flow transition	High-load or functional surface	Normal contact / structural stability
Complexity Tone	Neutral (mixed complexity)	Biased toward complexity	Simplified, uniform

Surface Quality Differentiation: Aesthetic surfaces are mainly the external faces of the base plate, which require smooth continuity, while functional surfaces include the internal cutout walls where precise geometry is more critical than high-order continuity. Transition challenges arise from mixed-radius fillet intersections, which generate complex curvature distributions.

Engineering Constraints Consideration. *Manufacturing Process Inference:* Precision machining or micro-molding suggested by tight tolerances (0.0002-0.001 features). Multi-axis machining required for complex internal geometries. EDM or laser cutting potential for intricate cutout features.

Quality Control Focus: Dimensional verification of critical interfaces (circular cutout Ø14.1mm). Fillet radius consistency at stress concentration points. Wall thickness integrity after multiple Boolean operations.

Design Optimization: Consider consolidating sketch planes to reduce computational complexity. Evaluate fillet hierarchy to minimize sharp transitions in high-stress regions. Assess potential for feature patternization if multiple instances exist in final assembly.

2. Semantic Diversity of Rich-Text Descriptions and Its Influence on Geometric Priors

This section investigates how different natural-language descriptions, paired with the same modeling code, influence the B-spline priors inferred by the large language model. Because our framework relies on linguistic reasoning rather than syntactic patterns alone, variations in how the design intent is articulated may shift the model's predictions on surface degree, control-point layout, curvature scale, and other structural parameters. To isolate this effect, we apply three semantically distinct rich-text descriptions—denoted Sample A, Sample B, and Sample C—to the same modeling script and analyze how their semantic differences propagate into geometric predictions.

To keep the semantic–geometric comparison grounded, all three rich-text descriptions (Samples A/B/C) are paired with the same underlying modeling script. The script defines a closed sketch composed of two line segments and two three-point arcs, extruded to a thin solid and finished with eight fillets applied to the tagged curve segments. This script is shown below:

```
sk0 = Sketch(plane={"normal": [0.0, -1.0, 0.0],
                    "origin": [0.0, 0.0, 0.0],
                    "x": [1.0, 0.0, 0.0]})
p0_0 = Profile()
p0_0.addLoop(Loop()
    .moveTo(-0.05,0.0)
    .lineTo(0.0,0.0).curveTag("JGB")
    .threePointArc((0.01561553,0.02), (0.0,0.04))
            .curveTag("JGR")
    .lineTo(-0.05,0.04).curveTag("JGF")
    .threePointArc((-0.06561553,0.02), (-0.05,0.0))
            .curveTag("JGV")
)
sk0.addProfile(p0_0)
shape0 = Extrude(sk0, distance=0.035)
shape0.Fillet(radius=0.002, entities=[
    {"referenceId": "JGF", "capType": "END"},
    {"referenceId": "JGB", "capType": "END"},
    {"referenceId": "JGV", "capType": "END"},
    {"referenceId": "JGR", "capType": "END"},
    {"referenceId": "JGF", "capType": "START"},
    {"referenceId": "JGV", "capType": "START"},
    {"referenceId": "JGR", "capType": "START"},
    {"referenceId": "JGB", "capType": "START"}])
```

Figure 2 shows the corresponding geometry.

2.1. Semantic Variations Across Descriptions

Although all three rich-text descriptions reference the same modeling code and follow a similar narrative template, each

Table 2. Trends in B-spline priors generated from different rich-text descriptions.

Feature	**Sample A**	**Sample B**	**Sample C**
Degree Trend (U/V)	4–8 mixed	8 or 5–6 (biased high)	All degree 4 (lowest)
Control-Point Trend	5–7	6–9	All 5
Curvature Class	medium/high mixed	**high (dominant)**	medium
Patch Consistency	Medium (visible variation)	Medium-high consistency	**Highest consistency**
Prior Complexity	**Medium-high**	**High**	**Low / conservative**

imposes a distinctly different semantic framing of the geometry. Sample A adopts a neutral and moderately technical tone, describing the object primarily as a precision guiding or fluid-transition component with smooth curvature and functional but not overly complex surfaces. Sample B, in contrast, introduces a noticeably more demanding engineering interpretation: it characterizes the same geometry as a high-performance mechanical or optical component, stresses G2/G3-level continuity, and implies stricter manufacturing tolerances. Sample C takes the opposite direction, presenting the geometry as a more regular, stable, and conventional structural piece with predictable curvature and standard manufacturability.

Table 1 distills these differences into key semantic categories. Rather than numerical information, the table highlights the qualitative tendencies—such as curvature emphasis, continuity demands, functional framing, and implicit manufacturing difficulty—that differentiate the three descriptions. Sample B consistently signals the highest surface sophistication, Sample C the most regularized and stable interpretation, while Sample A remains centered between them without strongly pushing complexity in either direction.

2.2. Influence on B-Spline Prior Prediction

These semantic differences directly influence the B-spline priors generated by the model. Even though the modeling code is identical, the inferred surface parameters diverge systematically across the three descriptions. Sample B—motivated by its emphasis on higher-order continuity, complex curvature, and high-performance roles—leads the model to predict higher B-spline degrees, larger ranges of control points, and curvature classes dominated by "high." Sample C, shaped by its preference for regularity and stability, consistently produces low-degree patches with uniform control-point counts and smooth, predictable curvature behavior. Sample A again occupies the middle ground, producing priors with moderate degrees and mixed curvature and control-point trends.

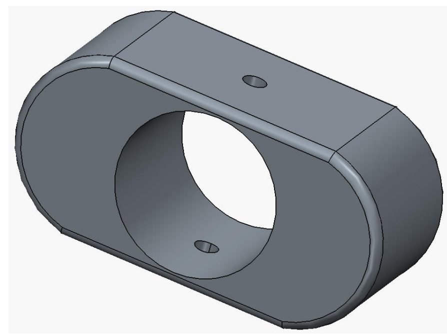

Figure 2. Base geometry in the semantic–prior analysis.

Table 2 summarizes these structural tendencies. Instead of exact solver parameters, the table presents the predicted ranges of degrees, control-point counts, and curvature classes. These coarse numerical ranges reflect how the inferred B-spline structures vary under different semantic framings. The results show a clear semantic-to-geometry mapping: descriptions emphasizing high-performance or high-continuity behavior lead to more complex B-spline structures, whereas conservative descriptions yield lower-degree and more uniform patches.

2.3. Surface Visualization Comparison

Figure 3 shows reconstructed fillet surfaces produced under the three sets of predicted B-spline priors. Although the global shapes remain close—since the modeling code prescribes the same boundary geometry—the local surface behavior does vary across Samples A, B, and C. These variations arise from differences in the predicted degrees, control-point layouts, and curvature tendencies, and they manifest as subtle but genuine changes in surface transitions and local smoothness. Standard shaded renderings make these distinctions visually modest, yet the underlying parameterizations are materially different.

The visualization thus complements the structural trends reported in Table 2. The three semantic descriptions lead to clearly distinct B-spline priors, which in turn produce

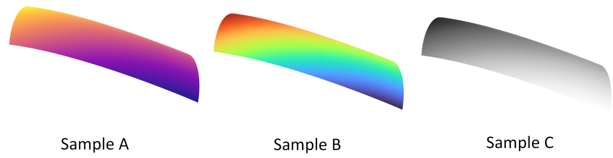

Figure 3. Comparison of reconstructed fillet surfaces generated using priors inferred from Samples A, B, and C. While the global shapes are similar due to identical modeling code, the underlying B-spline parameterizations differ, producing subtle but genuine variations in local smoothness and surface transitions.

localized geometric deviations even when the global form remains nearly unchanged. This alignment between numerical trends and rendered behavior reinforces that rich-text semantics do influence the resulting surface formulation in a meaningful and measurable way.

2.4. Local Surface Differences Under Distinct Semantic Priors

To complement the subtle variations observed in the global fillet reconstruction, we further illustrate a case where semantic prompting produces more pronounced geometric differences. Figure 4 compares two surfaces generated from distinct semantics—Smooth-transition intent (Semantic Prior A) and Functional-detail intent (Semantic Prior B). Although both are reconstructed from the same modeling history, their local curvature behavior differs significantly.

Under the smooth-transition prior, the resulting surface favors evenly distributed curvature and uniformly blended transitions. In contrast, the functional-detail prior introduces sharper curvature changes and more localized variation, reflecting its richer geometric interpretation. The zoomed-in regions in the figure highlight these deviations clearly: Semantic Prior B exhibits stronger directional curvature gradients and more expressive surface modulation, whereas Semantic Prior A remains smoother and more homogeneous.

These examples provide direct visual evidence that semantic cues do not merely shift numerical B-spline parameters—they meaningfully alter the resulting geometric behavior, particularly in regions where curvature or blending is semantically emphasized.

3. Dataset Statistics and Distribution Analysis

This section provides a comprehensive statistical overview of the B-spline surfaces extracted from our parsed modeling histories. The goal is not only to document the scale and diversity of the dataset, but also to reveal the structural regularities in geometric attributes and semantic properties that motivate the need for language-aware prior prediction.

Table 3. Operation type distribution and probability of producing B-spline surfaces.

Operation Type	Count	B-spline Produced	Probability
Fillet	40,165	38,491	95.83%
Chamfer	1,642	1,543	93.97%
Shell	639	375	58.69%
Boolean	65,032	0	0.00%

3.1. Geometric Attribute Distributions

Across all parsed files, we obtain a total of **121,811 unique B-spline surfaces**. The underlying modeling histories show wide variability—some scripts contain only a few operations, whereas others produce dozens of intermediate surfaces—indicating substantial structural complexity. When examining all parametric operations, we observe that only a subset actually yields B-spline surfaces. Boolean operations (the majority of steps) rarely contribute new B-splines, whereas fillet, chamfer, and shell operations almost always produce them. This reveals a natural, operation-level asymmetry: the presence of a B-spline surface is not uniformly distributed but tied to specific semantic categories.

Operation statistics. Table 3 summarizes the occurrence frequency of each operation type and the probability that it produces a B-spline surface. Fillet and chamfer steps exhibit extremely high generation probabilities (>93%), whereas boolean operations produce none. This highlights the semantic dependency between modeling intent and expected surface type.

Degree distributions. Figure 5 shows the distributions of B-spline degrees in the U and V directions. The U-degree distribution is highly concentrated at degree 2 (~92,000 instances), revealing a strong bias toward quadratic behavior. In contrast, the V-degree distribution is far more diversified, spanning degrees 2–14, with modes at 2, 5, and 14. This asymmetric pattern indicates different curvature behaviors and structural roles along the two parametric directions.

Control point distributions. Figure 6 visualizes control point counts along U and V. In the U direction, nearly all surfaces contain exactly three control points, forming an extremely narrow unimodal pattern. Meanwhile, the V direction exhibits a long-tailed distribution ranging from 5 to over 40 control points. This contrast suggests that the dataset contains surface families with relatively simple U-direction structure but highly variable V-direction complexity.

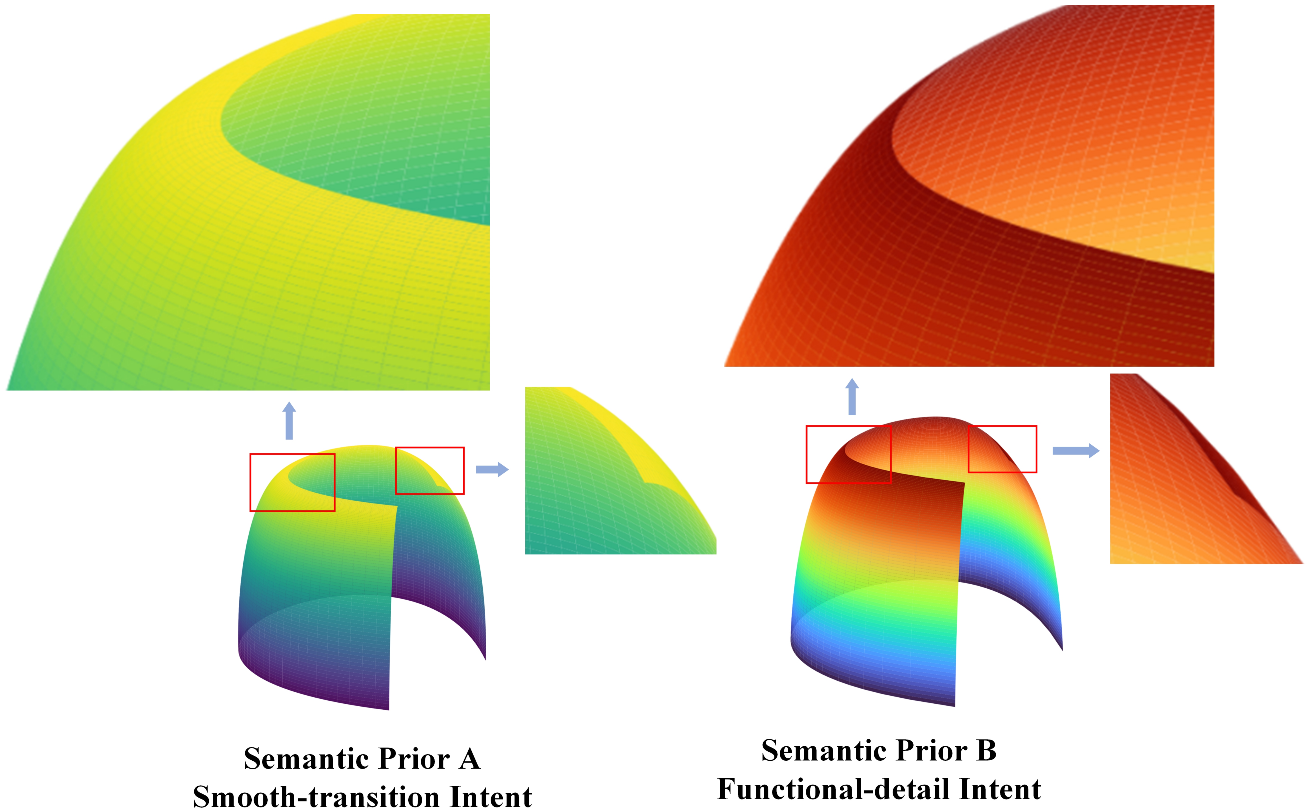

Figure 4. Local geometric differences induced by semantic priors. Both surfaces are reconstructed from the same procedural modeling history, but the "Smooth-transition" prior (left) results in uniform curvature and gentle blending, whereas the "Functional-detail" prior (right) produces more localized curvature variation. The zoomed-in regions highlight how semantic cues alter the underlying B-spline behavior.

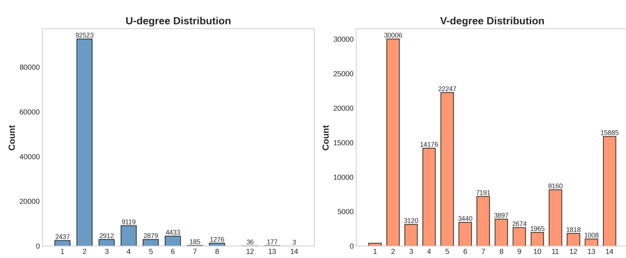

Figure 5. **Degree distributions.** (Left) U-degree is sharply peaked at 2, reflecting a dominant structural pattern. (Right) V-degree shows richer variability, covering degrees 2–14.

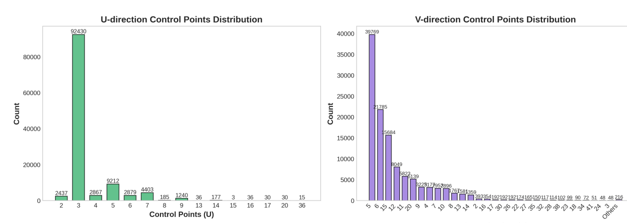

Figure 6. **Control point distributions.** (Left) U-direction control points are heavily concentrated at 3. (Right) V-direction exhibits long-tail behavior with substantial variability.

B-spline generation by shape index. Figure 7 shows the distribution of B-spline counts across "shape indices"—the internal identifiers of intermediate solids created throughout modeling histories. A single dominant shape (shape0) accounts for over 80,000 B-spline surfaces, reflecting that early-stage modeling steps often introduce numerous fillets and transitional surfaces. Subsequent shapes rapidly decrease in B-spline count, indicating that later steps tend to perform local refinements or boolean operations with minimal new surface creation.

3.2. Semantic Attribute Distributions

In addition to geometric parameters, each B-spline surface is associated with semantic attributes inferred from geometric regularities, including rationality, trimming status, con-

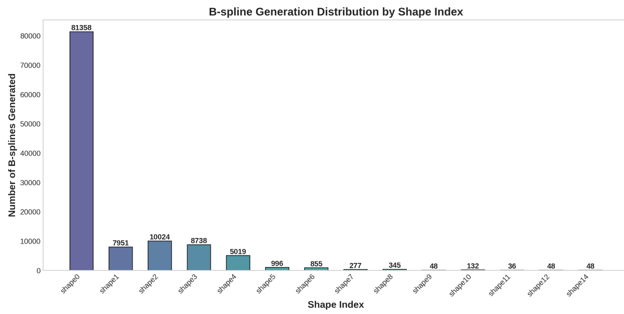

Figure 7. **B-spline generation by shape index.** B-spline creation is heavily concentrated in the earliest shapes, where most transitional and blending surfaces are introduced.

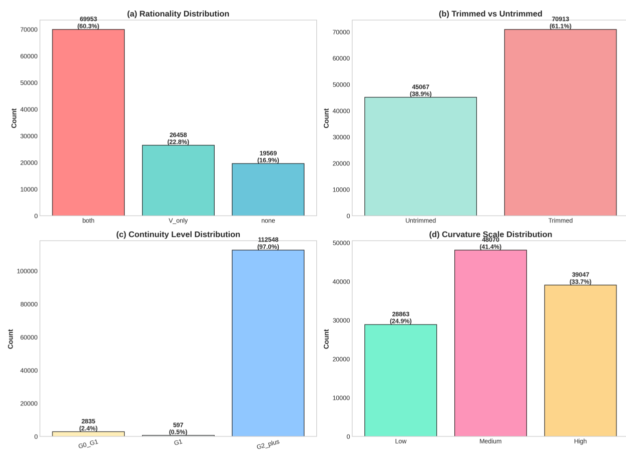

Figure 8. **Semantic attribute distributions.** (a) Rational vs. non-rational surfaces. (b) Trimmed vs. untrimmed distributions. (c) Continuity classes dominated by $G2$ surfaces. (d) Curvature scale categories.

tinuity classification, and curvature scale. These attributes capture behavior that is not directly observable from symbolic operation code but is crucial for accurate surface reconstruction.

Figure 8 summarizes their distributions. Most surfaces are rational in both parametric directions, and the majority are trimmed, reflecting the prevalence of boolean-induced boundaries. Continuity levels are overwhelmingly in the G^2 category, which is consistent with the dominance of fillet transitions observed earlier. Curvature scale spans low, medium, and high regimes, indicating substantial diversity in geometric behavior across the dataset.

Overall, these distributions reveal strong structural regularities—particularly in degrees, control points, and continuity levels—while also demonstrating considerable variability in shape complexity and curvature. Such characteristics highlight the need for semantic priors: many parameters cannot be reliably inferred from symbolic modeling code alone, and rich-text guidance plays a key role in resolving these ambiguities.

4. Implementation and Reproducibility

This section provides implementation details of the two–stage training framework and clarifies how the prompting strategy is designed to support geometric learning and semantic reasoning. Since the prediction task involves both structural parameters of B-spline surfaces and high-level semantic attributes derived from modeling intent, the training pipeline is organized into two complementary stages. Each stage uses a prompting scheme tailored to its learning objective, and the overall design reflects our goal of making the model acquire geometric regularities before engaging in semantic inference.

4.1. Stage I: Geometric Parameter Learning from Isolated Operations

The first stage focuses solely on the structural aspects of B-spline generation. Each data sample consists of a *single* CAD operation (e.g., `Fillet`, `Chamfer`, `Shell`), independent of its modeling context. This design eliminates ambiguities arising from long-range dependencies and enables the model to learn the mathematical constraints behind degrees, control-point counts, and rationality flags.

The prompt used in Stage I is intentionally restrictive and avoids any semantic content, allowing the model to concentrate on numerical consistency:

```
You are a CAD surface modeling expert. Given a single
CAD operation in isolation, determine whether this
operation directly creates a new B-spline surface.
If it does, predict only the surface's core structural
parameters (UDegree, VDegree, NbUPoles, NbVPoles, and
rationality flags). Do not infer control points or knots.
Return only JSON.
```

By removing references to modeling history, functional intent, or trimming semantics, Stage I becomes a pure geometric-learning process. The structured nature of the predictions helps stabilize training and establishes a reliable foundation for the semantic reasoning introduced in Stage II.

4.2. Stage II (Training): Joint Modeling of Code Structure and Semantic Descriptions

The second stage expands the input from isolated operations to full modeling scripts. Unlike Stage I, here the model must reason jointly over two complementary inputs:

(1) the parametric modeling code, which supplies operation order, shape indexing, and function-call patterns; (2) the rich-text semantic description, which captures functional intent, continuity expectations, trimming behavior, and surface roles.

The prompt explicitly introduces these semantic cues and instructs the model to link them to geometric outcomes:

```
The model receives two inputs: (1) the parametric model-
ing code, which defines operation order, shape indices,
and function call patterns; and (2) a rich-text semantic
description that explains the functional intent behind
each step. The model must combine both sources to infer
whether an operation introduces new B-spline surfaces,
and predict attributes such as trimming, closure, con-
tinuity class, curvature scale, and surface role. The
output is restricted to a fixed JSON structure used for
downstream geometric validation.
```

In this stage the model is encouraged to treat language as an interpretable signal for structural prediction. References in the rich-text such as "smooth transition", "critical interface", or "high curvature region" become soft constraints that influence continuity class, curvature scale, or trimming behavior. This enables the model to form stable semantic-to-geometric correspondences without requiring explicit supervision for every attribute.

4.3. Stage II (Inference): Strict JSON Output for Deterministic Evaluation

During inference we adopt a stricter prompt to ensure that the prediction results are deterministic and fully machine-parsable. The model is no longer permitted to output explanations or free-form text; instead, it must produce a fixed JSON schema so that each predicted surface can be passed to downstream geometric solvers:

```
Return ONLY valid JSON with fields:
{
  "shape_index": ...,
  "operation_sequence": ...,
  "operation_type": ...,
  "produces_bspline": ...,
  "num_new_bsplines": ...,
  "bspline_params_pred": [
    { UDegree, VDegree, NbUPoles, NbVPoles,
      IsURational, IsVRational, IsTrimmed,
      IsUClosed, IsVClosed, HasSeamEdge,
      ContinuityClass, ExpectedCurvatureScale,
      SurfaceRole }
  ]
}
```

The inference prompt acts as a formatting contract rather than a learning signal. By requiring strict structural alignment, it ensures that evaluations are reproducible and can be consistently integrated with geometric kernels or diagnostic tools.

4.4. Rationale Behind the Two-Stage Design

While the final task requires unified prediction of all geometric and semantic attributes, separating the learning process into two stages is essential. Properties such as polynomial degree and control-point counts exhibit strong geometric regularity and benefit from isolated training without semantic noise. In contrast, attributes such as continuity level, trimming status, and surface role depend on design intent and contextual interpretation, which emerge only when modeling code is paired with rich semantic descriptions.

Table 4. **Controlled comparisons under matched fine-tuning and backbone settings.** All variants are evaluated using the same metrics as in the main text. These results complement the ablation study by showing that the gains of LASP are not explained by model scale alone.

Model	Pole MAE↓	Cont F1↑	Role F1↑
Full LASP	**0.56**	**0.70**	**0.98**
Qwen3-14B (history-only)	0.61	0.59	0.86
Qwen3-7B (2-stage)	0.58	0.64	0.94
LLaMA2-13B (2-stage)	0.57	0.68	0.97
LLaMA2-13B (w/o Sem.)	0.60	0.61	0.89

The two-stage framework therefore allows the model to first internalize the structural constraints of B-spline geometry, and then refine its predictions through semantically-informed reasoning. This staged approach substantially improves stability and generalization, especially when dealing with long modeling histories and varied geometric contexts.

5. Additional Controlled Comparisons

This section provides additional controlled comparisons that complement the analyses reported in the main text. The first subsection reports matched fine-tuning and backbone comparisons to further disentangle the effects of semantic conditioning, staged training, and model scale. The second subsection reports several representative solver-supported configuration settings as supplementary references. These results are intended to complement, rather than replace, the main-text evaluations.

5.1. Matched Training and Backbone Comparisons

To further disentangle the source of LASP's gains, we report additional controlled comparisons under matched fine-tuning and backbone settings. These results complement the main-text ablation study by separating the effects of semantic conditioning, staged training, and model scale.

As shown in Table 4, the full two-stage LASP design consistently outperforms the history-only and semantic-reduced variants, indicating that the observed gains are not attributable to backbone scale alone. The reduced-capacity and alternative-backbone results further suggest that the benefits of semantic conditioning and staged training are stable across different model settings.

5.2. Additional Solver Configuration Comparisons

We additionally report several representative solver-supported configuration settings as supplementary references. These settings operate at a different problem level from LASP, since their choice and parameterization are themselves intent-dependent. They are therefore not treated as primary baselines in the main text, where the focus is

Table 5. **Additional solver-supported configuration comparisons.** These results are provided as supplementary references to distinguish solver strategy choices from the intent-to-configuration mapping studied in the main text.

Configuration	**RMS↓**	**HD↓**	**Med↓**
Default	0.0491	0.2746	0.0193
H-Strong	0.0722	0.3267	0.0271
EDR	0.0423	0.2485	0.0179
LASP	0.0141	0.0395	0.0103

on translating semantic context into solver-usable configuration.

As shown in Table 5, alternative kernel-supported settings may yield limited or inconsistent gains, whereas LASP provides substantially better fitting quality through semantically guided configuration. These results support the interpretation that LASP addresses an intent-to-configuration mapping problem, rather than a benchmark over solver strategies themselves.