

Language-Augmented Semantic Priors for B-Spline Surface Fitting

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

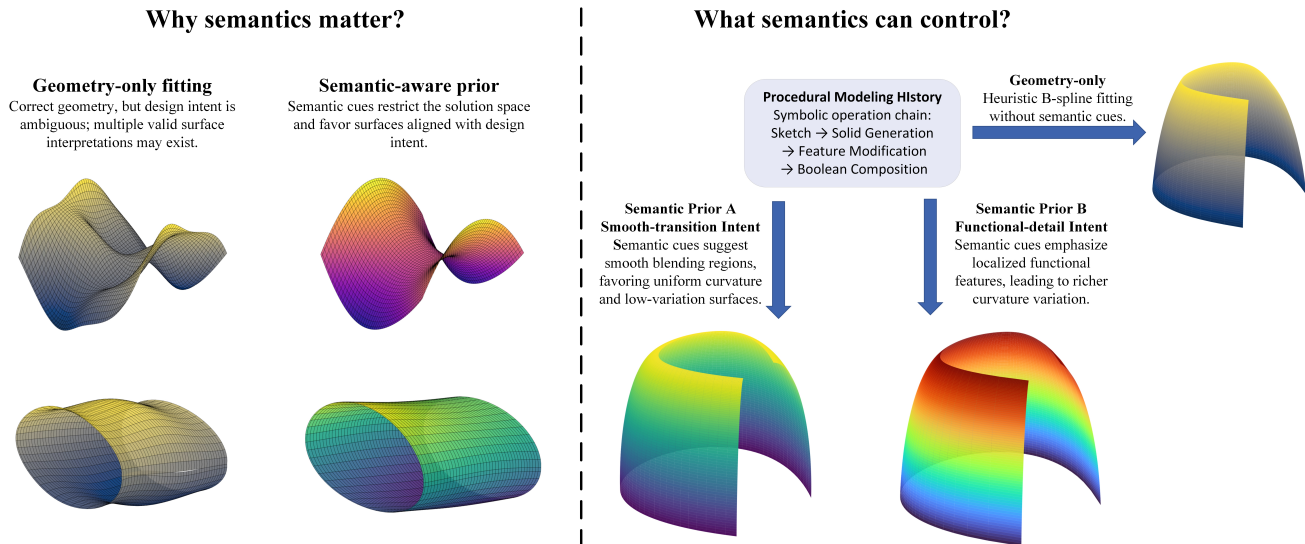


Figure 1. **Semantic information meaningfully constrains the geometric interpretation of B-spline surfaces.** When relying solely on geometric cues, a solver may produce multiple equally valid but intent-agnostic surface configurations. Incorporating semantic cues—such as smooth-transition intent or functional-detail emphasis—reduces this ambiguity and steers the fitted surface toward characteristics aligned with design intent.

Abstract

The use of B-splines and Non-Uniform Rational B-Splines (NURBS) surfaces constitutes the mathematical foundation of contemporary three-dimensional computer-aided design (CAD) systems. Despite long-term progress, geometric kernels in traditional CAD still rely heavily on predetermined heuristic initialization for surface fitting and parameterization. Meanwhile, the rich procedural semantics and design intent encoded in modeling histories (i.e., sequences of design operations) are largely ignored during geometry generation. This disconnect creates a critical gap between high-level design intent and solver-executable geometric configuration, often leading to suboptimal and semantically inconsistent fitting results. To bridge this gap, we introduce **LASP**, a Language-Augmented Semantic Priors framework that leverages large language models (LLMs) to infer structured, solver-usable B-spline priors from procedural modeling histories. Rather than modifying the geometric ker-

nel itself, **LASP** operates as a semantic reasoning layer above existing solvers. It first translates modeling histories into rich textual descriptions that capture design intent, geometric context, and functional relationships, and then uses a fine-tuned LLM to predict structured B-spline prior parameters. **LASP** is trained through a two-stage scheme that combines local geometric regularities with long-range contextual dependencies, producing priors that are both interpretable and semantically coherent. This approach furnishes enhanced inductive signals that direct the conventional B-spline fitting process toward solutions that more accurately encapsulate the intended design objectives and demonstrate heightened semantic coherence. Compared to traditional machine learning schemes, the experiments demonstrate that language-driven reasoning can serve as a powerful inductive bias for geometric solving, establishing a new paradigm of language-guided geometric optimization in modern CAD systems.

1. Introduction

B-spline and Non-Uniform Rational B-Splines (NURBS) surfaces provide the mathematical backbone of modern CAD systems [7, 14, 27], supporting smooth, editable, and topologically consistent geometric representations. Commercial kernels such as Parasolid and ACIS, as well as open-source systems like OpenCascade [31], implement decades of carefully engineered numerical routines to ensure robustness and validity in geometric computation. Despite their maturity, these solvers depend on fixed heuristic initialization—predefined degrees, uniform knot vectors, and static control-point layouts—before executing deterministic fitting procedures. While effective for stability, such initialization limits the solver to a narrow region of its feasible parameter space and prevents it from adapting to the semantics of the modeling operation. Procedural commands such as *Fillet*, *Shell*, or *Loft* encode rich information about curvature transitions, functional intent, and local geometry [15, 42, 43], yet this knowledge is rarely translated into solver-usable geometric configurations for downstream surface generation.

Closing this gap requires connecting two fundamentally different representations. Modeling histories are symbolic, rule-based, and hierarchical, as captured in recent analyses of parametric workflows [18]; geometric solvers, by contrast, operate over continuous numeric manifolds. This structural mismatch makes it difficult for solvers to exploit procedural context, even though such context often determines the intended geometric behavior. Prior learning-based approaches typically regress geometric parameters or reconstruct surfaces directly [38, 46, 49], leaving the solver unchanged and isolated from semantic guidance. Existing AI4CAD studies have predominantly focused on modeling-history reconstruction, or generating modeling histories and geometric outputs from design intent [24, 26], while the problem of translating designer-side semantics, or practical semantic proxies of them, into geometry-governing and solver-usable configurations remains largely underexplored. A more principled direction is therefore to endow solvers with context-conditioned priors that guide their initialization and restrict their search space toward semantically meaningful solutions—while retaining their deterministic optimization pipelines.

To address this challenge, we introduce **LASP** (*Language-Augmented Semantic Priors*), a framework that leverages large language models (LLMs) [5, 10] to infer semantically coherent B-spline priors from procedural modeling histories. LASP translates symbolic operations into rich-text semantic descriptions that capture multi-level cues about operation intent, geometric context, and functional relationships, aligning with recent language-driven CAD reasoning systems [39, 44, 45]. A fine-tuned LLM then predicts structured surface priors—degrees, control-point

grids, rationality, closure, trimming patterns, continuity class, and surface roles—forming a language-conditioned parameterization of B-spline surfaces. Importantly, LASP does not modify the solver itself or introduce a new geometric optimization method; instead, it operates as a semantic reasoning layer above existing geometric kernels, configuring the fitting process through solver-usable priors and biasing the numerical optimization toward geometrically valid and semantically consistent regions of the parameter space.

LASP is trained in two complementary stages. Stage I focuses on local geometric regularities by learning operation-specific parameter structures grounded in physically valid B-spline distributions [18]. Stage II extends this reasoning to full modeling histories, capturing long-range dependencies such as continuity propagation, curvature transitions, and semantic role consistency across operations. This aligns with recent advances showing that LLMs can model long-range symbolic dependencies [9, 28, 53]. Together, these stages enable the model to learn both geometric grounding and high-level design semantics.

Extensive experiments demonstrate that LASP substantially improves parameter prediction accuracy and semantic consistency over strong LLM baselines and Transformer-only architectures. We further conduct an engine-level study by mapping discrete semantic priors into solver-accessible configuration choices, showing that semantically guided parameterization significantly improves RMS, Hausdorff, and median fitting errors—highlighting the practical influence of symbolic semantics on geometric fitting behavior and complementing recent language-conditioned optimization pipelines [32, 48]. Our key contributions are summarized as follows:

- We show that semantic priors learned from procedural modeling histories serve as transferable inductive biases for geometric solvers, guiding numerical optimization toward semantically meaningful solutions without modifying solver internals.
- We present **LASP**, the first language-driven framework that predicts structured B-spline priors from symbolic modeling histories via rich-text semantic reasoning and a two-stage LLM-based learning pipeline.
- We demonstrate that semantic priors significantly improve both parameter-level prediction and solver-level reconstruction quality, providing quantitative and qualitative evidence that linguistic reasoning provides a strong inductive bias for intent-aware geometric configuration.

2. Related Work

2.1. LLMs for Structured and Symbolic Reasoning

Large language models have shown strong capabilities in processing structured and symbolic data beyond natural language [5, 10]. Transformer-based models have been

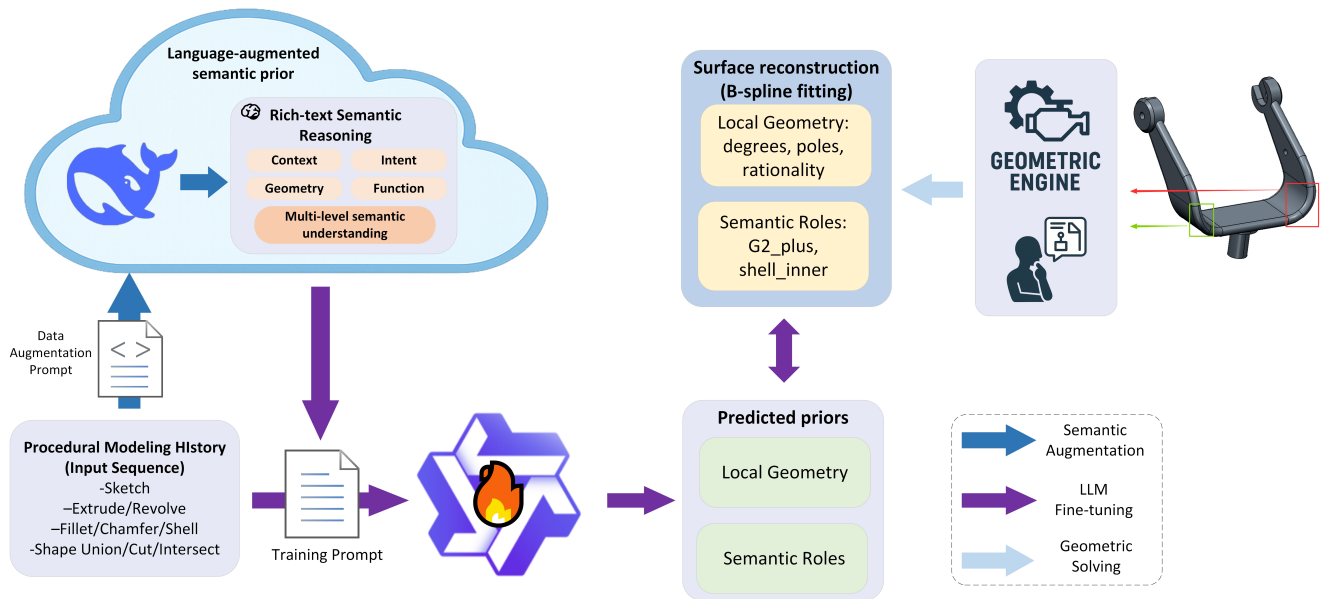


Figure 2. **LASP framework.** During training, given a procedural modeling history H , LASP first performs rich-text semantic reasoning to generate contextual descriptions R that capture operation intent, geometric characteristics, and functional cues. The combined input (H, R) is formatted into prompts and used to fine-tune a large language model to predict structured B-spline semantic priors \mathcal{P} , including local geometry and high-level surface roles. These priors serve as semantic initialization for downstream B-spline fitting within a geometric engine, guiding the solver toward semantically consistent surface reconstruction.

successfully applied to program synthesis, code completion, and symbolic querying [29], with systems such as Codex [9], CodeT5 [40], and SQL-PaLM [36] demonstrating that pretrained LLMs can capture discrete dependencies and multi-step reasoning patterns directly from serialized sequences [41]. Tasks involving text-to-SQL translation and knowledge-graph question answering further highlight that LLMs generalize well to symbolic domains when the underlying structure is appropriately encoded [16, 50].

Motivated by these findings, we treat procedural modeling histories as structured symbolic sequences. When enriched with contextual descriptions, they allow LLMs to infer geometric intent and operational semantics that conventional numerical solvers cannot access.

2.2. Semantic Modeling and Data-Driven CAD

Semantic enrichment of CAD representations has long been a challenge [35]. Classical formats such as BRep and CSG provide explicit geometry and topology but lack high-level semantic interpretability [4, 6]. Efforts such as ShapeNet [8] semantics and ontology-driven design models [12] aim to attach human-readable meaning to geometric entities, though typically through manual or rule-based processes with limited scalability.

Recent data-driven approaches have shifted CAD research toward learning from parametric workflows. DeepCAD [43] models distributions over CAD command sequences, while CAD-Llama [20] extends this direction with large language models for parametric CAD genera-

tion. Seek-CAD [22] and ReCAD [21] further explore self-refinement, multimodal reasoning, and reinforcement-enhanced generation in text or image conditioned CAD modeling. However, these methods primarily aim at generating CAD programs or modeling sequences, rather than inferring interpretable semantic priors that can directly configure downstream geometric solvers.

2.3. B-Spline Parameterization and Geometric Solving

B-spline and NURBS surfaces [27] remain the foundation of modern CAD kernels due to their mathematical flexibility and robustness [13, 47]. Solvers such as OpenCascade [3], Parasolid [19], and ACIS typically rely on fixed initialization schemes (uniform knots, predefined degrees) followed by deterministic least-squares optimization [51]. Although effective for geometric validity, these procedures operate independently of their procedural or functional context.

Prior efforts to improve B-spline solving—such as curvature-aware refinement, adaptive pole allocation, or energy-based fitting [52]—remain heuristic and local. In contrast, we inject semantic priors inferred by an LLM to guide solver initialization, enabling more meaningful convergence behavior without modifying kernel internals.

2.4. Hybrid Language–Geometry Reasoning

Hybrid reasoning frameworks that integrate language models with continuous numerical systems have shown promis-

ing results across generalist agents [30], molecular modeling [17], constraint solving [34], and text-driven 3D synthesis [53]. These works demonstrate that textual cues can shape optimization trajectories and solution spaces.

Inspired by this paradigm, recent CAD-oriented studies [44] explore the connection between symbolic reasoning and geometric processing. Our framework advances this direction by introducing structured semantic priors that bridge procedural semantics and B-spline parameterization, enabling solver-compatible guidance where language provides global intent and the geometric kernel enforces local validity.

3. Method

3.1. Overview and Problem Definition

We propose **LASP** (*Language-Augmented Semantic Priors*), a language-driven framework that bridges procedural modeling and geometric reasoning through contextual supervision. An overview of the framework is illustrated in Fig. 2. LASP introduces a semantic reasoning layer between symbolic modeling and numerical solving, learning to infer *semantic priors* that parameterize B-spline surface generation. This design enables contextual understanding of geometric operations while remaining fully independent of solver internals.

B-spline Surface Fitting. Given control points $\{P_{ij}\}$, knot vectors, and basis functions $N_{i,p}(u)$ and $N_{j,q}(v)$, a tensor-product B-spline surface is defined as[27]:

$$S(u, v) = \sum_{i=0}^{N_u} \sum_{j=0}^{N_v} N_{i,p}(u) N_{j,q}(v) P_{ij}, \quad (1)$$

where (p, q) denotes the polynomial degrees and (N_u, N_v) represents the control-point grid. Conventional geometric kernels determine these parameters heuristically and fit the surface by minimizing the least-squares energy[25]:

$$E_{\text{fit}} = \sum_k \|\mathbf{x}_k - S(u_k, v_k)\|^2, \quad (2)$$

which is highly sensitive to initialization and parameterization. LASP aims to infer a structured set of surface parameters — including polynomial degrees (p, q) , control-point grids (N_u, N_v) , rational flags (ρ_u, ρ_v) , and semantic attributes such as trimming and continuity — which act as semantic priors regularizing the fitting process and embedding design intent directly into the parameter space of geometric optimization.

Symbolic and Linguistic Representations. A procedural modeling history is represented as an ordered sequence of parametric operations:

$$H = \{o_t = (\text{type}_t, \text{param}_t)\}_{t=1}^T, \quad (3)$$

where type_t denotes the operation category (*Fillet, Shell, Chamfer, etc.*), and param_t specifies its geometric arguments. To enrich these symbolic operations with contextual, functional, and geometric semantics, a *rich text semantic representation*

$$R = \{R_t\}_{t=1}^T$$

is generated by a language reasoning module (corresponding to the ‘‘Rich-text Semantic Reasoning’’ block in Fig. 2). Each R_t captures multi-level cues about design context, intent, geometry, and function, serving as the linguistic complement to the symbolic history H .

Semantic Prior Prediction Task. Given a symbolic history H and its corresponding rich-text representation R , LASP learns to predict a structured prior for each operation:

$$P_t = \{d_u, d_v, N_u, N_v, \rho_u, \rho_v, \text{trim}, \text{closed}_u, \text{closed}_v, \text{cont}, \text{curv}, \text{role}\}. \quad (4)$$

Each P_t encodes both numerical and semantic attributes—such as polynomial degrees, control-point counts, rationality, trimming, continuity, and surface roles. The complete prior sequence $\mathcal{P} = \{P_t\}_{t=1}^T$ encapsulates geometric regularities and high-level semantic relations throughout the modeling process. Detailed field definitions and examples are provided in the supplementary material.

Generative Mapping and Training Objective. LASP learns a generative mapping:

$$f_\theta : (H, R) \rightarrow \mathcal{P} = \{P_t\}_{t=1}^T, \quad (5)$$

where f_θ is parameterized by θ and implemented using an instruction-tuned large language model. During fine-tuning, the model receives concatenated symbolic histories and their rich-text semantic descriptions as input and autoregressively generates structured priors in a text-to-structured-text format. Training is carried out in two stages (Sec. 3.2), but both stages share the same token-level objective, namely the negative log-likelihood (NLL):

$$\mathcal{L}_{\text{NLL}} = - \sum_{t=1}^T \sum_{i \in \mathcal{I}_t} \log p_\theta(y_{t,i} | y_{t,<i}, H, R), \quad (6)$$

where \mathcal{I}_t indexes valid output tokens for the t -th operation. This unified objective implicitly enforces both linguistic and geometric consistency, as each target sequence jointly encodes semantic reasoning and B-spline parameter values.

In summary, LASP formulates B-spline parameter inference as a language-conditioned generative process, where each procedural operation is enriched with multi-level semantic reasoning R and optimized through the autoregressive objective in Eq. 6.

3.2. Semantic Prior Prediction

Building on the generative mapping in Sec. 3.1, LASP learns to infer semantic priors $\mathcal{P} = \{P_t\}_{t=1}^T$ that encapsulate the structural configuration of B-spline surfaces generated during procedural modeling. Each P_t describes the expected parameterization pattern of the operation o_t (e.g., degrees, control-point counts, continuity, trimming, and surface role), serving as a high-level inductive bias for downstream geometric solving. To capture both local parameter regularities and long-range semantic dependencies across the sequence, we adopt a two-stage conditional generative modeling scheme:

$$p_\theta(\mathcal{P} | H, R) = \prod_{t=1}^T p_\theta(P_t | P_{<t}, H, R),$$

where H denotes the symbolic modeling history (design history), and R its rich-text semantic enrichment. A detailed prompt template, together with representative rich-text examples for R , is provided in the supplementary material, illustrating how symbolic operations are expanded into structured semantic descriptions.

Stage I: Local Prior Learning. The first stage focuses on capturing local geometric regularities that govern B-spline formation, independent of semantic intent. Given operations that directly produce new surfaces (e.g., *Fillet*, *Chamfer*, *Shell*), the model estimates a geometry-only prior distribution

$$p_{\theta_1}(P_t | o_t),$$

where o_t denotes an isolated procedural command. Each training pair (o_t, P_t^{GT}) supervises numerical attributes $(d_u, d_v, N_u, N_v, \rho_u, \rho_v)$, with small controlled perturbations for regularization. This stage enables the model to internalize transferable structural laws of B-spline generation, providing a stable inductive bias for subsequent semantic modulation.

Stage II: Global Contextual Prior Learning. Building upon Stage I representations, the second stage extends prior estimation to full modeling sequences, where operations interact through shared topology and design intent. The model learns

$$p_{\theta_2}(P_t | P_{<t}, H, R),$$

introducing autoregressive dependencies among surface priors. Initialization with θ_1 preserves low-level geometric consistency while the model learns higher-order relations—such as continuity propagation, curvature transitions, and semantic role coherence (e.g., “transition fillet” or “shell offset”). This stage enables LASP to reason over long-range structural dependencies and infer consistent prior distributions across the procedural hierarchy.

Prompt-Conditioned Training. Both stages are trained under the unified autoregressive objective in Eq. 6, with stage-specific prompt schemas that emphasize different reasoning scopes. Stage I prompts highlight local parameter inference and structural precision, whereas Stage II prompts incorporate richer contextual cues to express inter-operation relationships and semantic continuity. Although the underlying optimization objective remains identical, schema-specific masking and conditioning provide field-aware supervision, aligning geometric precision with semantic coherence within a unified generative representation.

LLM-based Implementation. The mapping f_θ is realized by fine-tuning an instruction-tuned large language model using a text-to-structured-text formulation. The model receives procedural histories augmented with their rich-text representations R as input and outputs structured JSON-like prior sequences. Stage-wise prompt conditioning gradually shifts the model’s focus from local numeric regularity to broader procedural-semantic context, allowing LASP to generate interpretable and context-aware B-spline priors that are more consistent with geometric constraints and high-level semantic context.

3.3. Theoretical Influence of Semantic Priors on Geometric Solving

The semantic priors predicted by LASP are not directly optimized inside the geometric kernel. Instead, they act as structured biases over solver-accessible configuration choices, such as polynomial degree, closure status, and trimming-related patterns. From a conceptual probabilistic perspective, this effect can be interpreted as favoring semantically plausible regions of the feasible configuration space during geometric solving.

Bayesian Formulation. Given geometric constraints C (e.g., adjacency, boundary, or continuity) and the semantic priors $\mathcal{P} = \{P_t\}_{t=1}^T$ predicted by LASP, the posterior of the reconstructed surface can be expressed as

$$p(S | H, R) \propto p(C | S) p(S | \mathcal{P}) p_\theta(\mathcal{P} | H, R), \quad (7)$$

where $p(C | S)$ measures geometric fidelity, $p(S | \mathcal{P})$ encodes the likelihood of the surface configuration under the semantic priors, and $p_\theta(\mathcal{P} | H, R)$ is the language-conditioned prior distribution learned in Sec. 3.2. Maximizing this posterior is equivalent (up to an additive constant independent of S) to

$$S^* = \arg \min_S \left[E_{\text{fit}}(S; C) + \lambda E_{\text{sem}}(S; \mathcal{P}) \right], \quad (8)$$

where $E_{\text{fit}} = -\log p(C | S)$ and $E_{\text{sem}} = -\log p(S | \mathcal{P})$. Rather than defining the exact objective optimized in our implementation, this expression provides an intuitive interpretation of how language-derived priors may bias the feasible configuration space considered during fitting.

Table 1. **Parameter-level prediction results.** Comparison across geometric, topological, and semantic B-spline attributes. Regression metrics use MAE; classification uses Accuracy and Macro-F1. Baselines include Transformer [37], GPT-5 [1], and DeepSeek-v3.2 [23].

Metric	Transformer	GPT-5	DeepSeek-v3.2	LASP (Ours)
Geometric Parameters				
BSpl-Det (F1 \uparrow)	0.507	0.559	0.667	0.950
Deg-Acc (U/V/Mean \uparrow)	0.436 / 0.416 / 0.429	0.520 / 0.501 / 0.507	0.622 / 0.593 / 0.608	0.883 / 0.824 / 0.854
Pole-MAE (U/V/Total \downarrow)	0.474 / 0.780 / 0.624	0.364 / 0.630 / 0.494	0.367 / 0.660 / 0.517	0.353 / 0.612 / 0.485
Rat-F1 (U/V \uparrow)	0.143 / 0.410	0.202 / 0.481	0.270 / 0.645	0.446 / 0.941
Topology				
Topo (Acc / F1 \uparrow)	0.593 / 0.070	0.606 / 0.096	0.710 / 0.141	0.967 / 0.233
Semantic Attributes				
Cont (MF1 / Acc \uparrow)	0.280 / 0.430	0.380 / 0.489	0.478 / 0.596	0.698 / 0.824
Curv (MF1 / Acc \uparrow)	0.121 / 0.385	0.174 / 0.460	0.222 / 0.554	0.327 / 0.765
Role (MF1 / Acc \uparrow)	0.528 / 0.588	0.607 / 0.625	0.716 / 0.730	0.979 / 0.985

Theoretical Implication. Equation 8 provides a conceptual probabilistic view of how semantic priors may influence geometric solving: rather than changing the solver objective itself, they bias the search toward semantically preferred configurations. In practice, our current framework does not explicitly optimize E_{sem} inside the geometric kernel. Instead, the engine-level study in Sec. 4.3 realizes this influence at the configuration level by instantiating discrete prior fields, such as degree, closure, and trimming patterns, as solver-accessible choices.

4. Experiments

4.1. Experimental Setup

Dataset. We adopt an extended subset of the ABC dataset [18], which provides paired procedural modeling histories and their corresponding BRep geometries. All operations capable of producing B-spline surfaces—primarily *Fillet*, *Chamfer*, and *Shell*—are extracted for Stage I local learning, so the current evaluation scope is mainly determined by the operation coverage of the dataset. After filtering, geometric verification, and controlled data augmentation, both Stage I and Stage II datasets contain approximately 35,000 valid samples. Minor random perturbations of degrees, pole counts, and curvature scales are used to increase sample diversity while keeping the statistical distribution consistent with real procedural data. Ground-truth B-spline parameters are directly derived from OpenCascade 7.8 to ensure geometric correctness.

Implementation. LASP is implemented using the **Qwen3-14B** backbone with full-parameter fine-tuning to enable complete adaptation to geometric reasoning. Training is conducted for 4 epochs using the AdamW optimizer (learning rate 2×10^{-5} , weight decay 0.01) on eight NVIDIA A100 GPUs. Each training sample consists of a

procedural code sequence and its automatically generated rich-text semantics, which are tokenized and concatenated as model input. All B-spline reconstructions and quantitative evaluations are executed with OpenCascade, including surface trimming, curvature sampling, and distance-based metric computation.

4.2. Parameter-Level Evaluation

Before integrating LASP with any geometric solver, we first assess whether the model can directly infer geometrically meaningful and semantically coherent B-spline parameters purely from procedural histories and textual cues. This experiment isolates the intrinsic geometric reasoning ability of the language model—evaluating its capacity to infer surface structure without relying on any downstream numerical optimization.

We evaluate three complementary aspects: *geometric parameters* (degrees, pole counts) that reflect surface complexity; *topological flags* (closure, trimming, rationality) that capture structural regularities; and *semantic attributes* (continuity class, curvature scale, surface role) that indicate whether textual semantics encode design intent. Regression metrics use mean absolute error (MAE), while classification is evaluated with accuracy and macro-averaged F1.

Quantitative Results. LASP consistently outperforms all baselines, as shown in Table 1. It achieves $F_1=0.95$ in B-spline detection and a mean degree accuracy of 0.85, indicating strong discriminability across geometric scales and structures. On higher-level semantics, LASP surpasses GPT-5[1] and DeepSeek-v3.2[23] on both *ContinuityClass* ($F_1=0.698$) and *SurfaceRole* ($F_1=0.979$), confirming that linguistic signals provide effective priors for inferring design functionality. Although some rare topology categories (e.g., *UClosed*, $< 5\%$ frequency) yield lower F1-scores, their high accuracies indicate that LASP avoids false pos-

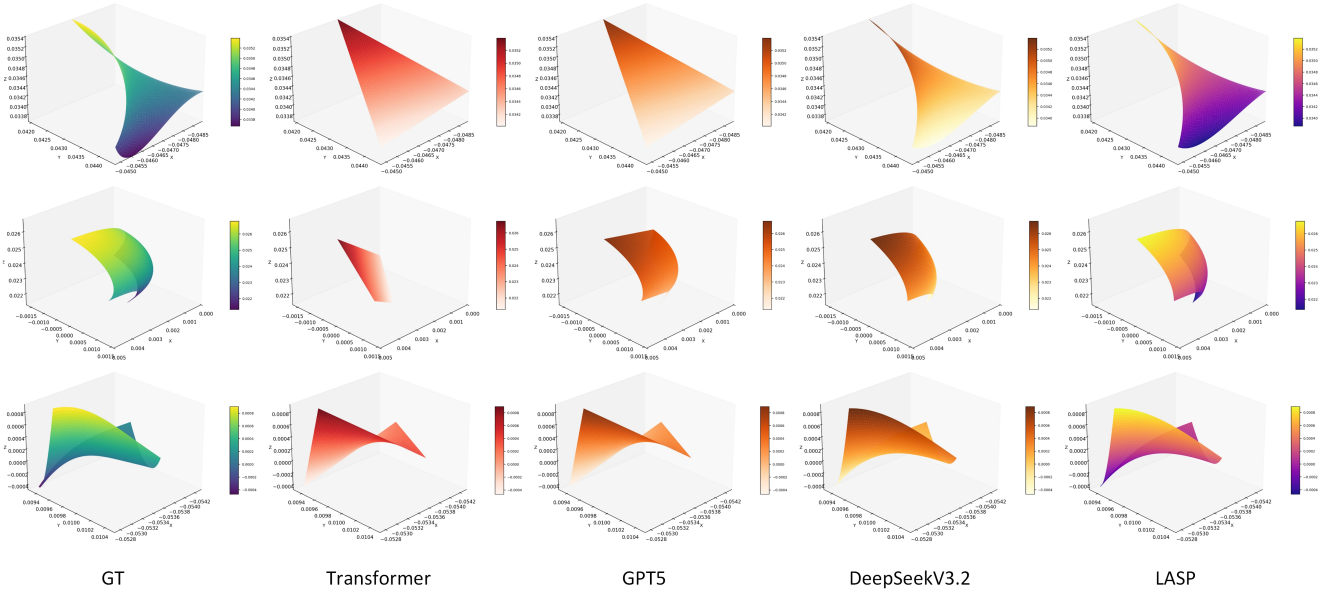


Figure 3. **Qualitative comparison of predicted B-spline parameters.** All methods reconstruct the surface using their predicted B-spline degrees and pole counts while reusing the ground-truth control points. This isolates the quality of the predicted priors, with LASP producing parameters that generate surfaces most consistent with the reference geometry.

itives even under extreme imbalance. For brevity, we use the following abbreviations in Table 1: **BSpl-Det** (B-spline detection F1), **Deg-Acc** (degree accuracy U/V/mean), **Pole-MAE** (pole-count MAE U/V/total), **Rat-F1** (rationality flags U/V), **Topo** (topology flag Acc/F1), **Cont** (continuity class MF1/Acc), **Curv** (curvature scale MF1/Acc), and **Role** (surface role MF1/Acc). Controlled attribution under matched settings, including a matched history-only variant, is further analyzed in Sec. 4.4.

Qualitative Visualization. To qualitatively assess the plausibility of predicted parameters, we reconstruct surfaces using the degrees and pole configurations generated by each model while keeping ground-truth control points fixed to eliminate solver effects. Figure 3 compares representative samples across the reference ground truth (GT), Transformer baseline, GPT-5, DeepSeek-v3.2, and LASP. Surfaces predicted by LASP exhibit the highest geometric fidelity, capturing fine curvature transitions and local continuity consistent with the ground truth. This demonstrates that LASP’s predicted priors encode semantically consistent and geometrically plausible structures, providing a solid foundation for the solver-level analysis in Sec. 4.3.

4.3. Engine-Level Evaluation via Solver Configuration Instantiation.

Modern geometric kernels encapsulate operation-level parameterization, making it difficult to directly inject semantic priors, such as control-point initialization or knot placement. To evaluate whether LASP’s predicted semantics can nevertheless provide practical guidance for downstream

Table 2. **Engine-Level Evaluation via Solver Configuration Instantiation.** B-spline fitting reconstruction error under different solver-accessible parameterization settings.

Config	RMS↓	HD↓	Med↓
Default	0.0491	0.2746	0.0193
+ Semantic Priors	0.0141	0.0395	0.0103

fitting, we instantiate discrete semantic priors into solver-accessible configuration choices for the B-spline fitting process, rather than directly modifying the kernel optimization loop. Surfaces generated by extrusion, revolution, or blending operations are fitted using shape-consistent parameter settings, while the default configuration relies on the kernel’s built-in parameters. Although the numerical kernel itself remains unchanged, the resulting fitting setup is biased toward semantically consistent parameter configurations, providing a practical instantiation of intent-aware configuration for downstream surface fitting.

We employ three complementary metrics to assess geometric fidelity: **RMS distance**[11] captures overall fitting quality, **Hausdorff distance**[2] highlights the worst-case deviation, and **Median distance**[33] provides robustness against outliers. Together, these metrics probe global accuracy, local stability, and robustness of the fitted surfaces.

As summarized in Table 2, the semantically guided fitting achieves significantly lower reconstruction errors across all metrics, reducing RMS deviation by 71.2% and Hausdorff distance by 85.6% ($p < 0.001$). These results

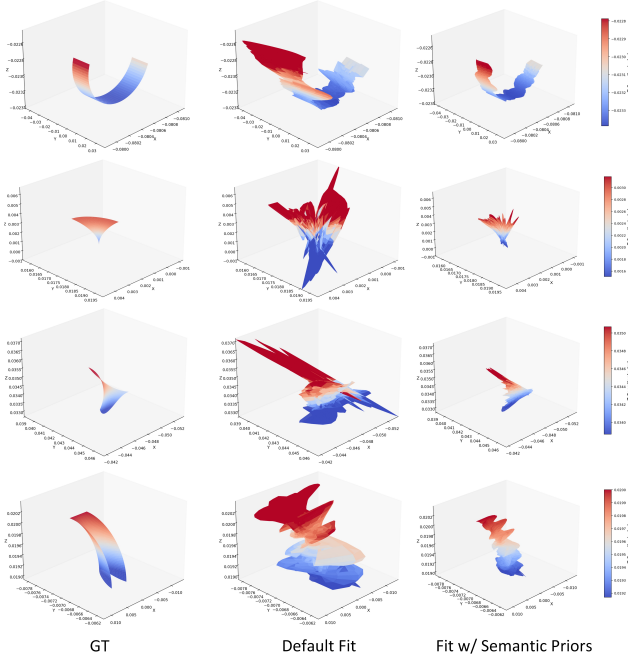


Figure 4. **Qualitative comparison of fitted B-spline surfaces.** Representative examples from extrusion, revolution, and blending operations. Columns show the ground truth surface, the default fitting configuration, and fitting guided by LLM-predicted semantic priors. Color shading indicates the z -axis height of the reconstructed surface.

demonstrate that semantic guidance can effectively improve downstream fitting quality through solver-accessible configuration choices that are more consistent with the procedural-semantic context, serving as empirical evidence that such priors can act as useful inductive biases without modifying solver internals.

Figure 4 provides qualitative comparisons across different operation types. Each triplet shows the reference ground truth, the default fitting result, and the fitting with LASP-derived semantic priors. Unlike Fig. 3, which visualizes parameter-level predictions under fixed control points, this figure directly shows solver-generated surfaces where the color variation reflects the surface height along the z -axis. Semantic priors yield smoother and more coherent reconstructions, particularly in high-curvature regions, confirming that operation-level semantics provide meaningful configuration guidance for the fitting process.

4.4. Ablation Study

To examine the contribution of each component within LASP and to better attribute the source of its gains, we conduct controlled ablation experiments together with a matched history-only comparison. Specifically, we evaluate a history-only variant under the same Qwen3-14B fine-tuning protocol, as well as variants that remove Stage I geo-

Table 3. **Ablation and controlled attribution analysis.** Key metrics showing the effect of a matched history-only variant under the same Qwen3-14B fine-tuning protocol, together with the effects of rich-text semantics, data augmentation, and Stage I pretraining.

Variant	Pole MAE↓	Cont F1↑	Role F1↑
Full LASP	0.56	0.70	0.98
Qwen3-14B(history-only)	0.61	0.59	0.86
w/o Rich-Text Sem.	0.59	0.62	0.91
w/o Augment.	0.58	0.63	0.92
w/o Stage I Pretrain.	0.66	0.68	0.95

metric pretraining, data augmentation, or rich-text semantics, respectively. All models are trained under identical optimization configurations and evaluated using the same parameter-level metrics as in Sec. 4.2.

As shown in Table 3, each component plays a distinct role. The matched Qwen3-14B history-only variant already falls clearly behind the full LASP model (Pole-MAE: 0.61 vs. 0.56; *ContinuityClass* F_1 : 0.59 vs. 0.70; *SurfaceRole* F_1 : 0.86 vs. 0.98), indicating that the gains cannot be explained by backbone scale or fine-tuning alone and that semantic conditioning is essential. Removing rich-text semantics within the two-stage LASP design yields the largest degradation in high-level reasoning (*SurfaceRole* F_1 : 0.98 \rightarrow 0.91; *ContinuityClass* F_1 : 0.70 \rightarrow 0.62), confirming that operation-level semantic context is crucial for capturing high-level procedural semantics. Disabling data augmentation produces similar declines, suggesting that moderate procedural diversity is necessary for robust generalization. Removing Stage I geometric pretraining primarily affects numerical precision (Pole-MAE: 0.56 \rightarrow 0.66), confirming that explicit geometric grounding stabilizes quantitative inference. Overall, LASP’s performance emerges from the combined effects of geometric supervision, semantic conditioning, and data diversity.

5. Conclusion

We introduced **LASP**, a language-augmented framework that brings contextual semantic reasoning into B-spline surface construction. Rather than modifying geometric kernels, LASP operates as an external semantic layer that transforms procedural modeling histories into rich-text descriptions and uses these to predict structured B-spline priors. These priors provide high-level inductive signals that guide conventional B-spline fitting toward more coherent and design-aligned solutions. A two-stage learning strategy allows LASP to capture both local geometric regularities and global contextual dependencies, producing priors that consistently improve surface quality and parameter stability. The results demonstrate that language-driven reasoning provides a powerful inductive bias for geometric solving, pointing toward deeper connections between symbolic intent and numerical optimization.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6
- [2] Nicolas Aspert, Diego Santa-Cruz, and Touradj Ebrahimi. Mesh: Measuring errors between surfaces using the hausdorff distance. In *Proceedings. IEEE international conference on multimedia and expo*, pages 705–708. IEEE, 2002. 7
- [3] Mladen Banović, Orest Mykhaskiv, Salvatore Auremma, Andrea Walther, Herve Legrand, and Jens-Dominik Müller. Algorithmic differentiation of the open cascade technology cad kernel and its coupling with an adjoint cfd solver. *Optimization Methods and Software*, 33(4-6):813–828, 2018. 3
- [4] Rafael Bidarra and Willem F Bronsvooort. Semantic feature modelling. *Computer-Aided Design*, 32(3):201–225, 2000. 3
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2
- [6] Gino Brunetti and Borut Golob. A feature-based approach towards an integrated product model including conceptual design information. *Computer-Aided Design*, 32(14):877–887, 2000. 3
- [7] Jorge D Camba, Manuel Contero, and Pedro Company. Parametric cad modeling: An analysis of strategies for design reusability. *Computer-aided design*, 74:18–31, 2016. 2
- [8] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3
- [9] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. 2, 3
- [10] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. 2
- [11] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. Metro: measuring error on simplified surfaces. In *Computer graphics forum*, pages 167–174. Wiley Online Library, 1998. 7
- [12] Francesco Furini, Marco Rossoni, and Giorgio Colombo. Knowledge based engineering and ontology engineering approaches for product development: methods and tools for design automation in industrial engineering. In *ASME International Mechanical Engineering Congress and Exposition*, page V011T15A032. American Society of Mechanical Engineers, 2016. 3
- [13] Md Shahid Hasan, Md Nur Alam, Md Fayz-Al-Asad, Noor Muhammad, and Cemil Tunç. B-spline curve theory: An overview and applications in real life. *Nonlinear Engineering*, 13(1):20240054, 2024. 3
- [14] Christoph M Hoffmann and GEORGE VANĚČEK JR. Fundamental techniques for geometric and solid modeling. In *Control and dynamic systems*, pages 101–165. Elsevier, 1991. 2
- [15] Imre Horvath. A treatise on order in engineering design research. *Research in engineering design*, 15(3):155–181, 2004. 2
- [16] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):8622–8642, 2024. 3
- [17] Yeonghun Kang and Jihan Kim. Chatmof: an artificial intelligence system for predicting and generating metal-organic frameworks using large language models. *Nature communications*, 15(1):4705, 2024. 4
- [18] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611, 2019. 2, 6
- [19] Thomas R Kramer. Extracting step geometry and topology from a solid modeler: Parasolid-to-step. Technical report, US Department of Commerce, National Institute of Standards and Technology, 1991. 3
- [20] Jiahao Li, Weijian Ma, Xueyang Li, Yunzhong Lou, Guichun Zhou, and Xiangdong Zhou. Cad-llama: leveraging large language models for computer-aided design parametric 3d model generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18563–18573, 2025. 3
- [21] Jiahao Li, Yusheng Luo, Yunzhong Lou, and Xiangdong Zhou. Recad: Reinforcement learning enhanced parametric cad model generation with vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6190–6198, 2026. 3
- [22] Xueyang Li, Jiahao Li, Yu Song, Yunzhong Lou, and Xiangdong Zhou. Seek-cad: A self-refined generative modeling for 3d parametric cad using local inference via deepseek. *arXiv preprint arXiv:2505.17702*, 2025. 3
- [23] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 6
- [24] Yunzhong Lou, Xueyang Li, Haotian Chen, and Xiangdong Zhou. Brep-bert: Pre-training boundary representation bert with sub-graph node contrastive learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1657–1666, 2023. 2
- [25] Weiyin Ma and Jean-Pierre Kruth. Parameterization of randomly measured points for least squares fitting of b-spline

- curves and surfaces. *Computer-Aided Design*, 27(9):663–675, 1995. 4
- [26] Weijian Ma, Shuaiqi Chen, Yunzhong Lou, Xueyang Li, and Xiangdong Zhou. Draw step by step: Reconstructing cad construction sequences from point clouds via multimodal diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27154–27163, 2024. 2
- [27] Les Piegel and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 2012. 2, 3, 4
- [28] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2
- [29] Bowen Qin, Binyuan Hui, Lihan Wang, Min Yang, Jinyang Li, Binhua Li, Ruiying Geng, Rongyu Cao, Jian Sun, Luo Si, et al. A survey on text-to-sql parsing: Concepts, methods, and future directions. *arXiv preprint arXiv:2208.13629*, 2022. 3
- [30] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. 4
- [31] Juergen Riegel, Werner Mayer, and Yorik van Havre. Freecad. *Freecadspec2002*, 7, 2016. 2
- [32] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023. 2
- [33] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, pages 519–528. IEEE, 2006. 7
- [34] Weichun Shi, Minghao Liu, Wanting Zhang, Langchen Shi, Fuqi Jia, Feifei Ma, and Jian Zhang. Constraintllm: A neuro-symbolic framework for industrial-level constraint programming. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16010–16030, 2025. 4
- [35] Ian Stroud. *Boundary representation modelling techniques*. Springer, 2006. 3
- [36] Ruoxi Sun, Sercan Ö Arik, Alex Muzio, Lesly Miculicich, Satya Gundabathula, Pengcheng Yin, Hanjun Dai, Hootan Nakhost, Rajarishi Sinha, Zifeng Wang, et al. Sql-palm: Improved large language model adaptation for text-to-sql (extended). *arXiv preprint arXiv:2306.00739*, 2023. 3
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 6
- [38] Ruiyu Wang, Yu Yuan, Shizhao Sun, and Jiang Bian. Text-to-cad generation through infusing visual feedback in large language models. *arXiv preprint arXiv:2501.19054*, 2025. 2
- [39] Siyu Wang, Cailian Chen, Xinyi Le, Qimin Xu, Lei Xu, Yanzhou Zhang, and Jie Yang. Cad-gpt: Synthesising cad construction sequence with spatial reasoning-enhanced multimodal llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7880–7888, 2025. 2
- [40] Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 8696–8708, 2021. 3
- [41] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 3
- [42] Karl D.D. Willis, Yewen Pu, Haoliang Luo, et al. Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences. *ACM Transactions on Graphics (TOG)*, 40(4):1–24, 2021. 2
- [43] Rundi Wu, Chang Xiao, and Changxi Zheng. Deepcad: A deep generative network for computer-aided design models. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6772–6782, 2021. 2, 3
- [44] Liu Xiaorui, Zhang Yuhao, Wang Leiqi, Gu Lexiang, Xu Yaning, Zheng Ke, Cai Qianqian, and Zhou Gang. Llm driven fusion ai-ad system for mechanical design: From understanding to generation. *Advanced Engineering Informatics*, 68:103745, 2025. 2, 4
- [45] Jingwei Xu, Chenyu Wang, Zibo Zhao, Wen Liu, Yi Ma, and Shenghua Gao. Cad-mllm: Unifying multimodality-conditioned cad generation with mllm. *arXiv preprint arXiv:2411.04954*, 2024. 2
- [46] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*, 2023. 2
- [47] Jieyin Yang, Xiaohong Jia, and Dong-Ming Yan. Topology guaranteed b-spline surface/surface intersection. *ACM Transactions on Graphics (TOG)*, 42(6):1–16, 2023. 3
- [48] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023. 2
- [49] Fukun Yin, Xin Chen, Chi Zhang, Biao Jiang, Zibo Zhao, Wen Liu, Gang Yu, and Tao Chen. Shapegpt: 3d shape generation with a unified multi-modal language model. *IEEE Transactions on Multimedia*, 27:4107–4120, 2025. 2
- [50] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3911–3921, 2018. 3
- [51] Guan-Jie Yuan, Hao Liu, Jian-Ping Su, and Xiao-Ming Fu. Computing planar and volumetric b-spline parameterizations

- for iga by robust mapping fitting. *Computer Aided Geometric Design*, 86:101968, 2021. [3](#)
- [52] Yuhua Zhang, Juan Cao, Zhonggui Chen, Xin Li, and Xiaoming Zeng. B-spline surface fitting with knot position optimization. *Computers & Graphics*, 58:73–83, 2016. [3](#)
- [53] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023. [2](#), [4](#)