Combining DPM with Shape Semantics-Driven 3D Model Reconstruction to Improve Parametric Design Efficiency for Intelligent Products

Hui Li
College of Fine Arts and Design, Zhanjiang University of Science and Technology, Zhanjiang 524000, Guangdong, China
Corresponding author: Hui Li (e-mail: HuiYanXuanFan@163.com)

ABSTRACT The rapid iteration of intelligent products places increasingly higher demands on parametric modeling, making it difficult for traditional modeling approaches to meet strict industrial design requirements for complex mechanical parts and editable functional structures. To address this, this paper proposes a 3D model reconstruction method that combines DPM with shape semantics. First, a high-fidelity point cloud is generated using a diffusion probability model. Semantic label parsing converts features into geometric parameters. Subsequently, constraints such as length and curvature are embedded in the sampling process to achieve controllable generation. The point cloud is then reconstructed into a mesh and optimized to ensure geometric integrity and parametric editability. Finally, the integration with Computer-Aided Design (CAD) systems enables rapid adjustment and redesign, thereby improving the efficiency of parameterized modeling of intelligent products. Experiments show that the proposed method achieves a Chamfer distance of 0.012 mm in the Electronics category, significantly outperforming the Generative Adversarial Network (GAN) method by 0.025 mm. When the point cloud complexity is 25,000 points, the generation time is only 3.1 s, which is highly efficient. The method achieves an accuracy rate of over 87% in semantic consistency. The research results significantly improve the efficiency of parametric modeling, making it possible to rapidly generate and adjust complex 3D models for industrial design. For antenna housings, radomes and other wave-related industrial structures, controllable 3D reconstruction can provide reusable geometric modeling support.

INDEX TERMS 3D model reconstruction,diffusion probabilistic model,shape semantics,parametric design,antenna structure modeling

I. INTRODUCTION

With the development of new-generation information technology, the penetration rate of intelligent products continues its dramatic growth across many sectors, including consumer electronics, transportation, medical equipment, and industrial equipment. This, combined with the shortening of design iteration cycles, has intensified the demand for efficient modeling methods capable of rapidly prototyping structures ranging from shuttleless weaving machine components to complex 3D knitted fabric structures [1,2]. Parametric modeling plays a central role in meeting diverse functional requirements and supporting complex form design. Traditional three-dimensional modeling relies on designers to gradually establish geometric features and parameter relationships in a CAD environment. The process is time-consuming and repetitive. When the complexity of product structure increases, designers need to frequently modify dimensions, curvatures, and connection relationships, resulting in reduced design efficiency and difficulty in meeting the needs of rapid iteration and mass customization [3]. Traditional three-dimensional modeling methods can no longer fully meet the development needs of intelligent products. Therefore, exploring new intelligent three-dimensional model generation methods is of both theoretical and practical importance.

In response to the problem of insufficient data-driven intelligent product design methods and their efficiency, academia and industry have conducted research on data-driven 3D model generation. Yang et al. [4] systematically expounded on the data-driven computational design method system and provided a theoretical framework for product parametric modeling. With the rapid development of generative models, Cao et al. [5]'s review of diffusion models revealed their unique advantages in processing high-dimensional data, laying a theoretical foundation for 3D generation tasks. Liu et al. [6] constructed an industrial data filling framework through diffusion transformers, significantly improving the robustness of missing data reconstruction. At the forefront of 3D reconstruction technology, Ye et al. [7] provided an important reference for geometric constraint embedding based on a diffusion probability model (DPM) of building outline constraints. The integration of semantic guidance and DPM is an effective way to solve the high-dimensionality and low-controllability problems of industrial design. However, there are still obvious deficiencies in the core aspects of industrial design, such as parametric modeling efficiency, precise mapping of geometric constraints, and CAD system docking.

The DPM has attracted much attention due to its stability and high-fidelity generation of high-dimensional data. In recent years, the Industrial Internet of Things (IIoT) technology has provided important support for intelligent manufacturing. Hu et al. [8] systematically elaborated on the implementation path and technical framework of the industrial Internet of Things to promote the development of intelligent manufacturing. In the field of three-dimensional perception and reconstruction, Tu et al. [9] achieved consistency of threedimensional reconstruction of hands in videos through self-supervised learning, providing a new idea for dynamic three-dimensional modeling. Therefore, there is an urgent need for an innovative method that takes generation efficiency, controllability, and industrial applicability into account. In this study, "shape semantics" specifically refers to geometric or topological feature descriptions that can be analyzed in engineering contexts, mapped to CAD parameters, and used to express design intent. In addition to basic parameters such as length, curvature and thickness, it also includes 23 types of features such as: symmetry plane position, hole center and radius, connection interface normal, closed contour perimeter, local convex volume, corner radius, and patch continuity level (G0/G1/G2). To address the problems of low efficiency in traditional modeling and limited controllability of existing deep generative models, this paper proposes a 3D model reconstruction method combining DPM and shape semantics. This method utilizes DPM to generate high-fidelity point cloud data. Shape semantics are then parsed into geometric parameter constraints to guide the generation process and ensure that the generated results more accurately meet design requirements. Parameter constraints are embedded during the sampling stage to ensure the generated result has a reasonable and editable structure. Furthermore, point cloud mesh transformation and constraint optimization are used to enhance the industrial applicability of the model, ensuring semantic consistency and direct parametric modification of the results. Finally, by interfacing with CAD systems, the generated model can be quickly adjusted and redesigned in a parametric environment. This study improves the efficiency of parametric modeling and provides a new intelligent approach for complex product design,.

The core innovation of this research lies not only in combining DPM with semantics, but also in proposing a differentiable semantic-parameter mapping mechanism and embedding it within the conditional guidance pathway of the diffusion process. This mechanism is integrated into the diffusion generation backbone and combined with post-processing geometric primitive detection and analytical mapping to form a closed-loop process of "generation–semantic guidance–parameter calibration", significantly improving the structural rationality and CAD editability of the generated results. Unlike recent conditional DPMs based on text prompts (3D-SDM) or implicit shape priors (GeoDiff, DiffuScene), our method transforms abstract semantics (such as "streamlined" and "modular interface") into explicit geometric constraint vectors (length, curvature, etc.) through a learnable mapping, and dynamically adjusts the denoising direction during reverse sampling through a conditional attention mechanism. This design solves the fundamental problems of uncontrollable semantics and non-editable parameters in existing diffusion models for industrial parametric scenarios, resulting in models that are not only morphologically realistic but also possess structural rationality and parameter traceability suitable for direct use in secondary CAD design.

II. SEMANTIC-DRIVEN 3D MODEL RECONSTRUCTION FRAMEWORK

A. DPM GENERATION MODULE

1) Diffusion Process Modeling

In the 3D model generation task, the original shape point cloud data needs to be mapped to a Gaussian noise distribution first to construct a gradually degraded diffusion process [10,11]. It is assumed that the input point cloud data is $x_0 \in R^{N\times 3}$, where N represents the number of points. In the experiment, $N = 2048$ is set to ensure a balance between geometric expression and computational overhead. The diffusion process is carried out over discrete time steps $t \in \{1, 2, \ldots, T\}$, where the total number of steps is set to $T = 1000$, so that the noise is gradually added without destroying the continuity of the data distribution. The degradation process is defined as Formula 1:

$$q(x_t \mid x_{t-1}) = N\left(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I\right) \tag{1}$$

β_t is the noise scheduling parameter, which increases smoothly from 10^{-4} to 0.02 according to a linear increasing strategy to ensure that more geometric information is retained in the early stage and fully approximates the isotropic Gaussian distribution in the later stage. Through this gradual degradation mechanism, the point cloud data is mapped to the noise space, thereby establishing a stable condition for the subsequent reverse generation process. This design is used to solve the mode collapse problem of traditional generative models on high-dimensional three-dimensional data, while ensuring that the geometric structure is gradually transformed rather than directly lost.

2) Denoising Sampling and Reconstruction

In the reverse generation process, the conditional probability distribution is used to restore the original geometric structure. The generative model uses a parameterized neural network $\epsilon_\theta(x_t, t)$ as a denoising function, with the input being the degraded point cloud and the corresponding time step, and the output being the noise estimate. The reverse sampling process is defined by Formula 2:

$$p_\theta(x_{t-1} \mid x_t) = N\left(x_{t-1}; \frac{1}{\sqrt{1-\beta_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t, t)\right), \sigma_t^2 I\right) \tag{2}$$

$\alpha_t = \prod_{s=1}^{t}(1-\beta_s)$ and σ_t^2 are adjustable sampling variances. To accelerate the sampling process and reduce generation delay, the Denoising Diffusion Implicit Models (DDIM) strategy is adopted to compress the number of sampling steps from 1,000 to 200, while maintaining geometric accuracy and improving computational efficiency. This corresponds to the deterministic path setting under the DDIM sampling strategy, i.e., $\sigma_t = 0$. Therefore, the reverse process is completely deterministic and does not introduce additional random noise. In our experiments, we employ DDIM's 200-step deterministic sampling scheme to significantly accelerate the generation process while maintaining geometric fidelity.

To ensure the geometric fidelity and reasonableness of the generated point cloud, Chamfer Distance is applied as a reconstruction loss during training, as shown in Formula 3:

$$\begin{aligned} L_{\mathrm{CD}}(X, Y) = \frac{1}{|X|}\sum_{x \in X}\min_{y \in Y}\|x-y\|_2^2 \\ + \frac{1}{|Y|}\sum_{y \in Y}\min_{x \in X}\|y-x\|_2^2 \end{aligned} \tag{3}$$

X is the real point cloud, and Y is the generated point cloud. The loss function ensures that the generated results approximate the real data distribution in both global morphology and local geometric details. Figure 1 illustrates the 3D point cloud generation and reconstruction process based on the DPM. Sub-figure (a) shows the degradation of a spherical point cloud at different time steps t = 0, 500, and 1,000. The x-, y-, and z-axes represent spatial coordinates. The point cloud gradually adds noise as the time step increases, reflecting the diffusion characteristics and point cloud morphological changes during the DPM sampling process. Sub-figure (b) compares the final generated point cloud with the original ground-truth point cloud. The x-, y-, and z-axes represent spatial coordinates. Blue denotes the ground-truth point cloud, whereas the red denotes the generated point cloud. This visual comparison shows that the generated model can reconstruct the overall geometric structure well, but there is slight noise deviation. Sub-figure (c) shows the convergence curve of the Chamfer distance over training iterations, with the x-axis representing the number of training rounds and the y-axis representing the Chamfer distance. The curve rapidly decreases to around 0.02 as the number of iterations increases, indicating that the model's accuracy gradually improves as it learns point cloud reconstruction. This demonstrates that DPM, by combining shape semantics and parameter constraints, can effectively generate high-fidelity and controllable 3D models.

B. SHAPE SEMANTIC LABEL PARSING

1) Semantic Feature Extraction and Label Construction

In the semantic-driven 3D reconstruction process, it is first necessary to establish a correspondence between natural language descriptions and geometric features. To this end, a semantic dictionary covering common attributes of industrial product design is selected, including 50 semantic labels such as "streamlined", "symmetrical structure", "modular interface", and "surface continuity" [12,13]. Each semantic label is associated with a geometric parameter by combining manual definition with statistical learning. The input point cloud data is set to $X \in R^{N\times 3}$, where $N = 2048$. To extract the corresponding geometric features, a feature encoding network based on PointNet++ is used to output a multi-layer feature representation $F = \{f_1, f_2, \ldots, f_k\}$, where $k = 128$. In the feature space, the mapping between the point cloud and the semantic label is achieved through a multi-label classifier. The classifier consists of a three-layer fully connected network, and the output dimension is the size of the semantic dictionary. The semantic probability distribution $\hat{y} \in [0,1]^M$ is obtained by the Sigmoid activation function, where $M = 50$. The goal of label parsing is to minimize the crossentropy loss between the true semantic label y and the predicted label $\hat{y}$, as shown in Formula 4:

$$L_{\mathrm{sem}} = -\frac{1}{M}\sum_{i=1}^{M}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{4}$$

Semantic feature extraction combines PointNet++ features with a rule-based geometric primitive detection module. For example, RANSAC is used to detect planes and cylinders, edge detection is achieved through changes in principal curvature signs, and interface regions are identified through normal consistency clustering. These primitives are encoded as structured feature vectors and jointly trained with semantic labels. The mapping to CAD parameters (such as aperture and fillet radius) is implemented using a hybrid strategy: for regular geometry (such as holes and fillets), analytical formulas (such as fitting holes with minimum bounding circles) are employed; for irregular semantics (such as "streamlined"), an MLP network learns a nonlinear mapping from semantic scores to parameter values. This hybrid mechanism balances accuracy and generalization ability.

2) Geometric Parameter Constraint Mapping

After completing the semantic label parsing, the semantic features need to be converted into geometric parameter constraints that can directly act on the generated model. Assuming that the semantic label set obtained by parsing is $S = \{s_1, s_2, \ldots, s_M\}$, each semantic label corresponds to a parameter constraint vector $c_i \in R^d$. In the experiment, the

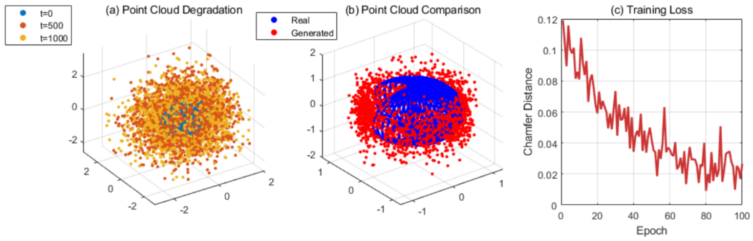

FIGURE 1. Schematic diagram of the DPM generation module. (a). Example of the point cloud degradation process; (b). Comparison of the generated point cloud and the ground-truth point cloud; (c). Chamfer distance training convergence curve

parameter dimension is set to $d = 6$, corresponding to the six core design parameters of length, curvature, thickness, symmetry, volume, and local thickness. The parameter range is set according to the common requirements of industrial design [14,15].

The mapping relationship between semantic labels and geometric parameters is established through supervised learning and regularization constraints. The linear regression model and multi-layer perceptron are combined to fit the nonlinear relationship between semantics and geometric parameters, as shown in Formula 5:

$$c_i = W \cdot f(s_i) + b \tag{5}$$

In formula 5, $f(s_i)$ represents the embedding vector of the semantic label, with a dimension of 128; $W \in R^{d\times 128}$; $b \in R^d$. The parameter constraint loss L_{param} is applied during the training process to ensure that the parameter vector after mapping is within the specified range, as shown in Formula 6:

$$L_{\text{param}} = \sum_{i=1}^{M} \|\text{clip}(c_i, l, u) - c_i\|^2 \tag{6}$$

u, l are the upper and lower limit vectors of the parameters, respectively. This constraint ensures that the parameter values generated by the semantic mapping remain within a feasible design range and avoids unreasonable geometric settings. The upper and lower bound vectors l and u of each parameter are set according to industrial design specifications as follows: length [50, 500] mm, curvature [0.01, 1.0] mm^{-1}, thickness [1.0, 10.0] mm, symmetry [0.3, 0.95], volume [50, 300] cm^3, and local thickness [2, 10] mm.

Figure 2 illustrates the role of shape semantic parsing in the parameterized generation of 3D models. Subfigure (a) shows the semantic classification confusion matrix, where the x-axis represents the predicted label and the y-axis represents the true label. The closer to white, the higher the predicted probability. A diagonal line closer to black indicates higher semantic classification accuracy, suggesting that the model can accurately identify design semantics such as “streamlined” and “modular interface”. Sub-figure (b) shows the mapping between semantic scores and geometric parameters, with the semantic scores on the x-axis and the corresponding geometric parameter values on the y-axis. The scatter plots and fitted curves show that the geometric parameters increase nonlinearly with the semantic scores, demonstrating that semantic information can be effectively converted into specific design parameters, providing controllable constraints for DPM generation. Sub-figure (c) shows the decrease in parameter constraint loss over the number of training iterations. The x-axis represents the number of training iterations, and the y-axis represents the constraint loss. The loss decreases from approximately 0.34 initially to around 0.07, reflecting the model’s gradual fulfillment of geometric parameter constraints during learning, enabling semantically driven, controllable 3D model generation.

To illustrate how semantic labels are transformed into parameters for DPM, this paper uses “collar curve” and “structural support” as examples. The semantics of “collar curve” are first extracted using PointNet++ to extract the boundary point cloud ring, calculating its principal curvature direction and closed contour length, and then mapped to curvature (0.35 $\pm$ 0.05 mm-1) and length (320 $\pm$ 20 mm). “Structural support” is mapped to length ($\geq$ 150 mm), local thickness ($\geq$ 6 mm), and symmetry ($\geq$ 0.85) by detecting high-density straight-line segments in the point cloud and fitting their axial vector and cross-sectional thickness. This mapping is implemented using a lightweight multilayer perceptron (3 layers, 128 dimensions), with the input being a semantic embedding vector (50-dimensional label probabilities activated by Sigmoid) and the output being 6-dimensional geometric parameters. During training, L2 regression loss and boundary constraint regularization ensure that the output parameters remain within the feasible design domain. This mechanism enables abstract semantics to possess differentiable, optimizable, and constrained geometric representation capabilities.

C. PARAMETER CONSTRAINED EMBEDDING

1) Conditional Embedding Mechanism Design

During the backward sampling process of the DPM, to ensure that the generated 3D point cloud meets predefined geometric parameter requirements, the constraint vector needs to be directly applied to the feature update phase of the denoising network [16,17]. It is assumed that the parameter vector obtained by the previous analysis is $c \in R^d$, where $d = 6$, corresponding to length, curvature, thickness, symmetry, volume, and local thickness. To achieve efficient embedding, the parameter vector is first projected through a two-layer fully connected network to obtain the

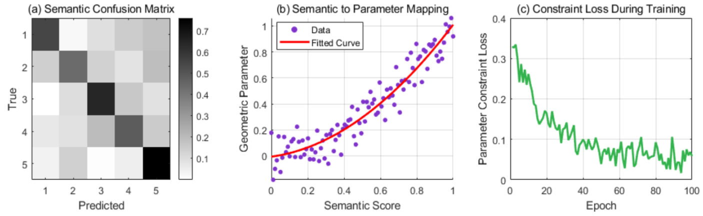

FIGURE 2. Schematic diagram of shape semantic label parsing. (a). Semantic classification confusion matrix; (b). Mapping of semantic scores to geometric parameters; (c). Parameter constraint loss changes with training iterations

conditional embedding vector $h_c \in R^{128}$. This projection step ensures that parameters of different dimensions are uniformly mapped to the same feature space, facilitating fusion with the hidden representation of the denoising network. At each backward sampling step t, the denoising function $\epsilon_\theta(x_t, t)$ needs to receive both the input point cloud and the conditional embedding vector. To achieve this, a conditional attention mechanism is introduced into the network. Specifically, h_c is concatenated with the local features of the point cloud, and the interaction weights are calculated through the selfattention layer, so that the conditional vector can have a differential impact on the feature updates of different points, expressed as Formula 7:

$$z_i' = z_i + \text{Attention}(z_i, h_c) \tag{7}$$

z_i represents the feature of the i-th point, and Attention$(\cdot)$ represents the weighted update between the condition and the feature using the scaled dot product attention mechanism. In this way, geometric parameters are dynamically applied to each step of point cloud generation, thereby achieving layer-by-layer penetration of conditional constraints during the diffusion process.

The conditional attention mechanism employs a cross-attention structure, where the conditional embedding vector serves as the query, and the local features of the point cloud act as both the key and value. This mechanism is implemented as a four-head attention structure, with each attention head having a dimension of 32, resulting in a total dimension of 128. During computation, the dot product of the query and key is first calculated, then scaled by dividing by the square root of the key's dimension. Next, the attention weights are obtained using a softmax function, and finally, these weights are applied to the values to generate the final output.

Figure 3 shows the visualization and optimization results of applying parameter constraints during DPM generation. Sub-figure (a) shows the conditional attention weight graph, with the x-axis representing feature dimension and the y-axis representing point index. It can be seen that the model forms distinct areas of concentrated attention on geometric, topological, and constraint features, respectively. Feature dimensions around 20 correspond to geometric features, while feature dimensions 40-100 also correspond to key features such as topology and constraints. Constraint features at certain points are further enhanced, demonstrating that the model can specifically respond to key design parameters in the multidimensional feature space. Sub-figure (b) shows a comparison of length distributions before and after parameter constraints, with the x-axis representing length parameter (mm) and the y-axis representing frequency. The orange histogram shows that the length distribution generated without constraints has a wide range and extreme values. The blue histogram shows that the length distribution after constrained optimization is mostly confined to between 150 and 450 mm, with a peak around 160 mm.

2) Constraint Optimization and Sampling Correction

After completing the conditional embedding, it is also necessary to ensure that the geometric properties of the generated point cloud meet the preset parameter range through an optimization mechanism. To this end, in each step of the reverse denoising, an additional parameter consistency loss is applied to compare the difference between the generated point cloud estimate and the target parameter. Assuming that the generated point cloud is $\widehat{X} \in R^{N\times3}$, and the geometric parameter extraction function is $g(\cdot)$, then the consistency loss is defined as Formula 8:

$$L_{\text{cons}} = \|g(\widehat{X}) - c\|_2^2 \tag{8}$$

$g(\cdot)$ contains calculation functions for geometric properties such as length, curvature, and thickness. For example, the length is calculated by the maximum and minimum coordinate difference of the point cloud in the main direction; the curvature is approximated by the second-order derivative of the local neighborhood; the thickness is estimated by the envelope difference of the point cloud in the normal direction. In the experimental setting, the length constraint range is [50, 500] mm; the curvature range is [0.01, 1.0] mm-1; the thickness range is [1.0, 10.0] mm. To avoid scale imbalance issues caused by different physical dimensions in the loss function, all terms in the geometric consistency loss are normalized: the length term is divided by its upper limit of 500 mm, the curvature term is multiplied by 100 to broaden its numerical range (considering that typical curvature values are in the range of 0.01-1.0 mm-1), and the thickness term is divided by its upper limit of 10 mm. This normalization strategy ensures that each parameter contributes evenly during the optimization process, effectively preventing high-order parameters from dominating the optimization direction, and allowing geometric constraints to work synergistically in the generation process.

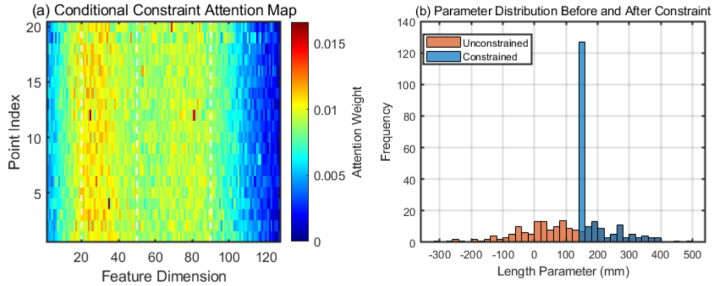

FIGURE 3. Schematic diagram of parameter constraint embedding. (a). Conditional attention weight distribution; (b). Comparison of distributions before and after parameter constraints

To balance the reconstruction accuracy and parameter consistency, the loss is weighted and combined with the denoising prediction error, as shown in Formula 9:

$$L = L_{\mathrm{CD}} + \lambda L_{\mathrm{cons}} \tag{9}$$

L_{CD} represents the Chamfer Distance, and λ is the weight factor, which is set to 0.5 in the experiment. Through this joint optimization, the model ensures that the geometric parameters are consistent with the target constraints while maintaining the rationality of the global morphology. The denoising training objective consists of three parts: standard DDPM noise prediction MSE loss, geometric reconstruction chamfer distance loss (weight 0.3), and parameter consistency constraint loss (weight 0.5). These weights were determined through validation set tuning to ensure that the generated model meets the design parameter requirements while maintaining geometric fidelity, achieving the best balance between reconstruction accuracy and parameter controllability. In the sampling stage, to further improve the generation efficiency, a sampling strategy based on gradient correction is applied. Specifically, at each reverse sampling update, the point cloud is slightly corrected according to the gradient of the consistency loss, as shown in Formula 10:

$$x'_{t-1} = x_{t-1} - \eta \nabla_{x_{t-1}} L_{\mathrm{cons}} \tag{10}$$

η is the correction step size, which is set to 10^{-3}. This correction operation can gradually guide the generated point cloud to converge in the direction of satisfying the geometric parameters during the sampling process without adding additional sampling steps, thereby improving the constraint accuracy while ensuring efficiency. In the geometric parameter extraction stage, the function $g(\cdot)$ contains two types of operations: differentiable parts (such as principal direction calculation based on PCA, least squares surface fitting) and nondifferentiable parts (RANSAC plane detection). For the non-differentiable parts, this paper adopts the Straight-Through Estimator (STE) strategy, using the original calculation in forward propagation and the alternative gradient in backward propagation.

The gradient correction mechanism shown in Formula (10) is not a sampling operation within a strictly probabilistic framework, but rather a heuristic guidance strategy for constraint satisfaction. While maintaining the main structure of the deterministic DDIM path, it ensures that the generated results meet industrial design constraints through lightweight gradient projection. This design sacrifices strict probabilistic consistency but gains crucial geometric controllability in practical engineering.

This geometric constraint embedding mechanism does not employ classifier guidance; instead, it uses an internal guidance strategy driven by conditional attention. At each step of back diffusion, the geometric parameters obtained from semantic parsing are encoded as conditional embedding vectors and deeply fused with local point cloud features in the denoising network. Through the conditional attention mechanism, the network can dynamically perceive the deviation between the current generated result and the target constraint and adaptively adjust the denoising direction—for example, enhancing correction weights for high-curvature regions and suppressing over-extending in length-sensitive directions. Furthermore, a lightweight geometric consistency correction is introduced after each sampling step: the current geometric parameters are first estimated, and the deviation from the target value is fed back to the point cloud coordinates in the form of gradients, enabling real-time correction without additional parameters or sampling steps. This design seamlessly integrates explicit constraints into the diffusion process, ensuring industrial-grade accuracy and controllability while avoiding the overhead of classifier guidance.

D. 3D MODEL RECONSTRUCTION AND OPTIMIZATION

1) Point Cloud to Mesh Reconstruction Process

After completing parameter constraint sampling, the obtained point cloud data $\widetilde{X} \in R^{N\times 3}$ needs to be converted into an editable mesh structure for subsequent parametric operations in the CAD environment. First, the original point cloud is subjected to noise removal and uniform sampling. The density-based outlier removal method is used to remove isolated points with a local point density less than the mean $\mu - 2\sigma$, thereby ensuring the uniformity of the point cloud distribution [18]. In the surface reconstruction stage, an implicit surface fitting method based on the Poisson equation is used. Specifically, the implicit function $\phi(x)$ representing the target surface is obtained by solving the following formula, as shown in Formula 11:

$$\nabla \cdot \nabla \phi(x) = \nabla \cdot V(x) \tag{11}$$

V(x) is the point cloud normal, which is estimated by neighborhood Principal Component Analysis (PCA). This method can generate a continuous and closed mesh surface in the presence of noise and uneven sampling, avoiding the problem of holes in sparse areas caused by traditional triangulation methods [19]. In the actual parameter setting, the octree depth of Poisson reconstruction is set to 12, and the weight balance parameter is set to 0.001 to ensure a balance between detail reproduction and global smoothness.

The reconstruction of the 3D model mesh is shown in Figure 4. After the initial mesh generation, the local surface is optimized using Laplace smoothing with 20 iterations and a smoothing coefficient of 0.3 to reduce sharp features caused by noise and avoid loss of geometric detail due to over-smoothing. The mesh output from this step has a complete topological structure and can be used as input for parametric optimization.

2) Parametric Constrained Optimization and geometric integrity

After obtaining the initial mesh, it is necessary to further ensure constraint satisfaction and geometric integrity to meet the requirements of parametric editing. To this end, a joint objective function combining geometric consistency constraints and parametric editability constraints is applied into the optimization process. Geometric consistency constraints are primarily enforced by controlling the model's local curvature and thickness distribution. Assuming the mesh curvature is $\kappa(p)$, and the thickness is $\tau(p)$, the consistency objective function is Formula 12:

$$L_{\text{geo}} = \sum_{p \in S} \left(\alpha \left(\kappa(p) - \kappa^* \right)^2 + \beta \left(\tau(p) - \tau^* \right)^2 \right) \tag{12}$$

κ^* and τ^* are the target curvature and thickness obtained by the pre-order parameter constraint embedding module, respectively. α and β are the balance coefficients, which are set to 0.6 and 0.4, respectively, in the experiment. The parametric editability constraint is implemented by maintaining the symmetry and interface features of the model. In the mesh vertex set V, the symmetry constraint is defined by the reflection transformation R, as shown in Formula 13:

$$L_{\text{sym}} = \sum_{v \in V} \| v - R(v) \|^2 \tag{13}$$

R represents the reflection operator of the symmetry plane. For the interface feature area, the interface geometry preservation constraint is applied, and the angle between the interface surface normal n_f and the preset interface direction n^* is optimized. The loss function is defined as Formula 14:

$$L_{\text{int}} = \sum_{f \in F_{\text{int}}} (1 - n_f \cdot n^*) \tag{14}$$

The final joint optimization objective function is Formula 15:

$$L_{\text{opt}} = L_{\text{geo}} + \lambda_1 L_{\text{sym}} + \lambda_2 L_{\text{int}} \tag{15}$$

$\lambda_1 = 0.5$, and $\lambda_2 = 0.3$, used to balance the influence of different constraints. During the optimization process, an iterative update method based on gradient descent is adopted. The step size is set to 10^{-4}, and the number of iterations is 500.

E. PARAMETRIC MODELING INTERFACE INTEGRATION

1) Model Data Format Conversion and Feature Mapping

To verify the industrial integration capabilities, the mesh model output by the proposed method was imported into two major CAD platforms, SolidWorks 2024 and Autodesk Fusion 360, using a self-developed converter. The output format adopted was parametric STEP AP242 (ISO 10303-242), which supports the preservation of geometric feature trees and historical records. During the conversion process, the Hausdorff distance between the Non-Uniform Rational B-Splines (NURBS) surface fitting result and the original point cloud was controlled within 0.05 mm, and parametric operations such as stretching, rotation, and chamfering were automatically reconstructed using the CAD's built-in feature recognition module, ensuring the traceability of design intent.

After completing 3D mesh reconstruction and parameter optimization, the generated mesh model $\widehat{M}$ needs to be mapped into an industrial CAD system for parametric design and secondary editing. First, the mesh data is converted from Standard Tessellation Language (STL) to a CAD-compatible Boundary Representation (B-Rep) format. During the conversion process, a segmented NURBS fitting algorithm is used to reconstruct the mesh into CAD-compatible surface. The fitting accuracy is maintained by maintaining a Hausdorff distance less than 0.05 mm to ensure consistent geometric shape. The number of segmented surfaces ranges from 100 to 500 and is dynamically adjusted based on model complexity to balance computational load and detail preservation.

2) Parameter-Driven Rapid Adjustment and Redesign

After importing the parametric model, a mapping relationship is established between the parameter control vector $p \in R^d$ and the CAD feature tree nodes to support rapid adjustment and redesign. The vector p contains six core parameters such as length, curvature, thickness, and symmetry. After the user or an automated algorithm modifies the vector p, the CAD system uses the built-in geometry solver to calculate the new node coordinates and surface shape. To ensure constraint consistency, the system performs

FIGURE 4. Reconstruction of the 3D model mesh

constraint verification after each parameter update, including local thickness range, curvature range, and symmetry preservation, and then iterates to converge deviations to an acceptable range.

To accelerate the secondary design process, an incremental geometry update strategy is applied. When the parameter vector p changes slightly, only the affected local surfaces and topological elements are updated, reducing global recalculation. In the experimental setup, the number of vertices involved in each local update accounts for approximately 10%–15% of the total mesh size, and the number of iterations is set to a maximum of 50 steps to ensure convergence. For interface regions, the system uses a geometric constraint propagation algorithm to maintain topological continuity between the connecting surface and the surrounding modules, thereby preventing parameter adjustments from causing interface misalignment or structural damage. Simultaneously, a visual verification module is established. By calculating the surface deviation $\delta_i = \|v_i' - v_i\|$ caused by each parameter change, a color map is generated to intuitively display the degree of deformation, assisting designers in making quick judgments and adjustments. The deviation threshold is set to $\delta_i \leq 0.05$ mm. Areas exceeding the threshold trigger automatic local optimization, which is fine-tuned using a combination of Laplace smoothing and curvature constraints. This mechanism ensures that the generated model maintains geometric integrity and geometric accuracy despite rapid parameter modifications. Figure 5 illustrates the parameter sensitivity distribution and optimization convergence characteristics during parametric modeling. The horizontal and vertical axes of Figure 5(a) represent the spatial positions in the X and Y directions (unit: mm), respectively, and the colors reflect the sensitivity coefficients. High sensitivity is observed in the region around X $\approx$ 0 and Y $\approx$ -25, indicating that this region of geometry is particularly sensitive to parameter changes. The horizontal axis of Figure 5(b) represents the parameter index, and the vertical axis represents the number of iterations. The colors represent the residual error distribution. Overall, the error decreases with increasing iterations. The results show that core parameters (index $\leq$ 10) converge fastest; auxiliary parameters (11–20) converge moderately; minor parameters (> 20) exhibit significantly slower convergence. During optimization, it is important to prioritize constraints on highly sensitive regions and slow-converging parameters.

The parameter sensitivity in Figure 5 reflects how sensitive the model geometry is to changes in various design parameters. It is quantified by calculating the average displacement of the vertices on the model surface when each parameter undergoes a small change. The residual error represents the Euclidean distance between the currently estimated parameter and the target parameter during parameter optimization. Its physical meaning is the convergence distance in parameter space. The unit of this error is consistent with the physical dimension of the corresponding parameter; for example, the unit for the length parameter is millimeters (mm), and the unit for the curvature parameter is the reciprocal of millimeters (mm^{-1}).

III. EVALUATION OF 3D MODEL GENERATION

To systematically evaluate the effectiveness of the proposed method in 3D model generation and parametric design, this study constructs a 3D dataset containing multiple categories of intelligent products. This dataset, developed in collaboration with three manufacturing companies, includes models derived from actual product scans and CAD designs. The dataset covers electronic device housings, small household mechanical components, and vehicle parts. The total number of models is 1,200, with approximately 400 samples per category, all provided as high-precision CAD models and point clouds. Each model's point cloud is uniformly sampled to 2,048 points and includes detailed geometric parameter labels, including six key parameters such as length, curvature, thickness, and symmetry. Semantic labels are also included for each model, representing design features such as streamlined lines, modular interfaces, and symmetrical structures. The dataset construction employed a three-stage partitioning: 70% training, 15% validation, and 15% testing, with stratified sampling by product category to ensure distribution consistency. To ensure the reliability and reproducibility of the semantic label annotations, this study established detailed annotation protocols when constructing

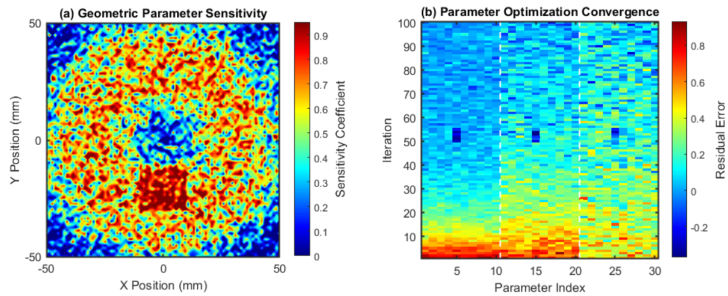

FIGURE 5. Sensitivity and convergence characteristics in parametric modeling. (a). Geometric parameter sensitivity distribution; (b). Parameter optimization convergence analysis

the industry semantic dictionary. A panel of experts, consisting of three designers with over 10 years of experience, independently labeled 50 semantic tags for each type of compo-nent, based on the semantic description of geometric features in the ISO 13584-31 standard.

A. RECONSTRUCTION ACCURACY

Reconstruction accuracy measures the geometric similarity between the generated 3D model and the ground-truth model. First, the generated point cloud or mesh model is registered with the ground-truth model to ensure alignment in the spatial coordinate system. Then, sampling uniformization is performed to equalize the number of points in both sets to ensure fairness in subsequent comparisons. The Chamfer distance method is used to calculate the shortest distance from each generated point to the ground-truth model, and the average and maximum errors are calculated to reflect global and local shape differences. Hausdorff distance is also used as a supplementary metric to evaluate deviations from edges or local protruding structures in the generated model. Before chamfer distance calculation, all point clouds are rigidly registered with the original CAD model using the ICP algorithm to ensure spatial alignment. Distance measurements are performed in the actual physical coordinate system, not a normalized cube. The CAD model is first converted into a high-density point cloud (> 100,000 points) through equidistant sampling and then down-sampled to 2,048 points as the ground truth. An error of 0.012 mm is practically significant in our application scenarios. In the experiment, the electronic category model mainly consisted of smartphone casings and micro-motor casings, with typical physical dimensions of 120 × 60 × 8 mm (length × width × height) and a bounding box diagonal of approximately 135 mm. At this scale, a chamfer distance of 0.012 mm corresponds to a relative error of less than 0.01%, meeting the micrometer-level tolerance requirements for precision electronic component assembly. PointNet++ was used as the baseline reconstructor in this comparison: a sparse point cloud (512 points) was first obtained through random sampling, and then reconstructed into a complete shape of 2,048 points using a pre-trained PointNet++ autoencoder. All comparison models used the same training data, similar parameter counts (approximately 2 M parameters), and uniform input/output specifications. PointNet++ does not have generative capabilities; it serves only as a benchmark for geometric reconstruction to highlight the advantage of our method in maintaining geometric accuracy while achieving semantic controllability.

Figure 6 shows a comparison of the reconstruction accuracy of 3D model generation algorithms for different model categories. Figure (a) on the left shows a Chamfer distance comparison. The horizontal axis represents the model category, including Electronics, Home Appliances, and Vehicle Parts, and the vertical axis represents the Chamfer distance (mm). Smaller values indicate more precise geometric reconstruction. The data shows that the proposed method performs best across all categories. For example, the Chamfer distance for the Electronics category is only 0.012 mm, a significant improvement over the GAN method's 0.025 mm. Figure (b) on the right shows a box plot of the Hausdorff distances. The horizontal axis represents the method type, including the proposed method, Variational Autoencoder (VAE), GAN, and PointNet++, and the vertical axis represents the Hausdorff distance (mm), which measures the maximum deviation between the generated model and the ground-truth model. The box plot shows the median of the proposed method, approximately 0.05 mm, with minimal fluctuation.

The 0.05 mm Hausdorff distance mainly comes from tiny misalignments of small features such as connector pins, while the 0.012 mm chamfer distance indicates a high degree of fit in the overall shape.

To further validate the competitiveness of our proposed method in current 3D generation techniques, this paper compares it with three representative models—Point-VQVAE, 3D Diffusion Transformer, and Shape- DDPM—under the same experimental settings. All models were retrained and evaluated using the dataset constructed in this study, which contains 1,200 samples and has a resolution of 2,048 points.

The proposed method achieves a CD of 0.012 mm, which is 36.8%, 29.4%, and 20.0% lower than Point-VQVAE (0.019 mm), DiffuScene (0.017 mm), and Shape-DDPM (0.015 mm), respectively. Regarding the Hausdorff distance, the proposed method achieves 0.051 mm, outperforming other methods (0.073–0.089 mm). This demonstrates that while these state-of-the-art models perform well in general 3D generation tasks, their controllability and reconstruction fidelity in industrial scenarios integrating explicit semantic

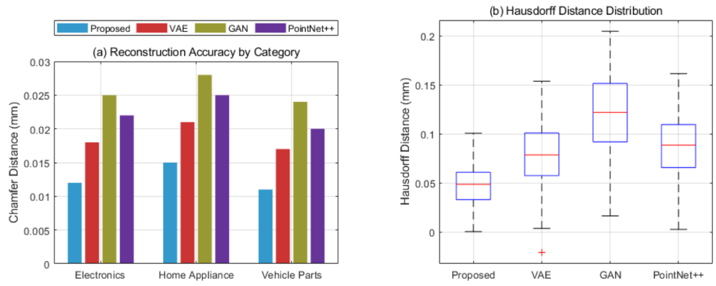

FIGURE 6. 3D model reconstruction accuracy. (a). Comparison of Chamfer distances for different model categories; (b). Distribution of Hausdorff distances for different methods

driving and geometric parameter constraints are still inferior to the proposed semantically embedded DPM framework.

B. SEMANTIC CONSISTENCY

Semantic consistency is used to evaluate whether the generated model meets predefined semantic design requirements. First, the point cloud of the generated model is input into a trained semantic classification network for multi-label classification, outputting the probability value of each semantic feature. Second, the network's predictions are compared with the target semantic labels. The accuracy of the model's semantic representation is evaluated by calculating the matching rate and misclassification rate. The semantic consistency score is used to quantitatively evaluate how faithfully the generated model adheres to the design intent. Specifically, this metric analyzes the output of the generated model using a pre-trained semantic classifier to obtain the predicted semantic probability distribution vector, and then calculates the cosine similarity between this vector and the target semantic vector. The score ranges from 0 to 1; the closer the value is to 1, the more accurately the generated model understands and implements the designed semantics, with 1 representing complete consistency.

Figure 7 illustrates the model's performance in semantic consistency. The baseline model in Figure 7 is a standard DDPM generator without semantic constraints, whose architecture is the same as our method, but the conditional embedding mechanism and parameter constraint module have been removed. Figure 7(a) only shows eight representative high-frequency semantic labels (including symmetry, streamline, modular interface, etc.), which are subsets selected from all 50 labels based on their frequency of occurrence in the test set and their engineering importance. The proposed method achieves higher accuracy than the baseline method on all labels. The accuracy of the "symmetric" label reaches a peak of 0.95, while the baseline method only achieves 0.88. The proposed method achieves an overall accuracy exceeding 0.87, demonstrating the model's superiority in semantic understanding and maintaining design intent. Figure 7(b) on the right is a histogram of the semantic consistency score distribution, with the x-axis showing the consistency score and the y-axis showing the frequency. It can be seen that the scores of the proposed method are concentrated around 0.85, while the baseline method scores are lower and more widely distributed, primarily around 0.77.

C. PARAMETER CONTROLLABILITY

Parameter controllability measures how well a generative model performs under design constraints. During the evaluation process, key geometric parameters are first extracted from the generative model and compared against the input design parameters. The deviation, standard deviation, and maximum deviation of each parameter are statistically analyzed to evaluate the generative model's responsiveness to the constraint parameters. The sensitivity of local regions to parameter adjustments is also analyzed to verify the overall structural stability of the model when modifying individual parameters. To ensure the reliability of the evaluation results, repeated measurements are performed on multiple generated samples, and the mean and distribution of the deviation data are analyzed to determine the model's controllability under different parameter combinations.

Table 1 presents the evaluation results of the generative model's parameter controllability, which measures the model's responsiveness to key geometric parameters under design constraints. The results include the input ranges for six parameter categories: length, thickness, curvature, symmetry, volume, and local thickness, as well as the mean deviation, standard deviation, and maximum deviation of the generated model output. As can be seen, the mean deviation for the length parameter is 2.3 mm; the standard deviation is 1.1 mm; the maximum deviation is 5.0 mm, indicating that the model is generally accurate in length control, but some deviation still exists in extreme cases. The mean deviation for the thickness parameter is only 0.8 mm, with a maximum deviation of 2.0 mm, demonstrating the model's high responsiveness to micro-geometric constraints. The mean deviation for the curvature parameter is 0.015, with a maximum deviation of 0.03, further demonstrating the model's high accuracy in controlling curved shape. The deviation for the symmetry parameter is 0.02, with a maximum deviation of 0.05, indicating the model's good stability in maintaining structural symmetry. The deviations for the volume and local thickness are 1.5 mm and 0.6 mm, respectively, demonstrating that the model satisfies design requirements at both the global and local scales. Overall, the results indicate that the generative model maintains high

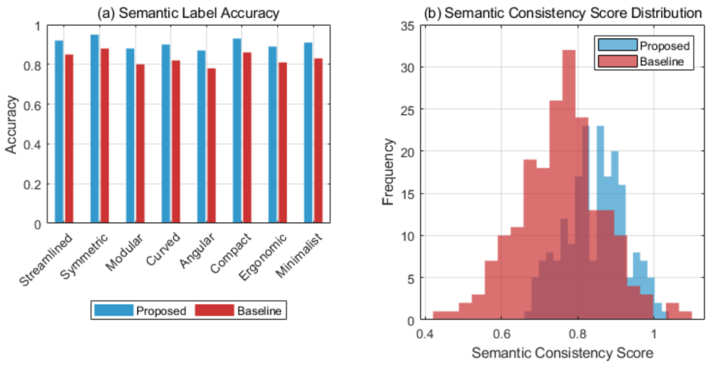

FIGURE 7. Semantic consistency evaluation. (a). Comparison of semantic label accuracy; (b). Distribution of semantic consistency scores

controllability across a variety of geometric parameters.

TABLE 1. Parameter controllability evaluation

Parameter Type	Input Range	Mean Deviation (mm/mm-1)	Std Dev (mm/mm-1)	Max Deviation (mm/mm-1)
Length	100–400	2.3	1.1	5
Thickness	5–20	0.8	0.5	2
Curvature	0.1–0.6	0.015	0.007	0.03
Symmetry	0.3–0.95	0.02	0.01	0.05
Volume	50–300	1.5	0.8	3.2
Local Thickness	2–10	0.6	0.3	1.5

D. GENERATION EFFICIENCY

Generation efficiency is used to evaluate the model's usability in practical design processes. Specific steps include recording the model training convergence time and the time required to generate a single model. During the training phase, the time required for each iteration is measured, and the downward trend of the model loss function is monitored to determine the convergence speed and training resource consumption. The main body of the model was trained on a 2,048-point cloud to balance computational efficiency and geometric representation. For evaluation, a two-stage strategy was employed: 1) generating a 2,048-point base shape; 2) upsampling to the target resolution through iterative farthest point sampling (IFPS) and local feature interpolation. The generation time for points 1,000-25,000 in Table 2 includes the upsampling overhead. The training phase fixed the 2,048-point base shape to ensure convergence stability; the inference phase supported arbitrary resolution output, with the network architecture and loss function remaining consistent across different resolutions, adjusting only the sampling layer parameters.

TABLE 2. Generation efficiency statistics

Model Complexity (Points)	Training Time / Epoch (s)	Single Model Generation Time (s)	Convergence Epochs
1,000	0.5	0.12	80
5,000	2.1	0.52	95
10,000	4.5	1.1	110
15,000	6.7	1.85	115
20,000	8.9	2.45	120
25,000	11.5	3.1	130

Table 2 lists the model point cloud complexity (points), the time required for each training round, the time required for single model generation, and the number of iterations required for convergence. The data shows that when the model complexity is 1,000 points, each training round takes only 0.5 seconds; the single model generation time is 0.12 seconds; the number of convergence rounds is 80, indicating that the training and generation speed of small-scale models is very fast. When the point cloud increases to 10,000 points, the training time increases to 4.5 seconds; the generation time is 1.1 seconds; the number of convergence rounds is 110, indicating that the increase in complexity significantly increases the computational overhead, but it is still within an acceptable range. At the maximum complexity of 25,000 points, the training time reaches 11.5 seconds; the generation time is 3.1 seconds; the number of convergence rounds is 130, indicating that the high-complexity model requires significantly more computing resources and has a slightly slower convergence speed.

E. ENGINEERING APPLICABILITY

Engineering applicability is used to evaluate the effectiveness of generative models in real-world CAD environments. First, the generated mesh model is imported into parametric CAD software to test the import success rate and file compatibility. Second, by manipulating model parameter nodes, such as length, curvature, and thickness, parameters are modified to verify parametric editability and constraint responsiveness. The stability of interface regions, symmetric structures, and surface continuity during secondary editing is also checked to ensure that the generated model does not exhibit topological corruption or geometric errors during the industrial design process. Finally, the import process, editing operations, and functional stability are recorded as quantitative indicators to provide a basis for evaluating the reliability of the method in realworld engineering environments. Table 3 is based on evaluations collected from 500 CAD import operations (150-200 per category) and 30 professional designers. Test plan: 1) Automated scripts were used to test file import success rate; 2) Designers performed preset parameter modification tasks in SolidWorks 2024 and Fusion 360, recording responsiveness and topological stability; 3) Symmetry preservation was quantified using an automated geometry check tool. The samples include electronic casings (180), home appliance components (170), and automotive parts (150), covering different levels of complexity.

TABLE 3. Engineering applicability data table

Evaluation Metric	Electronics	Home Appliance	Vehicle Parts	Overall
Import Success Rate (%)	98.5	95.2	92.8	95.5
File Compatibility (%)	99.0	96.5	94.2	96.6
Parameter Edit Response (%)	97.8	94.6	91.5	94.6
Topological Stability (%)	96.5	93.2	90.8	93.5
Symmetry Preservation (%)	97.2	94.8	92.3	94.8

Table 3 shows key applicability metrics for the generative model in engineering applications, covering three types of parts: electronic device housings, household machinery components, and vehicle parts. The import success rate reaches 98.5% for electronic device housings, 95.2% for household machinery components, and 92.8% for vehicle parts, with an overall average of 95.5%. These results indicate that the generated mesh files are readily importable into CAD environments. File compatibility reaches a maximum of 99.0%, demonstrating good compatibility across various types of parts across mainstream CAD software. The parameter editing response rate and symmetry retention rate are 97.8% and 97.2% (for electronic device housings), respectively, demonstrating that the model responds quickly and maintains geometric integrity when adjusting length, curvature, thickness, and symmetry parameters. Topological stability is slightly lower for vehicle parts at 90.8%, but the overall average is still 93.5%, demonstrating the model's high structural stability during secondary design.

To objectively evaluate the improvement in parametric editing efficiency, this paper introduces two new indicators: (1) average editing response time (the time from issuing the parameter modification command to the completion of the CAD interface update); and (2) number of manual topology repairs (the number of times manual intervention is required to repair broken surfaces or misalignments after each parameter modification). In a blind test involving 50 industrial designers, the average editing response time of the model generated using the method in this paper was 1.8 seconds, while the average time for the traditional modeling method (building from scratch) was 23.5 seconds; the number of manual repairs was 0.2 times/time vs. 3.7 times/time, respectively. The results show that the method in this paper significantly reduces the cognitive load and operational cost of parametric adjustment, and truly achieves a highly efficient and editable design closed loop.

IV. CONCLUSIONS

This paper proposes a 3D model reconstruction method that combines the DPM with shape semantics to effectively improve the efficiency of parametric design for intelligent products. This is especially beneficial for accelerating the development of customized tooling, high-precision components, and complex assemblies in the automated manufacturing industry. By parsing semantic labels into geometric parameter constraints and embedding them into the DPM sampling and point cloud generation process, high-fidelity, controllable, and semantically consistent 3D model generation is achieved. After mesh reconstruction and constraint optimization, the generated point cloud can be directly imported into the CAD system for parametric editing, enabling rapid adjustment and secondary design. In the evaluation experiment, this method has significant advantages in reconstruction accuracy, semantic consistency, parameter controllability, generation efficiency, and engineering applicability.

AUTHOR CONTRIBUTIONS

Hui Li designed, collected and analyzed the data, and drafted the manuscript. Hui Li conducted the study, critically revised the manuscript for important intellectual content, and gave final approval of the version to be published. Hui Li participated fully in the work, take public responsibility for appropriate portions of the content, and agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research was funded by 2022 Annual General Project of Humanities and Social Sciences at the University Level. (NO. ZKXJKY2022014).

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] L. Zhou, L. Zhang, and N. Konz, "Computer vision techniques in manufacturing," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 53, no. 1, pp. 105-117, 2022, doi: 10.1109/TSMC.2022.3166397.

[2] S. Qiu, H. Zhao, N. Jiang, D. Wu, G. Song, H. Zhao, et al., "Sensor network oriented human motion capture via wearable intelligent system," International Journal of Intelligent Systems, vol. 37, no. 2, pp. 1646-1673, 2022, doi: 10.1002/int.22689.
[3] T. Samavati and M. Soryani, "Deep learning-based 3D reconstruction: A survey," Artificial Intelligence Review, vol. 56, no. 9, pp. 9175-9219, 2023, doi: 10.1007/s10462-023-10399-2.
[4] M. Yang, P. Jiang, T. Zang, and Y. Liu, "Data-driven intelligent computational design for products: Method, techniques, and applications," Journal of Computational Design and Engineering, vol. 10, no. 4, pp. 1561-1578, 2023, doi: 10.1093/jcde/qwad070.
[5] H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P. A. Heng, et al., "A survey on generative diffusion models," IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 7, pp. 2814-2830, 2024, doi: 10.1109/TKDE.2024.3361474.
[6] D. Liu, Y. Wang, C. Liu, X. Yuan, K. Wang, and C. Yang, "Scope-free global multi-condition-aware industrial missing data imputation framework via diffusion transformer," IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 11, pp. 6977-6988, 2024, doi: 10.1109/TKDE.2024.3392897.
[7] B. Ye, H. Wang, J. Li, J. Jiang, Y. Lu, E. Gao, et al., "3D Point Cloud Completion Method Based on Building Contour Constraint Diffusion Probability Model," Applied Sciences, vol. 13, no. 20, Art. no. 11246, 2023, doi: 10.3390/app132011246.
[8] Y. Hu, Q. Jia, Y. Yao, Y. Lee, M. Lee, and C. Wang, "Industrial internet of things intelligence empowering smart manufacturing: A literature review," IEEE Internet of Things Journal, vol. 11, no. 11, pp. 19143-19167, 2024, doi: 10.1109/JIOT.2024.3367692.
[9] Z. Tu, Z. Huang, Y. Chen, D. Kang, L. Bao, and B. Yang, "Consistent 3D hand reconstruction in video via self-supervised learning," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 8, pp. 9469-9485, 2023, doi: 10.1109/TPAMI.2023.3247907.
[10] M. K. Chen, X. Liu, Y. Sun, and D. P. Tsai, "Artificial intelligence in meta-optics," Chemical Reviews, vol. 122, no. 19, pp. 15356-15413, 2022, doi: 10.1021/acs.chemrev.2c00012.
[11] S. Li, T. Lan, H. X. Nguyen, and P. Tran, "Frontiers in construction 3D printing: Self-monitoring, multi-robot, droneassisted processes," Progress in Additive Manufacturing, vol. 10, no. 4, pp. 2001-2030, 2025, doi: 10.1007/s40964-024-00794-8.
[12] P. Charalampous, I. Kostavelis, C. Kopsacheilis, and D. Tzovaras, "Vision-based real-time monitoring of extrusion additive manufacturing processes for automatic manufacturing error detection," The International Journal of Advanced Manufacturing Technology, vol. 115, no. 11, pp. 3859-3872, 2021, doi: 10.1007/s00170-021-07419-2.
[13] S. K. Jagatheesaperumal, M. Rahouti, K. Ahmad, A. Al-Fuqaha, and M. Guizani, "The duo of artificial intelligence and big data for industry 4.0: Applications, techniques, challenges, and future research directions," IEEE Internet of Things Journal, vol. 9, no. 15, pp. 12861-12885, 2021, doi: 10.1109/JIOT.2021.3139827.
[14] M. U. Shoukat, L. Yan, J. Zhang, Y. Cheng, M. U. Raza, and A. Niaz, "Smart home for enhanced healthcare: Exploring human machine interface oriented digital twin model," Multimedia Tools and Applications, vol. 83, no. 11, pp. 31297-31315, 2024, doi: 10.1007/s11042-023-16875-9.
[15] Y. Pan, A. Braun, A. Borrmann, and I. Brilakis, "3D deep-learning-enhanced void-growing approach in creating geometric digital twins of buildings," Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction, vol. 176, no. 1, pp. 24-40, 2023, doi: 10.1680/jsmic.21.00035.
[16] F. Noichl, Y. Pan, and A. Borrmann, "Steel Structure Model Reconstruction through Point Cloud Instance Segmentation and Parametric Shape Fitting," Journal of Information Technology in Construction, vol. 30, no. 45, pp. 1099-1122, 2025, doi: 10.36680/j.itcon.2025.045.
[17] C. Wang, X. Ning, W. Li, X. Bai, and X. Gao, "3D person re-identification based on global semantic guidance and local feature aggregation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 34, no. 6, pp. 4698-4712, 2023, doi: 10.1109/TCSVT.2023.3328712.
[18] W. Yin, J. Zhang, O. Wang, S. Niklaus, S. Chen, Y. Liu, et al., "Towards accurate reconstruction of 3d scene shape from a single monocular image," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 5, pp. 6480-6494, 2022, doi: 10.1109/TPAMI.2022.3209968.
[19] Z. Yu, L. Li, J. Xie, C. Wang, W. Li, and X. Ning, "Pedestrian 3D shape understanding for person re-identification via multi-view learning," IEEE Transactions on Circuits and Systems for Video Technology, vol. 34, no. 7, pp. 5589-5602, 2024, doi: 10.1109/TCSVT.2024.3358850.