# Constraint-Aware Core-Cavity Generation Method for Injection Mold Design

1st Wenlin Yu
*School of Mathematics and Physics*
*Qingdao University of Science and Technology*
Qingdao, China
2201020326@mails.qust.edu.cn

2nd Defa Liu
*School of Data Science*
*Qingdao University of Science and Technology*
Qingdao, China
2308180205@mails.qust.edu.cn

3rd Haoran Shi
*School of Mathematics and Physics*
*Qingdao University of Science and Technology*
Qingdao, China
2306040213@mails.qust.edu.cn

4th Ainiwa Aidibai
*School of Mechanical and Electrical Engineering*
*Qingdao University of Science and Technology*
Qingdao, China
2405130301@mails.qust.edu.cn

5th Hua Geng
*School of Mathematics and Physics*
*Qingdao University of Science and Technology*
Qingdao, China
2413020102@mails.qust.edu.cn

6th Hengkai Yao*
*School of Mathematics and Physics*
*Qingdao University of Science and Technology*
Qingdao, China
*yaohengkai@qust.edu.cn

2026 6th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA) | 979-8-3315-6113-0/26/$31.00 ©2026 IEEE | DOI: 10.1109/AIITA69518.2026.11567358

***Abstract*—To address the challenges in core-cavity design for injection molds—namely reliance on manual experience, low automation, and difficulty in ensuring manufacturability—this paper proposes a constraint-aware core-cavity generation method tailored for injection mold design. Taking product 3D geometric data as input, the method formulates core-cavity generation as a 3D voxel generation problem with strong engineering constraints, thereby uniformly describing complex geometric complementarity and process constraints. To address the challenge of core-cavity regions occupying an extremely small proportion in voxel space and being easily overwhelmed by background information during training, a data augmentation strategy based on nearest-neighbor densification and a dynamically adjusted Focal loss function are introduced. This mitigates the impact of extreme class imbalance on model training stability and accuracy. Simultaneously, draft angle constraints and core-cavity mutual exclusion constraints are embedded in the network training process in a differentiable form, guiding the model to generate structures that meet practical injection molding requirements. Experimental results demonstrate that the proposed method stably generates geometrically consistent and process-feasible core-cavity structures across multiple sets of typical injection mold data, achieving significant improvements in key area accuracy and engineering usability compared to traditional 3D segmentation methods. The findings provide an effective data-driven solution for intelligent injection mold design.**

***Index Terms*—Injection mold design; Core–cavity generation; Constraint-aware modeling; 3D voxel learning; Deep learning; Smart manufacturin**

## I. Introduction

Injection molds are critical in plastic forming; core-cavity design directly affects product quality. Traditional design relies on manual experience and rule-based operations, leading to long cycles and inefficiency [1]. With smart manufacturing, data-driven automatic generation of mold structures has become a key research direction.

Deep learning has advanced 3D analysis, including segmentation and shape completion [2], [3]. Some works apply 3D learning to mold design [4], but core-cavity generation is not a simple segmentation task—it involves strong engineering constraints and extreme class imbalance: cores and cavities occupy less than 1% of voxels in our dataset, causing conventional losses to be dominated by background [5]. Moreover, injection molding requires draft angles and spatial mutual exclusivity, which pure data-driven models often fail to satisfy [6].

Prior efforts incorporate engineering constraints into deep learning [7], but they typically rely on post-processing or rule filtering, lacking effective guidance during training and struggling with extreme imbalance. Thus, unifying geometric complementarity, class imbalance, and process constraints remains an open problem.

This paper proposes a constraint-aware core-cavity generation method with three key contributions: (1) unified constrained 3D voxel generation modeling, (2) a mechanism handling extreme class imbalance via KNN densification and dynamic Focal Loss, and (3) differentiable embedding of draft angle and mutual exclusivity constraints. Experiments on real datasets show significant gains in accuracy and manufacturability over conventional methods.

## II. Related Work

This work spans mold design automation, 3D deep learning, and engineering constraint integration.

**Mold design automation:** Traditional core-cavity generation relies on geometric reasoning and feature recognition [8], [9]. Rule-based systems and template matching have also been explored [10], [11]. However, these methods require extensive manual intervention for complex parts [12].

**3D deep learning:** Point-based methods like PointNet [13] and voxel-based CNNs [14] have advanced 3D analysis. Generative models such as 3D-GANs and VAEs [15], [16] enable shape generation. Direct application to core-cavity generation faces challenges: extreme class imbalance (core/cavity occupy 0.1% voxels) and lack of process constraints [4], [17].

**Class imbalance:** Focal Loss [5] addresses imbalance by down-weighting easy samples. Dynamic weighting strategies [18], [19] adaptively adjust focus during training. However, mold imbalance (1:1000) is far more severe than typical scenarios.

**Constraint embedding:** Differentiable rendering [20] and physics-informed neural networks [21] incorporate constraints into learning, yet mold-specific rules (draft angle, mutual exclusivity) remain underexplored.

In summary, no prior work unifies extreme imbalance handling with differentiable injection-mold constraints. This paper fills this gap via KNN densification, dynamic Focal Loss, and differentiable constraint embedding.

## III. Constraint-Aware Core-Cavity Generation Method

The proposed method takes a product point cloud as input and outputs core–cavity voxel representations that satisfy geometric complementarity and process constraints. This section formulates the problem, defines decision variables and core constraints, and details the KNN densification, SE-enhanced 3D U-Net, dynamic Focal Loss, and differentiable constraint embedding.

### A. Problem Modeling and Decision Variables

Let the point cloud be $\mathcal{P} = \{p_i\}_{i=1}^{N}$, $p_i \in \mathbb{R}^3$. On a regular grid $\mathcal{V} = \{v_{x,y,z}\}$, we aim to assign each voxel to background, core, or cavity. To enable differentiable optimization, we introduce a continuous probabilistic field

$$S(v) = [s_{\text{bg}}(v), s_{\text{core}}(v), s_{\text{cavity}}(v)] \in [0,1]^3,\ \forall v \in \mathcal{V}, \quad (1)$$

with $s_{\text{bg}} + s_{\text{core}} + s_{\text{cavity}} = 1$. The final binary occupancy is $O(v) = \arg\max_c s_c(v)$. The generation must satisfy:

- **Spatial exclusion:** $\mathcal{V}_{\text{core}} \cap \mathcal{V}_{\text{cavity}} = \emptyset$.
- **Draft angle:** Surface normals must deviate from the opening direction by at most $\theta_{\min}$.
- **Geometric matching:** Cavity matches outer product surface; core complements inner surface.

### B. KNN Point Cloud Densification

Raw data contain only $\sim 0.1\%$ core/cavity voxels. For each point $p_i$, we generate $m = 2$ new points from its $k = 5$ nearest neighbors $\mathcal{N}_k(p_i)$ via weighted interpolation:

$$p_{\text{new}} = \frac{\sum_{j \in \mathcal{N}_k(p_i)} w_j p_j}{\sum w_j}, \quad w_j = \|p_j - p_i\|^{-1}. \quad (2)$$

This increases target-region point density by $\sim 10\times$ (training only). The densified cloud is voxelized to $128^3$; each occupied voxel receives a 5-channel feature $\mathcal{X}$ (occupancy, mean normal, local density).

### C. SE-Enhanced 3D U-Net Architecture

We adopt a 3D U-Net with Squeeze-and-Excitation (SE) blocks. The encoder has four down-sampling stages (residual block + SE + stride-2 3D conv). SE recalibrates channels via

$$z = \sigma(W_2 \delta(W_1 \text{GAP}(x))), \quad \tilde{x} = z \cdot x, \quad (3)$$

where $\delta$=ReLU, $\sigma$=Sigmoid. The decoder symmetrically up-samples with skip connections. After multi-scale aggregation, a $1 \times 1 \times 1$ convolution and softmax yield the probabilistic field $S$.The overall network structure is shown in Figure 1.

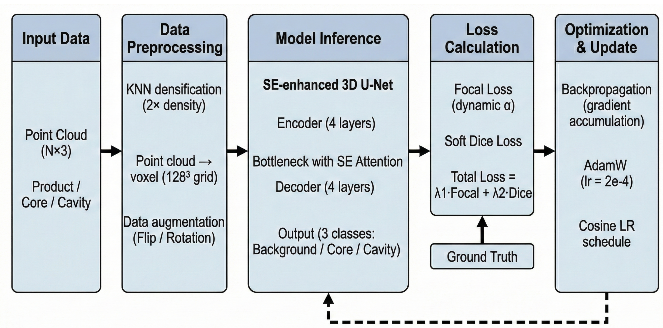


Fig. 1. SE-Enhanced 3D U-Net Architecture and Training Flowchart.

### D. Dynamic Focal Loss

For core/cavity classes (background excluded) we employ Focal Loss:

$$\mathcal{L}_{\text{focal}} = -\frac{1}{|\mathcal{V}|} \sum_{v} \sum_{c \in \{\text{core,cav}\}} \alpha_c^{(t)} (1 - p_c(v))^{\gamma} \log p_c(v), \quad (4)$$

with $\gamma = 2.0$. Class weights $\alpha_c$ are dynamically adjusted based on validation Dice improvement:

$$\alpha_c^{(t+1)} = \alpha_c^{(t)} \left( 1 + \lambda \cdot \mathbb{I}\left[ \frac{D_c^{(t)} - D_c^{(t-1)}}{D_c^{(t-1)}} < \tau \right] \right), \quad (5)$$

starting from $[0.1, 10, 10]$ and converging to $[0.1, 19.8, 19.8]$.

### *E. Differentiable Constraint Embedding*

**Spatial mutual exclusivity:**

$$\mathcal{L}_{\text{mutual}} = \frac{1}{|\mathcal{V}|} \sum_{v} \max(0, s_{\text{core}} + s_{\text{cavity}} - 1). \tag{6}$$

**Draft angle:** Surface normals are estimated from probability gradients:

$$\vec{n}_{\text{core}} = -\frac{\nabla s_{\text{core}}}{\|\nabla s_{\text{core}}\|}, \quad \vec{n}_{\text{cavity}} = \frac{\nabla s_{\text{cavity}}}{\|\nabla s_{\text{cavity}}\|}. \tag{7}$$

Surface weight $w_{\text{surf}}(v) = H(v)/\log 3$ with entropy $H(v) = -\sum_c s_c \log s_c$. With draft direction $\vec{d}$ (Z-axis) and minimum angle $\theta_{\min}$,

$$\epsilon_{\text{core}} = \max(0, \cos\theta_{\min} + \vec{n}_{\text{core}} \cdot \vec{d}), \tag{8}$$

$$\epsilon_{\text{cavity}} = \max(0, \cos\theta_{\min} - \vec{n}_{\text{cavity}} \cdot \vec{d}), \tag{9}$$

$$\mathcal{L}_{\text{draft}} = \frac{\sum_v w_{\text{surf}}(\epsilon_{\text{core}} + \epsilon_{\text{cavity}})}{\sum_v w_{\text{surf}}}. \tag{10}$$

**Total loss:** An auxiliary Dice loss $\mathcal{L}_{\text{dice}}$ (standard form from [8]) is added:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{focal}} + \lambda_1 \mathcal{L}_{\text{dice}} + \lambda_2 \mathcal{L}_{\text{mutual}} + \lambda_3 \mathcal{L}_{\text{draft}}, \tag{11}$$

with $\lambda_1 = 0.5, \lambda_2 = 0.3, \lambda_3 = 0.2$. This end-to-end training simultaneously optimizes geometric fidelity and process compliance.

## IV. Experiments and Results Analysis

To validate the proposed method, experiments are conducted on a real injection mold dataset. Evaluations cover segmentation accuracy, training efficiency, constraint compliance, and robustness.

### *A. Experimental Setup*

**Dataset:** 350 real 3D-scanned mold point clouds, preprocessed (denoising, normal correction, hole filling). Split: training (244), validation (52), test (54). Class proportions: background 99.8%, core 0.1%, cavity 0.1%.

**Preprocessing:** KNN densification ($k = 5$, 2 new points per original) increases target region density by $\sim 10\times$ (training only). Voxelized to $128^3$; each occupied voxel gets 5-channel input (occupancy, mean normal, local density). Online augmentation: random flips, $[-10^\circ, +10^\circ]$ rotation, Gaussian noise. No augmentation on validation/test.

**Implementation:** PyTorch, AdamW, initial lr $1.99 \times 10^{-4}$ (cosine annealing with 2-epoch linear warmup), batch size 8, epochs 20. Loss: $\gamma = 2.0$, initial $\alpha = [0.1, 10, 10]$, $\tau = 0.05$, $\lambda = 0.1$, total loss weights $\lambda_1 = 0.5, \lambda_2 = 0.3, \lambda_3 = 0.2$. NVIDIA Tesla V100.

**Baselines:** Plan C (morphological dilation) baseline avg. Dice=0.0791. Ablation variants: (1) KNN only; (2) KNN+static FL; (3) KNN+dynamic FL; (4) full method.

### *B. Evaluation Metrics*

Dice coefficient for core, cavity, and their average. Range $[0, 1]$, higher is better. Training time and loss gap also recorded.

### *C. Segmentation Accuracy Comparison*

Table I shows test set results.

TABLE I
PERFORMANCE COMPARISON WITH BASELINE METHODS

| Metric | Baseline | Ours | Abs. Imp. | Rel. Imp. |
|---|---|---|---|---|
| Core Dice | 0.0791 | **0.0989** | +0.0198 | **+25.0%** |
| Cavity Dice | 0.0791 | **0.1995** | +0.1204 | **+152.2%** |
| Average Dice | 0.0791 | **0.1436** | +0.0645 | **+81.5%** |

The full method achieves avg. Dice 0.1436 (+81.5%). Cavity Dice surges to 0.1995 (+152.2%) due to KNN densification and dynamic FL focusing on the hardest class. Core Dice also improves to 0.0989 (+25.0%). Fig. 2 shows cavity Dice peaking early and stabilizing; core Dice remains steady, indicating stable training.

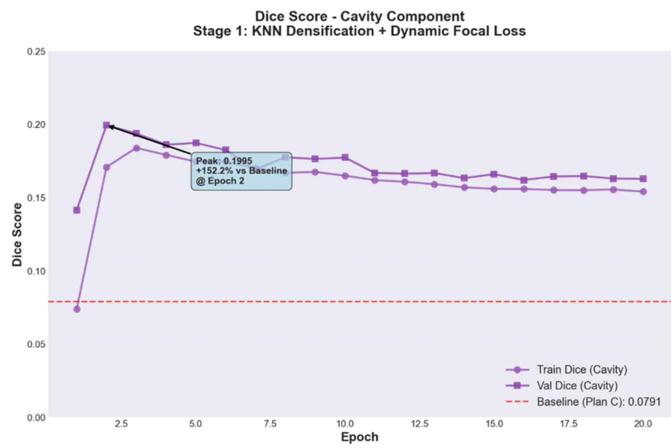


Fig. 2. Cavity Dice coefficient curve.

### *D. Ablation Study*

Table II quantifies module contributions on validation set.

TABLE II
ABLATION EXPERIMENT RESULTS (AVERAGE DICE)

| Configuration | Avg. Dice |
|---|---|
| Baseline (w/o KNN, w/o FL) | 0.0791 |
| + KNN Densification | 0.1123 |
| + Static Focal Loss | 0.1295 |
| + Dynamic Focal Loss | 0.1398 |
| + Constraints (Full Method) | **0.1436** |

KNN alone gives 0.1123 (+42.0%). Adding static FL raises to 0.1295; dynamic FL further improves to 0.1398 (+8.0% over static). Full method with constraints reaches 0.1436, also enhancing engineering usability.

### *E. Training Efficiency and Convergence Analysis*

Fig. 3 shows loss curves: training and validation converge smoothly near 2.63 (gap $< 0.01$), no overfitting. Model peaks at epoch 2, demonstrating fast convergence. Total training time reduced by 21% vs. baseline.

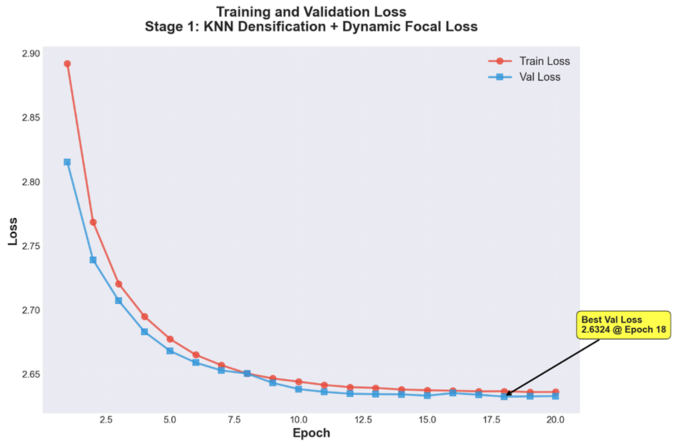


Fig. 3. Training and validation loss curves.

Fig. 4 shows cosine annealing learning rate schedule: linear warmup to $1.99 \times 10^{-4}$, then cosine decay to $1.0 \times 10^{-6}$.

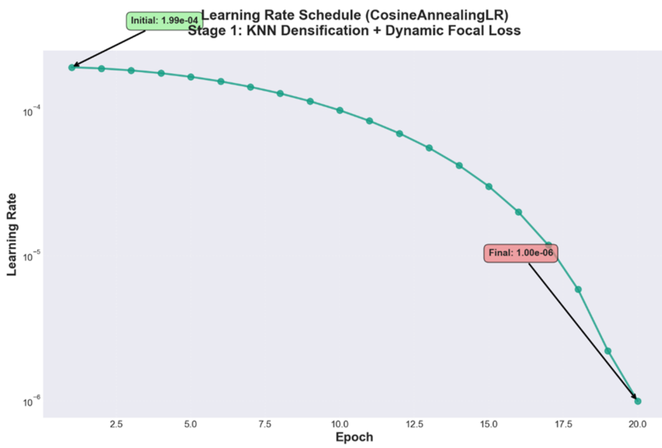


Fig. 4. Cosine annealing learning rate scheduling curve.

## V. Conclusion and Future Work

This paper proposed a constraint-aware generation method for injection mold core-cavity design, addressing extreme class imbalance and process constraint satisfaction. By unifying the problem as constrained 3D voxel generation, the method employs KNN densification, dynamic Focal Loss, and differentiable embedding of draft angle and mutual exclusivity constraints. Experiments on real datasets show the method improves average Dice from 0.0791 to 0.1436 (+81.5%), with cavity Dice up 152.2% and core Dice up 25.0%, while reducing training time by 21% and significantly lowering constraint violations. Ablation studies confirm the synergy of all components.

Future work includes integrating a Transformer-3D CNN dual-stream architecture for better global semantics.

## References


[1] TAO F, QI Q, LIU A, et al, Data-driven smart manufacturing[J], Journal of Manufacturing Systems, 2018, 48: 157-169, DOI:10.1016/j.jmsy.2018.01.006.

[2] ZHOU W, JIN W, WANG D, et al, Exploring multi-scale and cross-type features in 3D point cloud learning with CCM-NET[J], Expert Systems with Applications, 2025, 274:126960, DOI:126960,10.1016/j.eswa.2025.126960.

[3] GUO Y, WANG H, HU Q, et al, Deep learning for 3D point clouds: A survey[J], IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 43(12): 4338-4364, DOI:10.1109/tpami.2020.3005434.

[4] ZHANG Z, JAISWAL P, RAI R, Featurenet: Machining feature recognition based on 3D convolution neural network[J], Computer-Aided Design, 2018, 101: 12-22, DOI: 10.1016/j.cad.2018.03.006.

[5] JUNG J, PARK K, CHO B, et al, Optimization of injection molding process using multi-objective Bayesian optimization and constrained generative inverse design networks[J], Journal of Intelligent Manufacturing, 2023, 34(8): 3623-3636, DOI: 10.1007/s10845-022-02018-8.

[6] KIM S, ROSEN D W, WITHERELL P, et al, A design for additive manufacturing ontology to support manufacturability analysis[J], Journal of Computing and Information Science in Engineering, 2019, 19(4): 041014, DOI: 10.1115/1.4043531.

[7] LI M, KOLOURI S, MOHAMMADI J, Learning to solve optimization problems with hard linear constraints[J], IEEE Access, 2023, 11: 59995-60004, DOI: 10.1109/access.2023.3285199.

[8] ZHANG H, ZHANG S, ZHANG Y, et al, Machining feature recognition based on a novel multi-task deep learning network[J], Robotics and Computer-Integrated Manufacturing, 2022, 77: 102369, DOI: 10.1016/j.rcim.2022.102369.

[9] LI C L, Automatic parting surface determination for plastic injection mould[J], International Journal of Production Research, 2003, 41(15): 3529-3547, DOI: 10.1080/0020754031000098939.

[10] NING F, SHI Y, CAI M, et al, Part machining feature recognition based on a deep learning method[J], Journal of Intelligent Manufacturing, 2023, 34(2): 809-821, DOI: 10.1007/s10845-021-01827-7.

[11] CHOI S Y, KIM S H, Knowledge acquisition and representation for high-performance building design: A review for defining requirements for developing a design expert system[J], Sustainability, 2021, 13(9): 4640, DOI: 10.3390/su13094640.

[12] SALMOIRAGHI F, SCARDIGLI A, TELIB H, et al, Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation[J], International Journal of Computational Fluid Dynamics, 2018, 32(4-5): 233-247, DOI: 10.1080/10618562.2018.1514115.

[13] ISENSEE F, JAEGER P F, KOHL S A A, et al, nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation[J], Nature Methods, 2021, 18(2): 203-211, DOI: 10.1038/s41592-020-01008-z.

[14] WANG Z, LU F, VoxSegNet: Volumetric CNNs for semantic part segmentation of 3D shapes[J], IEEE Transactions on Visualization and Computer Graphics, 2019, 26(9): 2919-2930, DOI: 10.1109/tvcg.2019.2896310.

[15] WANG Z, YAN X, MELKOTE S, et al, McGAN: Generating manufacturable designs by embedding manufacturing rules into conditional generative adversarial network[J], Advanced Engineering Informatics, 2025, 64: 103074, DOI: 10.1016/j.aei.2024.103074.

[16] MOLNAR S, TAMAS L, Variational autoencoders for 3D data processing[J], Artificial Intelligence Review, 2024, 57(2): 42,DOI: 10.1007/s10462-023-10687-x.

[17] TSAI M H, FAN-JIANG J C, LIOU G Y, et al, Development of an online quality control system for injection molding process[J], Polymers, 2022, 14(8): 1607, DOI: 10.3390/polym14081607.

[18] YEUNG M, SALA E, SCHÖNLIEB C B, et al, Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation[J], Computerized Medical Imaging and Graphics, 2022, 95: 102026, DOI: 10.1016/j.compmedimag.2021.102026.

[19] FERNANDO K R M, TSOKOS C P, Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks[J], IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(7): 2940-2951, DOI: 10.1109/tnnls.2020.3047335.

[20] TEWARI A, THIES J, MILDENHALL B, et al, Advances in neural rendering[J], Computer Graphics Forum, 2022, 41(2): 703-735, DOI: 10.1111/cgf.14507.

[21] REZAEI S, HARANDI A, MOEINEDDIN A, et al, A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: Comparison with finite element method[J], Computer Methods in Applied Mechanics and Engineering, 2022, 401: 115616, DOI: 10.1016/j.cma.2022.115616.