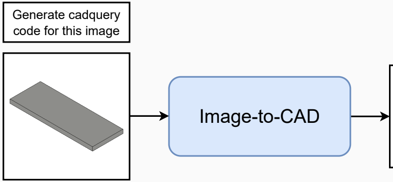
Bootstrapping Image-to-CAD Program Synthesis via Geometric Feedback

GIFT | ICML 2026

Giorgio Giannone, Anna Clare Doris, Amin Heyrani Nobari,
Kai Xu, Akash Srivastava, Faez Ahmed

DeCoDE Lab, MIT & AI Innovation Team, Red Hat & Core AI, IBM

Image-to-CAD Program Synthesis




```python
import cadquery as cq
# Generating a workplane for sketch 0
wp_sketch0 = cq.Workplane(cq.Plane(cq.Vector(-0.75, 0.296, 0.0), cq.Vector(1.0, 0.0, 0.0), cq.Vector(0.0, 0.0, 1.0)))
loop0=wp_sketch0.moveTo(0.671875, 0.0).lineTo(0.671, 0.247).lineTo(0.0, 0.247).lineTo(0.0, 0.0).close()
solid0=wp_sketch0.add(loop0).extrude(0.0312)
solid=solid0
```

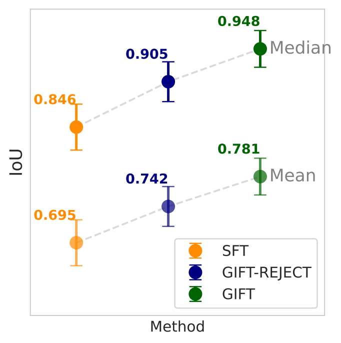
Goal: Convert a single image into an executable CAD program (CadQuery/Python).

Why programs? Unlike meshes or B-Reps, CAD programs provide a **compact**, **editable** representation naturally suited to **autoregressive LMs**. They are also inherently **parametric**, allowing engineers to easily modify dimensions.

The Bottleneck

Not model capacity, but **training data**: limited diversity of image-code pairs causes weak modality alignment, especially on complex geometries.

The Data Bottleneck: Why SFT Is Not Enough



SFT degrades on complex designs; GIFT maintains resilience.

Performance vs. task complexity (token length).

Key Observation

SFT uses one-to-one image-code pairs → narrow training signal → **brittle on complex geometries** → poor modality alignment

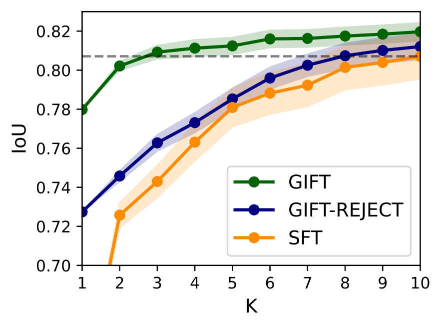
The Amortization Gap

Inference-Time Scaling (ITS) reveals a large gap between single-shot and oracle performance:

- Top-10% mean IoU: **0.734**
- Top-1 (best) IoU: **0.839**
- SFT pass@1 amortization gap: **15.5%**

Insight

The model *can* produce correct solutions, but standard decoding discards them. We need to **amortize** this into the weights.



GIFT matches SFT's peak with **80% less compute**.

Our Approach: Geometric Inference Feedback Tuning (GIFT)

Core Idea

Use offline geometric verification to convert **inference-time samples into augmented training data**, then retrain with standard SFT.

GIFT integrates two complementary augmentation mechanisms:

1. Soft Rejection Sampling (SRS)

Output augmentation: retains diverse valid programs ($0.9 \leq \text{IoU} < 0.99$) beyond exact ground-truth matches.

2. Failure-Driven Augmentation (FDA)

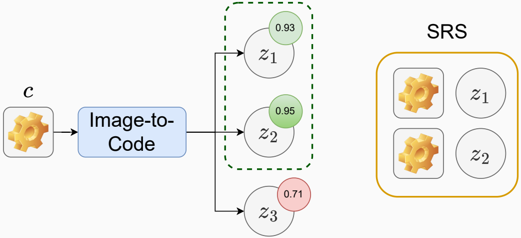
Input augmentation: renders near-miss programs ($0.5 \leq \text{IoU} < 0.9$) as synthetic images, paired with ground-truth code.

Soft Rejection Sampling (SRS): Output Augmentation

Standard rejection sampling keeps only exact matches. SRS retains **diverse valid alternatives**:

$$w_{\text{srs}}(\mathbf{z}) = \mathbb{1}\left[\tau_{\text{valid}} \leq f(\mathbf{z}) < \tau_{\text{match}}\right]$$

- $\tau_{\text{valid}} = 0.9$, $\tau_{\text{match}} = 0.99$
- Multiple valid programs per image
- Prevents collapse to single syntactic pattern
- Expands target distribution



SRS: sample, verify with CAD kernel, retain diverse valid programs.

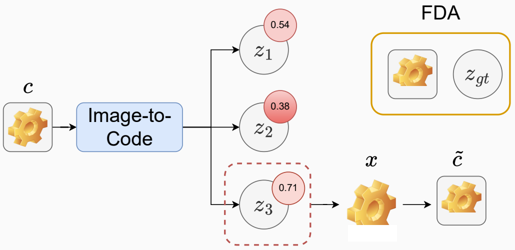
Failure-Driven Augmentation (FDA): Input Augmentation

Near-miss programs encode **structured errors**.
FDA uses them as training signal:

$$w_{\text{fda}}(\mathbf{z}) = \mathbb{1}\left[\tau_{\text{low}} \leq f(\mathbf{z}) < \tau_{\text{valid}}\right]$$

$$\tilde{\mathbf{c}} \leftarrow \phi(d(\mathbf{z}))$$

- Render failed program $\rightarrow$ synthetic image
- Pair with **original ground-truth code**
- Geometric denoising objective
- Targets hard cases the model struggles with



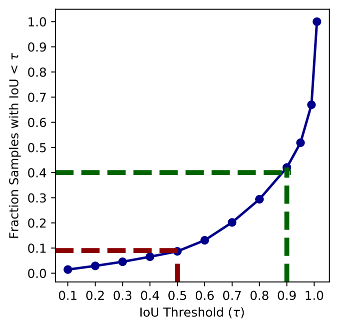
FDA: render near-miss $\rightarrow$ synthetic image, train to recover GT code.

The GIFT Objective

Combined Training Loss

$$\mathcal{F}_{\text{GIFT}}(\theta) = \underbrace{\mathbb{E}_{(\mathbf{c},\mathbf{z})\sim\mathcal{D}_{\text{SFT}}}[\log p_\theta(\mathbf{z}|\mathbf{c})]}_{\text{Base SFT}} + \underbrace{\mathbb{E}_{(\mathbf{c},\mathbf{z})\sim\mathcal{D}_{\text{SRS}}}[\log p_\theta(\mathbf{z}|\mathbf{c})]}_{\text{Output Diversity (SRS)}} + \underbrace{\mathbb{E}_{(\mathbf{c},\mathbf{z})\sim\mathcal{D}_{\text{FDA}}}[\log p_\theta(\mathbf{z}|\mathbf{c})]}_{\text{Input Robustness (FDA)}}$$

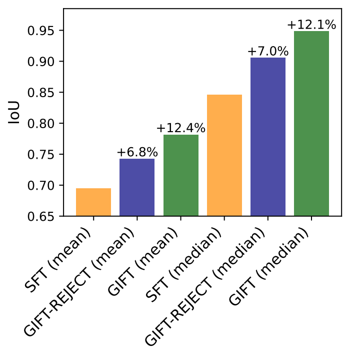
- **Offline**: no online RL needed
- SRS and FDA operate on **disjoint** IoU intervals
- Training set: 163k $\rightarrow$ **370k** samples
- Converges in 3–4 epochs



Results: IoU Across Compute Budgets

Method	1	2	4	6	8	10
SFT	.698	.725	.763	.788	.801	.807
GIFT-REJ	.732	.745	.773	.796	.807	.812
GIFT-FAIL	.761	.780	.791	.802	.803	.806
GIFT	**.779**	**.802**	**.811**	**.816**	**.817**	**.819**
Δ	+11.6	+10.5	+6.3	+3.5	+2.0	+1.6

Mean IoU on GenCAD test set. Δ is relative improvement over SFT (%).

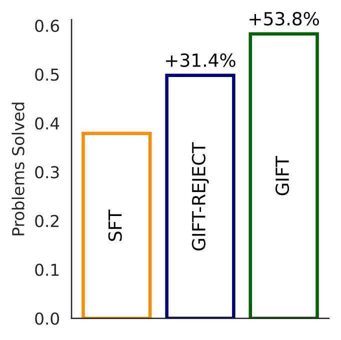


GIFT strategies: SRS adds diversity, FDA adds robustness.

Closing the Amortization Gap

	pass@1	pass@5	pass@10	Gap
SFT	.698	.776	.807	15.5%
GIFT-REJ	.732	.788	.812	10.9%
GIFT-FAIL	.761	.792	.806	5.9%
GIFT	**.777**	**.812**	**.820**	**5.2%**

Amortization gap reduced by **66.4%** vs. SFT baseline.



GIFT solves **53% more problems** than the SFT baseline.

Key Finding

GIFT shifts probability mass toward high-quality outputs, improving single-shot performance and reducing reliance on expensive test-time sampling.

Summary

Contributions

1. `GIFT`: verifier-guided data augmentation for Image-to-CAD program synthesis
2. `SRS`: diverse valid programs broaden output distribution
3. `FDA`: near-miss failures rendered as synthetic inputs improve robustness
4. **Results**: +12% IoU, 80% compute reduction, 53% more problems solved

Key Takeaway

The bottleneck in Image-to-CAD is **data, not models**. Geometric feedback turns inference-time compute into high-quality training data, amortizing test-time search into the weights.

Thank you!

ggiorgio@mit.edu