

# Flex3D: Feed-Forward 3D Generation with Flexible Reconstruction Model and Input View Curation

Junlin Han, Jianyuan Wang, Andrea Vedaldi, Philip Torr, Filippos Kokkinos Meta Al and University of Oxford



## TL;DR

A two-stage (1: multi-view generation + 2: feed-forward reconstruction) 3D generation framework that curates the optimal views from a large pool of 2D views before feeding them to a flexible reconstruction model designed to handle an arbitrary number of inputs.

#### Motivation

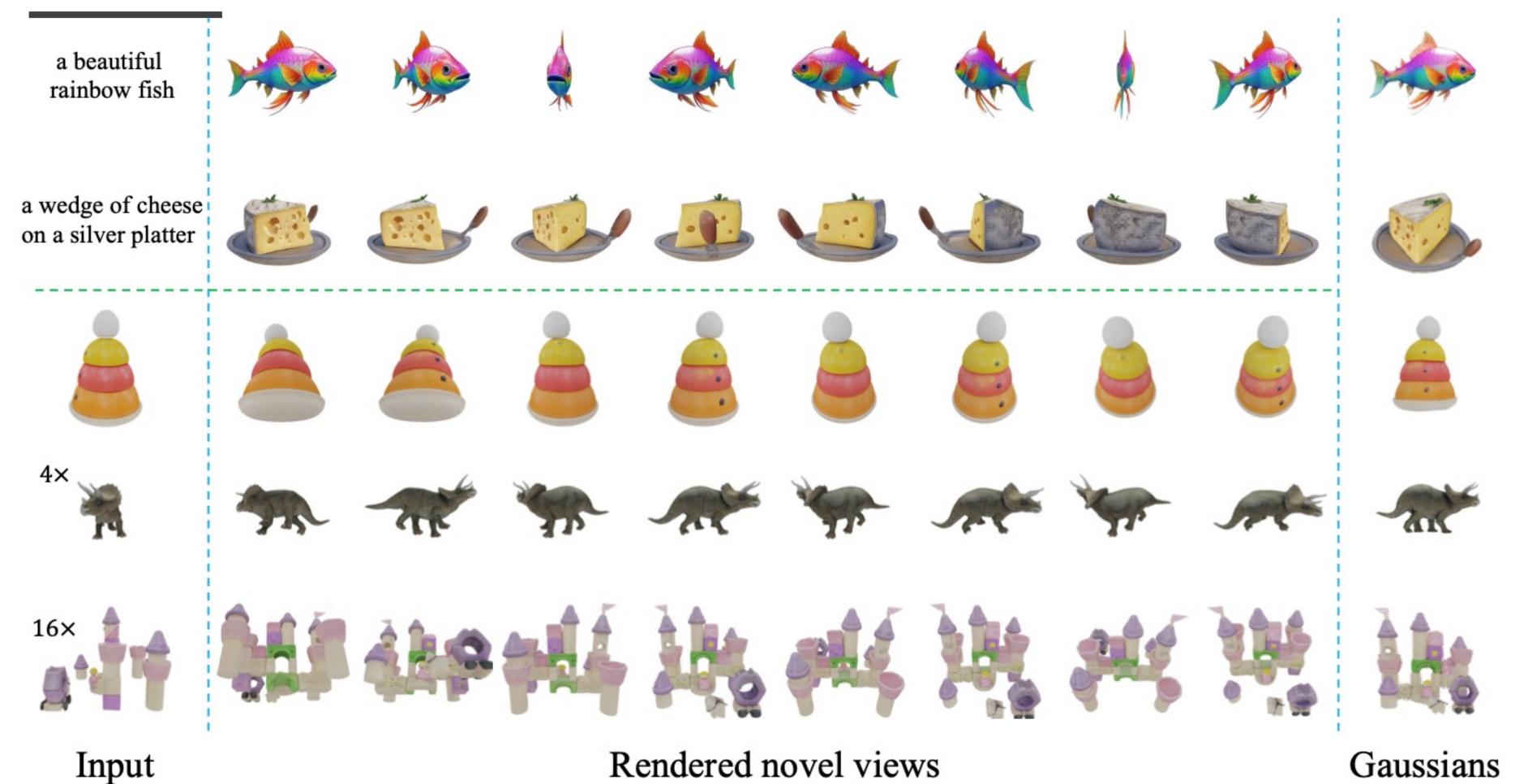
Existing widely used two-stage feed-forward 3D generation methods face common bottlenecks:

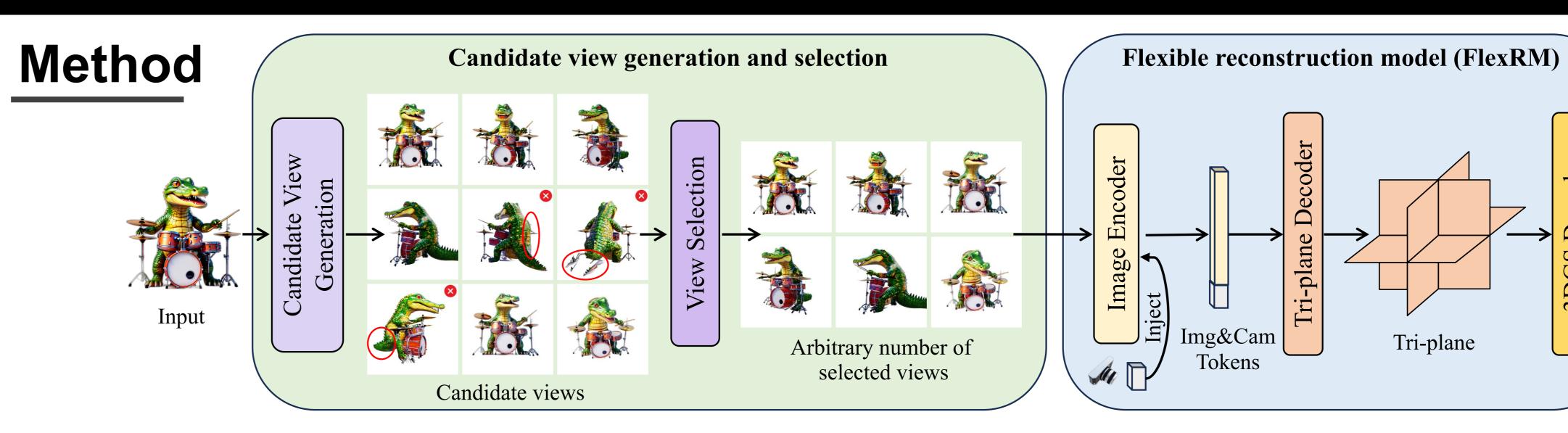
Stage (1): The multi-view generation stage often produces inconsistent and low-quality views.

Stage (2): The reconstruction model typically requires a small, fixed number of input views.

Together, these limitations cap the quality of the final 3D output.

## Generation results





We first generate a large pool of candidate views with two fine-tuned diffusion models, one for azimuth (16 views) and one for elevation (4 views), then select best views using a quality classifier and a feature matcher for geometric consistency.

# Comparison results



FlexRM is a fast, transformer-based model that generates 3D Gaussians from an arbitrary number of views. It can:

- (1): Handles a variable number of views from arbitrary poses, strengthened by dedicated camera embeddings.
- (2): Directly predicts 3D Gaussians from a tri-plane, using a residual scheme for high geometric accuracy.
- (3): It learns to handle imperfect inputs by being exposed to noise injected into its own generated 3D Gaussians.

### Reconstruction metrics

Method	Input views	PSNR↑	SSIM↑	LPIPS↓	CLIP image sim↑	CD↓	NC↑
OpenLRM VFusion3D	1	15.83 19.10	0.821 0.827	0.209 0.158	0.602 0.759	-	-
FlexRM	1	21.21	0.862	0.125	0.739	-	-
InstantMesh GRM FlexRM	4 4 4	21.33 25.03 <b>25.55</b>	0.859 <b>0.899</b> 0.894	0.133 0.102 <b>0.074</b>	0.809 0.869 <b>0.893</b>	1.496	0.841 0.866 <b>0.878</b>
FlexRM FlexRM FlexRM FlexRM	8 16 24 32	26.33 26.51 26.65 26.77	0.897 0.902 0.905 0.907	0.069 0.068 0.067 0.066	0.906 0.911 0.915 0.919	1.182 1.175	$\begin{array}{c} 0.881 \\ 0.884 \\ 0.886 \\ 0.888 \end{array}$