# DESIGNORBIT: EXPLORING THE UNSEEN SIDES OF A DESIGN IN CONTEXT FROM A SINGLE IMAGE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Early concepts for buildings, pavilions, and public installations often exist as a single sketch or image, yet judging them requires seeing their unseen sides and how they sit in their surroundings. We study generative design in context: from one image and a text description, we sample alternative designs as loop-closed $360^\circ$ orbit videos, each proposing plausible unseen sides while keeping the design and its surroundings consistent across the full orbit. We introduce DesignOrbit, which learns this behavior without any real architectural orbit videos. Procedural cuboid buildings rendered along known camera paths, combined with loop-closed background inpainting, provide initial supervision. A structure-from-motion verifier then selects the model's own generations for self-training based on frame registration and camera coverage, and later supplies preference pairs for direct preference optimization, extending training beyond the procedural shapes while increasing camera coverage. On architecture, camera coverage rises from $28.77^\circ$ to $342.13^\circ$, against $303.55^\circ$ for the strongest baseline, while diversity in unseen rear views is retained. Although trained only on synthetic building videos, the model also generates orbits for sketches, playgrounds, ships, interiors, and urban scenes without further fine-tuning. In a study with 50 participants, 80% with a background in architecture, design, or 3D visualization and over half with five or more years of experience, DesignOrbit was preferred over the baselines in 91% of ratings that expressed a preference.

## 1 INTRODUCTION

When designing buildings, pavilions and public art installations, early designs often exist as concept sketches or single images. Designers use such sketches or images to communicate ideas with their clients with the goal of rapid iteration. However, a single sketch or image is often insufficient to evaluate a design because it is a single view. The same front view may admit many possible rear designs. Sketching each of these alternatives along with the surroundings is too time-consuming and expensive, especially given that the design proposal is continuously being iterated upon.

Given advances in video generation, we ask: can video generative models help with this stage of the design process? In particular, we study *generative design in context*: given a single image and a text description of the design and its setting, we want a video depicting a $360^\circ$ orbit around the design (Fig. 1). If we can do this, then designers and their clients can rapidly explore design ideas from all sides *before* investing in detailed 3D modeling, thus enabling both informative communication and rapid iteration of the designs.

Unfortunately, we find that existing video generative models are not up to the task: they often produce incomplete orbits or inconsistent geometry as the camera moves. This is true even for models that offer additional motion or geometric guidance (Sun et al., 2024; Zhou et al., 2025; Yu et al., 2024; Ren et al., 2025). An alternative to video generation is 3D generative models (Voleti et al., 2024; Yang et al., 2024; Wang et al., 2026), but these often produce isolated foreground objects without the surrounding context, which is crucial information for interpreting and evaluating the design.

Why do video generative models struggle with this task? We posit that the key challenge is the lack of training data. Videos of $360^\circ$ orbits around large structures like buildings or art installations are



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

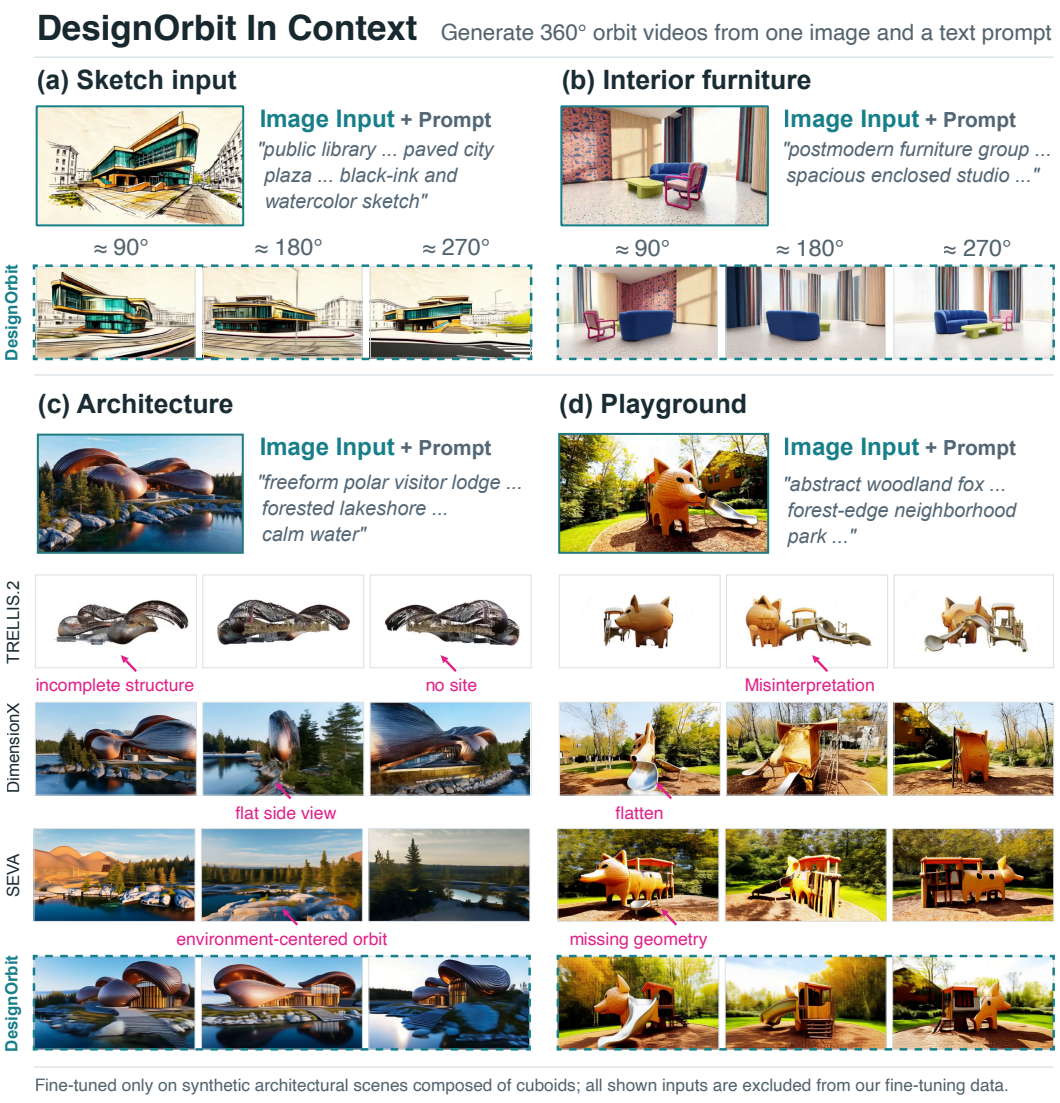


Figure 1: **DesignOrbit turns a single image and a text prompt into a full** $360°$ **orbit video of a design in its context.** Top half shows four prompts and the output of DesignOrbit: each strip shows frames at $\approx 90°$, $180°$, and $270°$ from one generated video. DesignOrbit produces complete orbits around high quality designs with consistent geometry and style. In comparison, bottom half shows three baselines that yield incomplete surroundings (TRELLIS.2), flat and unsatisfactory designs (DimensionX) or orbits that do not capture the geometry (Stable Virtual Camera).

either impossible to capture in real life owing to scene constraints, or require specialized equipment (like drones). As such, they are underrepresented on the web. Existing datasets of renderable 3D assets are biased towards objects (Deitke et al., 2022; 2023; Reizenstein et al., 2021; Liu et al., 2025) and particular landmarks (Li & Snavely, 2018; Tung et al., 2024), and as such are not general enough.

In this paper, we propose a new approach for training video generative models to perform complete orbits around large scene structures from single images. Our key insight is two-fold. First, synthetic data, in particular, $360°$ orbits around procedurally generated 3D artifacts on a generated canvas, are a cheap way to generate training data. Although this data is biased due to its synthetic nature, it suffices to train a good initialization. Second, even without reference videos, we can evaluate generated candidates for their geometric consistency and the completion of the orbits. This can be used to iteratively refine the model, encouraging good video orbits and discouraging bad ones.

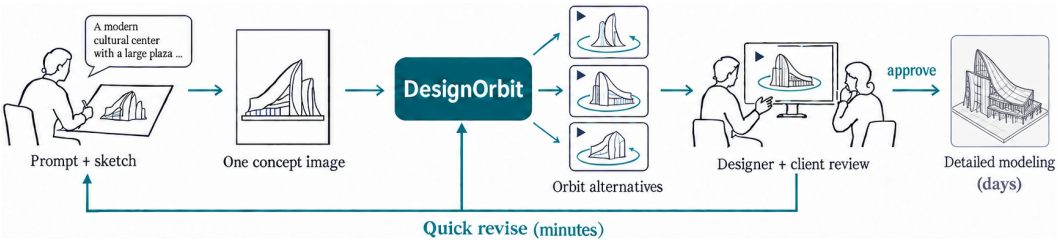


Figure 2: **DesignOrbit in the early design workflow.** A designer's prompt and sketch yield a single concept image, from which DesignOrbit generates several orbit alternatives within minutes. Designer and client review the orbits together, revise the prompt or sketch and regenerate as often as needed, and commit to detailed 3D modeling, which takes days, only once a design is approved.

We instantiate these insights in the context of architectural design, and test transfer to other kinds of designs. We first build a synthetic dataset by rendering procedural cuboid buildings along known orbits (Tsai & Hariharan, 2025) and add photorealistic surroundings through a novel split-and-condition video inpainting approach designed to preserve loop closure and background parallax. We train an initial model using this synthetic dataset. Then we generate videos from this initial model and use structure-from-motion to identify good generations (more complete orbits, consistent geometry) and bad ones. This allows us to self-train on good videos, and further improve the model using direct preference optimization (DPO) (Rafailov et al., 2023). Our final model, DesignOrbit, is able to produce more complete orbits around a diverse array of generated scenes.

We evaluate our approach and baselines on orbit completeness, consistency, visual quality, and diversity on architecture and other context-dependent designs. On architecture, camera coverage rises from $28.77°$ to $342.13°$, against $303.55°$ for the strongest baseline, while diversity in unseen rear views is retained. The model also generates orbits for sketches, playgrounds, ships, interiors, and urban scenes without further fine-tuning. In a study with 50 participants, 80% with a background in architecture, design, or 3D visualization and over half with five or more years of experience, DesignOrbit was preferred over the baselines in 91% of ratings that expressed a preference (Sec. 5.2).

In sum, our contributions are as follows:

- We formulate a new problem of *generative design in context*: from a single concept image and a text prompt, sample alternative loop-closed $360°$ orbit videos that propose plausible unseen sides of a design while keeping the design and its surroundings consistent across the full orbit.
- Focusing on architecture, we propose a new training approach that combines the complementary strengths of synthetic training data and self-training with an SfM-aided verifier in the loop.
- We demonstrate empirically that our model, DesignOrbit, is more preferred by users and produces more complete outputs with greater coherence and quality than prior art.

## 2 Related Work

**Generative tools for design exploration.** Text- and image-conditioned generators let designers explore the appearance of a concept (Rombach et al., 2022; Zhang et al., 2023), and architecture-specific methods generate building geometry (Tsai & Hariharan, 2025). Both operate on a single view or on the building alone. We study a complementary form of exploration: from one concept image, sampling alternative completions of the design and viewing each as a full orbit within its surroundings.

**3D assets and object-orbit synthesis.** TRELLIS.2 and Hunyuan3D-2 generate textured 3D assets (Xiang et al., 2025; Team, 2025), SV3D and Hi3D generate object-orbit views for subsequent reconstruction (Voleti et al., 2024; Yang et al., 2024), and Orbital conditions video generation on

latent features from a 3D foundation model (Wang et al., 2026). All of them treat the object in isolation, whereas contextual visualization additionally requires the surrounding scene, which motivates our joint modeling of a design and its setting.

**Scene and camera-controlled generation.** Multi-view diffusion models such as CAT3D synthesize novel views of a scene for subsequent reconstruction (Gao* et al., 2024). Camera-conditioned video models synthesize views along a specified trajectory: DimensionX and GenXD disentangle viewpoint changes from scene dynamics (Sun et al., 2024; Zhao et al., 2025), with DimensionX also supporting $360^\circ$ orbits; Stable Virtual Camera conditions directly on target cameras and supports looped trajectories (Zhou et al., 2025); and CameraCtrl and MotionCtrl inject camera parameters into pretrained video models (He et al., 2024; Wang et al., 2024). A second line guides generation with renders of an explicit 3D proxy: ViewCrafter, GEN3C, and TrajectoryCrafter render a point cloud along the target path (Yu et al., 2024; Ren et al., 2025; Yu et al., 2025), ReCamMaster redirects the camera of an input video (Bai et al., 2025), and OrbitForge completes missing orbit views with a frozen video prior and per-prompt Gaussian splatting (Fan & Favaro, 2026). From a single image, however, such proxies cover only the visible surfaces, and the unseen side must be completed generatively; ViewCrafter explicitly reports difficulty with large viewpoint changes under limited 3D cues (Yu et al., 2024). Specifying a full orbit therefore does not guarantee coherent geometry throughout the generated sequence. DesignOrbit targets this gap through complete-orbit supervision and geometric verification of self-generated training examples.

**Training data.** Objaverse and Objaverse-XL provide renderable assets (Deitke et al., 2022; 2023), CO3D and uCO3D provide real object-centric captures, with uCO3D targeting full $360^\circ$ coverage (Reizenstein et al., 2021; Liu et al., 2025), and MegaDepth and MegaScenes provide scene observations from Internet photo collections (Li & Snavely, 2018; Tung et al., 2024). Scene-video datasets such as RealEstate10K and DL3DV-10K supply posed camera trajectories, but they primarily capture walkthroughs rather than closed orbits around a single building with its site (Zhou et al., 2018; Ling et al., 2024). None of these offers loop-closed orbit videos of buildings together with their surroundings, let alone with useful design variety that exploration requires. We therefore construct architectural supervision from procedural buildings (Tsai & Hariharan, 2025) and loop-closed background inpainting that preserves parallax, jointly supervising the design and its surroundings.

**Geometry-guided model improvement.** VFusion3D trains a 3D generator on multi-view data produced by a video model (Han et al., 2024); we likewise learn from a video model's own generations, but filter them geometrically before training on them. Diffusion-DPO adapts preference optimization to diffusion models (Wallace et al., 2024), and Epipolar-DPO, VideoGPA, and VIGOR use geometric consistency signals to adapt video models (Kupyn et al., 2025; Du et al., 2026; Yin et al., 2026), but none of their objectives requires measured full-orbit coverage. DesignOrbit combines synthetic contextual orbit initialization with structure-from-motion (SfM) verification of frame registration and camera coverage, selecting generations for supervised self-training and ranking samples for preference optimization.

## 3 Problem Setup

Given a concept image $I_0$, a prompt $c$ describing the design and its surroundings, and random noise $\xi$, we seek a generator whose output $x = G(I_0, c, \xi)$ depicts a complete $360^\circ$ orbit of one coherent design within its setting. Different noise should propose plausible alternatives for unseen parts while preserving the design specified by the input. The target orbit must return to its starting viewpoint.

For the generations to be useful for design, we posit that they should satisfy the following properties:

1. The generated video should perform as much of a complete $360^\circ$ orbit as possible, so that the designer and client can visualize the whole structure,
2. The generated video should show not just the central structure but the surroundings as well so that users can visualize the structure in context,
3. The video should be consistent both geometrically and stylistically with the provided concept image,
4. The depicted structure should be a usable design, since that is the end goal, and

5. Multiple passes (with different noise) should yield a diverse set of designs that provide different design possibilities for the unseen parts of the scene.

Prior work does not satisfy one or more of these desiderata (see Sec.5). Current off-the-shelf video generative models often do not perform complete orbits or maintain geometric or stylistic consistency. 3D generative models do not visualize structures in context. Video generative models trained for camera controls often produce inconsistent geometry or unusable designs.

We posit that a video generative model can be trained to satisfy the requirements above, provided the right data and training paradigm. Our core contributions are in this training paradigm, which we describe next.

# 4 METHOD

We propose an approach to finetune existing video generative models to solve the generative design in context problem, satisfying the desiderata described above. Crucially, our adaptation does not use any captured orbit videos. Instead, it combines synthetic and model-generated videos. To do so, we propose a new approach to leverage procedural generation for producing synthetic training data (Section 4.1) and new methods for using Structure-from-motion to verify generations (Section 4.2) for self-training and preference optimization (Section 4.3).

Our overall pipeline is as follows (Fig. 3):

1. In **Stage I**, we use a procedural approach to produce a dataset of synthetic orbit videos (Section 4.1), and then filter them using our SfM-based verification (Section 4.2). On this dataset of 300 verified clips, we fine-tune Wan2.1-FLF2V-14B (Wan et al., 2025). We use the same image for both endpoint conditions to train the model to produce full orbits.
2. Our synthetic data is biased towards cuboidal buildings because of the limitations of the procedural approach. In **Stage II: self-training beyond cuboids**, we use the Stage I model to generate orbits of free-form building concepts and retain approximately 300 clips after SfM-based verification. We combine these with the synthetic training set and train a fresh LoRA from the original pretrained Wan checkpoint on approximately 600 clips. This expands the training shapes beyond procedural cuboids while retaining their orbit supervision.
3. Finally, **Stage III** applies Direct Preference Optimization (DPO) (Wallace et al., 2024) to push generations toward complete orbits: for each input we draw $K$ samples, rank them by SfM camera coverage, and form winner–loser pairs from the same input (Section 4.3).

In the subsections below, we delve deeper into the key technical contributions. Sampling budgets, acceptance rates, and hyperparameter settings are provided in the supplementary material.

## 4.1 SYNTHETIC ORBITS WITH CONTEXTUAL BACKGROUNDS

We bootstrap self-training with synthetic data because none of the approximately 300 videos sampled from the base pretrained video generation model passed our SfM verifier. To create this data, we first use an LLM to generate prompts describing buildings and their settings and then use these prompts to generate procedural cuboid buildings following (Tsai & Hariharan, 2025). We then render these procedural generations along known circular camera paths with identical start and end views.

The rendered clips are geometrically consistent across views but have no background. Training on them alone produces buildings that rotate against a static background (Sec. A.3). To make the background rotate with the building, we mask out everything except the building in every frame and inpaint the masked region with VACE (Jiang et al., 2025). The inpainting prompt is the LLM prompt originally used to generate the cuboid building, with the building description removed.

**Split-and-condition inpainting.** In our experiments, unconstrained single-pass inpainting fails to "close the loop": the beginning of the video is inconsistent with the end. The alternative of conditioning the whole sequence on identical endpoint images can produce a static background.

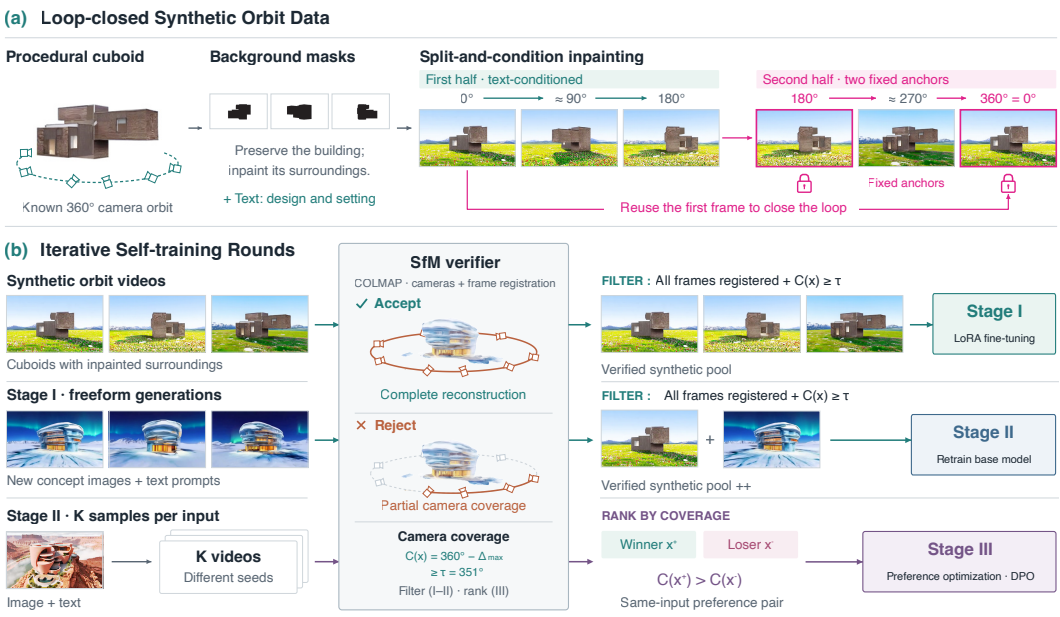


Figure 3: **Overview of DesignOrbit.** (a) We generate procedural cuboid buildings(Tsai & Hariharan, 2025), render them along a known 360° orbit and inpaint their surroundings to create initial training data (Sec. 4.1). (b) We use SfM-based verification (Sec. 4.2) to discard bad training data, be it synthetic data (Stage I) or model generations (Stage II). For the final stage, we use our verifier to produce preference data for DPO (Stage III, Sec. 4.3).

Instead, we split the orbit at $m = \lfloor (T-1)/2 \rfloor$ and first inpaint frames $0{:}m$, conditioned on the rendered foreground and setting prompt. We then inpaint frames $m{:}T-1$ using the completed midpoint and first frame as the starting and ending anchors, respectively, protecting both anchors and the foreground with masks. This retains the first half's camera-induced background motion while joining the second half to both anchors.

In spite of these measures, our approach might still introduce repeated textures or unwanted structures attached to the building. We therefore filter the completed clips with the verifier below, which discards clips that fail COLMAP SfM reconstruction and the camera-coverage check, and retain approximately 300 as $\mathcal{D}_{\text{syn}}$.

Qualitative examples appear in supplementary material.

### 4.2 SfM Verification of Consistency and Coverage

Our training pipeline relies on an structure-from-motion (SfM)-based verifier to both filter bad videos from the training set as well as to provide preference pairs for optimization. This verifier checks for both the consistency of geometry across the video as well as the completeness of the orbit. It works as follows.

For each candidate video $x$, we use COLMAP (Schönberger & Frahm, 2016) to estimate cameras and a sparse point cloud. We check that the number of registered frames, $N_{\text{reg}}(x)$, is equal to the length of the video $T = 81$. This supplies evidence of geometric consistency, but an incomplete orbit can also register successfully.

To measure coverage, we fit an orbital plane to the recovered camera centers by SVD, estimate the orbit center, and compute camera azimuths. We filter cameras for consistent travel direction and radius using the procedure, including $3\times$MAD outlier filtering (see details in supplementary). For the retained, azimuth-sorted cameras, let $\Delta_i$ denote consecutive angular gaps, including the gap between the last and the first. We define $C(x) = 360^\circ - \max_i \Delta_i$ to be the "length" of the orbit. Finally, we consider a video as verified if (a) the number of registered frames is equal to the length

**Algorithm 1** Coverage-ranked preference optimization (Stage III-DPO)

**Require:** Stage-II model $G_{\mathrm{II}}$; image–text inputs $\mathcal{P}$; seeds per image $K = 8$; margin $\delta = 30^\circ$

```
Define S(x) = C(x) if computable, and 0° otherwise
D_pref ← ∅
for p ∈ P do
    X_p ← {G_II(p; ξ_k)}_{k=1}^K
    x⁺ ← arg max_{x∈X_p} S(x)
    x⁻ ← arg min_{x∈X_p} S(x)
    if S(x⁺) − S(x⁻) ≥ δ then
        D_pref ← D_pref ∪ {(p, x⁺, x⁻)}
    end if
end for
Split D_pref by input into D_train and D_val
return DPO(G_II, D_train)
```

of the video, and (b) the length of the orbit is more than 351°, corresponding to at most a single missed frame.[1]

### 4.3 Coverage-ranked preference optimization

The binary filtering above discards many bad videos. But bad videos can serve as a useful *negative* examples. Therefore, in our final stage, we use Direct Preference Optimization to train the generative model to prefer generating good videos over bad ones.

Concretely, 90% of the videos generated by our Stage II model have all their frames registered during SfM, suggesting that they are geometrically consistent. However, many of them may not be complete orbits. We therefore rank videos by $C(x)$, which measures the viewing directions available for inspection

For each of 360 image–text inputs $p = (I_0, c)$, we sample $K = 8$ Stage-II videos. We set $S(x) = C(x)$ when coverage is computable and $S(x) = 0^\circ$ otherwise. Within each input, we pair the highest-scoring video $x^+$ with the lowest-scoring video $x^-$ and retain the pair if $S(x^+) - S(x^-) \geq 30^\circ$. Note that winners in each pair need not pass the binary gate $V$. This yields 350 pairs, one per retained input. We reserve approximately 15% of these inputs for validation and train on the remaining pairs (Alg. 1).

We adapt direct preference optimization (Rafailov et al., 2023) and Diffusion-DPO (Wallace et al., 2024) to the model's flow-matching parameterization (Lipman et al., 2023). For each pair $(p, x^+, x^-)$, we encode videos as latents $z^\pm$ and draw a shared Gaussian noise sample $\epsilon \sim \mathcal{N}(0, I)$ and noise fraction $t$, forming $z_t^\pm = (1-t)z^\pm + t\epsilon$ with velocity targets $u^\pm = \epsilon - z^\pm$. For each video, the policy and the frozen Stage-II reference model process the same noisy latents. We define

$$e_\theta^\pm = \mathrm{MSE}\big(v_\theta(z_t^\pm, t, p), u^\pm\big), \qquad d_\theta^\pm = e_\theta^\pm - e_{\mathrm{ref}}^\pm, \tag{1}$$

where MSE averages over latent elements and $e_{\mathrm{ref}}^\pm$ is the reference error. We minimize

$$\mathcal{L} = \mathbb{E}_{(p,x^+,x^-),t,\epsilon}\left[-\log\sigma\big(-\beta(d_\theta^+ - d_\theta^-)\big) + \lambda e_\theta^+\right], \tag{2}$$

where $\sigma$ is the logistic function. The preference term favors greater error reduction on winners relative to losers; the supervised term encourages low absolute error on winners. We merge the Stage-II LoRA into the base weights, freeze them, and train a fresh adapter initialized to produce zero weight updates. We sample $t$ at high noise levels; the exact sampling rule and training hyperparameters are in the supplementary material.

After training, we find that coverage rises from 318° to 342°, while keeping the rear-view both geometry and style coherent, with usable design alternatives (see Fig. 4 user study).

**Inference.** Given a sketch or concept image and its prompt, DesignOrbit directly samples orbit videos without requiring a pre-existing 3D model or additional views. We sample multiple candidate completions by varying the seed.

[1] For an ideal 81-frame orbit with a repeated endpoint, the spacing is $4.5^\circ$ and $C(x) = 355.5^\circ$.

Table 1: **User study.** We report the percentage of votes preferring **DesignOrbit** (bold) / the baseline; the remaining votes were "both/either", "neither", or "cannot judge". DesignOrbit wins every comparison. Single-video comparisons: 100 votes per baseline; video-set comparisons: 50 votes per baseline. TRELLIS renders are shown on a white background (white) or composited with an inpainted background (inpainted); the latter was not included in the set comparisons. See text for metrics.

| | % of votes preferring **DesignOrbit** / baseline | | | | | |
|---|---|---|---|---|---|---|
| | Single video | | | Video set | | |
| Baseline | Pref. | Shape consis. | Style consis. | Views | Unseen variety | Usable |
| DimensionX | **69** / 6 | **71** / 1 | **68** / 2 | **34** / 6 | **52** / 10 | **48** / 10 |
| Stable Virtual Camera | **66** / 7 | **70** / 0 | **55** / 5 | **54** / 4 | **58** / 16 | **68** / 4 |
| Gen3C | **74** / 5 | **80** / 3 | **67** / 9 | **80** / 2 | **62** / 8 | **76** / 2 |
| TRELLIS (white) | **78** / 7 | **28** / 12 | **39** / 11 | **38** / 2 | **38** / 18 | **46** / 12 |
| TRELLIS (inpainted) | **86** / 4 | **65** / 1 | **66** / 3 | – | – | – |
| All baselines | **75** / 6 | **63** / 3 | **59** / 6 | **51** / 4 | **52** / 13 | **60** / 7 |

# 5 EXPERIMENTS

## 5.1 SETUP

**Data.** We train on architectural scenes and evaluate on (a) *Architecture*, matching the training domain, and (b)*Other design-in-context*, which tests transfer beyond it: public art and installations, playgrounds, public and interior furniture, urban scenes, vehicles and vessels, and imagined settings such as spacecraft on the Moon or underwater subways. We additionally evaluate sketch inputs with users to test generalization across input representations.

**Inputs and protocol.** We use an LLM to generate a text prompt and a base text-to-video model to render a video. The first frame of the video is then the shared input image for DesignOrbit and every baseline. For Table 2, all outputs are converted to 15 fps, 81 frames and $320 \times 480$ (height$\times$width).

**Baselines.** We compare against what a designer could use today, in four families.

(i) *Camera-controlled video:* GEN3C (Ren et al., 2025), ViewCrafter (Yu et al., 2024) and TrajectoryCrafter (Yu et al., 2025) condition on a depth-warped point cloud, ReCamMaster (Bai et al., 2025) on source-video tokens, and Stable Virtual Camera (Zhou et al., 2025) on explicit camera poses without depth warping. (ii) *Reconstruction-oriented video* built to supply views for 3D reconstruction: DimensionX (Sun et al., 2024) and GenXD (Zhao et al., 2025). (iii) *Single-image 3D generators*, built for isolated objects: TRELLIS.2 (Xiang et al., 2025) and Hunyuan3D-2 (Team, 2025). These produce no environment, so we also compare to TRELLIS.2 with a background inpainted by our split-and-condition pipeline. (iv) *Base video models* without orbit training. Every baseline runs from its released code with its recommended settings.

## 5.2 USER STUDY

Our primary evaluation is a user study. We recruited 50 participants, 40 of whom have training or professional experience in design-related fields. Each participant was asked two kinds of questions: (1) *Single-view comparisons*, where participants were shown one video from two different methods and asked to judge which they would prefer to show the design in its surroundings (*Pref.*), which shows consistent geometry (*Shape consis.*), and consistent style(*Style consis.*)

(2) *Video-set comparisons*, where participants got a set of two videos from each method and judged which set revealed enough of the sides and back (*Views*), showed enough variety (*Unseen variety*) and gave two usable design alternatives (*Unseen variety*)

Results are shown in Table 1. We find that DesignOrbit was preferred in the vast majority of comparisons against all baselines, and across the majority of participants. Overall, 75% of the time, users preferred DesignOrbit over baselines for presenting the design in context. Further results and analysis are presented in the supplementary.

Table 2: Comparison of our final model with baselines and earlier training stages in terms of completeness of the orbit and consistency of the foreground geometry.

| | Complete orbits? | | | | Consistent |
|---|---|---|---|---|---|
| | Architecture | | Other design-in-context | | Geometry? |
| Method | Cov.$\geq 300^\circ \uparrow$ | Ave.$^\circ \uparrow$ | Cov.$\geq 300^\circ \uparrow$ | Ave.$^\circ \uparrow$ | Fg. pts $\uparrow$ |
| **DesignOrbit** | **79.80%** | **337.33**$^\circ$ | **74.60%** | **328.20**$^\circ$ | **1878** |
| DimensionX | 65.50% | 302.40$^\circ$ | 74.50% | 316.62$^\circ$ | 1792 |
| Stable Virtual Camera | 55.30% | 303.55$^\circ$ | 63.70% | 309.81$^\circ$ | 1450 |

### 5.3 Evaluating geometric consistency and complete orbits

To further understand why DesignOrbit is preferred over baselines, we next quantitatively evaluate two of our desiderata that can be measured objectively: whether the generated video performs a complete orbit around the structure, and whether the video is geometrically consistent enough to describe one scene.

To measure how complete the generated orbits are, we look at $C(x)$, defined earlier as the "angular length" of the orbit, or 360° minus the largest gap between the cameras. Concretely, we record both the fraction of generations for which $C(x) \geq 300^\circ$, and the average $C(x)$. To see whether the video is geometrically consistent, we look at the average number of foreground points in the Colmap reconstruction, with foreground determined by SAM2.

Results are shown in Table 2. Only two baselines, DimensionX and Stable Virtual Camera, produce complete orbits ($C(x) \geq 300^\circ$) more than half the time. For all other baselines, orbits are more often than not incomplete, with the base video model *never* producing a complete orbit (see supplementary for the full results).

We report results for only these two baselines in the main paper; see supplementary for others. Our final model, DesignOrbit, produces complete orbits more often than all baselines. This is true *even for design-in-context cases that are outside the training domain*. It also produces a consistent geometry as visible from the denser foreground reconstruction. This suggests that our approach is effective in teaching video generative models to produce complete orbits around consistent structures, even outside the training domain.

Table 3: Ablations of the various design choices

| Base model | Stage 2 | Stage 3 | Cov$\geq 300^\circ$ |
|---|---|---|---|
| Wan2.1-FLF2V-14B | SFT | DPO | 79.80% |
| Wan2.1-FLF2V-14B | SFT | SFT | 65.90% |
| Wan2.1-FLF2V-14B | SFT | - | 55.30% |
| Wan2.1-FLF2V-14B | - | - | 48.40% |
| HunyuanVideo | SFT | DPO | 65.40% |

### 5.4 Ablations

We ablate our various design choices in Table 6. Each stage of the training (rows 3 and 4 vs row 1) improve the completeness of the orbit. For the last stage, DPO yields larger gains (row 2). Finally, we show that our training approach can also work on other base models such as HunyuanVideo (row 5).

## 6 Conclusion

We explore the problem of generating design-in-context, aimed at visualizing designs early in the design workflow. We present a model that, given a single sketch of a 3D design and a text prompt, produces a complete 360° video around the scene with many alternative completions. We train this model using a novel training approach combining synthetic data and SfM-aided verification for self-training. Our results show that our model, DesignOrbit, produces videos preferred by designers that complete orbits and maintain geometric consistency.

## REFERENCES


Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. *arXiv preprint arXiv:2503.11647*, 2025.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023.

Hongyang Du, Junjie Ye, Xiaoyan Cong, Runhao Li, Jingcheng Ni, Aman Agarwal, Zeqi Zhou, Zekun Li, Randall Balestriero, and Yue Wang. VideoGPA: Distilling geometry priors for 3D-consistent video generation, 2026. URL https://arxiv.org/abs/2601.23286.

Chenrui Fan and Paolo Favaro. Orbitforge: Text-to-3d scene generation via reconstruction-anchored video synthesis, 2026. URL https://arxiv.org/abs/2606.24799.

Ruiqi Gao*, Aleksander Holynski*, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole*. Cat3d: Create anything in 3d with multi-view diffusion models. *Advances in Neural Information Processing Systems*, 2024.

Junlin Han, Filippos Kokkinos, and Philip Torr. Vfusion3d: Learning scalable 3d generative models from video diffusion models. *European Conference on Computer Vision (ECCV)*, 2024.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Orest Kupyn, Théo Uscidda, Marta Tintore Gazulla, Fabian Manhardt, Federico Tombari, and Christian Rupprecht. Epipolar geometry improves video generation models, 2025. URL https://arxiv.org/abs/2510.21615.

Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2041–2050, 2018.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169, June 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Xingchen Liu, Piyush Tayal, Jianyuan Wang, Jesus Zarzar, Tom Monnier, Konstantinos Tertikas, Jiali Duan, Antoine Toisoul, Jason Y. Zhang, Natalia Neverova, Andrea Vedaldi, Roman Shapovalov, and David Novotny. UnCommon Objects in 3D, 2025. URL https://arxiv.org/abs/2501.07574.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023. URL https://arxiv.org/abs/2305.18290.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10901–10911, 2021.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113, 2016.

Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion. *arXiv preprint arXiv:2411.04928*, 2024.

Tencent Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025.

I-Ting Tsai and Bharath Hariharan. 3D synthesis for architectural design. In *WACV*, pp. 4799–4809, 2025.

Joseph Tung, Gene Chou, Ruojin Cai, Guandao Yang, Kai Zhang, Gordon Wetzstein, Bharath Hariharan, and Noah Snavely. Megascenes: Scene-level view synthesis at scale. In *ECCV*, 2024.

Vikram Voleti et al. SV3D: Novel multi-view synthesis and 3D generation from a single image using latent video diffusion. In *European Conference on Computer Vision (ECCV)*, 2024.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2311.12908.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Rong Wang, Ruyi Zha, Ziang Cheng, Jiayu Yang, Pulak Purkait, and Hongdong Li. Towards realistic and consistent orbital video generation via 3D foundation priors. *arXiv preprint arXiv:2604.12309*, 2026. URL https://arxiv.org/abs/2604.12309.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, 2024. URL https://arxiv.org/abs/2312.03641.

Jianfeng Xiang et al. Native and compact structured latents for 3D generation. *arXiv preprint arXiv:2512.14692*, 2025.

Haibo Yang, Yang Chen, Yingwei Pan, Ting Yao, Zhineng Chen, Chong-Wah Ngo, and Tao Mei. Hi3d: Pursuing high-resolution image-to-3d generation with video diffusion models. In *ACM MM*, 2024.

Tengjiao Yin, Jinglei Shi, Heng Guo, and Xi Wang. VIGOR: VIdeo Geometry-Oriented Reward for temporal generative alignment, 2026. URL https://arxiv.org/abs/2603.16271.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 100–111, October 2025.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

Yuyang Zhao, Chung-Ching Lin, Kevin Lin, Zhiwen Yan, Linjie Li, Zhengyuan Yang, Jianfeng Wang, Gim Hee Lee, and Lijuan Wang. Genxd: Generating any 3d and 4d scenes. In *ICLR*, 2025.

Jensen (Jinghao) Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025. URL https://arxiv.org/abs/2503.14489.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.