# LAYOUT2VIDEO: PERSISTENT 3D LAYOUT CONTROL FOR INDOOR VIDEO GENERATION

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

## ABSTRACT

Text- and image-conditioned video diffusion models produce visually compelling videos, but their inputs do not explicitly specify scene-wide 3D object arrangements. Maintaining these arrangements as the camera reveals new regions therefore remains challenging. We present Layout2Video, a framework for generating indoor videos from 3D semantic layouts and text prompts without task-specific training. The framework first renders view-specific semantic and geometric conditions from the layout. A pretrained multi-view diffusion model uses these conditions and an initial reference image to generate candidate views, from which sparse anchors are selected. A video diffusion model then synthesizes segments between consecutive anchors, and the segments are concatenated into a single video. All generative components remain frozen. We also construct a benchmark containing 555 room layouts and introduce an instance-level evaluation protocol for object presence, layout adherence, and appearance consistency. Extensive experiments demonstrate that our method effectively preserves the specified object arrangement and appearance throughout generated videos, particularly as previously unseen regions become visible.

## 1 INTRODUCTION

High-quality videos of indoor scenes support a wide range of applications, including virtual reality, game development, interior design, and embodied AI. A conventional production pipeline first creates or reconstructs a renderable 3D scene and then renders it from a sequence of viewpoints (Mildenhall et al., 2021; Kerbl et al., 2023). Building such a scene requires detailed geometry and appearance, often involving substantial modeling or reconstruction effort. More recently, 3D generation approaches offer a more efficient solution for scene creation from text prompts or reference images (Höllein et al., 2023; Chung et al., 2023; Li et al., 2024a). Alternatively, recent advances in video diffusion offer a more direct route by synthesizing videos without first constructing a complete 3D asset (Blattmann et al., 2023; Bar-Tal et al., 2024; Yang et al., 2025).

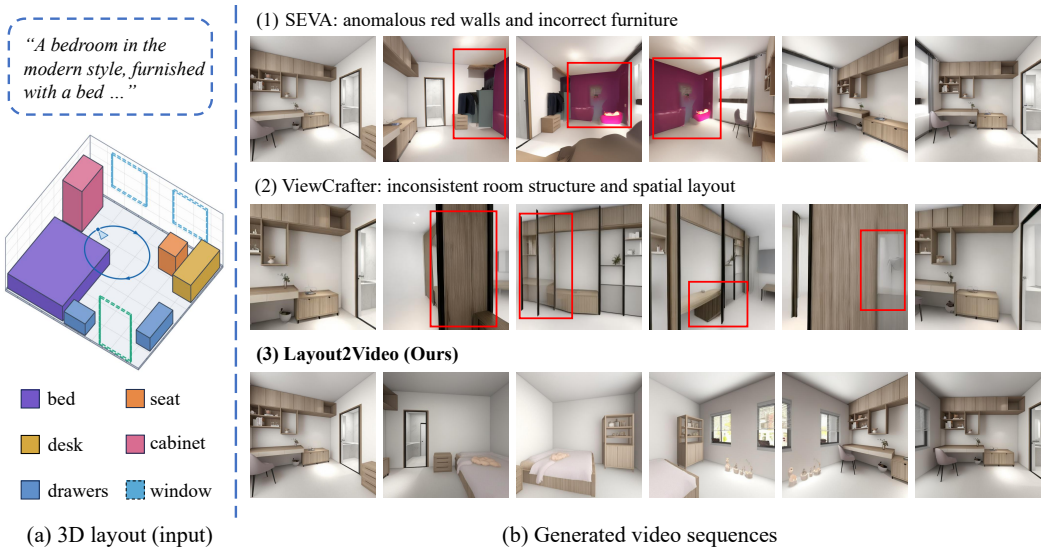
Text prompts and reference images are widely used to control the content of generated videos (Blattmann et al., 2023; Chen et al., 2024a; Yang et al., 2025). Camera-conditioned approaches guide viewpoint changes (Wang et al., 2024b; He et al., 2025), while depth, point clouds, and coarse geometry improve spatial control under camera motion (Yu et al., 2024; Ren et al., 2025; Kim et al., 2025). However, content and camera control alone cannot precisely specify 3D object arrangements. In contrast, a 3D semantic layout represents a scene as a collection of 3D semantic bounding boxes, providing precise spatial control without requiring dense geometry. While recent approaches demonstrate the effectiveness of 3D layouts (Chen et al., 2025; Yang et al., 2024; Fang et al., 2026), using these layouts for direct indoor video generation remains largely unexplored.

Unlike detailed 3D scene representations, a 3D semantic layout specifies object categories and bounding boxes but lacks fine-grained geometry or appearance. Generating a video from these sparse constraints requires maintaining temporal coherence and cross-view consistency as the camera moves (Ren et al., 2025; Kim et al., 2025). The generated sequence must preserve object identity, spatial relationships, and appearance as unseen regions enter the view.

In this paper, we present Layout2Video, a training-free framework that directly generates indoor videos from 3D semantic layouts. To provide scene information beyond a single viewpoint, we introduce Layout-Guided Anchor Generation. This stage generates layout-consistent anchors from



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.




(a) 3D layout (input) (b) Generated video sequences

Figure 1: **Indoor video generation from a 3D semantic layout.** The input layout and representative generated views are shown alongside results from SEVA (Zhou et al., 2025) and ViewCrafter (Yu et al., 2024). Compared to image-conditioned methods, Layout2Video better preserves room structure and furniture arrangement across viewpoint changes. Red annotations highlight errors.

different viewpoints of the same 3D semantic layout. To connect these sparse anchors, we introduce Sparse-Anchor Video Generation to synthesize intermediate frames between consecutive anchors. Both components build on frozen pretrained generative models, so our framework requires no task-specific training of a large video backbone. As the viewpoint changes, later anchors provide visual conditions for objects and regions that are not visible in the initial view. Figure 1 shows structural and object inconsistencies in image-conditioned baselines, whereas Layout2Video better preserves the specified layout as new regions enter the view.

We further introduce a benchmark and an instance-level evaluation protocol for layout-guided video generation. Standard measures of visual quality and temporal consistency (Huang et al., 2024) do not directly quantify agreement with a specified 3D layout. Our protocol therefore evaluates the presence of expected objects, the containment of tracked object masks within projected layout boxes, and appearance similarity to reference crops of the same objects.

Our main contributions are summarized as follows:

- We introduce a two-stage framework that converts 3D semantic layouts into sparse multi-view anchors and uses frozen pretrained models to generate video segments between consecutive anchors.
- We establish a task-specific benchmark for 3D semantic layout-guided video generation and develop an instance-level protocol for evaluating layout adherence throughout generated videos.
- Extensive experiments demonstrate that our method preserves object identity and appearance consistency while following the specified layout throughout generated videos, particularly when previously unseen regions become visible.

## 2 Related Work

**Text-to-Video Generation.** Diffusion- and transformer-based models substantially improve the visual quality and temporal coherence of text-to-video generation (Chen et al., 2024a; Bar-Tal et al., 2024; Yang et al., 2025; Team Wan et al., 2025). Text prompts offer flexible control over semantic content, appearance, and high-level actions, but provide only coarse specifications of object arrangements. This limitation matters for indoor videos driven by camera motion, where scene structure should remain consistent as new regions enter the view. Frame-wise conditioning and keyframe

guidance provide additional visual control (Zhu et al., 2025; Huang et al., 2025; Jang et al., 2026). Our framework uses layout-conditioned anchors to supply such guidance at multiple viewpoints.

**Geometry-Guided Video Generation.** Geometry-guided methods use 3D information to constrain video generation under viewpoint changes. One line of work uses rendered geometric information to guide novel-view video synthesis. ViewCrafter (Yu et al., 2024) reconstructs a point cloud from a single image or sparse views and renders it along a target camera trajectory to condition a video diffusion model for novel-view synthesis. MVSplat360 (Chen et al., 2024b) reconstructs 3D Gaussians from sparse views and uses latent feature renderings to condition video diffusion for 360-degree novel-view synthesis. Gen3C (Ren et al., 2025) builds a 3D cache from input images and renders it to guide video diffusion. WVD (Zhang et al., 2025a) jointly models RGB frames and per-pixel XYZ coordinates for camera-controlled video generation. Another line of work constructs videos through geometry-guided keyframe generation and interpolation. I2V3D (Zhang et al., 2025b) and VideoFrom3D (Kim et al., 2025) generate sparse keyframes from 3D geometry and use them as conditions for geometry-guided video interpolation.

These methods motivate the use of 3D information and sparse visual conditions for video generation. Our framework obtains its spatial specification from semantic bounding boxes and converts it into image anchors.

**Layout-Guided Scene and Video Generation.** Prior work explores layout guidance in both video and 3D scene generation, but the layout representations differ. In video generation, several methods use bounding boxes, tracklets, or blob representations to control object composition and motion (Wang et al., 2024a; Li et al., 2025; Feng et al., 2025). LVD (Lian et al., 2024) generates dynamic scene layouts from text and conditions video diffusion on the resulting layouts. DrivingDiffusion (Li et al., 2024b) generates multi-view driving videos conditioned on 3D spatial layouts. For 3D scene generation, scene-level 3D semantic layouts specify object arrangements and guide scene synthesis. SceneCraft (Yang et al., 2024), ControlRoom3D (Schult et al., 2024), and Ctrl-Room (Fang et al., 2025) condition indoor scene generation on 3D semantic layouts. Layout2Scene (Chen et al., 2025) employs 3D semantic bounding boxes to guide geometry and appearance generation in separate stages.

These studies establish layout guidance for both video synthesis and 3D scene generation. Our work focuses on indoor video generation from a static, scene-level 3D semantic layout, using layout-conditioned sparse anchors to connect multiple viewpoints.

## 3 METHOD

We present Layout2Video, a training-free framework for generating videos from 3D semantic layouts. We represent the objects in a scene layout as a collection of oriented 3D bounding boxes with semantic labels:

$$\mathcal{L} = \{o_n\}_{n=1}^{N}, \qquad o_n = (c_n, \mathbf{t}_n, \mathbf{s}_n, \mathbf{R}_n), \tag{1}$$

where $c_n$ denotes the semantic category of object $n$, $\mathbf{t}_n \in \mathbb{R}^3$ is the center of its oriented 3D bounding box, $\mathbf{s}_n \in \mathbb{R}^3_+$ gives the box dimensions, and $\mathbf{R}_n \in \mathrm{SO}(3)$ specifies its orientation. For rendering, we augment the object layout with walls, a floor, and a ceiling constructed from the room boundary and height. Given this layout $\mathcal{L}$ and a text prompt $y$, we aim to generate a video $\mathcal{V} = (v_1, \ldots, v_T)$ whose visible objects follow the specified categories and spatial arrangement while retaining coherent appearances across views.

The layout provides a shared spatial specification for the generated views. As illustrated in Figure 2, our framework converts this specification into video conditions in two stages. **Layout-Guided Anchor Generation** renders view-specific semantic and geometric conditions, generates candidate scene images, and selects four of these images as sparse anchors. **Sparse-Anchor Video Generation** then conditions a video diffusion model on each consecutive pair of anchors and concatenates the resulting segments. We denote the models for joint candidate generation and video generation by $G_A$ and $G_V$, respectively. All pretrained generative components remain frozen during this process.

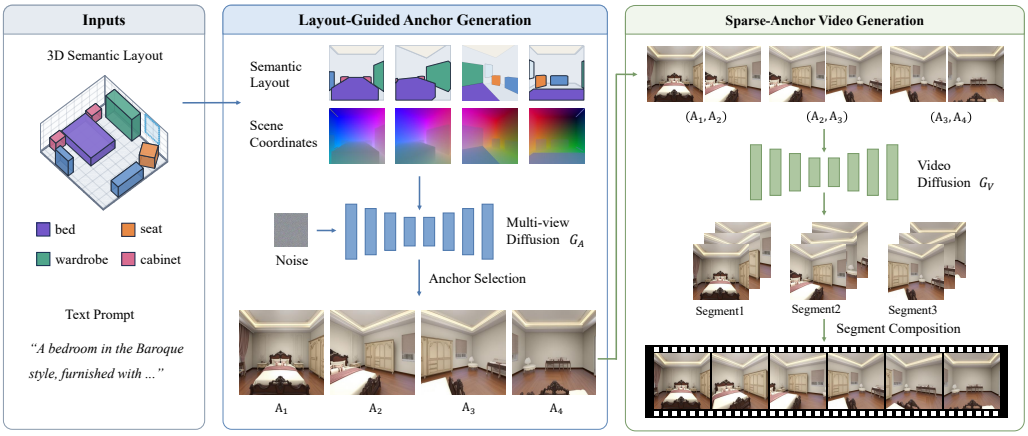



Figure 2: **Overview of Layout2Video.** Given a 3D semantic layout and a text prompt, the first stage renders semantic and geometric conditions for an ordered sequence of views. An image diffusion model generates an initial reference image from the text prompt and a wireframe projection of the layout. Conditioned on the reference image and the per-view layout conditions, $G_A$ jointly generates candidate images for the remaining views. The reference image and three selected candidates form the sparse anchors. In the second stage, $G_V$ uses consecutive anchors as first and last frame conditions to generate video segments, which are concatenated into the final video.

### 3.1 Layout-Guided Anchor Generation

A single image does not directly observe objects outside its field of view. A 3D semantic layout, however, specifies the categories and bounding boxes of objects across the represented room. We render this layout from an ordered sequence of views and generate a corresponding image for each view. Conditioning these images on a shared layout and reference image is intended to encourage consistent object arrangements and appearances. We then select a subset of the generated images as sparse anchors for video generation.

**Layout conditions for individual views.** We define an ordered sequence of camera views $\mathcal{C}$ within the room:

$$\mathcal{C} = (C_1, \ldots, C_M), \tag{2}$$

where $C_m$ contains the intrinsic and extrinsic parameters of the $m$-th view. We place the views along a smooth open arc inside the room and orient each camera toward the room interior, producing a gradual change in viewpoint. For each view, we project the oriented 3D bounding boxes together with the room structure to render a semantic layout map and a layout depth map:

$$S_m = R_S(\mathcal{L}, C_m), \qquad D_m = R_D(\mathcal{L}, C_m), \tag{3}$$

where $R_S$ and $R_D$ denote the semantic and depth renderers, respectively. The semantic map $S_m$ encodes the categories and projected regions of visible layout elements, while $D_m$ records the depths of their rendered proxy surfaces. We unproject $D_m$ using the camera parameters to obtain a scene coordinate map in a shared 3D coordinate system:

$$P_m = \Phi(D_m, C_m), \tag{4}$$

where $\Phi$ denotes depth unprojection. For each valid pixel $\mathbf{u} = (u, v)$, $P_m(\mathbf{u}) \in \mathbb{R}^3$ records the 3D position of the corresponding visible layout surface. The tuple $(S_m, P_m, C_m)$ forms the layout condition for the $m$-th view. All conditions are derived from the same 3D layout and share a common semantic and spatial reference.

**Anchor generation and selection.** At the initial view $C_1$, an image diffusion model generates a reference image $I_{\text{ref}}$ from a wireframe projection of the layout and the text prompt $y$. We set $I_1 = I_{\text{ref}}$ and use this image as the appearance reference for the remaining views. The diffusion

model $G_A$ (Fang et al., 2026) jointly generates the remaining candidate images conditioned on $I_{\text{ref}}$ and the per-view semantic, geometric, and camera conditions:

$$\{I_m\}_{m=2}^{M} = G_A\big(I_{\text{ref}}, \{S_m, P_m, C_m\}_{m=1}^{M}\big)\,. \tag{5}$$

We select $K$ images at uniformly spaced indices in the ordered candidate sequence, including the first and last images. Let $i_k$ denote the index of the $k$-th anchor in this sequence. The anchors are defined as:

$$\mathcal{A} = (A_1, \ldots, A_K), \qquad A_k = I_{i_k}, \quad k = 1, \ldots, K, \tag{6}$$

In our implementation, $M = 16$ and $K = 4$, giving anchor indices $(1, 6, 11, 16)$. These anchors provide the endpoint conditions for three consecutive video segments.

### 3.2 Sparse-Anchor Video Generation

The $K$ sparse anchors define $K-1$ video segments. We use consecutive anchors $(A_k, A_{k+1})$ as the first and last frame conditions for the video diffusion model $G_V$. The model generates the segment between them according to the text prompt $y$:

$$\mathcal{V}_k = G_V(A_k, A_{k+1}, y) = \{v_{k,t}\}_{t=0}^{T_s-1}, \qquad k = 1, \ldots, K-1, \tag{7}$$

where $T_s$ denotes the number of frames in each segment. The same video model and text prompt are used for all segments. A later anchor may contain objects or regions that are not visible in the preceding anchor. Conditioning on both anchors provides the model with visual information about both endpoints when synthesizing the intermediate frames.

Adjacent segments use a common anchor as an endpoint condition: $A_{k+1}$ conditions the end of $\mathcal{V}_k$ and the beginning of $\mathcal{V}_{k+1}$. We concatenate the segments in temporal order, omitting the first frame of each segment after the first. Let $\oplus$ denote the concatenation operation:

$$\mathcal{V} = \mathcal{V}_1 \oplus \bigoplus_{k=2}^{K-1} (v_{k,1}, \ldots, v_{k,T_s-1}). \tag{8}$$

For $K-1$ segments of $T_s$ frames each, the final sequence contains $T = (K-1)(T_s-1)+1$ frames. This construction provides layout-derived visual guidance through the anchors, but does not explicitly enforce the 3D layout at every intermediate frame.

## 4 Experiments

### 4.1 Dataset and Experimental Setup

We construct our benchmark from the Habitat Synthetic Scenes Dataset (HSSD), which contains manually designed 3D scenes with instance-level semantic annotations (Khanna et al., 2024). From 160 HSSD scenes, we extract individual rooms and convert the annotated objects into a unified 3D semantic layout representation. After filtering by room area, object count, and category diversity, we obtain 555 layouts across bedrooms, dining rooms, kitchens, and living rooms.

Each layout is aligned with its source scene, enabling the rendering of corresponding RGB, depth, and semantic sequences. Our main comparison evaluates all methods on the same subset of 30 rooms, covering all four room types.

We provide every baseline with the same initial reference image used by Layout2Video. We generate this image using FLUX.1-dev with a pretrained LoRA. For the first–last variants, we additionally provide the final anchor generated by our layout-guided anchor generation stage.

For joint candidate generation, we use the pretrained multi-view multi-modal diffusion model (Fang et al., 2026) based on Stable Diffusion 2.1. We uniformly select four views from the ordered candidate sequence as anchors, including the first and last views. For video generation, we use Wan2.1-FLF2V-14B (Team Wan et al., 2025). For each pair of consecutive anchors, we generate a 29-frame segment using 50 sampling steps. The three segments are concatenated by omitting the first frame of each segment after the first, yielding an 85-frame video at $512 \times 512$ resolution and 16 FPS.

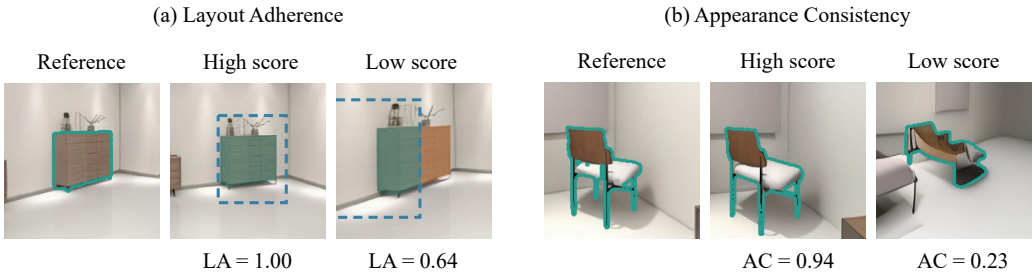



Figure 3: **Visual interpretation of Layout Adherence (LA) and Appearance Consistency (AC).** (a) Higher LA corresponds to better containment of the tracked object mask within the projected target layout box, whereas lower LA reveals visible spatial misalignment (See Blue dashed box). (b) Higher AC corresponds to closer visual resemblance to the reference object, whereas lower AC reveals pronounced differences in object shape and appearance (See Green outlines).

### 4.2 Evaluation Protocol

Our protocol compares objects in a generated video with those specified by the input 3D layout. For each video, we use VGGT (Wang et al., 2025) to recover camera poses from sampled frames and align them to the layout coordinate system. From the aligned poses, we determine which layout objects should be visible and project their 3D boxes onto each frame. At each input anchor, we query Grounding DINO (Liu et al., 2024) with the layout categories and use SAM 2.1 (Ravi et al., 2025) to segment the detected objects. We use the Hungarian algorithm to match the resulting masks to the projected boxes and then track the matched objects throughout the video with SAM 2.1.

We evaluate three aspects of agreement with the layout: whether expected objects are present, whether their image regions lie within the projected layout boxes, and whether their appearances resemble available references.

**Object Presence (OP).** OP is the fraction of expected object instances recovered by the detection-and-tracking pipeline. It measures detected presence and therefore depends on the reliability of detection, matching, and tracking.

**Layout Adherence (LA).** For a nonempty tracked object mask $M$ and a projected layout box region $B$, we define $\mathrm{LA}(M, B) = |M \cap B|/|M|$, where $|\cdot|$ denotes pixel area. LA measures mask containment within the projected box. A high score does not by itself establish correct object scale, complete box coverage, or accurate 3D geometry.

**Appearance Consistency (AC).** AC is the maximum DINOv2 cosine similarity between a tracked object crop and valid anchor crops of the same object. It therefore measures similarity to at least one available reference, rather than agreement across all frames or references.

Tracking failures receive zero, and scores are averaged over objects expected to be visible. These measures characterize object-level agreement with the layout and anchor references; they do not directly measure overall video quality or adherence to a target camera trajectory.

Figure 3 illustrates LA and AC with selected generated object crops. The higher-LA cabinet mask lies within its projected layout box, whereas the lower-LA mask extends beyond it. The higher-AC chair more closely resembles its reference, while the lower-AC chair differs visibly in shape.

We evaluate all three metrics every fourth frame and exclude each method's input frames. We also report these metrics at different stages of each video.

### 4.3 Comparisons with the State of the Art

We compare our method with representative methods for novel-view synthesis and video generation from images. These include ViewCrafter (Yu et al., 2024) and SEVA (Zhou et al., 2025) for novel-view synthesis, and Wan2.2-5B and Wan2.2-A14B from the Wan2.2-Fun-InP family (Team Wan et al., 2025) for video generation from images.

Table 1: **Quantitative comparisons of video generation methods.** Wan2.2-5B and Wan2.2-A14B denote the corresponding Wan2.2-Fun-InP variants.

| Method | Direct 3D Layout Input | Object Presence ↑ | Layout Adherence ↑ | Appearance Consistency ↑ |
|---|---|---|---|---|
| ViewCrafter | – | 0.4569 | 0.3725 | 0.2724 |
| SEVA | – | 0.5689 | 0.5001 | 0.3440 |
| Wan2.2-5B (single image) | – | <u>0.7012</u> | <u>0.6316</u> | <u>0.5878</u> |
| Wan2.2-5B (first–last) | – | 0.6260 | 0.2910 | 0.4774 |
| Wan2.2-A14B (single image) | – | 0.6545 | 0.6037 | 0.5174 |
| Wan2.2-A14B (first–last) | – | 0.4583 | 0.2443 | 0.3437 |
| **Layout2Video (Ours)** | ✓ | **0.8818** | **0.7738** | **0.6828** |

Table 2: **Ablation results for the number of anchors and the anchor generation strategy.**

**(a) Number of Anchors**

| $K$ | OP ↑ | LA ↑ | AC ↑ |
|---|---|---|---|
| 2 | 0.6101 | 0.3091 | 0.4460 |
| 3 | 0.7809 | 0.6400 | 0.5950 |
| **4** | **0.8585** | **0.7113** | **0.6643** |

**(b) Anchor Generation Strategy**

| Strategy | Anchor Consistency ↑ | MUSIQ ↑ |
|---|---|---|
| Independent | 0.3124 | **73.92** |
| Joint | **0.3562** | 70.62 |

**Quantitative Comparisons.** Table 1 presents the quantitative comparison with the existing approaches. Among the compared methods, ours is the only one that takes a 3D semantic layout as input and achieves the highest scores on all three evaluation metrics. Specifically, our method achieves 0.8818 in Object Presence, 0.7738 in Layout Adherence, and 0.6828 in Appearance Consistency. Wan2.2-5B with a single input image is the strongest baseline across all three metrics, but its generated videos cover only a narrow range of viewpoints.

These results show that explicit 3D layout guidance helps preserve the specified objects, their spatial placement, and their appearance as the viewpoint changes.

**Qualitative Comparisons.** Figure 4 compares representative video frames across four scenes, with red boxes highlighting visual artifacts. As the viewpoint changes, Wan2.2-A14B alters counter arrangements and room structures. ViewCrafter produces distorted boundaries and repeated furniture, while SEVA exhibits deformed objects and changes in their appearance. In these selected examples, Layout2Video preserves the specified object arrangement more faithfully and exhibits fewer changes in object shape and spatial relationships as additional regions become visible.

**Performance Across Video Stages.** Figure 5 compares the performance of all methods across five stages of the generated videos. Our method ranks first on all three metrics at every stage and retains a clear lead through the final stage. Wan2.2-5B with a single input image remains relatively stable, consistent with the limited range of viewpoints in its generated videos. The remaining methods generally decline as the videos progress. Both Wan2.2 models conditioned on the first and last frames show higher scores near the endpoint.

The stage-wise results are consistent with the motivation for sparse-anchor conditioning: intermediate anchors provide visual references beyond the initial view and divide the viewpoint transition into shorter segments. The anchor-count ablation below provides a more direct comparison of this design choice.

### 4.4 Ablation Studies

**Effect of the Number of Anchors.** We compare $K \in \{2, 3, 4\}$ on the same 30 rooms used in the main evaluation, generating 81-frame videos with 20 sampling steps.

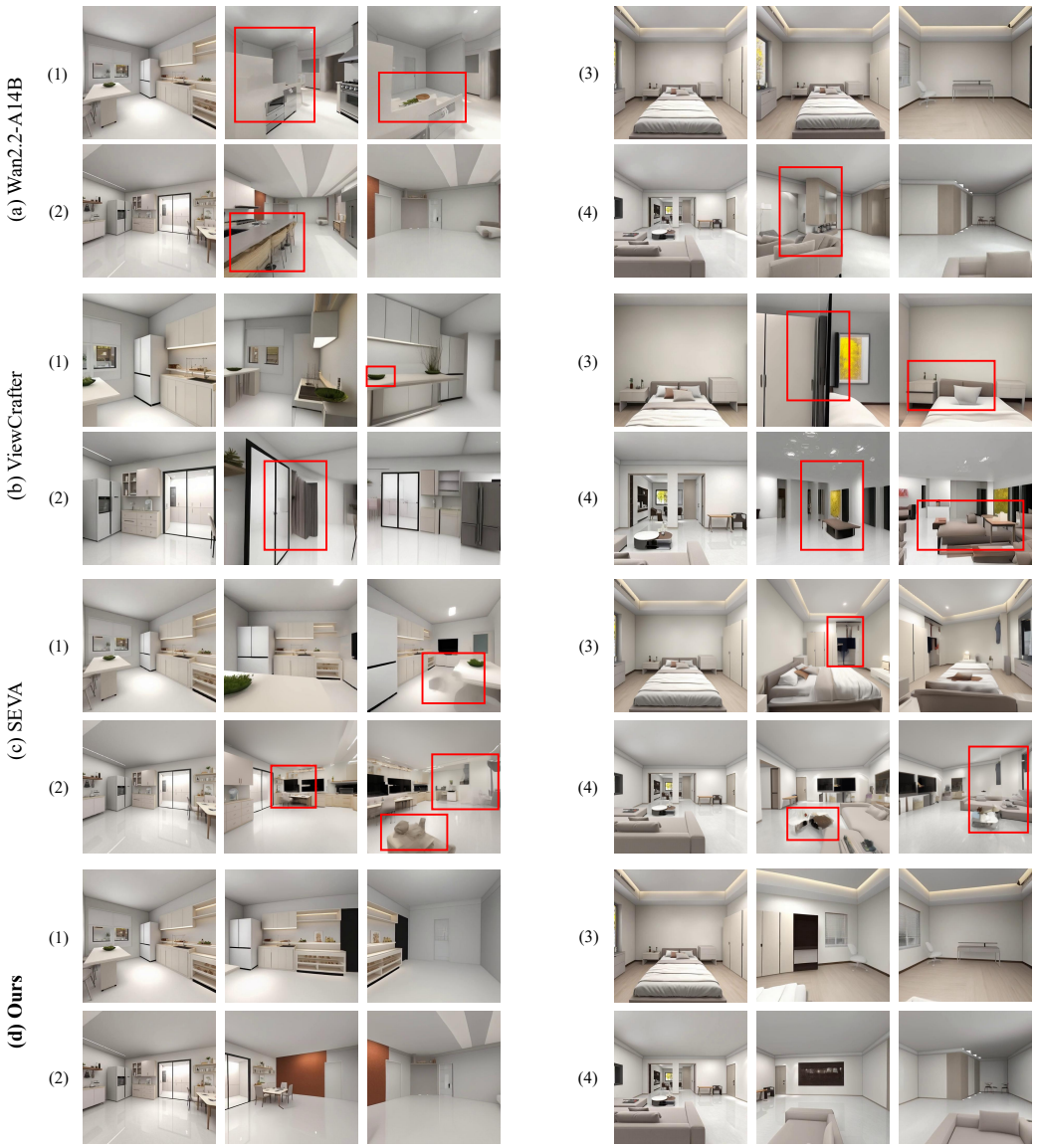



Figure 4: **Qualitative comparisons of generated videos across four scenes:** (1) kitchen, (2) dining room, (3) bedroom, and (4) living room. Red boxes mark representative structural and object-level errors in the baseline results.

Table 2(a) shows that all three metrics improve as more anchors are added and reach their highest values with four anchors. In particular, Layout Adherence increases from 0.3091 with two anchors to 0.7113 with four anchors.

**Effect of Anchor Generation Strategies.** We next examine how joint generation affects anchor consistency across views. The independent strategy applies the image generator used for the reference image to each remaining view separately, whereas our method generates the corresponding images jointly. We define Anchor Consistency as the DINOv2 similarity of the same object between consecutive anchors. Table 2(b) shows that the joint strategy obtains a higher Anchor Consistency score than the independent strategy (0.3562 versus 0.3124), but a lower MUSIQ score (70.62 versus 73.92). The two metrics therefore favor different strategies: the joint configuration improves measured cross-view appearance consistency, whereas the independent configuration receives a higher no-reference image-quality score. These results motivate our choice of the joint configuration for anchor generation.

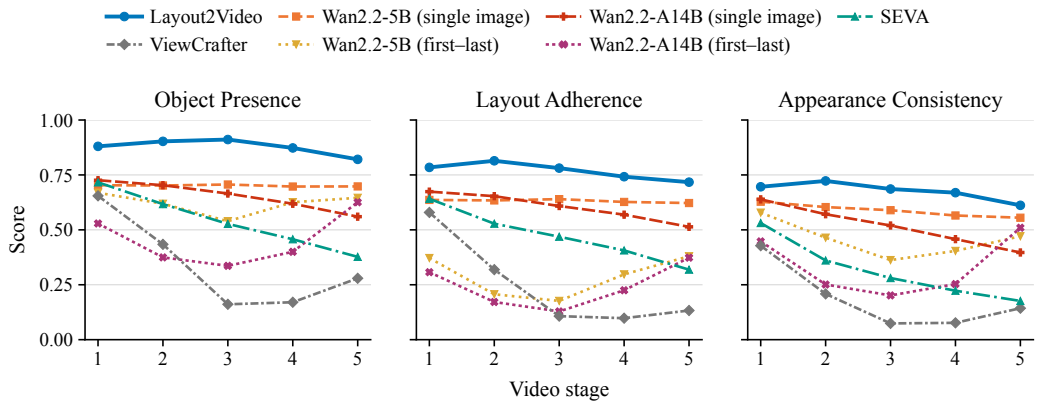



Figure 5: **Performance across five video stages.** We report Object Presence (OP), Layout Adherence (LA), and Appearance Consistency (AC) for all compared methods at each stage.

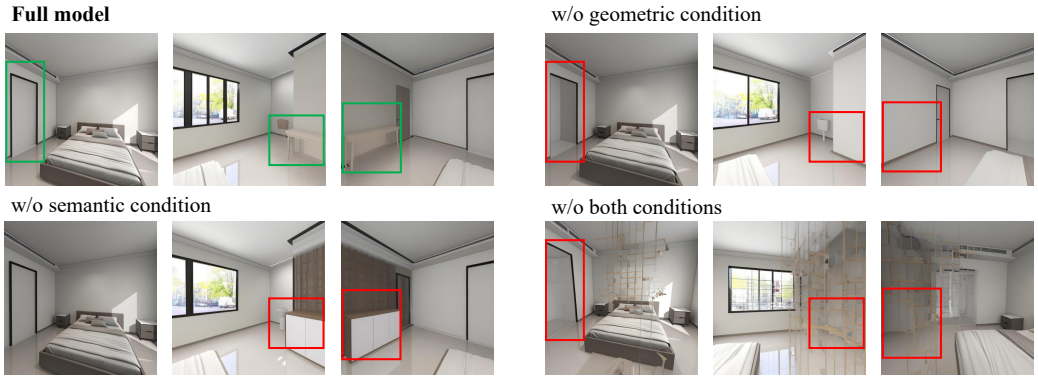



Figure 6: **Qualitative ablation of semantic and geometric conditions in Layout-Guided Anchor Generation.** Each variant shows three matched views generated with the same reference image, camera poses, and random seed. Green boxes mark reference regions in the full model, while red boxes indicate the corresponding regions in the ablated outputs where representative errors occur.

**Effect of Semantic and Geometric Conditions.** Figure 6 presents a qualitative ablation of the semantic and geometric conditions used in Layout-Guided Anchor Generation. We zero the semantic condition, geometric condition, or both at inference while keeping the model weights, reference image, camera poses, and random seed fixed. Removing the semantic condition changes the furnishings and wall appearance. Without the geometric condition, the desk is absent from the corresponding views, while removing both produces grid-like artifacts. The full model retains more intact structures and furnishings in this example.

## 5 Conclusion

We present Layout2Video, a framework for generating indoor videos from 3D semantic layouts and text prompts using frozen pretrained models. The framework generates layout-conditioned anchors at multiple viewpoints and synthesizes video segments between consecutive anchors. We also construct a benchmark containing 555 layouts and introduce an instance-level evaluation protocol. Experimental results show that Layout2Video achieves the best results on all metrics and maintains its advantage across video stages. These findings support the use of 3D semantic layouts and sparse anchors for spatially guided indoor video generation.

### AI USE STATEMENT

We used generative AI tools for language polishing. The authors reviewed all AI-assisted edits and remain responsible for the content of the manuscript.

## REFERENCES

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. In *Proceedings of the SIGGRAPH Asia Conference Papers*, pp. 1–11, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable Video Diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. VideoCrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7310–7320. IEEE, 2024a.

Minglin Chen, Longguang Wang, Sheng Ao, Ye Zhang, Kai Xu, and Yulan Guo. Layout2Scene: 3D semantic layout guided scene generation via geometry and appearance diffusion priors. *arXiv preprint arXiv:2501.02519*, 2025.

Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. MVSplat360: Feed-forward 360 scene synthesis from sparse views. *Advances in Neural Information Processing Systems*, 37:107064–107086, 2024b.

Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. LucidDreamer: Domain-free generation of 3D Gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023.

Chuan Fang, Yuan Dong, Kunming Luo, Xiaotao Hu, Rakesh Shrestha, and Ping Tan. Ctrl-Room: Controllable text-to-3D room meshes generation with layout constraints. In *International Conference on 3D Vision (3DV)*, pp. 692–701. IEEE, 2025.

Chuan Fang, Heng Li, Yixun Liang, Jia Zheng, Yongsen Mao, Yuan Liu, Rui Tang, Zihan Zhou, and Ping Tan. SpatialGen: Layout-guided 3D indoor scene generation. In *International Conference on 3D Vision (3DV)*, pp. 1392–1402. IEEE, 2026.

Weixi Feng, Chao Liu, Sifei Liu, William Yang Wang, Arash Vahdat, and Weili Nie. BlobGEN-Vid: Compositional text-to-video generation with blob video representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12989–12998. IEEE, 2025.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 100433–100464, 2025.

Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2Room: Extracting textured 3D meshes from 2D text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7875–7886. IEEE, 2023.

Hsin-Ping Huang, Yu-Chuan Su, and Ming-Hsuan Yang. Generating long-take videos via effective keyframes and guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3709–3720. IEEE, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Sangwon Jang, Taekyung Ki, Jaehyeong Jo, Jaehong Yoon, Soo Ye Kim, Zhe Lin, and Sung Ju Hwang. Frame Guidance: Training-free guidance for frame-level control in video diffusion models. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 60977–61003, 2026.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5128–5137. IEEE, 2021.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat Synthetic Scenes Dataset (HSSD-200): An analysis of 3D scene scale and realism tradeoffs for ObjectGoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16384–16393. IEEE, 2024.

Geonung Kim, Janghyeok Han, and Sunghyun Cho. VideoFrom3D: 3D scene video generation via complementary image and video diffusion models. In *Proceedings of the SIGGRAPH Asia Conference Papers*, pp. 1–11, 2025.

Haoran Li, Haolin Shi, Wenli Zhang, Wenjun Wu, Yong Liao, Lin Wang, Lik-hang Lee, and Peng Yuan Zhou. DreamScene: 3D Gaussian-based text-to-3D scene generation via formation pattern sampling. In *European Conference on Computer Vision (ECCV)*, pp. 214–230. Springer, 2024a.

Pengxiang Li, Kai Chen, Zhili Liu, Ruiyuan Gao, Lanqing Hong, Dit-Yan Yeung, Huchuan Lu, and Xu Jia. TrackDiffusion: Tracklet-conditioned video generation via diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3539–3548. IEEE, 2025.

Xiaofan Li, Yifu Zhang, and Xiaoqing Ye. DrivingDiffusion: Layout-guided multi-view driving scenarios video generation with latent diffusion model. In *European Conference on Computer Vision (ECCV)*, pp. 469–485. Springer, 2024b.

Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. LLM-grounded video diffusion models. In *International Conference on Learning Representations (ICLR)*, volume 2024, pp. 50207–50227, 2024.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, pp. 38–55. Springer, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 28085–28128, 2025.

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 (CVPR)*, pp. 6121–6132. IEEE, 2025.

Jonas Schult, Sam Tsai, Lukas Höllein, Bichen Wu, Jialiang Wang, Chih-Yao Ma, Kunpeng Li, Xiaofang Wang, Felix Wimbauer, Zijian He, et al. ControlRoom3D: Room generation using semantic proxy rooms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6201–6210. IEEE, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566*, 2024a.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *Proceedings of the ACM SIGGRAPH Conference Papers*, pp. 1–11, 2024b.

Xiuyu Yang, Yunze Man, Jun-Kun Chen, and Yu-Xiong Wang. SceneCraft: Layout-guided 3D scene generation. *Advances in Neural Information Processing Systems*, 37:82060–82084, 2024.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 83048–83077, 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.

Qihang Zhang, Shuangfei Zhai, Miguel Angel Bautista Martin, Kevin Miao, Alexander Toshev, Joshua Susskind, and Jiatao Gu. World-consistent video diffusion with explicit 3D modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21685–21695, 2025a.

Zhiyuan Zhang, Dongdong Chen, and Jing Liao. I2V3D: Controllable image-to-video generation with 3D guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13360–13371. IEEE, 2025b.

Jensen 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12405–12414. IEEE, 2025.

Tianyi Zhu, Dongwei Ren, Qilong Wang, Xiaohe Wu, and Wangmeng Zuo. Generative inbetweening through frame-wise conditions-driven video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27968–27978. IEEE, 2025.

# A Additional Qualitative Results

## A.1 Different Styles from the Same Layout

Figure 7 shows videos generated from the same 3D semantic layout with different style prompts. With the layout input held fixed, the generated views exhibit distinct furniture designs, materials, and decorative details across styles.




Figure 7: **Different styles from the same layout.** Left: the input bedroom layout. Right: video frames in **(a)** modern, **(b)** Chinese, and **(c)** Baroque styles, ordered from left to right in time.

## A.2 Qualitative Effect of the Number of Anchors

Figure 8 complements the anchor-count comparison in Table 2(a) with selected living-room and dining-room examples. The highlighted differences provide a visual counterpart to the improvements in OP, LA, and AC observed as the number of anchors increases.

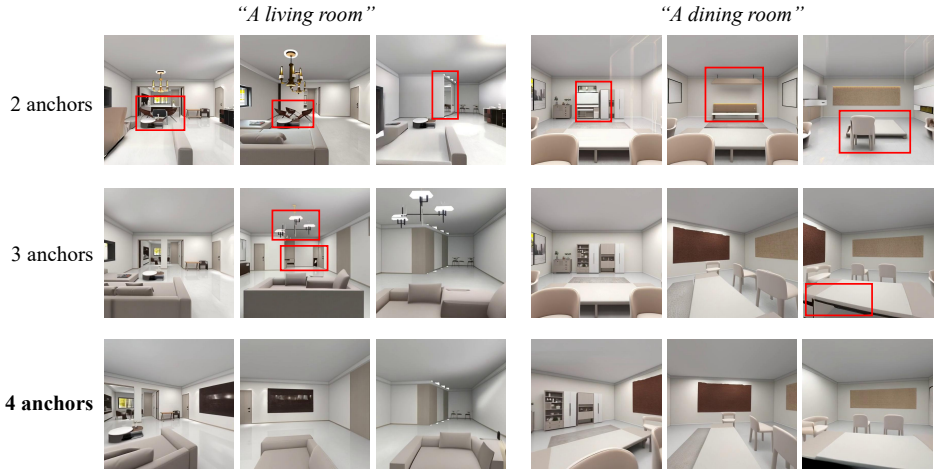



Figure 8: **Qualitative comparison of different numbers of anchors.** Rows use 2, 3, and 4 anchors from top to bottom; the left and right groups show a living room and a dining room, respectively. Red boxes highlight selected inconsistencies in furniture, room structure, or lighting in the 2- and 3-anchor examples.

### A.3 Independent versus Joint Anchor Generation

Figure 9 compares independent generation with FLUX.1-dev and its wireframe LoRA against joint generation with SpatialGen. Furniture appearance is more consistent across the jointly generated views in these examples, complementing the Anchor Consistency comparison in Table 2(b).




Figure 9: **Independent versus joint anchor generation.** Rows **(a)–(d)** show four room examples, with independent generation on the left and joint generation on the right. Each group shows three anchor views.

## B Benchmark Construction and Evaluation Subset

### B.1 Room Selection

We index 742 rooms from 160 HSSD scenes (Khanna et al., 2024) across four room types. Rooms qualify if their indexed area is 8–80 $\mathrm{m}^2$, with 3–12 layout objects spanning at least two semantic categories. This filtering yields an eligible pool of 555 room layouts from 156 scenes; Table 3 gives the counts by room type.

### B.2 Scene-Aligned Layouts

Each layout represents objects as oriented 3D bounding boxes with semantic labels, specifying their categories, positions, sizes, and orientations. Source-instance identifiers preserve the correspondence between these boxes and objects in the original scene. The prepared inputs include semantic maps $S_m$ and layout depth maps $D_m$ at the candidate cameras $C_m$. As in Section 3.1, depth is unprojected to scene coordinate maps $P_m = \Phi(D_m, C_m)$. The tuple $(S_m, P_m, C_m)$ supplies the semantic, geometric, and camera conditions for generation. The instance correspondence also links generated content to the expected layout objects for object-level evaluation.

### B.3 Evaluation Subset

The main comparison and anchor-count ablation use the first 30 rooms in the pre-existing 52-room evaluation order. These rooms come from 30 distinct scenes and cover all four room types, with the composition reported in Table 3. All compared methods have outputs for this common room set.

Table 3: Room counts before and after filtering, and in the fixed evaluation subset.

| Room type | Indexed rooms | Eligible layouts | Evaluation rooms |
|---|---|---|---|
| Bedroom | 368 | 315 | 8 |
| Dining room | 81 | 61 | 6 |
| Kitchen | 165 | 89 | 8 |
| Living room | 128 | 90 | 8 |
| Total | 742 | 555 | 30 |

### B.4 Scope of the Main Comparison

Table 1 compares complete pipelines with their respective pretrained backbones and input conditions. The direct-layout-input column describes the model interface: every baseline receives the same layout-derived initial reference image, and first–last variants additionally receive our final anchor. The main Layout2Video setting uses 85-frame videos at 16 FPS (approximately 5.3 seconds), with layout guidance supplied through sparse anchors. The evaluation covers this sequence length and the open-arc camera configuration described in Section C.

## C Generation Implementation Details

### C.1 Pretrained Components

The reference-image generator combines FLUX.1-dev with the published FLUX.1-Wireframe-dev-lora. The joint anchor generator uses the released SpatialGen-1.0 model (Fang et al., 2026). These components and Wan2.1-FLF2V-14B-720P are used with frozen weights; training-free refers to their reuse without task-specific training in our pipeline.

### C.2 Candidate Cameras

The 16 candidate views sample a room-specific elliptical open arc over $180^\circ$, including both endpoints, at uniform increments of the ellipse parameter angle. The ellipse is centered at the room polygon centroid. Its initial horizontal radii are 20% of the room extents along the two axes, clipped to $[0.35, 1.20]\,\mathrm{m}$, and reduced together when needed to keep the sampled camera positions inside the room. The starting angle is selected from 16 orientations using projected layout-box visibility and occlusion criteria. Cameras are $1.20\,\mathrm{m}$ above the floor and face the room centroid at the same height. At $512 \times 512$, the horizontal field of view is $90^\circ$, with $f_x = f_y = 256$ and principal point $(256, 256)$. These parameters define the input candidate cameras; intermediate video views are generated by the video model.

### C.3 Main Video Sampling

For the main 85-frame configuration, each of the three 29-frame segments uses 50 sampling steps, guidance scale 5.0, timestep shift 16.0, and seed 43. The same scene/style prompt is used across the three segments of a video. The prompt requests slow, steady camera motion and stable furniture, geometry, materials, lighting, and exposure; the negative prompt suppresses abrupt motion, cuts, flicker, and scene changes.

### C.4 Endpoint Handling and Concatenation

Each segment is generated using its two anchors as first- and last-frame conditions. The export and stitching code retains the decoded frames returned by the video model, without replacing them with the input anchor images. For zero-based segment frames $\mathcal{V}_k = (v_{k,0}, \ldots, v_{k,T_{s,k}-1})$, the concatenation is explicitly

$$\mathcal{V} = \mathcal{V}_1 \oplus \bigoplus_{k=2}^{K-1} (v_{k,1}, \ldots, v_{k,T_{s,k}-1}). \tag{9}$$

Thus, each later segment loses its first frame at the shared anchor time. The stitching operation concatenates the trimmed video segments without boundary blending. The shared anchor supplies a common image condition for adjacent segments. Boundary pixel values and camera velocities are determined by the independently generated segments.

## D Instance-Level Evaluation Details

### D.1 Evaluation Targets, Frames, and Camera Alignment

**Expected objects.** We evaluate layout objects that should be visible in each scored frame. Visibility is computed by depth-buffer rasterization of the source-aligned layout cuboids and structural mesh at the aligned camera pose, with a $0.05\,\mathrm{m}$ near plane. The visible-pixel count therefore accounts for occlusion by this geometry and is independent of detections in the generated image. Objects meeting the projected-area and visible-pixel thresholds remain in the evaluation even if detection or tracking fails. At the reference resolution of $512 \times 320$, the minimum projected area, visible area, and accepted mask area are 64, 48, and 64 pixels, respectively. Each threshold is multiplied by the evaluated image area divided by $512 \times 320$ and rounded up; at $512 \times 512$, the thresholds are 103, 77, and 103 pixels.

**Scored frames.** We sample every fourth video frame and exclude the method's input anchors. For the 85-frame Layout2Video sequence, removing anchors $(0, 28, 56, 84)$ leaves 18 scored frames. Single-image methods exclude the first frame; first–last methods exclude both endpoints.

**Camera alignment.** To project layout boxes into generated frames, we express recovered camera poses in layout coordinates. VGGT (Wang et al., 2025) processes every second frame within Layout2Video intervals $[0, 28]$, $[28, 56]$, and $[56, 84]$. For each interval, the first anchor fixes orientation, position, and metric scale $s_z$. We center-crop layout depth and apply the RGB padding transform to match the $518 \times 518$ VGGT input. The scale $s_z$ is the median layout-to-predicted depth ratio after trimming the lowest and highest 5%. Valid ratios require finite positive layout depth, finite predicted depth greater than $10^{-8}$, and finite predicted-depth confidence of at least 1.0. We exclude pixels adjacent to a layout-depth change greater than $0.10\,\mathrm{m}$ and require at least 1,000 valid ratios before trimming.

Let $(R_i, c_i)$ denote recovered camera-to-world poses and $(\bar{R}_a, \bar{c}_a)$, $(\bar{R}_b, \bar{c}_b)$ the known poses at the starting and ending anchors. We first align the interval to its starting anchor using $Q = \bar{R}_a R_a^{\mathsf{T}}$:

$$\begin{aligned} \hat{R}_i &= Q R_i, \\ \hat{c}_i &= \bar{c}_a + s_z Q (c_i - c_a). \end{aligned} \tag{10}$$

We then distribute the remaining endpoint discrepancy across the interval. Define the position residual $d = \bar{c}_b - \hat{c}_b$ and rotation correction $D = \bar{R}_b \hat{R}_b^{\mathsf{T}}$. For frame index $f_i$, let $\alpha_i = \frac{f_i - f_a}{f_b - f_a}$ denote its relative position between the anchors. The corrected poses are

$$\begin{aligned} R_i^* &= \mathrm{Exp}(\alpha_i \,\mathrm{Log}(D))\, \hat{R}_i, \\ c_i^* &= \hat{c}_i + \alpha_i d. \end{aligned} \tag{11}$$

Here Exp and Log are the exponential and logarithm maps on $\mathrm{SO}(3)$. We project rotations onto $\mathrm{SO}(3)$, fix endpoint poses to known anchors, and average shared-boundary intrinsics. Projection uses VGGT-predicted intrinsics mapped back to the evaluated image coordinates by subtracting

preprocessing padding from the principal point and undoing the horizontal and vertical resize factors. Single-image baselines use initial-anchor alignment; first–last variants also correct endpoint drift.

Alignment was developed on 20 separate rooms and frozen before evaluation; all 30 evaluation rooms passed its geometric checks. It uses geometry and anchor correspondences, without detections or metric scores.

### D.2 Instance Correspondence across Anchors

**Assigning object identities.** Layout instance identifiers link the same object across anchors. Grounding DINO Tiny detects layout categories in each anchor, and SAM 2.1 segments the detections. Hungarian matching pairs these masks with category-compatible projected layout boxes using their spatial agreement and detection confidence. Matches passing the assignment, mask-area, and containment thresholds inherit their layout instance identifiers, distinguishing objects of the same category across views. We use detection box/text thresholds of 0.20/0.18, same-category non-maximum suppression at 0.55 IoU, and duplicate-mask containment at 0.85. For a compatible mask–box pair, the assignment score is

$$S = 0.65u + 0.15h + 0.10v + 0.05\max(0, 1 - d/0.5) + 0.05q, \tag{12}$$

where $u$ is the fraction of mask pixels inside the box, $h$ indicates whether the mask centroid lies inside it, $v$ is mask–box IoU, $d$ is the centroid-to-box-center distance divided by the image diagonal, and $q$ is detection confidence. Accepted matches require $S \geq 0.10$, $u \geq 0.50$, and the mask-area threshold above.

**Tracking and appearance references.** SAM 2.1 Hiera Base Plus propagates each accepted anchor mask forward and backward through the video. Each scored frame uses the track from the temporally nearest accepted anchor for that instance, preferring the earlier anchor in ties. Different anchors can therefore provide references for objects visible in different parts of the camera trajectory. Only accepted input-anchor detections initialize tracks; non-anchor frames do not introduce new detection seeds. For appearance comparison, we expand the tight mask bounding box by 10% on each side, fill pixels outside the mask with RGB $(124, 116, 104)$, and center-pad the crop to a square with the same background. The crop is resized to $224 \times 224$ by bicubic interpolation and normalized using ImageNet channel means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$ after scaling pixels to $[0, 1]$. DINOv2 ViT-B/14 (Oquab et al., 2023) supplies the $\ell_2$-normalized class-token feature for comparison with identically processed accepted anchor crops of the same layout instance.

### D.3 Scoring and Interpretation

Object presence (OP) records whether an expected object has an accepted anchor reference and a tracked mask meeting the area threshold. Layout adherence (LA) measures the fraction of the tracked mask inside the object's projected layout box. This gives a 2D containment measure, distinct from a full assessment of object geometry. Appearance consistency (AC) takes the maximum DINOv2 cosine similarity to valid anchor crops of the same instance, as in Section 4.2. In the evaluation implementation, this score is clipped to $[0, 1]$. It measures appearance agreement with the anchor references; temporal appearance drift is a separate property.

Let $\mathcal{E}_{mr}$ contain expected object–frame observations for method $m$ in room $r$. Their OP, LA, and AC scores are $p_e \in \{0, 1\}$, $\ell_e$, and $a_e$, respectively. Observations without valid anchor references or masks meeting the area threshold receive zero for all three scores and remain in the denominator. This includes an expected object with no accepted input-anchor reference, even if it appears in a generated frame. Frames with no expected objects contribute no observations to the pooled sums; a score with an empty denominator is left undefined.

With $N_m = \sum_r |\mathcal{E}_{mr}|$ observations across all rooms, the aggregate scores are

$$\mathrm{OP}_m = \frac{\sum_r \sum_{e \in \mathcal{E}_{mr}} p_e}{N_m}, \quad \mathrm{LA}_m = \frac{\sum_r \sum_{e \in \mathcal{E}_{mr}} \ell_e}{N_m}, \quad \mathrm{AC}_m = \frac{\sum_r \sum_{e \in \mathcal{E}_{mr}} a_e}{N_m}. \tag{13}$$

Each expected object–frame observation has equal weight in these averages. Methods share rooms and visibility rules, but recovered views and excluded anchors can yield different denominators.

**Five-stage analysis.** For a video with $T$ frames, a scored frame with zero-based index $f$ belongs to stage $1 + \lfloor 5f/T \rfloor$. The stages partition normalized video time into five equal intervals; the sampling and anchor-exclusion rules above still apply. Within each stage, we pool expected object–frame observations across rooms using the same aggregation as above. For our 85-frame videos, the five stages contain 4, 3, 4, 3, and 4 scored frames, respectively.

# E Ablation Protocols

## E.1 Number of Anchors

The 81-frame videos in Table 2(a) use one 81-frame segment for $K = 2$, two 41-frame segments for $K = 3$, and three segments of 29, 29, and 25 frames for $K = 4$. After omitting the first frame of each segment after the first, the total length is

$$T = 1 + \sum_{k=1}^{K-1} (T_{s,k} - 1), \tag{14}$$

where $T_{s,k}$ is the length of segment $k$. For equal-length segments, this reduces to the expression in Section 3.2. The four-anchor ablation uses a shorter final segment and yields $29 + 28 + 24 = 81$ frames, whereas the main configuration uses three 29-frame segments and yields $29 + 28 + 28 = 85$ frames.

**Generation and scoring settings.** The $K = 2, 3, 4$ configurations select candidate views $(0, 15)$, $(0, 8, 15)$, and $(0, 5, 10, 15)$, respectively, with zero-based indexing. The zero-based candidate indices $(0, 5, 10, 15)$ correspond to the one-based indices $(1, 6, 11, 16)$ in Section 3.1. All use Wan2.1-FLF2V-14B-720P at $512 \times 512$ and 16 FPS, with 20 sampling steps per segment, guidance scale 5.0, timestep shift 16.0, and seed 43. Their anchor-frame indices are $(0, 80)$, $(0, 40, 80)$, and $(0, 28, 56, 80)$, respectively. Evaluation samples every fourth frame and excludes the union $(0, 28, 40, 56, 80)$ for all three configurations, yielding the same 16 scored frames. Each configuration uses its own input anchors for camera alignment and instance references as described in Section D.

## E.2 Anchor-Generation Configurations

Table 2(b) uses 51 rooms from the 52-room pool, excluding one room used for preliminary pipeline checks. Both configurations share the initial reference image and target cameras. The independent configuration uses the FLUX.1-dev image generator with its pretrained wireframe LoRA, whereas the joint configuration uses SpatialGen. This comparison evaluates the two complete anchor-generation configurations. Anchor Consistency is computed for layout objects visible in both views of each adjacent pair $(0, 5)$, $(5, 10)$, and $(10, 15)$, using zero-based candidate-view indices. Object crops and DINOv2 features follow Section D.2. Each observation receives the cosine similarity clipped to $[0, 1]$ if the object is matched in both views, and zero otherwise. We average these observations within each room and then across rooms. MUSIQ (Ke et al., 2021) is averaged over the three generated target anchors and then across rooms; the shared reference is excluded.

## E.3 Semantic and Geometric Conditions

For Figure 6, we zero the features encoding the semantic maps $S_m$, scene coordinate maps $P_m$, or both. Zeroing is applied after VAE encoding and before concatenation into the denoiser input. The intervention is applied in both anchor-generation passes, preserving the input channel structure, reference RGB, camera poses, masks, random seed, and frozen model weights. Each variant recomputes the reprojection used by its second pass from its own first-pass output. This tests inference-time sensitivity to the encoded conditions. The shared reference image still contains layout information when both condition features are removed.

# F Confidence Intervals for Evaluation Results

We report 95% percentile confidence intervals from 10,000 room-level bootstrap resamples of the 30 evaluation rooms, using seed 2024. Each resample draws 30 rooms with replacement, retains all observations from each selected room, and recomputes the pooled numerator-to-denominator ratio defined in Section D.3. The interval endpoints are the 2.5th and 97.5th percentiles of these bootstrap estimates. These intervals quantify room-sampling uncertainty conditional on the generated outputs; they do not include variation across repeated generation seeds.

## F.1 Main Comparison

Table 4 supplements Table 1 with confidence intervals for all seven configurations. The point estimates and evaluation protocol are unchanged.

Table 4: Main-comparison scores and 95% room-bootstrap confidence intervals. Each cell gives the point estimate above its interval; values are rounded to four decimal places.

| Method | OP | LA | AC |
|---|---|---|---|
| ViewCrafter | 0.4569<br>[0.3482, 0.5695] | 0.3725<br>[0.2840, 0.4664] | 0.2724<br>[0.2073, 0.3452] |
| SEVA | 0.5689<br>[0.4798, 0.6559] | 0.5001<br>[0.4169, 0.5828] | 0.3440<br>[0.2781, 0.4092] |
| Wan2.2-5B (single image) | 0.7012<br>[0.6076, 0.7896] | 0.6316<br>[0.5426, 0.7160] | 0.5878<br>[0.5069, 0.6663] |
| Wan2.2-5B (first–last) | 0.6260<br>[0.5137, 0.7269] | 0.2910<br>[0.2234, 0.3565] | 0.4774<br>[0.3882, 0.5643] |
| Wan2.2-A14B (single image) | 0.6545<br>[0.5675, 0.7348] | 0.6037<br>[0.5148, 0.6862] | 0.5174<br>[0.4422, 0.5870] |
| Wan2.2-A14B (first–last) | 0.4583<br>[0.3663, 0.5521] | 0.2443<br>[0.1750, 0.3160] | 0.3437<br>[0.2726, 0.4193] |
| Layout2Video | 0.8818<br>[0.8202, 0.9241] | 0.7738<br>[0.7195, 0.8139] | 0.6828<br>[0.6325, 0.7218] |

Table 5: Pointwise 95% room-bootstrap confidence intervals for OP across the five video stages. Interval endpoints are rounded to three decimal places.

| Method | Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 |
|---|---|---|---|---|---|
| ViewCrafter | [0.560, 0.743] | [0.278, 0.573] | [0.065, 0.323] | [0.059, 0.362] | [0.029, 0.634] |
| SEVA | [0.629, 0.795] | [0.509, 0.724] | [0.422, 0.634] | [0.343, 0.569] | [0.288, 0.474] |
| Wan2.2-5B (single image) | [0.618, 0.784] | [0.608, 0.791] | [0.611, 0.797] | [0.599, 0.793] | [0.599, 0.793] |
| Wan2.2-5B (first–last) | [0.570, 0.759] | [0.496, 0.735] | [0.403, 0.668] | [0.455, 0.784] | [0.500, 0.785] |
| Wan2.2-A14B (single image) | [0.642, 0.804] | [0.609, 0.789] | [0.575, 0.749] | [0.526, 0.707] | [0.471, 0.645] |
| Wan2.2-A14B (first–last) | [0.404, 0.651] | [0.237, 0.518] | [0.226, 0.445] | [0.271, 0.535] | [0.517, 0.737] |
| Layout2Video | [0.817, 0.933] | [0.847, 0.947] | [0.842, 0.963] | [0.788, 0.943] | [0.725, 0.903] |

Table 6: Pointwise 95% room-bootstrap confidence intervals for LA across the five video stages. Interval endpoints are rounded to three decimal places.

| Method | Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 |
|---|---|---|---|---|---|
| ViewCrafter | [0.490, 0.669] | [0.173, 0.473] | [0.027, 0.252] | [0.016, 0.233] | [0.000, 0.367] |
| SEVA | [0.559, 0.719] | [0.421, 0.632] | [0.367, 0.572] | [0.303, 0.512] | [0.239, 0.410] |
| Wan2.2-5B (single image) | [0.551, 0.714] | [0.542, 0.721] | [0.547, 0.727] | [0.534, 0.717] | [0.528, 0.713] |
| Wan2.2-5B (first–last) | [0.289, 0.451] | [0.144, 0.266] | [0.111, 0.246] | [0.192, 0.417] | [0.258, 0.516] |
| Wan2.2-A14B (single image) | [0.589, 0.752] | [0.557, 0.741] | [0.512, 0.695] | [0.476, 0.658] | [0.424, 0.601] |
| Wan2.2-A14B (first–last) | [0.215, 0.402] | [0.088, 0.269] | [0.061, 0.212] | [0.116, 0.334] | [0.287, 0.473] |
| Layout2Video | [0.719, 0.841] | [0.761, 0.862] | [0.721, 0.833] | [0.655, 0.818] | [0.622, 0.797] |

Table 7: Pointwise 95% room-bootstrap confidence intervals for AC across the five video stages. Interval endpoints are rounded to three decimal places.

| Method | Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 |
|---|---|---|---|---|---|
| ViewCrafter | [0.362, 0.494] | [0.133, 0.280] | [0.028, 0.150] | [0.021, 0.176] | [0.011, 0.316] |
| SEVA | [0.460, 0.599] | [0.282, 0.440] | [0.203, 0.363] | [0.148, 0.299] | [0.123, 0.232] |
| Wan2.2-5B (single image) | [0.550, 0.704] | [0.522, 0.683] | [0.507, 0.670] | [0.481, 0.649] | [0.472, 0.636] |
| Wan2.2-5B (first–last) | [0.490, 0.663] | [0.363, 0.566] | [0.252, 0.472] | [0.279, 0.537] | [0.360, 0.580] |
| Wan2.2-A14B (single image) | [0.560, 0.713] | [0.486, 0.650] | [0.437, 0.596] | [0.383, 0.527] | [0.327, 0.466] |
| Wan2.2-A14B (first–last) | [0.336, 0.556] | [0.147, 0.367] | [0.123, 0.278] | [0.166, 0.336] | [0.424, 0.598] |
| Layout2Video | [0.638, 0.749] | [0.672, 0.768] | [0.628, 0.734] | [0.599, 0.728] | [0.535, 0.679] |

### F.2 Five-Stage Comparison

Tables 5–7 give the intervals corresponding to Figure 5. Stages and aggregation follow Section D.3. The intervals are pointwise, rather than simultaneous confidence bands over the five stages, and are reported as uncertainty summaries rather than pairwise significance tests.