

# WorldScore: A Unified Evaluation Benchmark for World Generation

## Supplementary Material

### A. Additional Details on World Specification

We provide additional details on world specification pre-processing  $w_{\text{proc}}$  in Eq. 1. We evaluate models across 3D scene generation, 4D scene generation, and video generation, each with distinct input requirements. For instance, 3D/4D scene generation models [90, 91] accept precise camera poses as input, whereas video generation models do not. Also, among these models, some are T2V models [14, 41], which rely solely on text-based control, while others are I2V models [20, 58, 86, 90, 91], which accept image control signals. To accommodate these variations,  $w_{\text{proc}}$  ensures that each model receives inputs in its appropriate format.

Specifically,  $w_{\text{proc}}$  standardizes the inputs as follows:

- **Reference image I:** The image for current scene  $\mathcal{C}$  is center-cropped and resized to match the resolution required by each model (see Table S1 for the specific resolutions). This serves as both a visual style reference and a necessary input for I2V models. Notably, T2V models are treated as I2V models that ignore image-based control signals.
- **Layout  $\mathcal{L}$ :** The world specification module generates a predefined precise camera trajectory  $\mathcal{T}$  (which serve as ground truth for camera controllability) and corresponding textual descriptions  $\mathcal{Y}$  (e.g., “camera moves left”) as world layout  $\mathcal{L}$ .  $w_{\text{proc}}$  gives models that accept explicit camera control signals the transformed camera poses  $\mathcal{T}'$ , ensuring alignment across different camera types, while models without explicit camera control receive textual descriptions  $\mathcal{Y}$  instead.
- **Next-scene prompt  $\mathcal{N}$ :** For 3D/4D models which all accept camera matrices as input,  $w_{\text{proc}}$  does not adapt the prompt  $\mathcal{N}$ . For video models that do not accept camera matrices as input,  $w_{\text{proc}}$  processes the next-scene prompt  $\mathcal{N}$  by adding camera movement text to it.

### B. Additional Details on Dataset Curation

#### B.1. Image Filtering

To construct a high-quality and diverse image dataset as our starting current scene images, we source from both existing datasets and supplement them with Unsplash [7]. Existing scene datasets [8, 38, 57, 62, 67, 69, 74, 98] (Table S2) are designed for scene understanding [8, 57, 69, 74]. Many of the images in these datasets are not suitable as the current scene image, as they may contain excessive redundancy, unusual viewpoints, and narrow-angle perspectives. Therefore, we apply filtering based on several criteria (see Figure S1 for visualization of the filtering):

**Quality.** We employ CLIP-IQA [75] and CLIP Aesthetic [63] predictors to filter out images with poor visual quality.

**Perspective.** To ensure appropriate viewpoint composition, we utilize the Perspective Fields [28] to model the local perspective properties (e.g., yaw, pitch, and FOV). We filter out images with extreme roll or pitch angles and those with a narrow FOV, aiming to retain open-angle, front-facing perspectives.

**Similarity.** Since many datasets contain redundant sequential images, we use CLIPSIM [53] to remove visually similar images.

**Brightness.** To exclude overly dark images, we compute image brightness and filter out those below a predefined threshold.

**Human Judgment.** Finally, we conduct a manual review to refine the selection, ensuring the curated images align with human perception and the intended use case.

#### B.2. Stylized Image Generation

After filtering and categorization, we obtain our photorealistic image dataset. Then, for each photorealistic image, we generate a stylized counterpart image using a text-to-image model [55].

**Predefined style sets.** To ensure diversity of visual style, we curate a predefined style set by referencing visual art history [59], supplemented with commonly used visual styles from SDXL [64]. Our final selection includes: *anime*, *cyberpunk*, *Chinese ink painting*, *ukiyo-e*, *impressionism*, *post-impressionism*, and *minecraft*. See example images in Figure S2.

#### B.3. Next-Scene Text Prompts Curation

We use GPT-4o [51] for scene description generation, with distinct approaches for static and dynamic scenarios. Specifically, for the static world generation task, we employ an auto-regressive process using the following task specification  $\mathcal{J}_{\text{static}}$  for system calls:

“You are an intelligent scene generator. Imaging you are wondering through a sequence of scenes, please tell me what sequentially next scene would you likely to see? You need to generate 1 to 3 most prominent entities in the scene. The scenes are sequentially interconnected, and the entities within the scenes are adapted to match and fit with the scenes. You also have to generate a brief scene description. If needed, you can make reasonable guesses. Please ensure the output is in the following JSON format: { ‘Entities’: [‘entity-1’, ...], ‘Prompt’: ‘scene description’}.”

Method	Version	Ability	Resolution	Length (s)	FPS	Open Source	Speed <sup>†</sup>	Camera <sup>§</sup>
Gen-3 [58]	24.07.01	I2V	1280×768	10	24	✗	1 min	✗
Hailuo [20]	24.08.31	I2V	1072×720	5.6	25	✗	3.5 min	✗
DynamiCrafter [84]	23.10.18	I2V	1024×576	5	10	✓	2.5 min	✗
VideoCrafter1 [9]	23.10.30	T2V	1024×576	2	8	✓	7 min	✗
		I2V	512×320	2	8	✓	2 min	✗
VideoCrafter2 [10]	24.01.17	T2V	512×320	2	8	✓	2 min	✗
		T2V-Turbo [41]	512×320	3	16	✓	5 s	✗
EasyAnimate [86]	24.05.29	I2V	1344×768	6	8	✓	16 min	✗
		T2V	720×480	6	8	✓	2.4 min	✗
CogVideoX [88]	24.08.12	I2V	720×480	6	8	✓	2.4 min	✗
		T2V	720×480	6	8	✓	2.4 min	✗
Allegro [97]	24.10.20	I2V	1280×720	6	15	✓	0.5 h	✗
Vchitect-2.0 [97]	25.01.14	T2V	768×432	5	8	✓	2.8 min	✗
LTX-Video [19]	25.05.05	I2V	768×512	4	30	✓	2.4 min	✗
SceneScape [16]	23.02.02	T2V	512×512	5	10	✓	11.4 min	✓
Text2room [24]	23.03.21	I2V	512×512	5	10	✓	12.4 min	✓
LucidDreamer [11]	23.11.22	I2V	512×512	5	10	✓	6.4 min	✓
WonderJourney [90]	23.12.06	I2V	512×512	5	10	✓	6.3 min	✓
InvisibleStitch [12]	24.04.30	I2V	512×512	5	10	✓	2.3 min	✓
WonderWorld [91]	24.06.13	I2V	512×512	5	10	✓	10 s	✓
4D-fy [3]*	23.11.29	T2V	256×256	4	30	✓	3 h	✓

Table S1. **Further details of the world generation models in our benchmark.** <sup>†</sup> The reported values indicate the average generation time per instance. All generations were conducted on H100 and L40S GPUs. <sup>§</sup> This indicates whether the model accepts precise camera poses as input. \* For 4D-fy, it takes about 20 hours for each generation, so we decrease the iteration steps to save time. While these models use different output resolutions and aspect ratios, our validation shows that WorldScore metrics are robust against these differences (Sec. D).



Figure S1. **Filtering.** We apply the filtering based on several criteria to remove undesired images. Besides automatic metrics, we also apply a final manual inspection to remove infeasible world generation starting scenes such as the mid-air city image in the 4th column.

For the dynamic world generation task, we use the task specification  $\mathcal{T}_{\text{dynamic}}$  for single system call:

*“You are an intelligent motion dreamer, capable of identifying the objects within an image that can exhibit dynamic motion. I will provide you with an image, and your task is to identify the most prominent object(s) that have the potential for dynamic movement. You also have to briefly describe how the object(s) move. If needed, you can make reasonable*

*guesses. Please ensure the output is in the following JSON format: {‘Objects’: [‘object\_1’, ...], ‘Prompt’: ‘description of how the object(s) move’}.”*

We show an example of generated next-scene prompts in Table S3.





Figure S2. **Examples of stylized images.** Our predefined style set contain 7 different visual art styles.

## C. Additional Details on Metrics

### C.1. Camera Controllability

As formulated in Eq. 3, we combine  $e_\theta$  and  $e_t$  with geometric mean to calculate the camera error. Specifically, we estimate

Scene Type	Dataset	Image Type	Res.	# Images
Indoor	InterviorVerse [98]	Synthetic	640×480	50,000
	Hypersim [57]	Synthetic	1024×768	77,400
	SUN-RGBD [69]	Real	640×480	10,000
	Matterport3D [8]	Real	1280×1024	194,400
	DIODE-indoor [74]	Real	1024×768	9,052
	ETH3D-indoor [62]	Real	6214×4138	597
Outdoor	LHQ [67]	Real	1024×1024	90,000
	EDEN [38]	Synthetic	640×480	300,000
	Argoverse-HD [42]	Real	1920×1200	70,000
	DIODE-outdoor [74]	Real	1024×768	18,206
	ETH3D-outdoor [62]	Real	6214×4138	301

Table S2. Statistics of the scene datasets we source from.

### Generated Next-Scene Prompt $\mathcal{N}$

#### Static world generation

##### # Scene 1

{“Entities”: [“yellow armchair”, “bookshelf”], “Prompt”: “A Cozy Reading Nook with a plush Yellow Armchair surrounded by a towering Bookshelf filled with books.”}

##### # Scene 2

{“Entities”: [“potted plants”], “Prompt”: “A serene Tranquil Garden Patio featuring a cozy Yellow Armchair surrounded by lush Potted Plants gently swaying in a soft, breezy atmosphere.”}

##### # Scene 3

{“Entities”: [“wooden rail”], “Prompt”: “A Rustic Balcony Retreat featuring a cozy Yellow Armchair and a classic Wooden Rail bathed in the warm glow of the setting sun.”}

#### Dynamic world generation

{“Objects”: [“windmill”, “cloud”, “sea”], “Prompt”: “The windmill blades spin in a circular motion driven by the wind, creating a consistent rotational movement. The clouds drift slowly across the sky, pushed gently by the breeze. The sea surface ripples and undulates, as small waves ripple across its surface.”}

Table S3. An example of generated next-scene prompt for static and dynamic world generation. The “prompt” in the above box is the next-scene prompt  $\mathcal{N}$ . The “entities” are the objects to detect when computing object controllability. The “objects” are used to help annotate the motion masks for computing motion accuracy.

the frame-wise camera poses using DROID-SLAM [72]. Then we compute the angular deviation between the ground

truth and the estimated camera rotations (in degrees):

$$e_\theta = \arccos\left(\frac{\text{tr}(\mathbf{R}_{\text{gt}}\mathbf{R}^T) - 1}{2}\right) \cdot \frac{180}{\pi}, \quad (\text{S1})$$

and the scale-invariant Euclidean distance between ground truth and estimated camera positions:

$$e_t = \|\mathbf{t}_{\text{gt}} - s\mathbf{t}\|_2, \quad (\text{S2})$$

where  $\mathbf{R}_{\text{gt}}, \mathbf{R} \in SO(3)$  denote the ground truth and estimated rotation matrices,  $\mathbf{t}_{\text{gt}}, \mathbf{t} \in \mathbb{R}^3$  denote the ground truth and estimated camera positions, and  $s$  denotes the least-square scale.

The final camera controllability error for a model is computed by averaging the error  $e_{\text{camera}}$  over all frames of all generated videos.

## C.2. 3D Consistency

To quantify the 3D consistency of generated videos, we use DROID-SLAM [72] to do the reconstruction and calculate the reprojection error. One key advantage of DROID-SLAM is its dense nature. Unlike sparse methods such as COLMAP [60, 61], which rely on selecting “good” feature matches while discarding the rest, DROID-SLAM employs a differentiable Dense Bundle Adjustment (DBA) layer. This layer continuously refines camera poses and dense, per-pixel depth estimates to ensure consistency with the current optical flow. By leveraging all available points, rather than focusing on partial matches, this dense approach aligns with our goal of assessing 3D consistency across the entire scene. This evaluation dimension ensures a more comprehensive understanding of the spatial coherence in generated videos.

Specifically, we calculate the reprojection error after DBA layer refinement:

$$e_{\text{reproj}} = \frac{1}{|\mathcal{V}|} \sum_{(i,j) \in \mathcal{V}} \|\mathbf{p}_{ij}^* - \Pi(\mathbf{P}_{ij})\|_2, \quad (\text{S3})$$

where  $\mathcal{V}$  denotes the valid set of co-visible points,  $\mathbf{p}_{ij}^*$  is the observed point on the ground truth image,  $\mathbf{P}_{ij}$  is the reconstructed 3D point, obtained from refined depth and camera pose,  $\|\cdot\|_2$  calculates the Euclidean distance.

## C.3. Photometric Consistency

The photometric consistency metric is to quantify the model capability to generate stable visual appearances. We estimate the optical flow between consecutive frames and compute the Average End-Point Error (AEPE). Specifically, given two consecutive frames  $A$  and  $B$ , we first track a set of center-cropped points  $\mathbf{p}_A$  from frame  $A$  to frame  $B$  using forward optical flow  $\mathcal{F}_{A \rightarrow B}$ :

$$\mathbf{p}_B = \mathbf{p}_A + \mathcal{F}_{A \rightarrow B}(\mathbf{p}_A). \quad (\text{S4})$$



	Visual Style	Scene Type	Category	# Samples
Static	Photorealistic	Indoor	Dining, Living, Passage, Public, Work	5 × 100
		Outdoor	City, Suburb, Aquatic, Terrestrial, Verdant	5 × 100
	Stylized	Indoor	Dining, Living, Passage, Public, Work	5 × 100
		Outdoor	City, Suburb, Aquatic, Terrestrial, Verdant	5 × 100
Dynamic	Visual Style	Motion Type		# Samples
	Photorealistic	Articulated, Deformable, Fluid, Rigid, Multi-Motion		5 × 100
	Stylized	Articulated, Deformable, Fluid, Rigid, Multi-Motion		5 × 100
# Total Samples				3000

Table S4. **Dataset Statistics.** We curate a dataset of 3000 test samples that span diverse worlds: static and dynamic, photorealistic and stylized, indoor and outdoor. The static subset is further divided into 5 indoor and outdoor scene categories, while the dynamic subset is categorized by 5 motion types.

We then track the same points back from frame  $B$  to frame  $A$  using backward optical flow  $\mathcal{F}_{B \rightarrow A}$ :

$$\mathbf{p}'_A = \mathbf{p}_B + \mathcal{F}_{B \rightarrow A}(\mathbf{p}_B). \quad (\text{S5})$$

Ideally, if the object remains photometrically consistent, the tracked points should return to their original locations, *i.e.*,  $\mathbf{p}'_A \approx \mathbf{p}_A$ . we quantify the deviation using the AEPE:

$$e_{\text{photometric}} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{p}_{A,i} - \mathbf{p}'_{A,i}\|_2, \quad (\text{S6})$$

where  $N$  is the number of sampled points. A higher AEPE indicates greater photometric inconsistency, signaling anomalies such as identity shifts, texture flickering, or object disappearances. Finally, the photometric consistency error is computed by averaging  $e_{\text{photometric}}$  over all consecutive frame pairs of all generated videos.

#### C.4. Subjective Quality

Numerous trained image quality assessment metrics exist, such as CLIP-Aesthetic [63] and QAlign-Aesthetic [82], which focus on factors like layout composition, color harmony, realism, and artistic appeal. Additionally, image quality predictors like MUSIQ [35] and CLIP-IQA [75] evaluate distortions such as overexposure, noise, and blur.

Our goal is to use automatic metrics that align well with human perception to evaluate the subjective quality of generated scenes. To identify the (combination of) best subjective quality predictors, we systematically conduct a human preference study to pick the one that best matches human perception on world generation quality. We find that the combination (arithmetic mean) of CLIP-IQA+ [75] and CLIP Aesthetic [63] works the best. We show more details in Sec. D.

#### C.5. Motion Accuracy

We assess whether motion occurs in the intended regions by:

$$s_{\text{motion-acc}} = \max(\mathbf{F} \odot \mathbf{M}) - \max(\mathbf{F} \odot \bar{\mathbf{M}}), \quad (\text{S7})$$

where  $\mathbf{F} \in \mathbb{R}^{H \times W}$  denotes the magnitude of optical flow between a pair of consecutive frames in the generated video  $\mathbf{V}$  estimated by SEA-RAFT [79],  $\mathbf{M} \in \{0, 1\}^{H \times W}$  denotes the segmentation masks at the former frame which has 1 at the pixels of dynamic objects, and the  $\max$  operator picks the maximum value among all the entries of a matrix. We track the mask of dynamic objects  $\mathbf{M}$  using SAM2 [54], where the first-frame segmentation masks are provided in our dataset. The final motion accuracy score is computed by averaging  $s_{\text{motion-acc}}$  across all pairs of consecutive frames of all generated videos.

#### C.6. Motion Magnitude

Some models take a “conservative” approach, generating only subtle motion. While the output appears visually smooth and high-quality, the motion is often minimal and uninteresting. Some models even produce near-static videos despite prompts explicitly describing motion. We measure this with  $s_{\text{motion-mag}}$ , defined as the median value of all the entries of  $\mathbf{F}$ , and the final motion magnitude metric is the average of  $s_{\text{motion-mag}}$  across all pairs of consecutive frames of all generated videos.

#### C.7. Motion Smoothness

We leverage the motion priors from a standard video frame interpolation models [93] to evaluate the smoothness of generated motion. Specifically, given a generated video consisting of frames  $\{\mathbf{f}_0, \mathbf{f}_1, \mathbf{f}_2, \dots\}$ , we drop the odd-indexed frames  $\{\mathbf{f}_1, \mathbf{f}_3, \dots\}$  to obtain a lower frame rate video, and

then we use video frame interpolation to infer the dropped frames. Finally, we compute the mean squared error, SSIM [80], and LPIPS [94] between the reconstructed frames and the original dropped frames. After each metric score is computed and normalized (Supp. C.9), we average them to get the motion smoothness metric.

### C.8. Empirical Bounds

In this section, we discuss how we calculate the empirical bounds for each evaluation dimension, which will be used for linear normalization in Supp. C.9.

**Empirical bounds for camera controllability.** Since the camera controllability metric calculates the deviation between the ground truth and estimated camera poses, the empirical minimum is naturally 0, which also represents the theoretical lower bound. To approximate the highest achievable values, we use a sequence of fixed cameras as a baseline. This effectively penalizes poorly performing world generation that fails to exhibit any camera movement.

**Empirical bounds for object controllability.** Since we evaluate object controllability using the object detection rate, the empirical minimum and maximum are naturally 0 and 100%, respectively, which also represent the theoretical bounds.

**Empirical bounds for 3D consistency, style consistency, and photometric consistency.** To establish empirical bounds for these frame-wise metrics, we randomly sample image pairs from our dataset and generate videos by interpolating intermediate frames using a video frame interpolation model [93]. This serves as a baseline exhibiting significant style shifts, low 3D consistency, and poor photometric stability. We define this baseline as empirical maximum for all three metrics, while the empirical minimum for each is set to 0, which is also theoretical minimum.

**Empirical bounds for motion smoothness.** To determine empirical values for *motion smoothness*, we leverage high-quality real-world videos. Given that most world generation models produce 3-10 second videos, we retrieve comparable video clips from OpenVid-1M [50], a large-scale, high-quality video dataset. Specifically, for each prompt in our benchmark, we retrieve the top five OpenVid-1M videos with the highest semantic similarity using CLIP-based text feature matching. Only 3-10 second clips are considered to ensure consistency with the length of generated videos.

Then, we use the retrieved videos as a reference. We manually drop the odd frames and apply bilinear interpolation to reconstruct them. This serves as a baseline, where the resulting interpolated videos represent the “empirical worst” (empirical maximum for MSE and LPIPS and empirical minimum for SSIM). The “empirical best” is set to 0, indicating perfectly smooth motion.

**Empirical bounds for content alignment, subjective quality, motion accuracy, and motion magnitude.** For these

four metrics, defining appropriate empirical bounds is challenging. To address this, we apply z-score rescaling, setting the empirical best and worst values so that the performance of selected models falls within the 25 to 75 range. This approach enhances differentiation and ensures a more reliable evaluation.

### C.9. Score Normalization and Mapping

The detailed formulation for score normalization and mapping is as follows:

$$s^{\text{norm}} = \begin{cases} \left\langle \frac{s - b^{\min}}{b^{\max} - b^{\min}} \right\rangle, & \text{if higher better,} \\ \left\langle 1 - \frac{s - b^{\min}}{b^{\max} - b^{\min}} \right\rangle, & \text{if lower better,} \end{cases} \quad (\text{S8})$$

where  $s$  denotes the raw value of a given metric,  $b^{\min}$  and  $b^{\max}$  denote the empirical bounds of the metric, and  $\langle \cdot \rangle$  denotes the clip function, making sure the normalized score  $s^{\text{norm}}$  is within the range  $[0, 1]$ , where a higher value corresponds to better performance.

## D. Validation with Human Preference

We validate the WorldScore metrics by human preference study for three purposes: Firstly, we use human preference to select the best combination of subjective quality metrics (e.g., image quality assessment metrics and aesthetic metrics) to form a single “subjective quality”. Secondly, we use human preference to validate other WorldScore metrics. Lastly, we measure how robust are the metrics to different resolutions and aspect ratios. In particular, we use the following agreement score.

**Human preference agreement score.** To measure how well each metric aligns with human preferences, we adopted a probabilistic agreement score. Given a video pair (A, B), a participant is forced to choose one video that appears to have higher subjective quality to them, a.k.a. 2-alternative forced choice (2AFC). We denote the portion of all participants who preferred A as  $p$ , therefore the portion of all participants who preferred B is  $1 - p$ . Then, consider an automatic assessment metric  $m$ :

- If the metric  $m$  assigned a higher score to A, i.e.,  $\text{score}_m(A) > \text{score}_m(B)$ , then the agreement score for this pair (A, B) is  $p$ .
- If the metric  $m$  assigned a higher score to B, i.e.,  $\text{score}_m(A) < \text{score}_m(B)$ , then the agreement score for this pair (A, B) is  $1 - p$ .
- If the metric assigned equal scores to A and B, then the agreement score was set to 0.5.

The final agreement score for each metric was obtained by averaging the agreement scores across all human-rated pairs.

To prepare the pairs of videos for human participants, we randomly sampled videos generated from CogVideoX-12V,

VideoCrafter1-I2v, DynamiCrafter, WonderJourney, and InvisibleStitch. Each comparison consisted of a pair of videos from different models. We recruited 400 participants for the human study.

Note that in our human preference study, we only use a single question, asking the participant “which video has higher quality”. While there are possibly different dimensions of subjective quality such as aesthetic quality and perceptual quality, our preliminary human preference study indicates that general human raters often struggle to differentiate between specific dimensions, yielding a very high correlation between aesthetic quality and perceptual quality. Therefore, we only use a single question.

**Agreement results on subjective quality.** We show the agreement results in Table S5. Since the combination (arithmetic mean) of CLIP-IQA+ [75] and CLIP Aesthetic [63] metrics yield the highest agreement, we use this combination to compute our subjective quality.

**Agreement results on other metrics.** To validate other metrics, we divide them into different score buckets, i.e.,  $90 \pm 5$ ,  $60 \pm 5$ , and  $30 \pm 5$ ; and then we compare between buckets. We show results in Table S6. The 2AFC results show that our metrics align well with human perception, so that a higher score (both “90 over 60” and “60 over 30”) consistently correlate to a higher human preference.

**Robustness against different resolutions and aspect ratios.** We validate if the metrics are robust to different resolutions and aspect ratios because models vary in these aspects. We use the videos generated by the highest-resolution model (EasyAnimate,  $1344 \times 768$ ) and apply center-cropping and resizing to create a version with small resolution ( $256 \times 256$ ). We evaluate both versions and show results in Table S7. The differences in all metrics are very small ( $\leq 0.83$ ), suggesting that our metrics are robust to these differences.

## E. Further Visualization

Our WorldScore metrics provide a comprehensive assessment by decomposing the broad concept of “world generation capability” into 10 independent dimensions. The typical examples for each metric are presented in Figure S3 and Figure S4. Each row showcases the evaluation of a metric on two generated results, highlighting how WorldScore metrics effectively differentiate model performance.

We show performances of selected models on WorldScore-Dynamic in Figure S5 and WorldScore-Static in Figure S6. Figure S5 highlights the challenges that current video generation models face, with significant variations across different dimensions. Notably, all video generation models (e.g., Hailuo, VideoCrafter1-I2V, EasyAnimate, T2V-Turbo) exhibit very low *camera controllability*, indicating difficulty in following predefined camera trajectories. Additionally, models (e.g., T2V-Turbo) that perform well in

Metric	Correlation
CLIP-IQA	0.596
CLIP-IQA+	0.602
QAlign Quality	0.581
QAlign Video Quality	0.571
MUSIQ	0.530
CLIP Aesthetic	0.628
QAlign Aesthetic	0.479
QAlign Video Aesthetic	0.556
CLIP-IQA+ & QAlign Quality	0.582
CLIP Aesthetic & QAlign Video Aesthetic	0.629
CLIP-IQA+ & CLIP Aesthetic	<b>0.637</b>
Upper Bound	0.772

Table S5. **Agreement of automatic assessment metrics with human preference.** The upper bound is the highest possible agreement score when a metric always agrees with the majority vote for every 2AFC pair.

	Cam Ctrl	Obj Ctrl	3D Consist	Photo Consist	Motion Mag
$60 \pm 5$ over $30 \pm 5$	71.2%	96.3%	91.7%	91.6%	91.8%
$90 \pm 5$ over $60 \pm 5$	73.5%	87.7%	97.3%	95.1%	76.2%

Table S6. 2AFC on WorldScore metrics with score difference 30.

Res.	Cam Ctrl	Obj Ctrl	Content Align	3D Consist	Photo Consist	Style Consist	Subject Qual	Motion Acc	Motion Mag	Motion Smooth
1344x768	25.72	54.50	49.81	67.29	46.65	73.05	49.66	75.00	37.76	40.32
256x256	25.69	53.78	50.32	67.41	47.06	73.88	48.99	74.89	36.90	39.62

Table S7. **Robustness** to resolution and aspect ratio differences.

*motion magnitude* tend to struggle with *motion smoothness*, suggesting a trade-off between large movements and temporal stability.

In Figure S6, the evaluation of static world generation shows that 3D scene generation models (e.g., WonderWorld) achieve high *camera controllability*, *3D consistency* and *photometric consistency*. However, they may struggle in *subjective quality*, indicating that while they excel in maintaining geometric and photometric coherence, they may generate less visually appealing results.






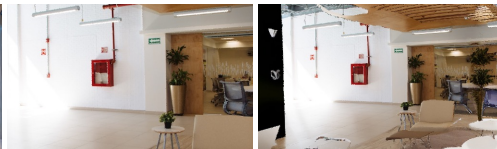
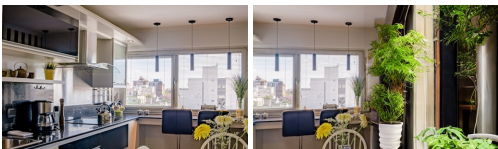

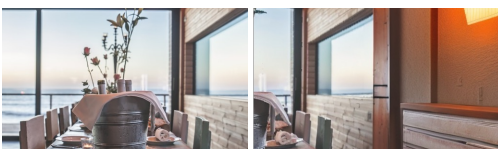
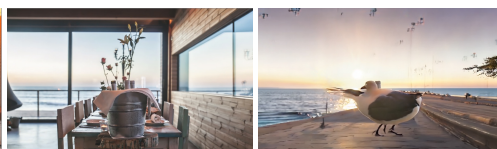
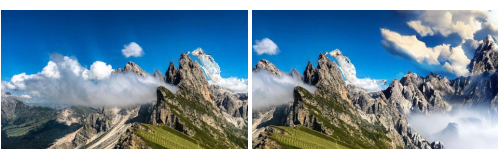
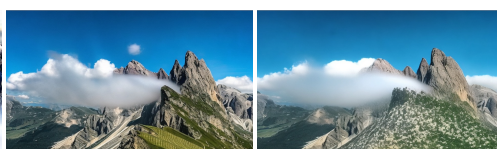
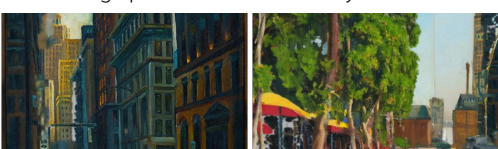
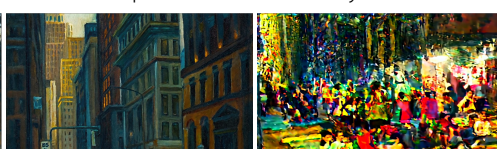


Next scene	Good examples		Bad examples	
"Camera pans right"				
	High camera controllability: 99.96		Low camera controllability: 0.00	
"Seating area, plants, cityview, chairs"				
	High object controllability: 100.00		Low object controllability: 25.00	
"Balcony featuring plant"				
	High content alignment: 71.13		Low content alignment: 0.00	
"Seaside dining, seagull"				
	High 3D consistency: 92.88		Low 3D consistency: 0.00	
"Peaks, clouds"				
	High photometric consistency: 94.28		Low photometric consistency: 11.95	
"Urban streetview"				
	High style consistency: 84.80		Low style consistency: 0.00	
"A bright modern kitchen"				
	High subjective quality: 100.00		Low subjective quality: 25.27	

Figure S3. **Typical examples from controllability and quality aspects.** Each row showcases the evaluation of a metric on two generated results, where the good example is shown on the left, and the bad example is shown on the right.

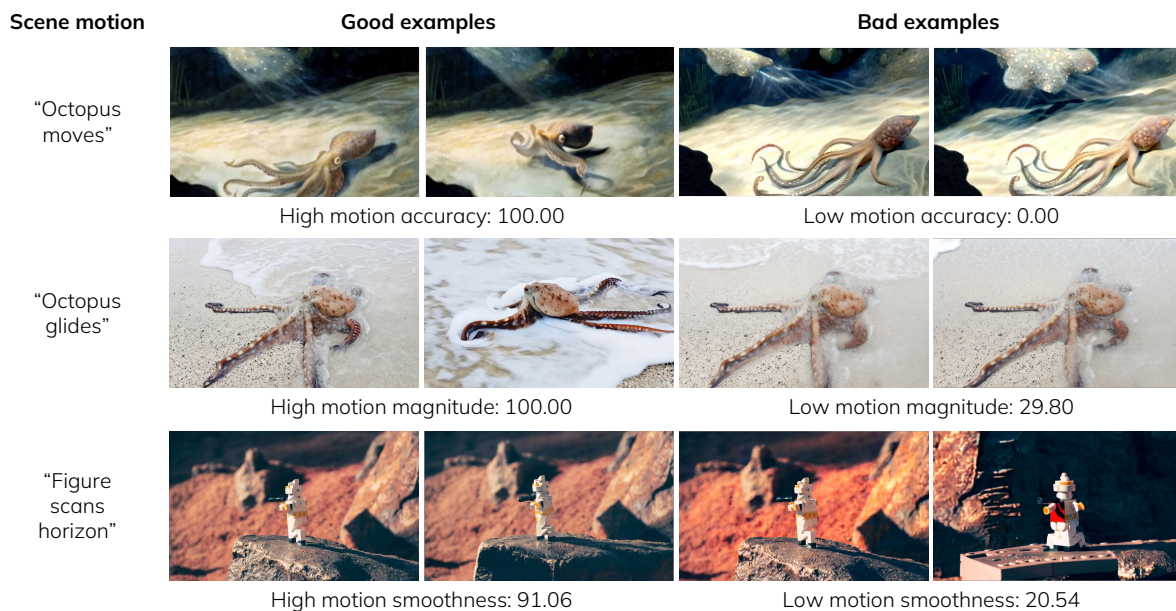


Figure S4. **Typical examples from dynamics aspect.** Each row showcases the evaluation of a metric on two generated results, where the good example is shown on the left, and the bad example is shown on the right.

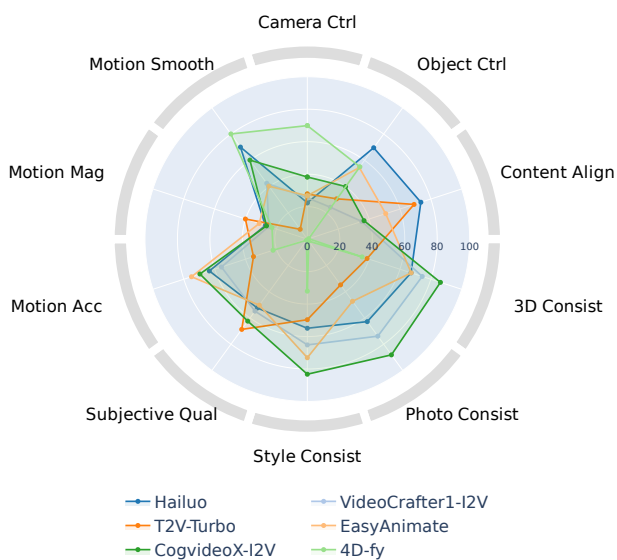


Figure S5. Evaluation results of WorldScore-Dynamic on selected models.

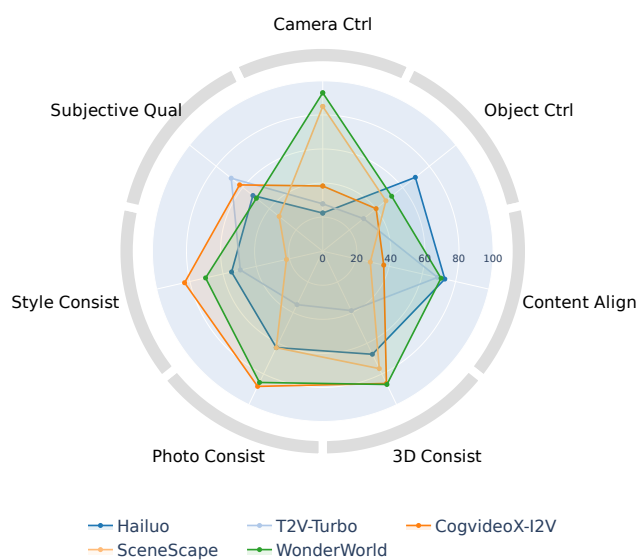


Figure S6. Evaluation results of WorldScore-Static on selected models