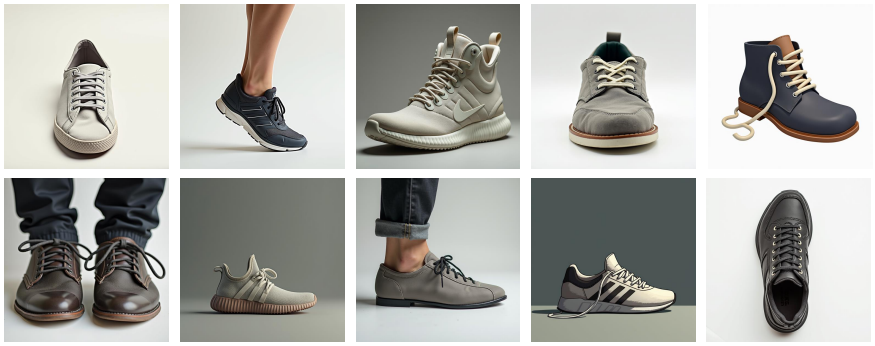
Rethinking Prompt Design for Inference-time Scaling in Text-to-Visual Generation

Supplementary Material

A. Additional Analysis

A.1. Qualitative Examples of Common Failures

We present qualitative examples of the identified common failure patterns for text-to-image generation in Figure 7 and for text-to-video generation in Figure 8.



Original prompt
: "A shoe with no laces, standing alone"

Decomposed elements
1.There is a shoe (core)
2.The shoe has no laces (core)
3.The shoe is standing alone (core)

Common failure patterns
: 2. The shoe has no laces

Original prompt
: "On a wooden table, both the spoons and plates are made of wood, only the fork is not made of wood."

Decomposed elements
1.There is wooden dining table (core)
2.There are spoons present (core)
3.There are plates present (core)
4.There is a fork present (core)
5.The spoons are made of wood (core)
6.The plates are made of wood (core)
7.The fork is not made of wood (core)
8.The spoons, plates, and fork are located on the wooden dining table

Common failure patterns
: 7. The fork is not made of wood

Figure 7. **Qualitative examples of recurring misalignments** when generating multiple images from a fixed prompt, with decomposed elements and common failures identified by EFC.

Original prompt
: “The glass car window changed into a wooden car window”

Decomposed elements
1. A car window is present in the scene (core)
2.The car window initially appears to be made of glass (core)
3.The car window later appears to be made of wood (core)
4.The transformation of the car window occurs sequentially over time (core)
5.The glass car window transforms into a wooden car window (core)

Common failure patterns
: 3. The car window later appears to be made of wood,
4. The transformation of the car window occurs sequentially over time,
5. The glass car window transforms into a wooden car window

Original prompt
: “A person is turning on the desk lamp”

Decomposed elements
1.There is a person in the scene (core)
2.There is a desk lamp on the desk (core)
3.The person is positioned near the desk lamp (extra)
4.The desk lamp is initially turned off (core)
5.The desk lamp has a switch that can be activated (core)
6.The person’s hand moves towards the desk lamp (core)
7.The person’s fingers interact with the lamp switch (core)
8.The desk lamp transitions from being off to on after the person interacts with the switch (core)

Common failure patterns
: 4. The desk lamp is initially turned off,
8. The desk lamp transitions from being off to on after the person interacts with the switch

Figure 8. **Qualitative examples of recurring misalignments** when generating multiple videos from a fixed prompt, with decomposed elements and common failures identified by EFC. We illustrate the first and the last frame for each generated video.

A.2. Details of Element-level Factual Correction (EFC)

We present a detailed overview of the visual verification process in our verifier, Element-level Factual Correction (EFC) in Figure 9.

Element-level factual correction. The goal of this process is to provide fine-grained and interpretable feedback on whether each part of a prompt is faithfully realized in the generated visuals. Given a prompt and its corresponding outputs (images or videos), EFC first decomposes the prompt into multiple disjoint semantic elements using a system prompt. For each element, it also constructs an open-ended question, where the element itself serves as the expected answer. Next, EFC verifies the fulfillment of these elements in the generated visuals through factual correction. Instead of relying on visual question answering, our key idea is to perform text-based comparison between the semantic elements and the visuals. To enable this, EFC first extracts captions from the generated visuals and then applies natural language inference (NLI) to classify each element as entailment, neutral, or contradiction.

Open-Ended visual probing. For elements classified as neutral, where captions are missing or ambiguous, EFC reuses the previously generated open-ended questions, queries the visual input again, and applies a second NLI step to the corresponding free-form answers, assigning a final label of either entailment or contradiction. Unlike direct QA, this procedure asks open-ended questions and compares their answers with the target elements to determine whether the expected element is present, rather than relying on yes/no responses. This removes affirmation bias inherent in binary QA and avoids providing contextual cues that may cause the verifier to rely on textual hints instead of extracting information directly from the visuals. Through this process, EFC pinpoints which parts of the prompt are faithfully represented and which are contradicted, thereby enabling accurate and interpretable fine-grained feedback for generated visuals.

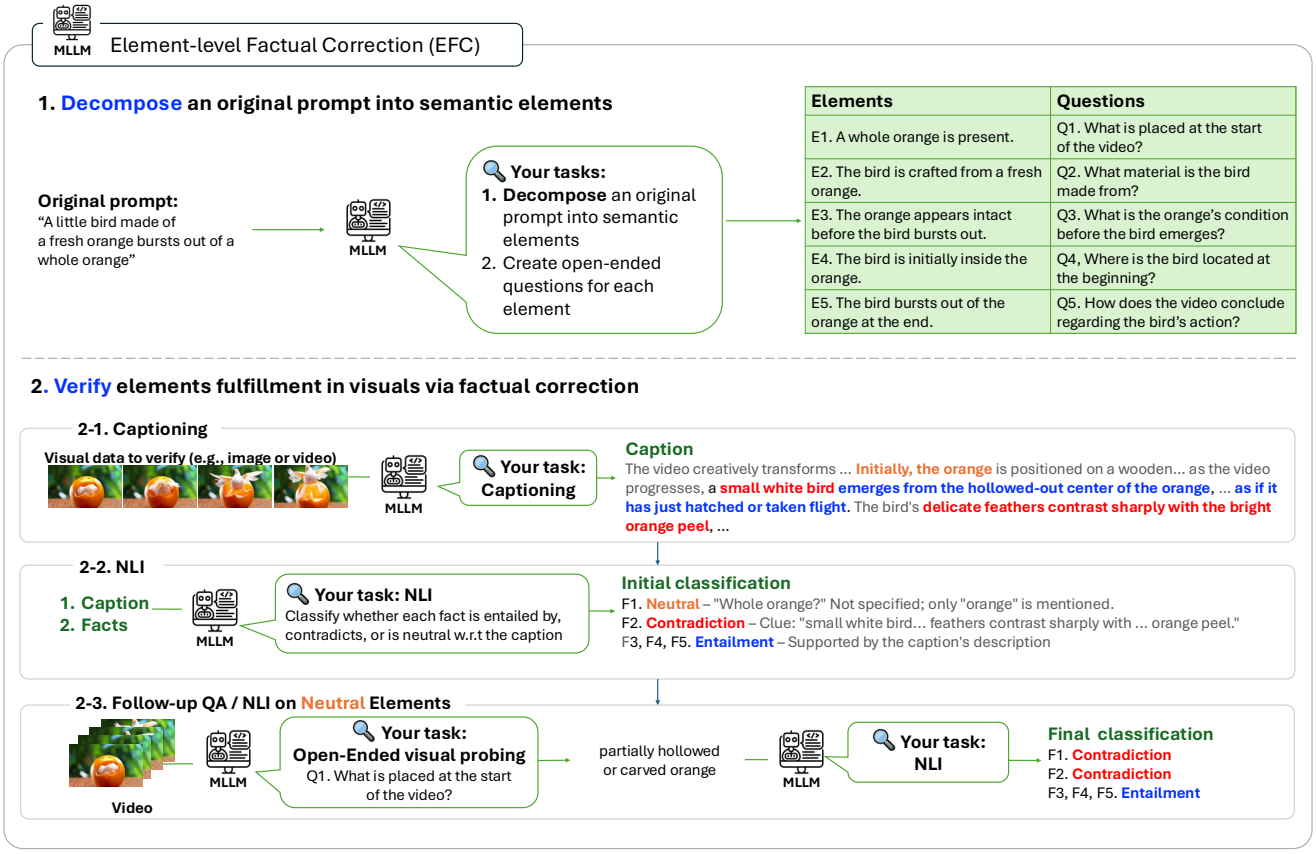


Figure 9. **Illustration of EFC.** The figure illustrates how EFC provides fine-grained, interpretable verification of prompt adherence. It first decomposes the prompt into semantic elements, then generates captions from the visuals, and applies factual correction to classify each element as entailment, neutral, or contradiction. Elements initially labeled neutral (due to missing mentions in the caption) are reevaluated to decide between entailment and contradiction. This design avoids direct QA, leading to more accurate verification.

A.3. Details on Integration Beyond BoN

This section provides additional details on the integration of our framework with visual scaling methods, complementing Section 4.2.

Figure 10. **Qualitative artifact results with RBF.** RBF alone often generates visuals where the prompt text is directly rendered on the image due to reward over-optimization, whereas combining RBF with our method substantially alleviates this issue.

Experiments on text-to-image generations. We integrate our approach with two inference-time scaling methods focused on visuals: DAS [20] and RBF [20]. Following their original experimental protocols, we use SDXL [32] for DAS and Flux.1-schnell [23] for RBF. In both settings, we generate a total of 8 samples, divided into two batches of 4. When combined with our method, the first batch of 4 samples is generated, the prompt is revised, and another 4 samples are generated, ensuring that the total number of function evaluations remains equivalent.

In addition to Table 4 and Figure 4 in the main manuscript, Figures 11 and 12 demonstrate that our integrated results achieve substantially better prompt adherence than visual scaling alone, for DAS and RBF, respectively. This indicates that advanced visual scaling methods can be further enhanced when combined with scaled prompts. It is also noteworthy that scaling visuals alone often leads to undesired outcomes caused by reward over-optimization (see Figure 10). In such cases, the model may even render the textual prompt itself, since these images achieve artificially high reward scores. For example, Figure 10 shows that RBF frequently generates images where the prompt text is printed directly. By contrast, our method mitigates this issue: the revised prompt guides the generator, while the original prompt is used only for the reward signal. This separation effectively reduces over-optimization artifacts and yields more faithful generations, even when PRIS is combined with RBF.

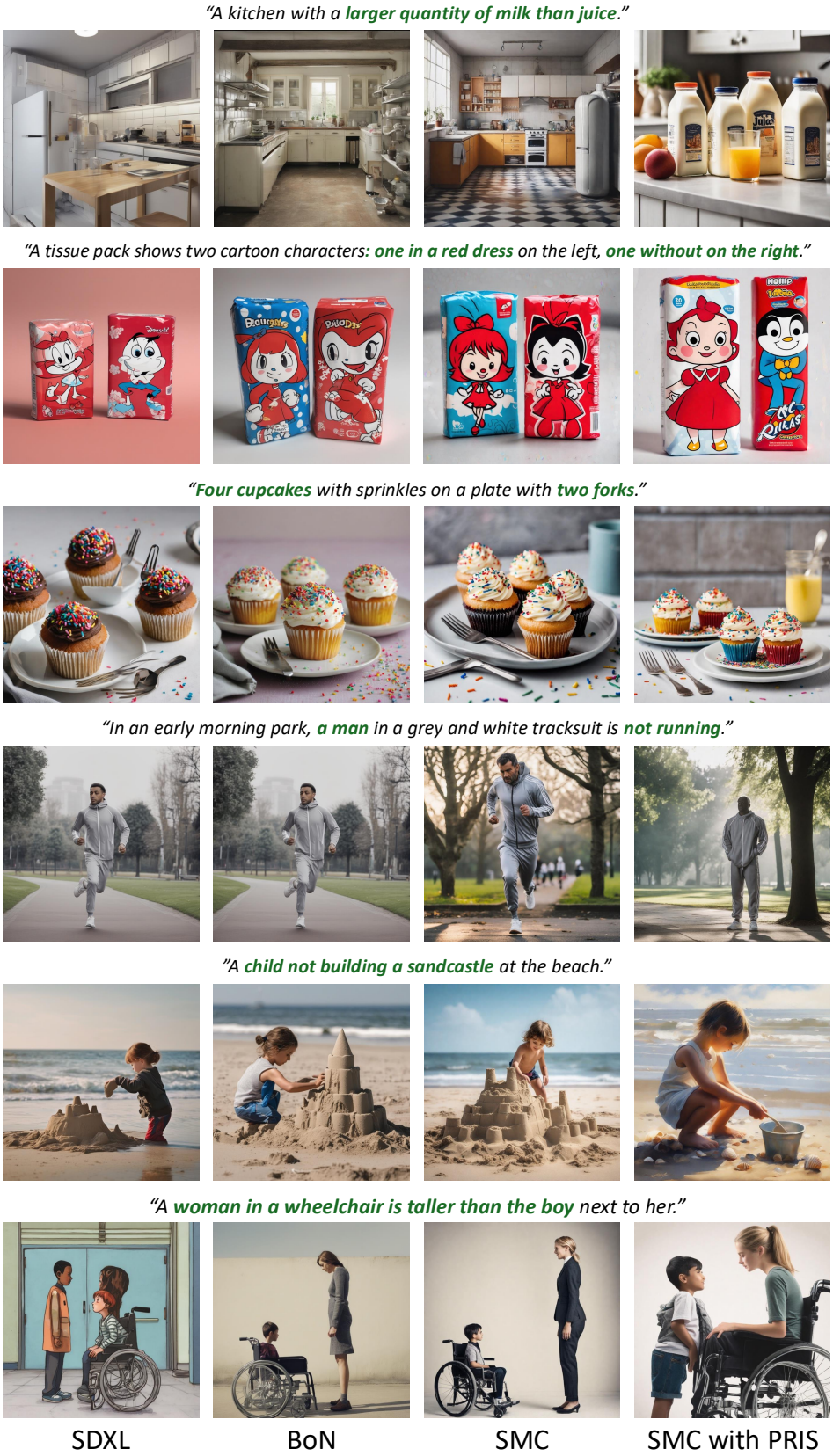


Figure 11. **Qualitative comparisons when integrating our method with SMC** under the same compute budget. Our approach more faithfully follows the prompt, effectively enabling SMC to scale visuals.

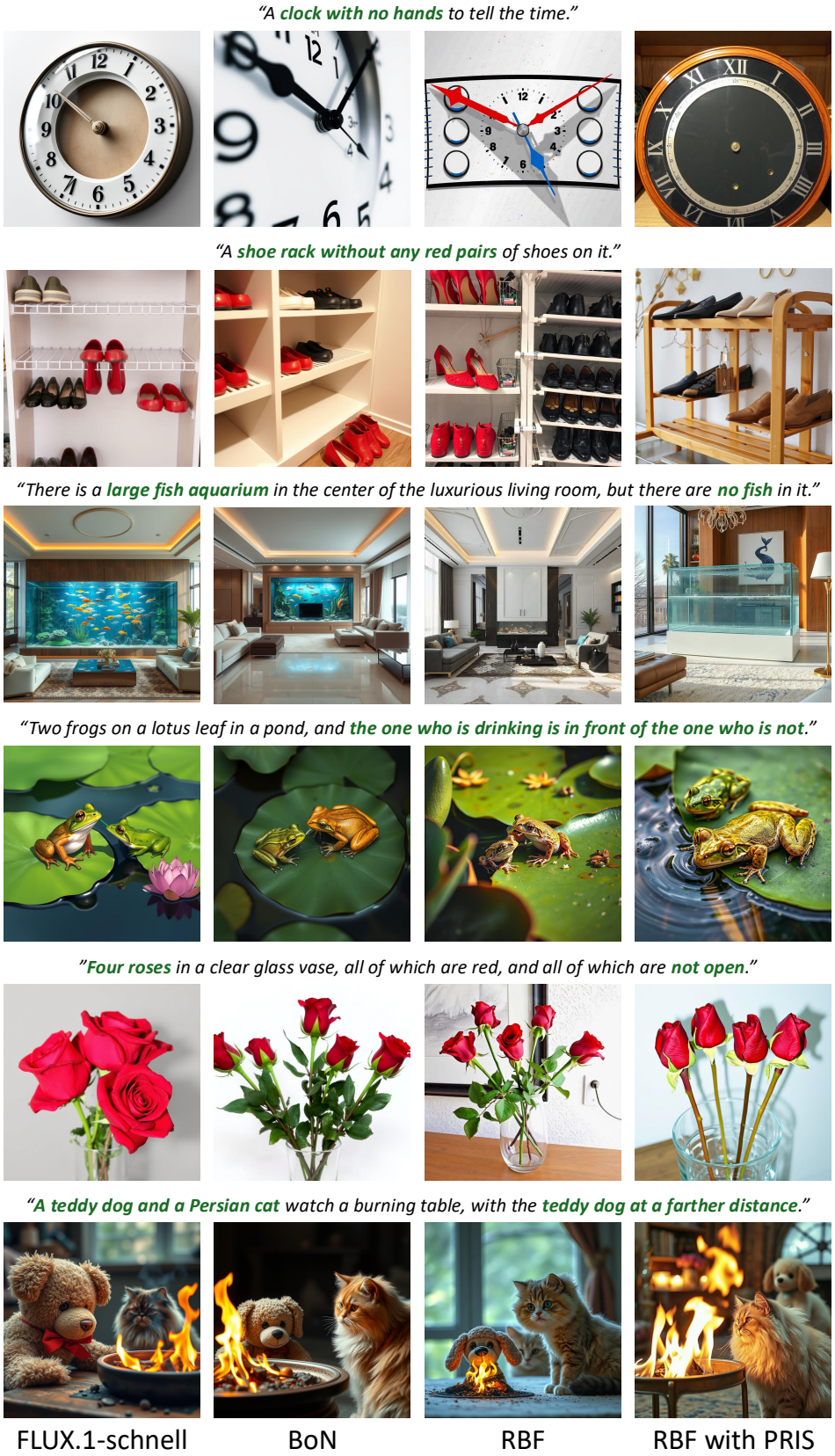


Figure 12. **Qualitative comparisons of RBF integrated with our method** under the same compute budget. Our method adheres more closely to the prompt and further improves RBF's visual scaling.

Experiments on text-to-video generations. We integrate our approach with EvoSearch [13], following its original setup on Wan2.1-1.3B. EvoSearch uses an evolution schedule of [5, 20, 30, 45] and a population schedule of [10, 5, 5, 5], totaling 2,000 NFEs. For integration, we first generate 10 samples with 50 steps (1,000 NFEs), then allocate the remaining 940 NFEs with [5, 30] for the evolution schedule and [5, 4] for the population schedule, resulting in 60 fewer NFEs than EvoSearch. As in the main manuscript, we evaluate on VBench2.0 with VideoAlign as the guiding reward.

Table 8 and Figure 13 present the quantitative and qualitative results. Unlike EvoSearch, which was evaluated on relatively simple prompts, our experiments employ more complex ones. In this setting, EvoSearch scores degrade after scaling, suggesting limited generalization to the unseen reward of VBench2.0. By contrast, when integrated with our method, it achieves improved average scores on VBench2.0.

Table 8. **Quantitative T2V results on VBench2.0, comparing EvoSearch alone with EvoSearch integrated with PRIS**. EvoSearch fails to generalize to unseen rewards, whereas integration with PRIS improves performance.

Method	Motion Rationality	Motion Order Understanding	Dynamic Attribute	**Average**
Wan2.1-1.3B	38.10	**52.87**	46.67	45.88
EvoSearch	32.14	51.72	43.33	42.20 ↓ −3.68
EvoSearch + PRIS	**53.57**	48.28	**60.00**	**53.95** ↑ +8.07

Figure 13. **Qualitative examples comparing EvoSearch and EvoSearch+PRIS**. In the first case, EvoSearch fails to change the butterfly's wing color despite scaling, whereas our method succeeds. In the second case, EvoSearch depicts the window as already open before the person attempts to open it, while our method correctly shows the window opening as the person reaches out.

A.4. Detailed Computational Time Analysis

In this section, we provide a detailed breakdown of verification and generation time, complementing Section 4.4. All measurements are conducted on a single NVIDIA H100 80GB GPU. For images, generating a single sample resolution $(1024, 1024)$ with Flux.1-dev takes on average 13 seconds, while verification with our verifier, EFC, requires 41 seconds. This implies that each verification is computationally equivalent to generating approximately three additional images. To balance this overhead, we set the number of function evaluations (NFE) to 4000 for BoN and 1000 for our method, corresponding to 40 and 10 images, respectively (with 50 sampling steps and classifier-free guidance). For videos, generating an 81-frame sequence at resolution $(480, 832)$ with Wan2.1-1.3B requires 105 seconds on average, while verification takes 100 seconds, approximately equivalent to one additional video generation. Accordingly, we set the NFE to 4000 for BoN (40 videos) and 2000 for our method (20 videos), again under 50 sampling steps with classifier-free guidance.

Our verifier is intentionally built on a pretrained MLLM without task-specific optimization, demonstrating that strong results can be achieved without additional training. Nonetheless, fine-tuning the base MLLM remains a promising direction for reducing verification time and improving efficiency.

A.5. Comparison with ReflectionFlow

We compare our method with ReflectionFlow [46], which relies on a trained reflection model to iteratively edit each generated sample. Our approach differs in three fundamental ways. First, we revise the prompt itself based on common failure patterns across samples, rather than reacting to individual errors. Second, our method is entirely training-free and does not rely on any auxiliary editing models. Third, it is applicable to any text-conditioned generator, whereas ReflectionFlow requires model-specific training. For a favorable comparison, we allocate 3840 NFEs ($N = 64$) to ReflectionFlow, following its default configuration, while ours uses only 2000 NFEs ($N = 20$). Even under this compute-advantaged setup for ReflectionFlow, our method consistently produces more accurate and semantically aligned results, as shown in Figure 14. PRIS outperforms ReflectionFlow across diverse categories, including comparison, counting, attributes, and negation, highlighting the advantage of correcting prompts based on shared failure modes rather than relying on per-sample post-hoc edits.

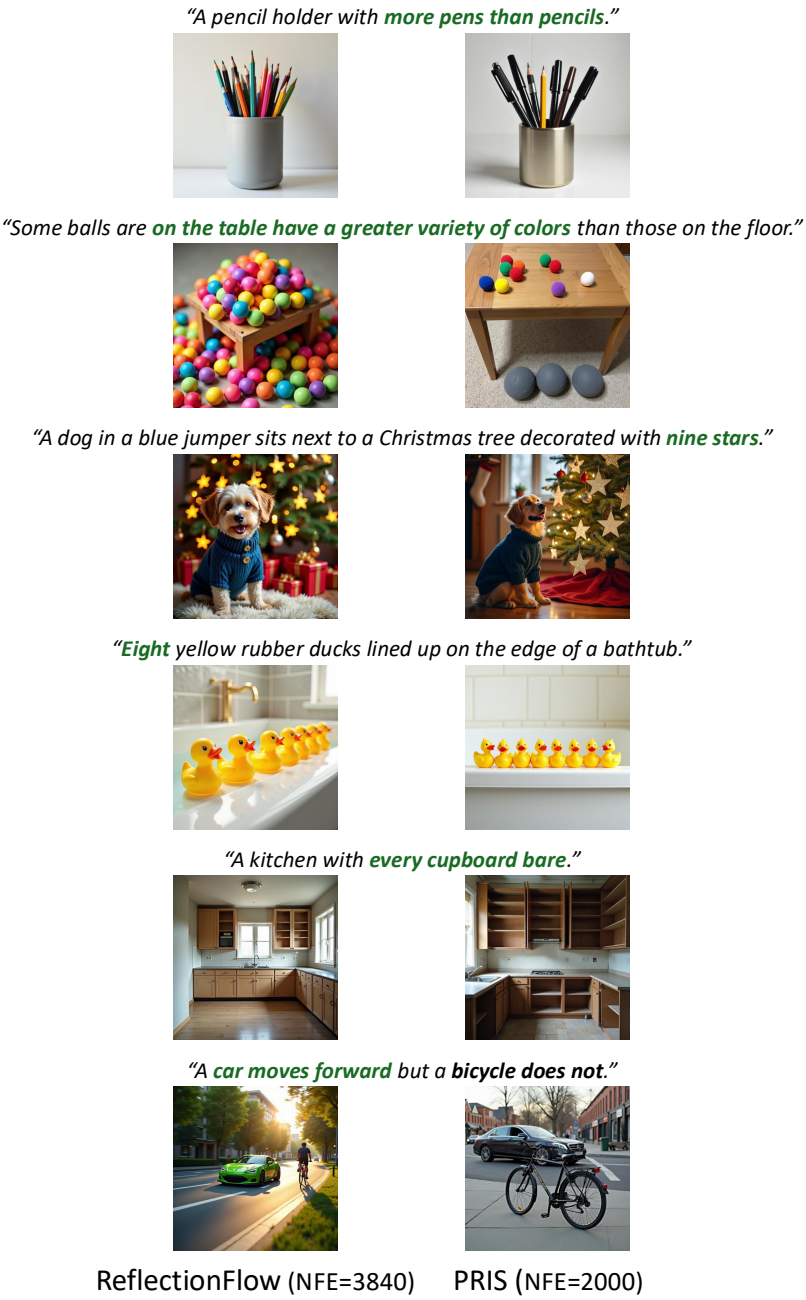


Figure 14. **Qualitative comparisons with ReflectionFlow.** Despite being training-free, our method markedly outperforms the learned approach, underscoring the effectiveness of *correcting prompts using shared failure patterns across samples*.

A.6. Comparison with T2I-Copilot

We further compare PRIS with prior prompt optimization methods, including T2I-Copilot, a recent state-of-the-art approach, in Table 9. Our results show that PRIS consistently outperforms T2I-Copilot across all scaling regimes, achieving higher VQA scores for all values of N. These results demonstrate that population-level scaling in PRIS is more effective than per-sample methods, despite differences in their refinement mechanisms.

Table 9. Comparison with state-of-the-art prompt optimization methods.

	VQA-Score (N: # of generated samples)		
Method	$N = 10$	$N = 20$	$N = 30$
BoN [31]	0.764	0.783	0.794
T2I-Copilot [5]	0.825	0.844	0.854
PRIS (ours)	**0.834**	**0.854**	**0.865**

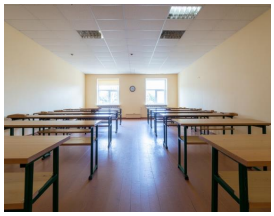
A.7. Component Ablations

To validate the design choices of PRIS, we conduct component ablations on aggregation and selection in Table 10. Reusing seeds complements population-level failure aggregation by mitigating residual misalignments beyond common failure modes, providing a more informative initialization than random sampling. For refinement triggering, we adopt a 50% success threshold per element. A stricter 25% threshold fails to identify elements requiring refinement, while a relaxed 75% threshold leads to unnecessary updates of already stable components, both resulting in degraded performance. For selection, $k=5$ ($\lceil N/4 \rceil$) achieves the best balance between selection pressure and diversity. Increasing to $k=7$ introduces more noisy samples, while reducing to $k=3$ overly restricts the search space.

Table 10. Quantitative ablations of PRIS on T2V experiments. The first row represents our **default configuration**.

Variant	Seed choice	Success prob.	Top-k	Score
PRIS (Default)	Reusing	50%	5	**0.754**
Seed choice	Random	50%	5	0.725 ↓ 0.029
Success prob.	Reusing	25%	5	0.737 ↓ 0.017
	Reusing	75%	5	0.726 ↓ 0.028
Top-k	Reusing	50%	3	0.745 ↓ 0.009
	Reusing	50%	7	0.721 ↓ 0.033

A.8. Prompt Transferability and Future Work



Original Prompt

Revised Prompt

Original Prompt: In a classroom, the clock is not on the wall

Revised Prompt: In a classroom, **the clock is placed on a polished wooden desk**, its round face softly illuminated, while the walls remain unadorned, free of any other timepieces.

Original Prompt

Revised Prompt

Original Prompt: A little boy with a ping pong paddle looks more excited than a little girl without one.
Revised Prompt: A young boy holding a bright yellow ping pong paddle beams with **wide eyes and an open smile**, while nearby, a **calm little girl** gazes at him with a curious expression, **her hands resting by her side.**

Figure 15. **Qualitative example of prompt transferability**. Prompts revised for Flux1.dev are applied to Firefly Image 4 Ultra. By clarifying vague instructions, specifying object presence and absence, and reinforcing contextual cues, the revised prompts yield visuals with stronger adherence compared to those generated from the original prompts.

We observe that our revised prompts are not only effective for the original generator but also transferable to other models, demonstrating their generalizability. This stems from the fact that our revisions resolve ambiguities in the original prompts, making them more precise and robust. Although different generators may specialize in certain aspects, such as producing fine-grained details or maintaining object counts, they often exhibit overlapping weaknesses. Addressing these weaknesses through prompt revision thus benefits multiple models simultaneously.

Figure 15 illustrates this transferability. The prompts originally revised for Flux1.dev are successfully applied to Firefly Image 4 Ultra. For example, the revised prompts clarify vague or underspecified instructions (e.g., replacing "not on the wall" with "the clock is placed on a polished wooden desk"), making object presence and absence explicit (e.g., reformulating "the girl is without a ping pong paddle" into "her hands resting by her side"), and reinforcing contextual cues. These findings suggest a promising research direction: fine-tuning LLMs or other prompt-rewriting systems on pairs of naïve user-provided prompts and failure-focused revisions. By learning systematic transformations from short, underspecified, and loosely written prompts into precise, detailed, and effective ones, rather than relying on random expansions, such models could reduce verification costs and inference-time overhead, accelerating the discovery of high-quality prompts from the outset.

A.9. Future work and limitations.

Our core idea—identifying shared failure patterns with high precision and addressing them through targeted prompt revision—is broadly applicable to any text-conditioned generative model. We believe extending this idea to other modalities and tasks is a compelling direction for future research, potentially challenging existing inference-time scaling laws. In addition, our benchmark provides a new avenue for evaluating verifiers at the attribute level. We also find that prompts refined on one model often generalize well to others (see Appendix A.8). This observation suggests a promising direction: fine-tuning LLMs or other prompt-rewriting models using paired data consisting of randomly expanded prompts and their failure-focused revisions. Such training resources could reduce verification overhead and inference-time costs, enabling more efficient discovery of high-quality prompts from the start.

B. Benchmark Construction and Evaluations

B.1. Benchmark Category

Details about benchmark constructions. Existing visual evaluation datasets are mostly limited to human-preference annotations. While useful for coarse quality assessment, such datasets are insufficient for our focus: selecting the best-aligned videos from among multiple misaligned candidates, which lies at the core of inference-time scaling. To address this limitation, we construct a new benchmark explicitly tailored for inference-time scaling and use it to evaluate our verifier, its ablations, and existing baselines. Beyond serving as a testbed for our study, this benchmark also provides a valuable resource for future research on visual prompt-adherence verification.

In our benchmark, each prompt is paired with multiple generated videos, with at least one ground-truth (GT) aligned reference and others containing slight misalignments, thereby forming a mid-quality candidate pool. In total, the benchmark comprises 410 prompts. We collect prompts showcased in demos of both popular open-source [36] and closed-source video models [9, 21], and categorize them into two broad groups: motion (120 prompts) and physics (144 prompts). To further enrich the evaluation, we also adopt prompts from VBench 2.0, spanning three fine-grained motion-related categories: dynamic attributes (47 prompts), motion order (68 prompts), and motion rationality (31 prompts). For each prompt, we generate videos using multiple text-to-video models [9, 21, 36] as well as image-to-video models [36], ensuring the inclusion of both GT-aligned and misaligned outputs. Each video is independently annotated by three human evaluators as GT or non-GT, and the final label is assigned by majority vote.

Detailed analysis of verifiers on our benchmark per category. In addition to the overall accuracy reported in Table 5 of the main manuscript, we present per-category accuracy in Table 11. As the results show, EFC consistently achieves the highest accuracy across all categories. Compared to the decomposed binary VQA baseline, which shares our decomposition strategy but replaces our text-to-text verification with binary VQA, EFC yields a substantial performance gain, underscoring the advantage of our text-based approach over visual QA methods. When compared to learned reward models (i.e., MLLM-based verifiers fine-tuned on human-preference datasets), including VideoAlign (the strongest among them and used as our tie-breaker), EFC still maintains a significant lead. Notably, it achieves this performance without any additional training on preference datasets, but rather through a systematic zero-shot verification process. Furthermore, we attribute this gap to the fact that reward models are typically trained on human-preference data, where subtle aspects such as frame quality, motion smoothness, or stylistic biases often dominate judgments, even when they are not directly related to prompt adherence. In contrast, EFC focuses explicitly on verifying semantic alignment with the prompt, making it both more accurate and interpretable.

Table 11. **Quantitative results of verifier accuracy per prompt category on our constructed dataset. Bold** indicates the best result.

Method	Motion	Physics	Dynamic Attributes	Motion Rationality	Motion Order Understanding	**Average**
VisionReward [42]	0.650	0.569	0.319	0.662	0.452	0.571
UnifiedReward [40]	0.492	0.507	0.298	0.588	0.581	0.498
VideoAlign [29]	**0.792**	0.660	0.511	0.794	0.516	0.693
Decomposed binary VQA	0.733	0.667	0.617	0.809	0.613	0.700
PRIS (Ours)	**0.792**	**0.764**	**0.638**	**0.838**	**0.677**	**0.763**

While our study focuses on prompt-adherence verification, we believe that our verification framework can be extended to other important axes of evaluation, such as motion quality, NSFW filtering, and bias detection, by replacing prompt decomposition with task-specific decomposition strategies. This flexibility offers promising directions for future research.

C. Experiments Details

C.1. Detailed Setup

For GenAI-Bench, since many prompts within the same categories (e.g., counting, differentiation, comparison, negation, universal) are similar but differ only in objects, we randomly subsample 20% to reduce redundancy. For selecting k, we set $k = N//4$, as $N//2$ samples are first generated for review before prompt revision, and half of them are used as top-performing seeds.

C.2. Base Model Selection

To ensure that our study focuses on the effect of prompt redesign in inference-time scaling, we first measure the degree of prompt adherence across candidate leading open-source video models such as Wan, LTX, and Hunyuan. This step is necessary because if a model fails to follow the prompt at all, there is little need to apply prompt redesign. Specifically, we compute the text embedding similarity between the original prompt and the generated video caption. We use Qwen-32B for captioning and employ the SentenceTransformer model (`intfloat/e5-mistral-7b-instruct`) to measure embedding similarity. We present the similarity score in Table 12.

Table 12. **Quantitative results of prompt adherence** across different text-to-video models, used to exclude base models with poor alignment and retain only those with acceptable adherence.

Metric	Method	Motion Rationality	Motion Order Understanding	Dynamic Attribute	**Average**
VideoAlign	LTX	0.764	-0.153	-0.977	-0.122
	Hunyuan	0.904	0.212	-0.775	0.114
	Wan	**1.475**	**0.940**	**-0.397**	**0.673**
Text Similarity	LTX	0.635	0.642	0.600	0.626
	Hunyuan	0.678	0.671	0.616	0.655
	Wan	**0.717**	**0.702**	**0.631**	**0.683**

Based on this analysis, we selected Wan as our primary video model, since it demonstrates a reasonable level of prompt adherence while leaving room for improvement through verification and redesign. In contrast, models such as LTX and Hunyuan were excluded, as their low adherence made them unsuitable for evaluating prompt redesign at inference-time scaling, particularly on complex prompts in VBench2.0 that involve status changes or multiple consecutive events within a single video.

D. Additional Qualitative Experimental Results

D.1. Text-to-Image Generation

We provide additional qualitative results beyond Figure 3, demonstrating that our prompt redesign improves coherence of the final visual outputs under the same NFE budget (2000, as in the main experiments). As shown in Figure 16, which compares the top-scoring outputs generated from the original GenAI-Bench prompts, our method performs particularly well on prompt sets containing ambiguous attributes, numerical specifications, or subtle constraints (e.g., "without," "greater variety"), effectively elaborating them into more faithful visual realizations than baselines.

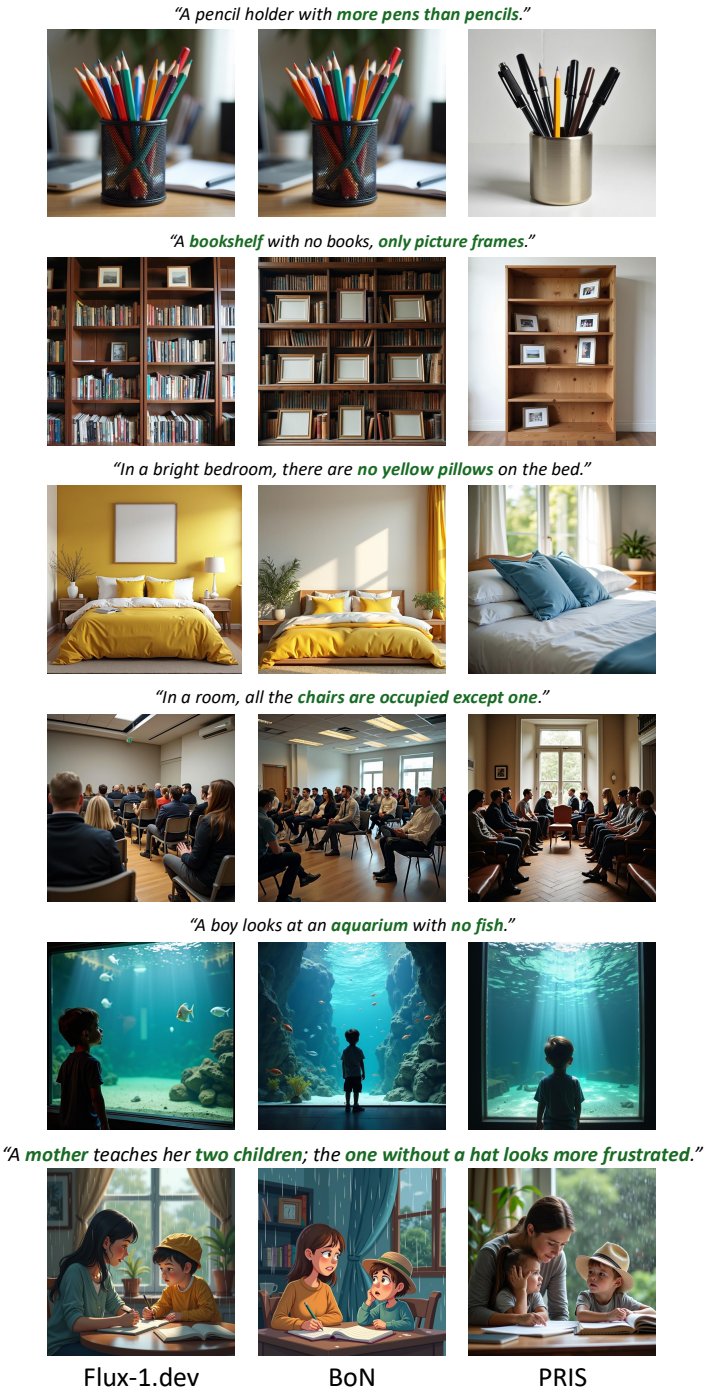


Figure 16. **Qualitative comparisons on T2I generation** where visual generation is (initially) conditioned on the original prompts.

We also compare with standard prompt expansion in Figure 17, where ours achieves substantially higher prompt fidelity compared to the baselines. Unlike standard prompt expansion, which cannot target or identify the most challenging semantic elements, our joint scaling of visuals and prompts more faithfully preserves the intended semantics by adaptively revising the prompt based on recurring failure modes.

Figure 17. **Qualitative comparisons on T2I geneation** where visual generation is (initially) conditioned on standard prompt expansion.

D.2. Text-to-Video Generation

In addition to Figure 4, we present additional qualitative top-scoring examples in Figure 18. As shown, our method more faithfully follows the intent of the original prompt. The final top-scoring visuals generated with our PRIS demonstrate significantly stronger prompt adherence compared to baselines. Specifically, BoN often misses key events or produces unnatural temporal order. For example, it may depict only a single motion (e.g., morphing without differentiating "cleaning the kitchen" in the 1st visual) or assign different motions to different people (in the 4th visual). BoN also frequently fails to capture dynamic changes, generating only static states (3rd and 6th visuals). Furthermore, BoN often does not correctly realize sequential actions, such as repeatedly attempting to break chocolate pieces, whereas our method generates coherent sequences where the person both attempts the action and displays the broken pieces (5th visual).

D.3. More Visualizations

We include an HTML file to the attached zip file. To explore the generated visuals and comparisons with baselines alongside their corresponding prompts, please open `visuals/index.html` in a Chrome browser (This file is located in the `visuals` directory within the attached zip file). This visualizes the generated visuals, including images and videos, in the `visuals/resources` folder.

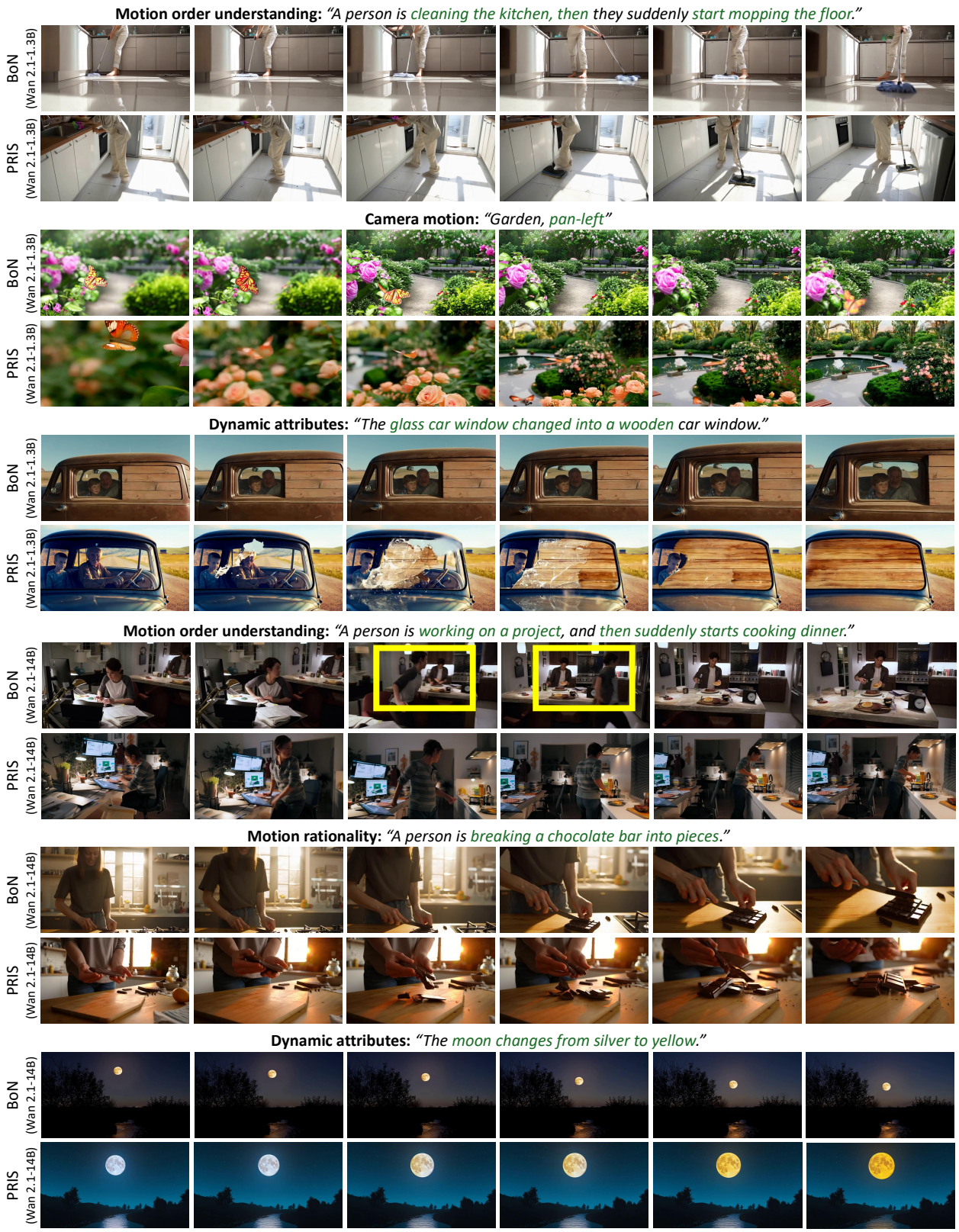


Figure 18. **Qualitative comparisons on T2V generation** where visual generation is (initially) conditioned on standard prompt expansion, with Wan2.1-1.3B (top) and Wan2.1-14B (bottom).