MatchDiffusion: Training-free Generation of Match-Cuts

Supplementary Material

We encourage the reader to visit our website `https://matchdiffusion.github.io` to better visualize the results shown in the paper.

This `website` provides additional insights and visualizations to complement the document. We show multiple videos of results that we showed as frames in the manuscript. We also include many more interesting Match-Cuts generated by MatchDiffusion. Additionally, we show a few examples of famous match cuts found in TV shows and movies to give a deeper understanding of what a match-cut is. In the following document we show a few more analysis of our method and a few more frame visualizations of match-cuts. However, **all these qualitative results are also included in the `website`.** We create visualizations by concatenating the first half of one video with the second half of the other. The method of combining these halves is user-selectable; on our website, we showcase examples using Straight Cuts, Alpha Blending, and Flickering transitions.

A. Alternative Methods for Match-Cuts

Our results demonstrate that **MatchDiffusion** outperforms existing alternatives for match-cut synthesis. However, we further analyze why these alternative methods, while seemingly viable, do not fully address the challenges of match-cut generation. We present both a conceptual discussion and additional experimental evidence to highlight the fundamental differences between match-cut synthesis and common video editing tasks.

Why Video Editing Methods Fall Short. We articulate further our reasoning for the baselines chosen in the main paper. Recent video editing tasks, such as video-to-video (V2V) translation [3, 27] and motion transfer [50, 54], could potentially synthesize match cuts by generating compositionally similar videos. However, as explained before, both approaches impose constraints that limit their applicability to match-cut generation. In contrast, match-cuts require alignment in either structure, motion, or both, *without enforcing one-to-one constraints on the generated videos*. This makes match-cut creation fundamentally different from any existing video morphing task. Unlike interpolation or morphing, which focus on smooth transformations between frames, match cuts demand independent synthesis while preserving a strong visual relationship between two separate scenes. This is the reason behind our design choice to synthesize a pair of videos, and later join them in a cut with post-processing.

Method + Backbone	CS	Mot	LPI
UniEdit [3] + LaVie [47]	0.30	0.67	0.46
MatchDiffusion (ours) + LaVie [47]	**0.31**	**0.76**	**0.28**

Table A2. Comparison of MatchDiffusion on LaVie with a Video-to-Video baseline.

Our method allows the model to naturally impose constraints on motion, structure, or both, depending on the input prompts, by leveraging the Joint Diffusion mechanism 3.2, while still allowing each prompt to be followed independently during Disjoint Diffusion 3.3. This enables the synthesis of match cuts without the structural rigidity of V2V methods or the motion constraints of motion transfer-based approaches.

MatchDiffusion on other backbones and V2V variations. We implemented MatchDiffusion on one of the best performing models at the time of submission, *i.e.* CogVideoX-5B, along baselines. At the time of submission CogVideoX-5B supported as V2V method SDEdit, which we used for our experiments. To extend our analysis to different V2V pipelines, we revisit the quantitative comparison from Table 1 in the main paper and extend it with an additional experiment using a more recent V2V baseline, UniEdit [3]. UniEdit uses LaVie as backbone in the official implementation. For a fair comparison, we implement our MatchDiffusion sampling using LaVie as a base T2V model (the same used by UniEdit) and use it for comparison.

As shown in Table A2, MatchDiffusion consistently outperforms UniEdit across all metrics, reinforcing the key insight that match-cut generation is not merely a variation or extension of existing video morphing or editing tasks. These results further justify the need for dedicated methodologies designed specifically for the unique constraints of match cuts. Furthermore, they demonstrate how MatchDiffusion can be seamlessly integrated into any video diffusion baseline. Additionally, just as the metrics indicate the superiority of MatchDiffusion, the qualitative results presented in Figure A10 further support this conclusion. The match-cuts produced by MatchDiffusion exhibit significantly more coherent transitions between the two scenes, underscoring the effectiveness of our approach in both numerical evaluations and visual quality. We invite the reader to see these results our `website` .

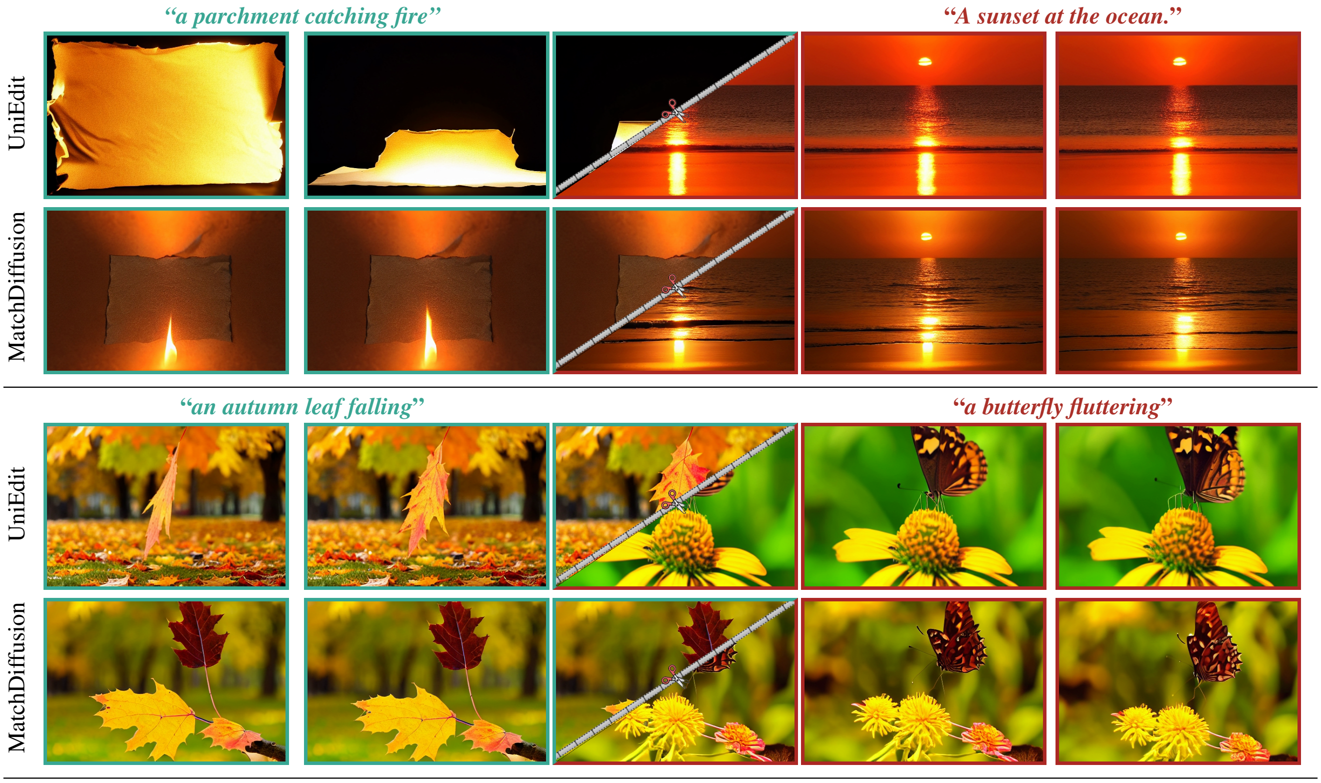

Figure A10. **Generated match-cuts with UniEdit vs MatchDiffusion on LaVie.** UniEdit results are in rows 1 and 3. MatchDiffusion results are in rows 2 and 4. Although UniEdit does preserve some structural characteristics (note the scene composition in the leaf to flower transformation), MatchDiffusion shows more consistent transitions preserving structure while diverging in semantics.

B. Additional Analysis.

Effect of classifier-free guidance. We analyze here the effect of the CFG (classifier-free guidance) parameter when making match-cuts with MatchDiffusion. Here we fix the K and analyze the different metrics when varying CFG. In Figure A11, we observe that larger CFGs tend to drop the CLIPScore but also make the entanglement (Motion) of motion and structure (LPIPS) to be stronger. Similar to the K, parameter there is a sweet spot in which Motion and Structure and shared across the two videos, while still following the prompt. We found that a CFG between 5 to 7 works well for the majority of the cases. In rare occasions, we found CFG $= 10$ also performing well for specific prompts.

Different combination function f. In Section 3.2 we defined f as the average of the estimates from the two paths. However, one could try a different strategy to combine the two path estimates. In Figure A12 we show the results but this time combining the two paths by linearly decaying the weight of one another until making them independent. This would change the previous approach of the combination of the two paths from a step function to a simple linear decay. The results, show that variations of K (dif-

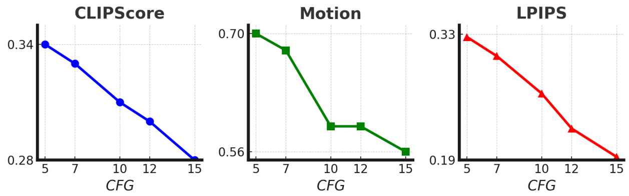

Figure A11. **Effects of CFG.** By increasing classifier-free guidance, we report significantly degraded performance in all metrics. We tune optimally the parameter between 5 and 7 for most cases.

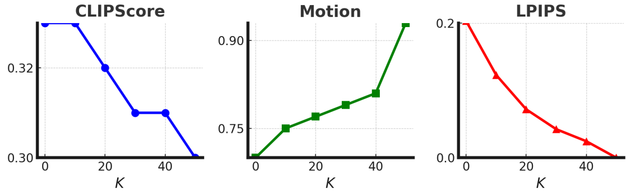

Figure A12. **Different f.** We replace the f used in the main paper with linearly interpolating between the two diffusion paths depending on K.

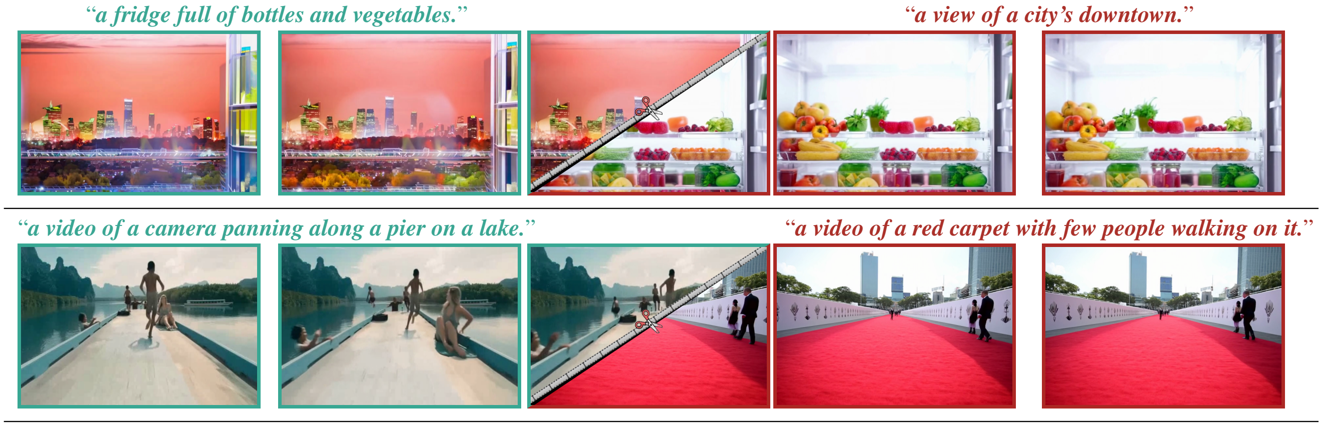
"a fridge full of bottles and vegetables."
"a view of a city's downtown."
"a video of a camera panning along a pier on a lake."
"a video of a red carpet with few people walking on it."

Figure A13. **Sampling match-cuts.** MatchDiffusion can automatically synthesize match-cuts based on the prompts in green and red. Each row shows a different sample coming from the same pair of prompts, providing the user with more alternatives for the same match-cut.

fusion step in which the decay starts) yield more motion-entangled results, quantified by the higher values in motion fidelity (middle plot). We advocate anyways that having more flexibility in the motion (hence with lower motion fidelity) allows to generate more variable videos, assuming outputs respecting the definition of a match-cut. Hence, we still selected averaging as our f of choice.

Visual quality From the analysis in the main paper, Figure 9, we notice that applying MatchDiffusion has negligible quality impact on CLIPScore with respect to the original model (*i.e*. the case in which $K = 0$). Here, we extend the discussion on the visual quality of the generated outputs. We calculate frame-wise aesthetic/technical quality with NIMA [43], obtaining **4.40/4.61** for base CogVideoX and 4.14/4.49 for Ours, totaling a marginal performance decrease (-5.9%/-2.6%). We also evaluate the blurriness of outputs, to understand if the generation of match-cuts would lead the network to generate more ambiguous (i.e. blurred) videos. To do so, we process frame-wise the generated videos with the blur detector proposed in [2] and average the obtained blurriness masks to get a single value. We report **0.149**/0.144 for CogVideoX-5B baseline/ours (-3.35%). *Performance degradation is negligible*, demonstrating that MatchDiffusion can benefit from the visual quality of highly performing video diffusion models.

Sampling We show results of different match-cuts produced for the same prompt and the same parameters, by just sampling with different seeds. We observe that sampling from the method can help at creating different interpretations of the same matching concepts. We show sampling from our method in Figures A14, A15, A16, and A17.

Cost of K optimization One may argue that K still requires tuning, hence a further discussion is needed. In our experiments, we did not experiment with more than 3 different values of K, that were often sufficient to find an aesthetically pleasant match-cut. Compared to the significant costs associated to a creation of a match-cut without MatchDiffusion, the cost of optimization is minimal. Moreover, let us stress again *results highly depend on the user's aesthetic perception*. In practice, any K would result in something potentially usable as a match-cut. This is also further proved by our experiment in Section 4.5.

C. Future directions

We briefly explored two potential directions to improve the controllability of our method. The first, illustrated in the top row of Figure A13, involves altering the colors of one video using an edge-based ControlNet. Our observations indicate that ControlNet, when combined with our approach, can introduce additional stylistic effects while still preserving the intended match cut. A second, and perhaps more promising, direction is conditioning on an input video to generate a synchronized counterpart from a text prompt, such that the pair forms a match cut. This task is considerably more challenging, as the input video imposes constraints on how the pre-trained diffusion model can achieve the cut. We attempted a simple strategy by replacing x'_t with the noisy encoding of the reference video at each denoising step. Although the resulting quality is lower compared to our main method, the approach can still produce reasonable match cuts. We present one successful example in Figure A13, where the system takes an existing advertisement video as input and generates a red-carpet scene to create a compelling match cut.

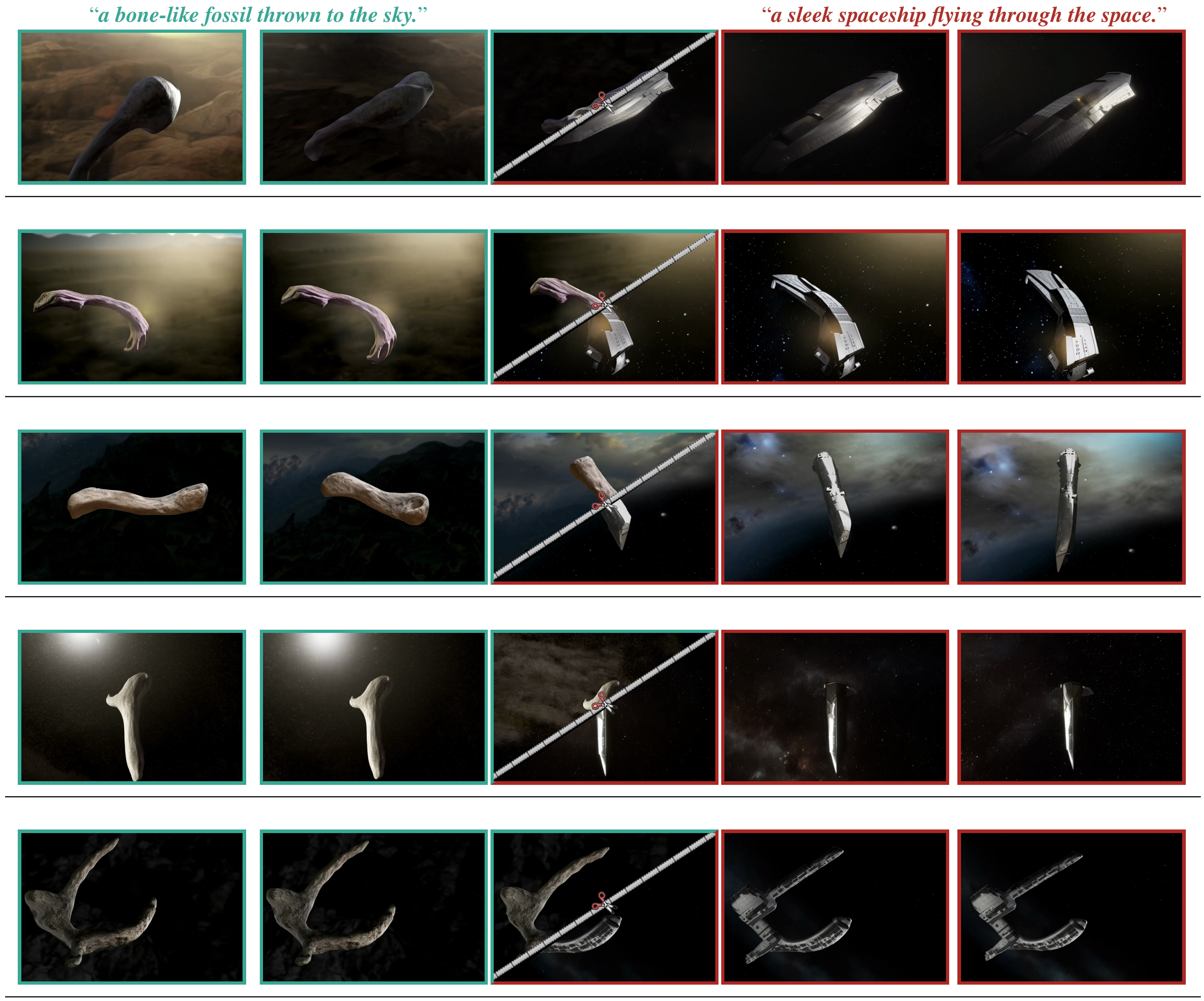
“a bone-like fossil thrown to the sky.”
“a sleek spaceship flying through the space.”

Figure A14. **Sampling match-cuts.** MatchDiffusion can automatically synthesize match-cuts based on the prompts in green and red. Each row shows a different sample coming from the same pair of prompts, providing the user with more alternatives for the same match-cut.

D. Limitations

A key limitation of our method lies in its reliance on prompt quality and creative input. While the system can generate visually appealing match-cuts, achieving truly compelling results often depends on carefully crafted prompts and sampling. We found that prompts inspired by existing match-cuts, such as those from iconic film scenes or curated blog posts, significantly improve the system's success rate, whereas randomly devised prompts frequently fail. This underscores that the creative process heavily relies on human ingenuity to guide the system. Currently, the system autonomously determines key aspects of the match-cut, including structure, color, layout, and motion. Future work could focus on providing users with finer control over these elements, enabling a more deliberate and customized match cut generation process.

E. Application on images

Although our paper focuses on match-cuts, we also found that by using an Image-Diffusion model like Stable Diffusion 1.5 [38], we can create couples of images that also share structural similarities while being semantically divergent. We did not include these results in the main manuscript as we are unsure whether this has any applications on real-world problem. However, the results look visually appealing, and as such they may enable creative use-cases. We show some results of MatchDiffusion using SD1.5 as backbone in Figures A18, A19.

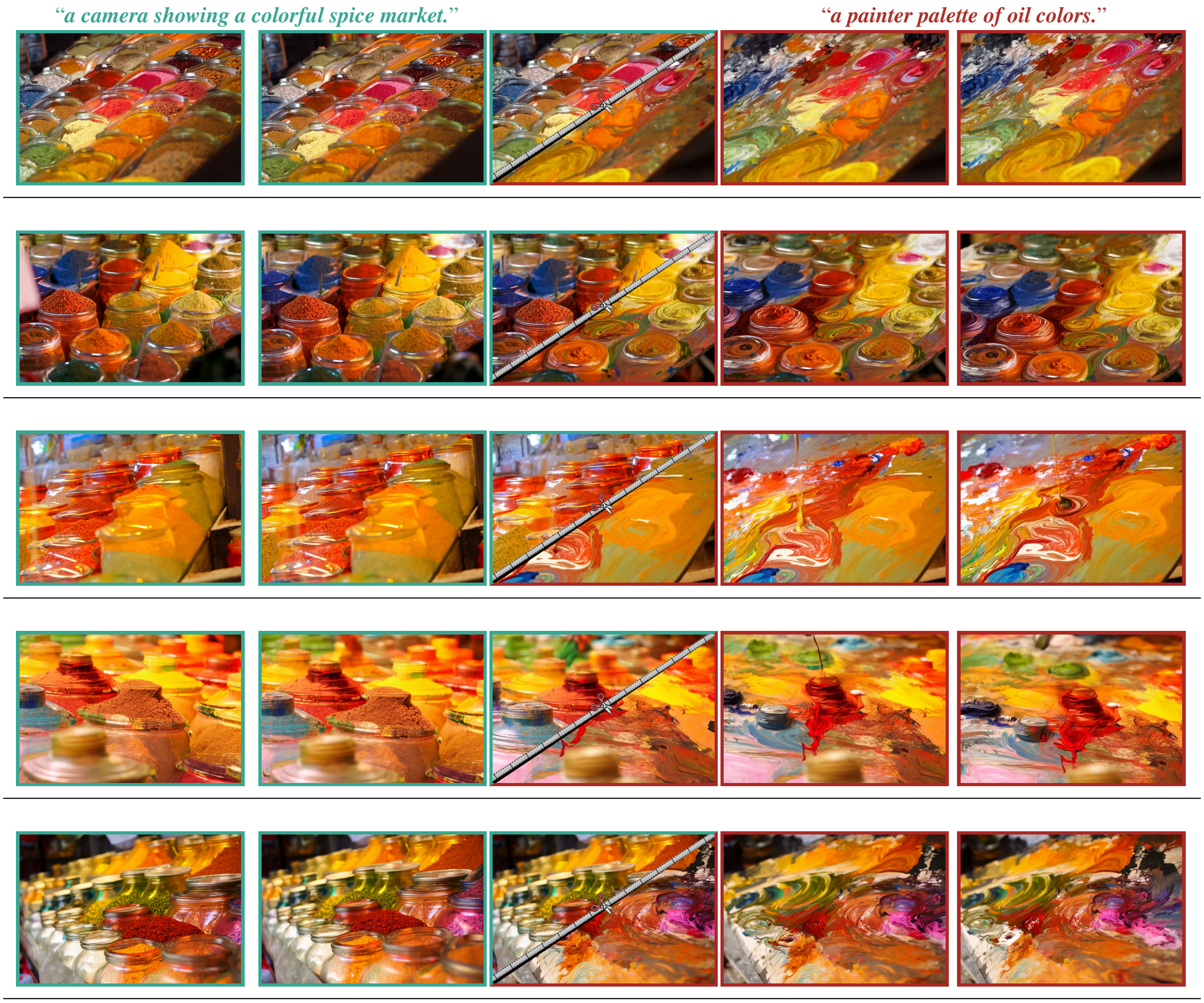
"a camera showing a colorful spice market."
"a painter palette of oil colors."

Figure A15. **Sampling match-cuts.** MatchDiffusion can automatically synthesize match-cuts based on the prompts in green and red. Each row shows a different sample coming from the same pair of prompts, providing the user with more alternatives for the same match-cut.

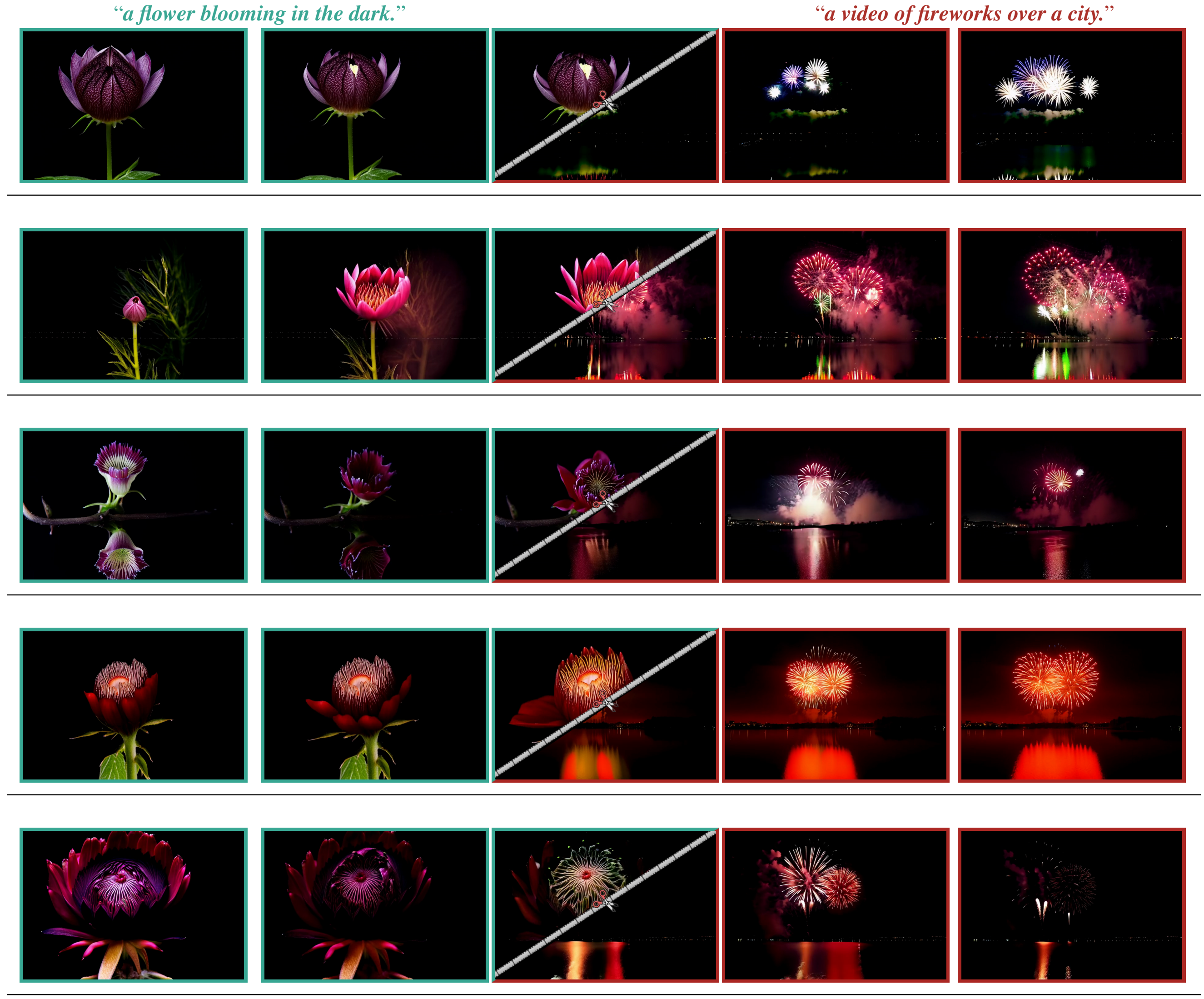
"a flower blooming in the dark."
"a video of fireworks over a city."

Figure A16. **Sampling match-cuts.** MatchDiffusion can automatically synthesize match-cuts based on the prompts in green and red. Each row shows a different sample coming from the same pair of prompts, providing the user with more alternatives for the same match-cut.

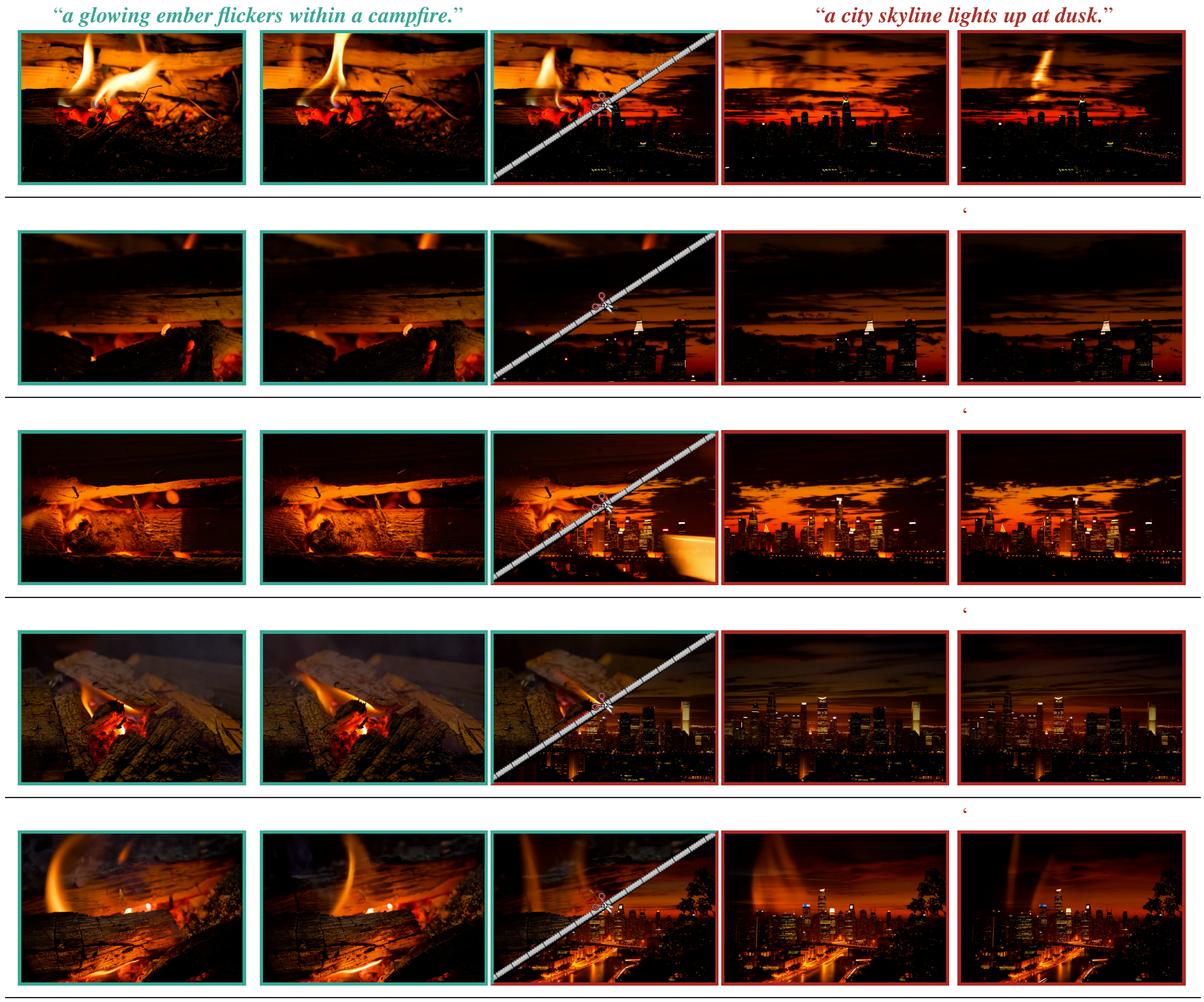
"a glowing ember flickers within a campfire."
"a city skyline lights up at dusk."

Figure A17. **Sampling match-cuts.** MatchDiffusion can automatically synthesize match-cuts based on the prompts in green and red. Each row shows a different sample coming from the same pair of prompts, providing the user with more alternatives for the same match-cut.

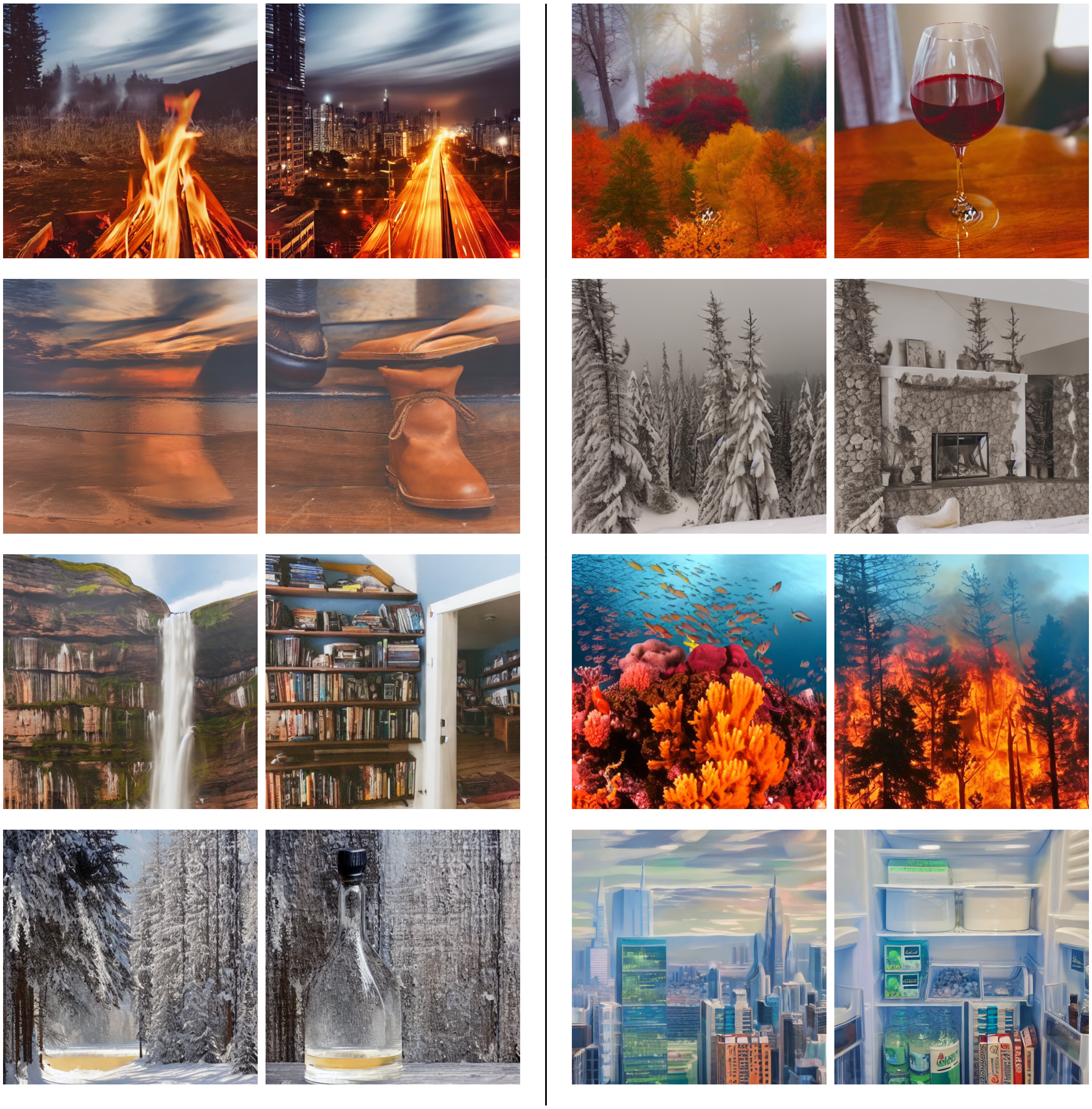

Figure A18. **Examples of MatchDiffusion with Stable Diffusion 1.5.**

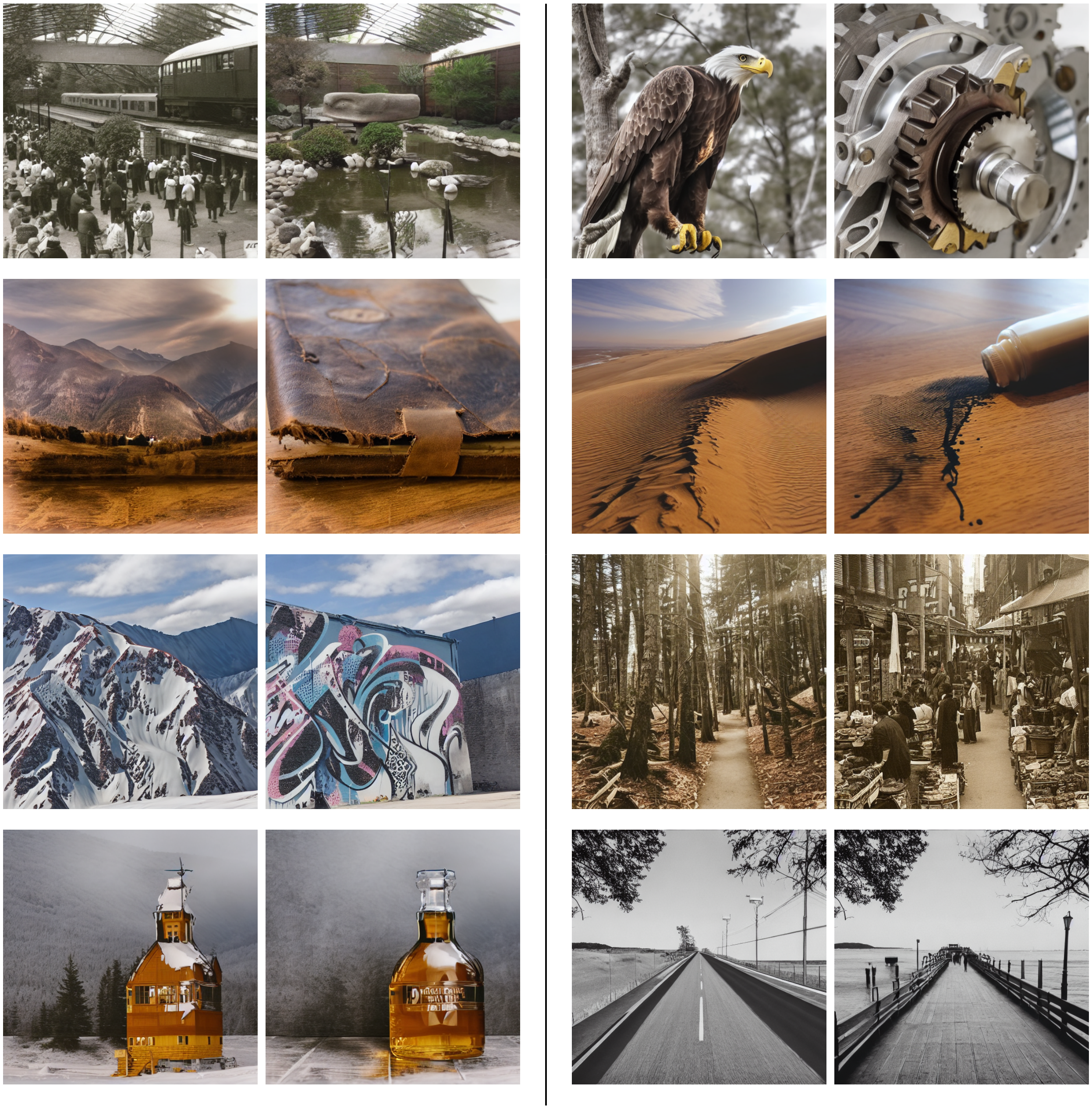

Figure A19. **Examples of MatchDiffusion with Stable Diffusion 1.5.**