How to Take a Memorable Picture? Empowering Users with Actionable Feedback

Supplementary Material

In this supplementary material, we provide additional details on MemBench and MemCoach. Section A presents qualitative examples from MemBench dataset and describes its construction pipeline. Section B gives additional implementation details of our proposed MemCoach method with a discussion regarding potential implications of our work, while Section C provides preliminary user study experiments. Finally, in Section D, we demonstrate the consistency of our framework across different editing models and memorability predictors, including an analysis of feedback quality generated by MemCoach.

A. MemBench Additional Details

A.1. Data Examples

Fig. 10 and 11 present data examples from the MemBench dataset. For each image pair, we show (from left to right) the source image (red frame), the destination image (blue frame), and the corresponding feedback generated by the multimodal model. The memorability scores assigned by the predictor $\mathcal{M}$ are shown beneath each image.

A.2. Construction Details

Images within each scene are ranked using the memorability predictor $\mathcal{M}$. Pairs of least and most memorable images are then selected to construct contrastive training data. Evaluation is performed on a *random held-out set of unseen scenes*, where feedback is generated starting from the scene's least memorable image.

A.3. Image Pre-processing

PPR10K [43] provides images in RAW format, with an average image file size of 42 MB. To reduce storage requirements, enable efficient processing, and ensure compatibility with MLLM vision processors, we convert all RAW files to JPEG format while preserving their original aspect ratio. Conversions are performed using `rawpy` and `PIL` Python libraries, both executed with default parameters.

A.4. Memorability Predictor

To build the memorability predictor $\mathcal{M}$, we follow the approach proposed in [74] where a frozen visual feature extractor is followed by an MLP head trained for regression. The final model outputs a single continuous value in $[0, 1]$ corresponding to the predicted memorability score.

Training. We train the regressor $\mathcal{M}$ on three widely used memorability datasets: LaMem [32], MemCat [18], and SUN [17]. Each image is associated with a ground-truth memorability score in the range $[0, 1]$. For feature extraction, we employ the vision tower of OpenCLIP [25], specifically the `ViT-SO400M-14-SigLIP-384` model [76] pretrained on WebLI [10]. The resulting visual embeddings (dimension 1152) serve as input to the regression head. The MLP consists of two fully connected layers: a 256-dimensional hidden layer with a ReLU activation [1], followed by a 1-dimensional linear output layer. The model is trained end-to-end only on the MLP parameters, using the mean squared error (MSE) loss. We train for 100 epochs using the Adam optimizer [33] with a learning rate of 1×10^{-4} and weight decay set to 0.0. Empirically, we find that using the raw (unnormalized) OpenCLIP features leads to improved Spearman's rank correlation [75]. A summary of the model performance is reported in Tab. 5.

Table 5. **Model $\mathcal{M}$ performance.** Comparison of memorability predictor models trained on CLIP-like embeddings. We report feature dimensionality, whether feature normalization is applied, and Spearman's rank correlation. The used predictor model achieves the highest correlation among all evaluated variants. (*) is reported from [74].

Model	Fts dim	Normalization	Spearman Rank (↑)
Human*	*n.d.*	*n.d.*	0.68
ViT-L-14-quickgelu	768	✓	0.73
ViT-L-14-quickgelu	768	×	0.81
ViT-SO400M-14-SigLIP-384	1152	✓	0.76
MemBench $\mathcal{M}$	1152	×	**0.82**

Validation. To assess the effectiveness of our memorability predictor $\mathcal{M}$, we compare it against state-of-the-art memorability models [17, 21, 58, 63, 74] on the LaMem test set (Tab. 6). While prior approaches are typically trained solely on LaMem, except for [58], our model $\mathcal{M}$ leverages three datasets (LaMem, MemCat, and SUN), achieving the highest Spearman's rank correlation among all evaluated methods.

Limitations. Our automatic pipeline relies heavily on the initial ranking of images, which is determined by a memorability predictor $\mathcal{M}$. Although this dependency introduces a potential source of bias, we treat the memorability model as a well-established black-box component, consistent with prior literature [8, 18, 26–28, 32, 38, 48, 52, 74]. In this sense, our framework is agnostic to the specific predictor used: given any target scoring criterion, it may substitute an alternative ranking signal and construct systems tailored to their own objectives.

Table 6. **Comparison with state-of-the-art memorability predictors.** Spearman Rank correlation on the LaMem test set. Our model, trained on LaMem, MemCat, and SUN, achieves the highest correlation among all methods. (*) is reported from [74].

Model	Pretrained	Spearman Rank (↑)
MemNet* [63]	LaMem	0.64
AMNet* [17]	LaMem	0.67
Human*	*n.d.*	0.68
ViTMem* [21]	LaMem	0.71
Henry [58]	LaMem + MemCat + SUN	0.72
Henry [58]	LaMem	0.74
PerceptCLIP [74]	LaMem	0.74
MemBench $\mathcal{M}$	LaMem + MemCat + SUN	**0.82**

A.5. Editing Baseline

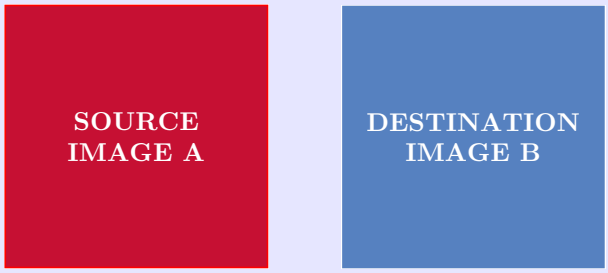
For in-context image editing, we employ FLUX.1 KONTEXT [37], as described in Sec. 3.2, via `diffusers` library [66] from HuggingFace (open-source model version with tag `FLUX.1-Kontext-dev`). We use the default configuration recommended by the original authors, setting the number of inference steps to 28, the guidance scale to 2.5, and fixing the seed generator to 0 to ensure reproducibility. The default aspect ratio of the model is 1:1, specifically 1024×1024 pixel resolution.

A.6. Prompting and Structured Feedback

Feedback elicitation prompt. To generate the full MemBench dataset, we rely on prompt p_a, presented in Sec. 3.2. The complete prompt p_a is reported below.

System: You are an observer.

User:



Your task is to determine the actions required to transform Image A into Image B.
Strictly avoid both explicit and implicit references to the images when suggesting action items, and ensure that each action item is fully self-contained.

Produce a structured JSON object that must include:
– `actions`: a list of precise and well-informative semantic actions.

Respond with a valid JSON object and no explanation.

Output:

```
{
    "actions": [
        <#1 sub-action>,
        <#2 sub-action>,
        ...,
        <#k sub-action>
    ]
}
```

The resulting output is then parsed as a JSON object.

During early experimentation, models frequently produced feedback that referred directly to the target image rather than describing the transformation itself (*e.g.* "*Adjust the brightness to match the one in Image B*"). To address this issue, we refined the prompt to explicitly forbid image-referential phrasing and require self-contained action descriptions.

In cases where multiple source images receive the same memorability score, ties are resolved by sorting according to the filename identifier in descending order.

Structured, formally valid JSON outputs. To ensure consistently structured outputs while maintaining flexibility in the definition of output fields, we adopt the `outlines` library [68] for constrained decoding; compatible with the `transformers` library [69], it allows to enforce a predefined output schema. For extracting the feedback divided into subactions, we define the following class specification:

```
class ActionListOutput(BaseModel):
    actions: List[str] = Field(
        description="A list of actions.",
        min_items=1,
        max_items=10
    )
```

Listing 1. Class specification to ensure valid JSON schema.

This setup enforces syntactic validity, guarantees reliable parsing, and enables systematic storage of feedback samples in JSON format. The schema-based design also allows for straightforward modifications, such as adding or removing fields when extending the output format.

A.7. Feedback Sub-actions Categorization

To categorize the atomic sub-actions contained in each feedback instance, we employ GPT-5 MINI [50] as an automatic annotator. The model is prompted with a taxonomy covering six high-level categories: *Framing*, *Lighting*, *Posing*, *Semantics*, *Intent*, and *Aesthetics*. The prompt used for annotation is reported below:

User:
Consider the following action that a photographer can do. Categorize the action into:
FRAMING – Zoom/Crop/Reframe, Angle and Viewpoint, Balance and Symmetry
LIGHTING – Lighting direction/strength/temperature, Exposure adjustment, Shadows control
POSING – Pose adjustments, Facial expressions, Subjects interaction, Clothes
SEMANTICS – Add/remove objects or people, Change background, Include contextual cues
INTENT – Change narrative emphasis, Mood and atmosphere
AESTHETICS – Color grading/filters, Contrast/sharpness, Blur and focus
Here the action: `<input action>`

This setup guarantees consistent labeling across all sub-actions, enabling downstream analysis of category frequencies and co-occurrence patterns.

B. MemCoach Additional Details

MemCoach is a training-free method that applies activation steering to modulate the internal representations of a multimodal model. All experiments were conducted using the PyTorch framework on a single NVIDIA A100 GPU (64 GB).

Below, we provide the implementation details for generating and injecting the steering vector, complementing the description in Sec. 4.2.

Extraction stage. We target the residual module within a specific language Transformer block of the multimodal backbone (*e.g.*, layer $l = 55$ in our best-performing configuration). For a chosen layer l, we register a forward hook to capture the activation tensor $h^{(l)}$ for each input sequence i. Each input sequence, as defined in Eq. 2, is first tokenized into input IDs. For each input i, we compute the mean over the sequence dimension to obtain a single activation representation. No normalization or additional post-processing is applied. Since we operate in batches, we record the starting index of padding tokens and exclude padded positions from the mean computation. No generation is performed during this stage: the model is only run in forward mode. The memorability steering vector $\mathbf{r}^{(l)}$ is computed as described in Sec. 4.2.

Inference stage. Inference requires selecting a steering coefficient α and the target layer l. During the model's forward pass, the vector $\mathbf{r}^{(l)}$ is injected, scaled by α, at the specified layer, following Eq. 4. Injection is applied uniformly along the sequence dimension. The forward computation then proceeds as usual, but with altered activation patterns at layer l, steering the model toward memorability-aware behaviour.

Different models configuration. Tab. 7 reports the optimal layer index l and steering coefficient α identified for the four open-source models we evaluate. Hyperparameters are optimized on a held-out split of the training set.

Table 7. **Optimal steering configuration across models.** Best-performing layer index l and steering coefficient α obtained via hyperparameter tuning for each evaluated open-source multimodal model. We also report the language-model depth (LM Depth) and the corresponding IR score.

Model	Layer	Coefficient	LM Depth	IR
***MemCoach*-IDEFICS**	13	30	32	0.75
***MemCoach*-LLAVA**	20	143	28	0.73
***MemCoach*-QWEN**	12	26	28	0.74
***MemCoach*-INTERNVL**	12	55	36	0.80

Implication of optimizing memorability. Enhancing memorability raises ethical concerns, including potential manipulation, undue influence on viewer perception, and the risk of homogenizing visual expression by favoring conventional cues over diversity. At the same time, in assistive and controlled contexts, *e.g.* education, creative exercises, or personal photography coaching, memorability optimization can enhance communication, reinforce learning recall, and provide actionable guidance without overriding individual intent. Balancing these risks and benefits is essential, emphasizing the need for transparency, user agency, and context-aware application to ensure that memorability interventions remain supportive rather than prescriptive.

C. User Studies

The following experiments are preliminary user studies designed to evaluate the effectiveness of MemCoach. They aim to provide early validation and insights on the effect of MemCoach on human memorability (Sec. C.1), evaluate the effectiveness of the approach for real-life guidance (Sec. C.2), and probe the quality of feedback according to users (Sec. C.3).

C.1. Human Memorability Alignment

To assess how images from different settings drive human memorability, we conduct a memorability experiment with 47 valid users (*avg.* 15.6 annotations/image), following previous work [19].

Experiment setup details. Participants completed a continuous recognition (repeat-detection) visual memory task in which they viewed a stream of images and pressed the space bar whenever they detected a repeat of an image previously shown within the same session. Each session consisted of 150 images presented sequentially, with each image displayed for 600 ms followed by an 800 ms blank interstimulus interval. The sequence contained 40 target pairs, where each target image appeared once and was repeated after 22-93 intervening images. An additional 10 vigilance pairs were included as attention checks, in which the repeat occurred after 1-4 intervening images. The remaining 50 images were fillers presented only once to maintain spacing and reduce predictability of repeats. Participants could respond at any time during the 1400 ms trial window (image plus blank interval). The same task protocol was used across three stimulus variants, *i.e.* MemCoach, INTERNVL3.5 , and source images x_s, with identical timing and sequence structure; only the image sources differed. Sequence order was deterministic and reproducible, using a fixed seed. Participants were instructed not to repeat the game after completing one. If participants missed more than 50% of the vigilance repeats in a run, user results were excluded from the analyses.

Results. Consistent with the editing experiments, Figure 9 shows both INTERNVL3.5 and MemCoach increase the average memorability *wrt* source images x_s, with MemCoach achieving a large margin of improvement. The gap between methods indicates that MemCoach more effectively shifts images toward higher memorability regimes, while INTERNVL3.5 provides only moderate improvements. Overall, results highlight the importance of explicitly injecting memorability-aware signals to achieve stronger memorability outcomes.

C.2. Human-in-the-loop Evaluation

To evaluate real-world effectiveness, we conduct a preliminary human-in-the-loop evaluation measuring whether users can successfully follow the generated feedback to produce more memorable images.

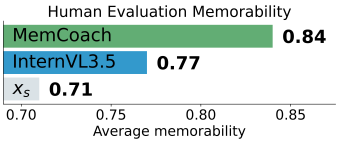


Figure 9. **Human memory performance** across three different image settings.

Experiment setup details. We implemented a mobile app allowing a user to use MemCoach in real-life scenarios. The app presents a live camera viewfinder and allows users to point their phone at any scene of their choice. Upon capturing a frame, users could request either a memorability score alone or a memorability score accompanied by actionable textual feedback generated by MemCoach. The feedback is displayed directly on screen as a natural-language suggestion, guiding the user toward a more memorable composition. Users are free to implement the feedback by adjusting their framing or scene and re-capturing the image, observing how the memorability score changed between attempts. When the feedback pertained to the objects or subjects of the scene, the phone was kept as stable as possible between captures to isolate the effect of compositional changes. Each submitted image, together with its predicted score and generated feedback, was logged for offline analysis. The app requires no installation and runs entirely in the browser, served over a secure HTTPS connection via a public tunnel to ensure accessibility across devices and operating systems.

Results. We collected 27 scenes and evaluated them using the memorability predictor $\mathcal{M}$. Despite the domain shift, MemCoach consistently improves memorability, achieving an IR of 0.52 and a relative gain (RM) of $+4.9\%$. These improvements reflect both the frequency and magnitude of successful memorability enhancements, indicating that the generated feedback effectively guides actionable changes even in previously unseen scenarios. Qualitative inspection confirms that suggestions focus on semantically meaningful adjustments, such as subject positioning, gaze direction, and interaction cues, rather than superficial edits. Overall, these results demonstrate the potential of human-in-the-loop memorability optimization and motivate future exploration of larger-scale, user-centered studies and strategies for robust real-world deployment.

C.3. Feedback Quality Evaluation

To understand the effectiveness of memorability feedback beyond score improvements, we conducted a human study with 28 participants (381 annotations), rating the generated feedback from MemCoach on a 1–5 Likert scale along three dimensions: *Clearness* (clarity of steps), *Relevance* (scene-specificity), and *Feasibility* (realism of applying it).

Experiment setup details. Participants completed a feedback-rating task in which they viewed a source image together with feedback and provided three scalar judgments about the feedback. For each item, participants were asked to rate:

(i) *Clearness*. "How clear are the steps needed to change the photo?" (1: Not at all, 5: Very clear);
(ii) *Relevance*. "How specific is the advice for this scene?" (1: Very general, 5: Very specific);
(iii) *Feasibility*. "How realistic and sensible is this feedback to apply in real-world conditions (e.g., given physical constraints, tools, and context)?" (1: Not realistic or sensible, 5: Completely realistic and sensible).

Each item yielded three numeric ratings, and items were presented in a randomized order with identical instructions and scale anchors across conditions, ensuring consistent evaluation of clarity, scene-specificity, and real-world actionability.

Results. A summary of the experiment performance is reported in Tab. 8. MemBench Oracle achieves consistently high scores across all dimensions, confirming the advantage of access to memorability-aware privileged information. MemCoach maintains strong clearness while improving feasibility, indicating that its suggestions are generally easier to interpret and implement in practice. This gain, however, comes with a slight reduction in scene-specificity, suggesting a tendency toward more generic but broadly applicable guidance. Overall, the results highlight a trade-off between precision and practicality, where MemCoach favors actionable and reliable feedback over highly tailored but less consistently executable suggestions.

Table 8. **Feedback quality results.**

Model	*Clearness*	*Relevance*	*Feasibility*
Oracle	4.19	4.30	4.11
MemCoach	3.96	3.76	4.32

D. Additional Analyses

D.1. Positive and Failure Cases

To analyze the factors underlying successful and unsuccessful memorability feedback, we designed a structured annotation pipeline based on an LLM-as-judge approach that classifies each sample according to a fixed taxonomy.

Experiment setup details. For each sample, the judge receives three inputs: the original source image, the generated feedback text, and the measured memorability outcome (Improved or Worsened) as determined by our memorability predictor $\mathcal{M}$. Based on the outcome, one of two dedicated prompt templates is selected. Both templates frame the model as an expert annotator performing classification rather than creative analysis, but present mutually exclusive category sets tailored to the direction of change. For improved samples, the judge selects among five improvement categories: (i) Posing / Body Configuration, (ii) Framing / Composition, (iii) Lighting / Visibility, (iv) Semantic Clarity, and (v) Emotional or Social Salience. For worsened samples, it selects among five failure categories: (i) Template Over-Normalization, (ii) Distinctiveness Suppression, (iii) Feasibility / Actionability Failure, (iv) Attention Dilution, and (v) Perceptual Degradation. In both cases, the model is required to first produce a one-sentence justification grounded strictly in the visible image content and the provided feedback, serving as a reasoning step before the final category assignment, followed by a single primary category, with explicit prohibitions against introducing new categories, referencing memorability scores, or speculating beyond the observable evidence. Annotations have been generated using GPT-5 MINI [50] model.

Results. Positive effects are driven primarily by posing (74.9%), which plays the dominant role, followed by emotional saliency (23.11%), and to a much smaller extent by framing (1.59%). This distribution indicates that improvements are largely attributable to how subjects are positioned and the emotional cues conveyed, with only marginal influence from compositional framing. Conversely, failures are predominantly caused by over-normalization (76.19%), which emerges as the principal limiting factor, along with distinctiveness suppression (22.22%), which further reduces the effectiveness of the outcome. Additional failure modes contribute only marginally, including perceptual degradation (1.59%), while instances of attention dilution or action infeasibility occur only rarely and have a negligible overall impact.

D.2. Editing Instruction-Following

To decouple feedback quality from the editor's instruction-following, we compare memorability changes from (i) human-in-the-loop ground truth (Sec. C.2) and (ii) edited images using the same feedback. Similar performance is noted (0.52 IR, $+4.9\%$ RM ground truth *vs* 0.55 IR, $+2.19\%$ RM edited) and the moderate correlation ($\rho = 0.51$) between destination image memorabilities confirms the editor as a valid proxy for automated evaluation.

D.3. Leveraging Predictor Biases

To mitigate shortcut learning, we ran a cross-predictor experiment using different memorability predictors for steering, *i.e.* ViTMem (VM) [21], and evaluation (our memorability predictor $\mathcal{M}$, MB). As shown in Tab. 9, using VM alone or the cross-predictor setup (VM $\rightarrow$ MB) consistently confirms the effectiveness of MemCoach, indicating that the observed improvements are not an artifact of a single predictor. These results demonstrate that MemCoach 's feedback remains robust across varying evaluation criteria, effectively enhancing memorability regardless of the specific predictor used, and mitigating concerns of shortcut learning or predictor-specific bias.

Table 9. **Performance of our framework on different settings**. We report the results using different editing models for the evaluation and different memorability predictors.

Model	Edit Model		Mem Predictor	
	Qwen-IE	FLUX.2-k	VM	VM $\rightarrow$ MB
Edit model	0.69	0.68	0.64	0.69
Zero-shot baselines				
GPT-5-MINI	0.78	0.80	0.73	0.76
LLAVA-OV	0.59	0.68	0.76	0.71
IDEFICS3	0.69	0.68	0.73	0.73
QWEN2.5VL	0.54	0.61	0.73	0.69
INTERNVL3.5	0.78	0.74	0.68	0.73
MemCoach				
***MemCoach*-LLAVA**	0.80	0.73	0.69	0.74
***MemCoach*-IDEFICS**	0.85	0.74	0.77	0.76
***MemCoach*-QWEN**	0.82	0.76	0.77	0.81
***MemCoach*-INTERNVL**	**0.88**	**0.83**	**0.83**	**0.82**

D.4. Multiple Editing Models

In Tab. 9, we evaluate MemCoach across multiple editing backbones, including Qwen-Image Edit[1] [70] and FLUX.2-klein[2] [7]. Across all editors, MemCoach consistently increases both the frequency and magnitude of memorability improvements compared to baseline zero-shot and default feedback. Gains are robust to variations in model architecture and editing style, indicating that the proposed approach generalizes across different latent spaces and editing mechanisms. Qualitative inspection confirms that MemCoach directs edits toward semantically meaningful transformations rather than generic low-level changes, producing feedback that is both actionable and visually effective. These results reinforce that the benefit of memorability-aware guidance is model-agnostic and not confined to a specific editing pipeline.

[1]Model card: https://huggingface.co/lightx2v/Qwen-Image-Lightning. Weight name: Qwen-Image-Edit-2509-Lightning-8steps-V1.0-fp32

[2]Model card: https://huggingface.co/black-forest-labs/FLUX.2-klein-9B

D.5. Generalization

As a first study, we focus on human-centric images, given the strong influence of human presence and attributes on image memorability. To assess how well this setting generalizes beyond such content, we conduct preliminary experiments on non-human images using the same editing proxy pipeline introduced in the main paper. Results on objects and landmarks from the Yo'LLaVA dataset [49] indicate that MemCoach performs on par with QWEN2.5VL (0.79 IR), *i.e.*, with no degradation in performance relative to the human domain. Exploring a broader extension beyond human-centric images is left for future work.

Mem score: 0.525

Mem score: 0.992

Feedback a:
1. Rotate the perspective to face forward.
2. Bring the hands up to cover the mouth.
3. Hold a small yellow flower between the fingers.
4. Adjust the hair to frame the face evenly.

Mem score: 0.753

Mem score: 0.803

Feedback a:
1. Adjust the position of the person on the left to face forward with a slight smile.
2. Raise the head of the person on the right and have them look slightly to the side with a gentle smile.
3. Ensure both individuals are standing upright and close together, with the person on the right holding the handlebars of the scooter.
4. Maintain the floral arrangement and attire of both individuals as they are.

Mem score: 0.885

Mem score: 0.965

Feedback a:
1. Adjust the position of the left arm to rest on the hip, ensuring the hand is placed firmly on the hip bone.
2. Shift the weight slightly to the left leg to create a more dynamic stance.
3. Tilt the head slightly to the right to add a subtle angle to the gaze.
4. Ensure the right hand is relaxed by the side, with fingers gently curved.
5. Adjust the lighting to maintain consistency in shadows and highlights across the outfit.

Mem score: 0.578

Mem score: 0.981

Feedback a:
1. Crop the image to focus on the upper body of the person, removing the background elements such as the airplane and the chair.
2. Adjust the angle to a closer, more intimate shot, centering the person's face and upper torso.
3. Ensure the person's hand is positioned near the face, with the index finger touching the lips, maintaining the same pose but in a tighter frame.
4. Remove any elements that are not part of the person's attire or immediate surroundings, such as the grass and distant trees.
5. Enhance the lighting to create a softer, more even illumination on the person's face and clothing.

Figure 10. A set of qualitative examples from MemBench.

Mem score: 0.716

Mem score: 0.980

Feedback a:

1. Stand up from the seated position on the railway track.
2. Hold a wicker basket with both hands in front of the body.
3. Adjust the hair to fall naturally over the shoulders.
4. Shift the gaze to the side while maintaining a smile.
5. Ensure the background changes to a green train car with visible rust and metal details.

Mem score: 0.710

Mem score: 0.752

Feedback a:

1. Adjust the person's posture to a standing position.
2. Raise both arms to hold flowers above the head.
3. Ensure the person is surrounded by lush greenery.
4. Remove the basket and gardening tools from the scene.
5. Adjust the lighting to create a brighter and more vibrant atmosphere.

Mem score: 0.807

Mem score: 0.892

Feedback a:

1. Reposition the couple from a seated to a lying down pose on a grassy surface.
2. Adjust the angle of the shot to an overhead view.
3. Remove the stone wall and window background, replacing it with a grassy area.
4. Ensure the bride's dress and veil spread out naturally on the grass.
5. Place the groom's jacket and pants neatly on the grass beside them.
6. Position the bride's bouquet on the grass near her hand.

Mem score: 0.749

Mem score: 0.829

Feedback a:

1. Adjust the position of the person so they are standing more centrally within the frame.
2. Shift the perspective slightly to the right to include more of the water and the sculpture on the left.
3. Reduce the brightness and contrast to create a softer, more subdued lighting effect.
4. Reposition the person's arm so it is relaxed by their side, not holding onto the structure.
5. Ensure the reflection on the water is more prominent by adjusting the angle of the light source.

Mem score: 0.798

Mem score: 0.990

Feedback a:

1. Reposition the individuals so that they are facing each other, with one person lifting the other into their arms.
2. Adjust the arms so that the lifted person's arms are wrapped around the other person's neck.
3. Ensure the lifted person's legs are bent at the knees and held by the other person.
4. Shift the gaze of both individuals to look at each other affectionately.
5. Maintain the scenic background with mountains and clouds, but adjust the angle slightly to accommodate the new pose.

Figure 11. A set of qualitative examples from MemBench.