Cycle-Consistent Tuning for Layered Image Decomposition

Supplementary Material

A. Implementation Details

A.1. Training and Inference Details

We implement our approach with PyTorch. Our data curation involves a 5-round iterative generation stage (∼5k samples) followed by a 5-round self-improving stage (∼5k samples), yielding a final corpus of ∼10k samples.

In each round of iterative data collection, the model is optimized for 4K steps with the learning rate set to 1. The LoRA rank and alpha is set to 32 for the IC-LoRA adapter. We train the cycle-consistent model for 5K to 10K steps in each round, with the step count increasing across rounds as the dataset grows. The learning rate gradually decays from 1 to 0.5. The LoRA rank and LoRA alpha are also set to 32.

We use Qwen-VL-7B [**?**] as a VLM-based filter to evaluate and select high-quality generated examples. Inspired by Zhang et al. [**?**], the newly added high-quality samples in each round are assigned double the sampling weight.

Experiments are conducted on eight NVIDIA L40 GPU. The inference takes 35 seconds per image with 50 steps, which is within the normal inference time of Flux-Fill. Notably, our method does not require multiple runs. The logo and object are produced together in one generation.

A.2. Cycle Consistency Loss

In our proposed cycle consistency loss, consider track 1, the model first performs decomposition and then composition. The outputs from the first stage serve as the inputs to the second stage. Since flow-based Diffusion Models do not directly predict images but velocity fields instead, we must convert the first-stage outputs into a clean latent required for the second-stage. To achieve this, in our implementation, we approximate a clean latent after the first stage using the following procedure:

$$\overline{x}^0 \approx x_t - t \cdot v_\theta\left([x_t, \mathcal{E}_{img}(X), M_d], t, \mathcal{E}_{txt}(T_d)\right), \quad (1)$$

where v_θ denotes the model, $\mathcal{E}_{img}$ and $\mathcal{E}_{txt}$ are image and text encoders, M_d and T_d are binary mask and text prompts for the decomposition task. Below we show that this approximation is reasonable.

During each training step, a random timestep $t \in [0, 1]$ is sampled to construct a noisy latent by linearly mixing the clean latent x_0 and the random noise x_1 by:

$$x_t = (1 - t) \cdot x_0 + t \cdot x_1. \quad (2)$$

The model then predicts the velocity v conditioned on x_t, together with auxiliary inputs including the masked image, the binary mask, etc. The objective is encouraging the model prediction v to match the direction between x_1 and x_0 by minimizing the following term:

$$\mathcal{L} = ||v - (x_1 - x_0)||_2^2. \quad (3)$$

When this objective becomes sufficiently small (i.e., closely heading zero), combining Equation (2) and Equation (3) yields an approximation of the clean latent:

$$x_0 \approx x_t - t \cdot v. \quad (4)$$

Given this approximation, we can recover an estimate of the clean latent and feed it into the second stage, which is both computationally efficient and empirically stable. In practice, we sample two timesteps independently for track 1 and track 2 to strengthen the cycle constraints from different noise levels. Algorithm 1 provides a Pytorch-like pseudo-code to calculate the cycle consistency loss in one step.[1]

B. LLM/VLM Prompts and Templates

B.1. Seed Data Generation

Given a isolated logo image and a clean object image (which is also synthetic), we use the following instruction to generate the seed dataset under the assistance of GPT-4o.

> *I will give you two images, showing a logo, and a product, respectively. Your task is to generate the following image: an image of the product with logo printed.* {*Logo Image*} {*Object Image*}.

B.2. Iterative Data Generation

LLM-Generated Prompts. In the iterative data generation process, we adopt IC-LoRA [**?**] to collect sufficient 1×3 grid-like images. After one training round is finished, we use the following instruction to collect a large and diverse set of structured text prompts using Qwen3 [**?**].

> *You are a text-to-image prompt generation assistant. Your task is to generate diverse and high-quality text prompts for branded product decomposition images.*
>
> *Below are some existing examples that have been successfully generated:* {*Last five generated prompts*}.
>
> *Your goal is to create a new prompt that is different from these examples in terms of: Product type/category; Brand name and style; Logo design elements; Background setting and color.*
>
> *The generated prompt should maintain the format: A three-panel image grid illustrating the decomposition of a branded product.[LEFT] [MIDDLE] [RIGHT].*

[1] In this pseudo-code, diffusion latents x_0, x_1, x_t are written as x^0, x^1, x^t to reserve subscripts for other indices.

ALGORITHM 1: Pytorch-like pseudo-code for the calculate of cycle-consistent loss in one training step

```
# X_{1×3}:  ground truth image
# M_d, M_c:  binary masks
# τ_d, τ_c:  text embeddings

Function get_pred(X, M, t, τ, x^0=None, x^1=None)
    # prepare latents
    if x^0 is None then
        x^0 ← enc_img(X)
    end
    if x^1 is None then
        x^1 ← N(0, I)
    end
    x^t ← (1 − t) · x^0 + t · x^1
    c ← [enc_img(M ⊙ X), M]

    # predict velocity
    pred ← transformer(x^t, c, t, τ)
    tgt ← x^1 − x^0
    x̄^0 ← x^t − t · pred

    return (pred, tgt, x̄^0, x^1)
end

Function cycle_step(X_{1×3}, M_d, M_c, τ_d, τ_c)
    # sample timesteps
    t_1, t_2 ← σ(N(0, 1))

    # track 1:  decompose
    (p_d, g_d, x̄^0_d, x^1_d) ← get_pred(X_{1×3}, M_d, t_1, τ_d)
    # track 2:  compose
    (p_c, g_c, x̄^0_c, x^1_c) ← get_pred(X_{1×3}, M_c, t_2, τ_c)
    # track 1:  decompose & compose
    (p̃_c, ·, ·, ·) ← get_pred(X_{1×3}, M_c, t_2, τ_c, x̄^0_d, x^1_c)
    # track 2:  compose & decompose
    (p̃_d, ·, ·, ·) ← get_pred(X_{1×3}, M_d, t_1, τ_d, x̄^0_c, x^1_d)

    # reconstruction loss
    L_rec ← mse(p_d, g_d) + mse(p_c, g_c)
    # cycle consistency loss
    L_cyc ← mse(p_d, p̃_d) + mse(p_c, p̃_c)

    return L_total ← L_rec + L_cyc
end
```

Templated Prompts. During the self-improving process, we no longer construct new 1×3 ground-truth triplets beyond those obtained in the iterative data generation phase. Instead, we focus on synthesizing increasingly diverse composite images. These composites are then fed into our cycle model, which performs decomposition and recomposition to yield supervision with higher quality and diversity.

> *A high-quality photo of a {material} {function} featuring a prominently visible logo of type {logo type}. The logo is {logo style}, {logo content}, and rendered in a {logo color scheme} scheme. It appears {logo texture}, positioned at the {logo placement} of the product, naturally following the {geometry} shape and {surface state} texture of the surface. The logo conforms realistically to the material — showing subtle distortions, shading, and reflections that match the underlying {material} surface. The overall composition emphasizes the tactile interaction between the logo and the object, revealing slight curvature, depth, and perspective alignment consistent with a real printed or embedded mark.*
>
> *The object is captured with a {viewpoint} viewpoint and is framed in a {composition style} composition, under {lighting type} lighting from the {lighting direction} direction, featuring {lighting quality} illumination in a {environment} setting. The background is {background}. The exposure is {exposure} with a {color balance} color balance. This image is sourced from a {data source}, showcasing realistic physical integration of the logo with the product surface.*

B.3. VLMScore Evaluation

We adopt Qwen2.5-VL [**?**] to evaluate the decomposition results automatically using the following instruction.

> *Your task is to analyze a 1x3 grid image containing three square images arranged in one row (left, middle, right).*
>
> *Answer these four questions by outputting scores ranging from 1-5 (1 for "No", 2 for "Probably No", 3 for "Uncertain", 4 for "Probably Yes", and 5 for "Yes"):*
>
> *1. Does the middle image show only a logo with no product visible?*
> *2. Is the logo in the middle image the same as the logo visible in the left image?*
> *3. Does the right image show only a product with no logo visible?*
> *4. Is the product in the right image identical to the product shown in the left image (same item, shape, design, structure)?*

The results for the four questions above are averaged over the entire test set and reported as quantitative metrics for logo isolation, logo consistency, object isolation, and object consistency, respectively.

C. Additional Results

C.1. Decompose&Compose

Figure S1 illustrates the effectiveness of decompose & compose operation in the self-improving process. Before the cycle model is introduced, the generated samples may exhibit inconsistent decomposition results (red boxes), which lead to imperfect supervision signals. After enabling the recomposition, the obtained grid provides consistency logo

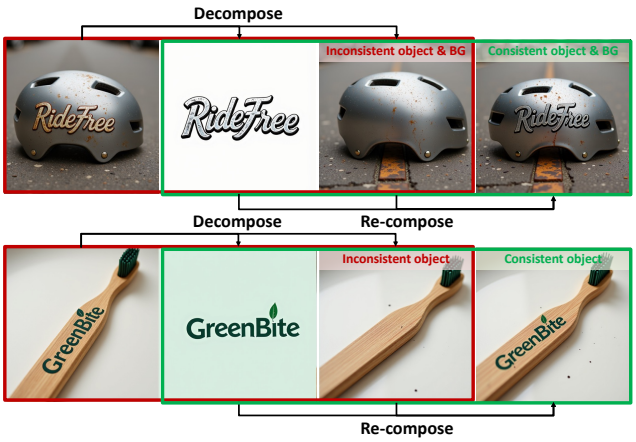

Figure S1. Illustration of the decompose&compose operation. With the bidirectional generation capability of the cycle model, we are able to collect high-quality samples with improved consistency.

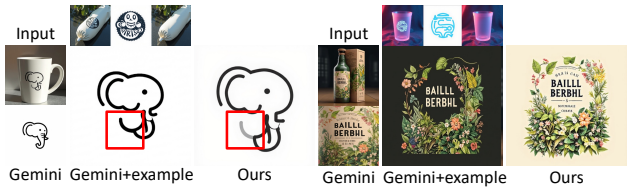

Figure S2. Comparison with Gemini with additional example.

and objects (green boxes), allowing us to create higher-quality samples to refine the training set.

C.2. Providing Additional Context to Baselines

In Figure S2, we evaluated Gemini both with and without an in-context demonstration, and observe that providing additional context improves its decomposition quality. Importantly, even under this strengthened setting, our method continues to produce more coherent decompositions in the presence of non-linear layer interactions.

C.3. Comparison with Baselines on Composition

Figure S3 shows additional comparison on compositing objects and logos with Gemini and GPT-4o. Compared with Gemini, our method is less affected by object color (in the first row) and perspective distortions (in the second row). Relative to GPT-4o, we obtain more consistent logo isolation and recombination.

C.4. Comparison with Baselines on Decomposition

Figure S4 presents additional comparison of our approach with the baseline methods. Our method produces more faithful logo and cleaner object layers under a wide range of

Figure S3. Comparison on logo-object composition. The first two rows are object and logo image, followed by outputs from Gemini, GPT-4o and our approach.

challenging cases, including non-uniform lighting, perspective distortion, complex surfaces and transparent objects.

C.5. Decomposition on Synthetic Images

Figure S5 shows additional decomposition results on our synthetic test dataset. These input images offer various challenging scenarios in lighting, perspective, material, and transparent objects. The results produce clean and consistent layers, demonstrating the effectiveness of our approach.

C.6. Decomposition on Real-world Images

Figure S6 presents additional results on real-world photographs. In the first five rows, we show decomposition results on well-known brand logos. The next five rows show decomposition results on less common logos.

C.7. Intrinsic Decomposition

Figure S7 shows more results on intrinsic decomposition, where our approach produces reasonable layers of albedo and shading. These results further illustrate that the same in-context generation mechanism can generalize to physics-related decompositions without hand-designed priors.

C.8. Foreground Background Decomposition

Figure S8 presents additional results on foreground-background decomposition. Our approach successfully isolates the salient objects while producing coherent background layers with the effects of objects (e.g., shadows) removed. We believe these results highlight the potential of large foundation models to handle graphic tasks that require contextual consistency with internal correspondence.

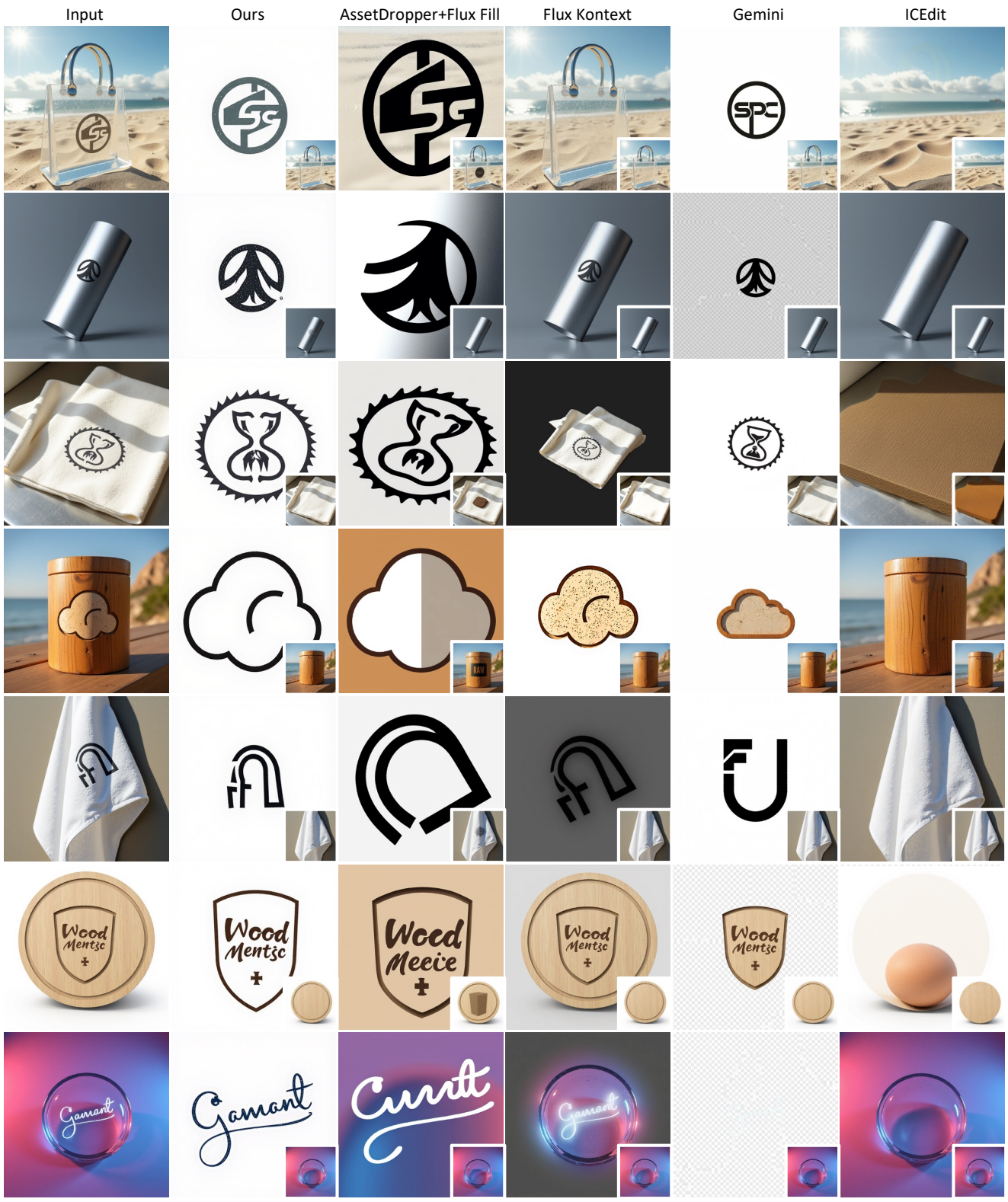

Figure S4. Additional qualitative comparison on challenging scenarios on synthetic data. The first column shows the inputs, while the following columns present results from our approach and baselines. The decomposed objects appear at the bottom-right of each sample.

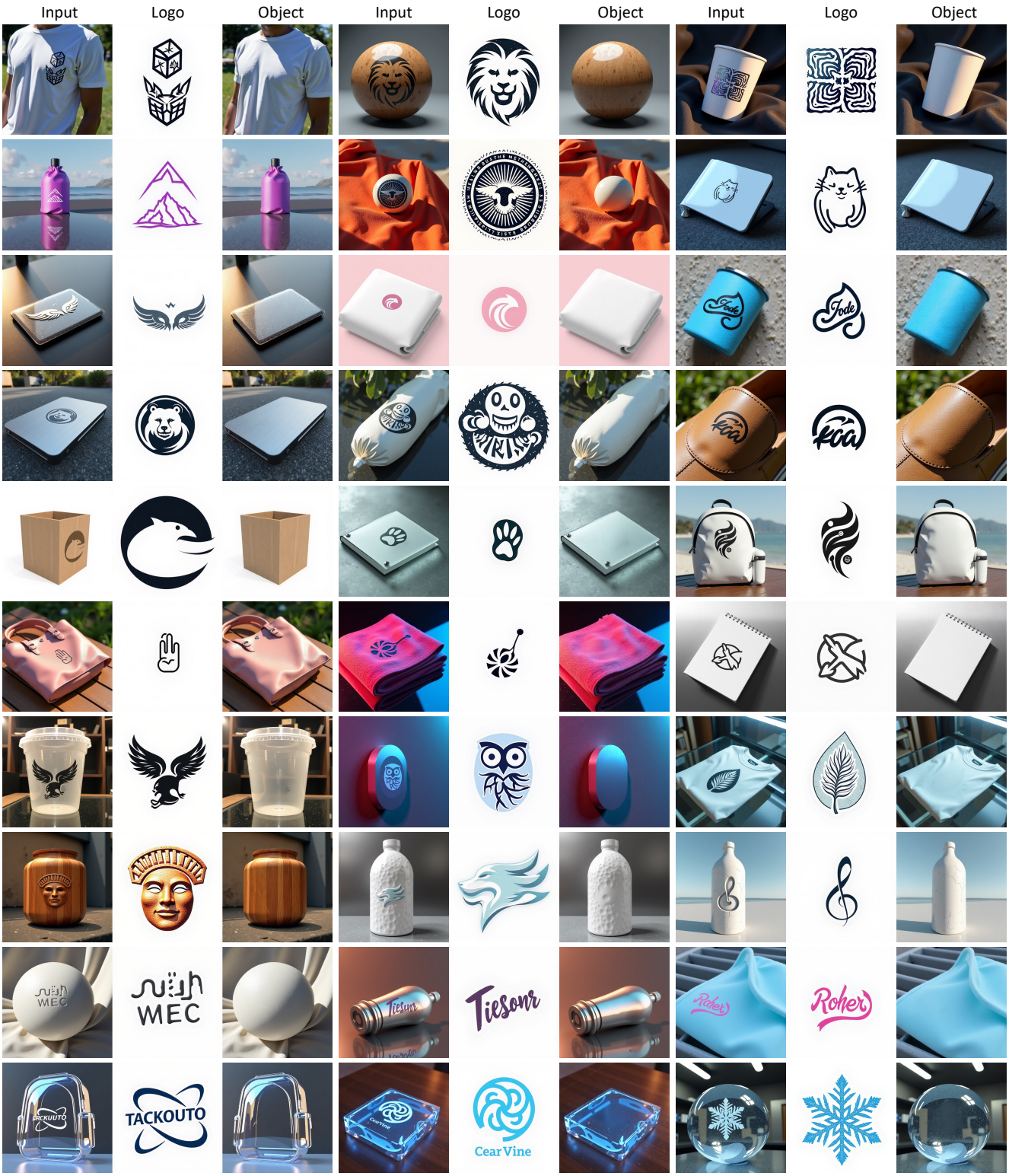

Figure S5. Additional decomposition results on synthetic images.

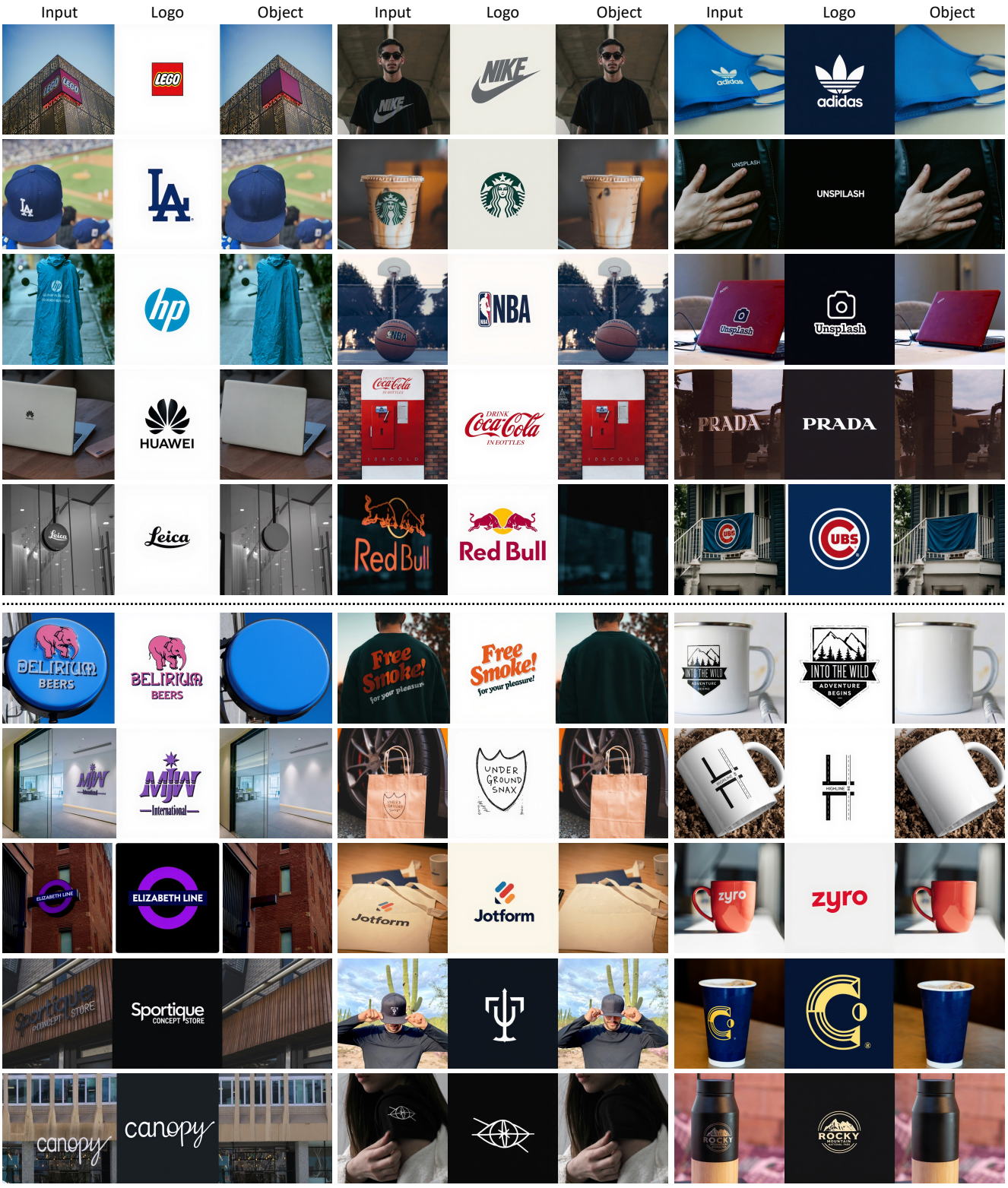

Figure S6. Additional decomposition results on real images. The first five rows show decomposition results for several well-known logos, while the last five rows present decomposition results for other in-the-wild logos.

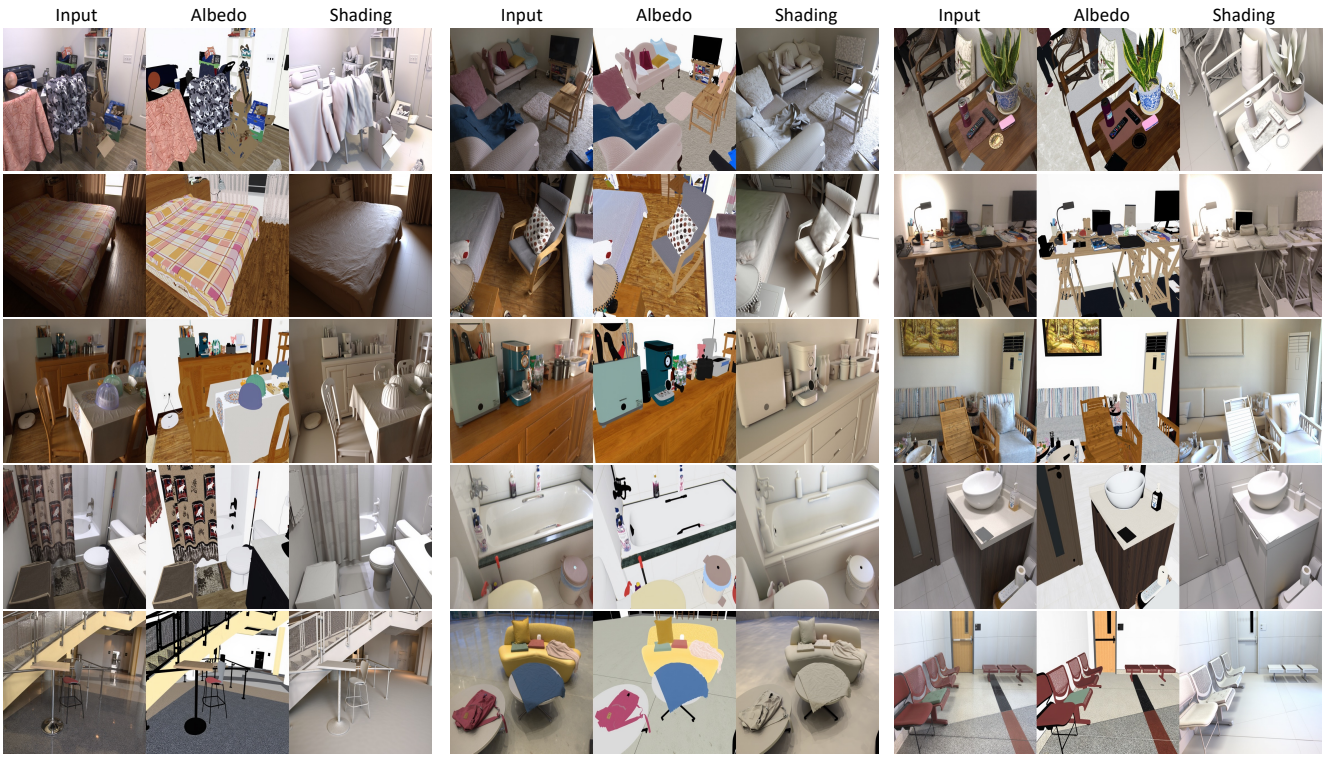

Figure S7. Additional results on intrinsic decomposition.

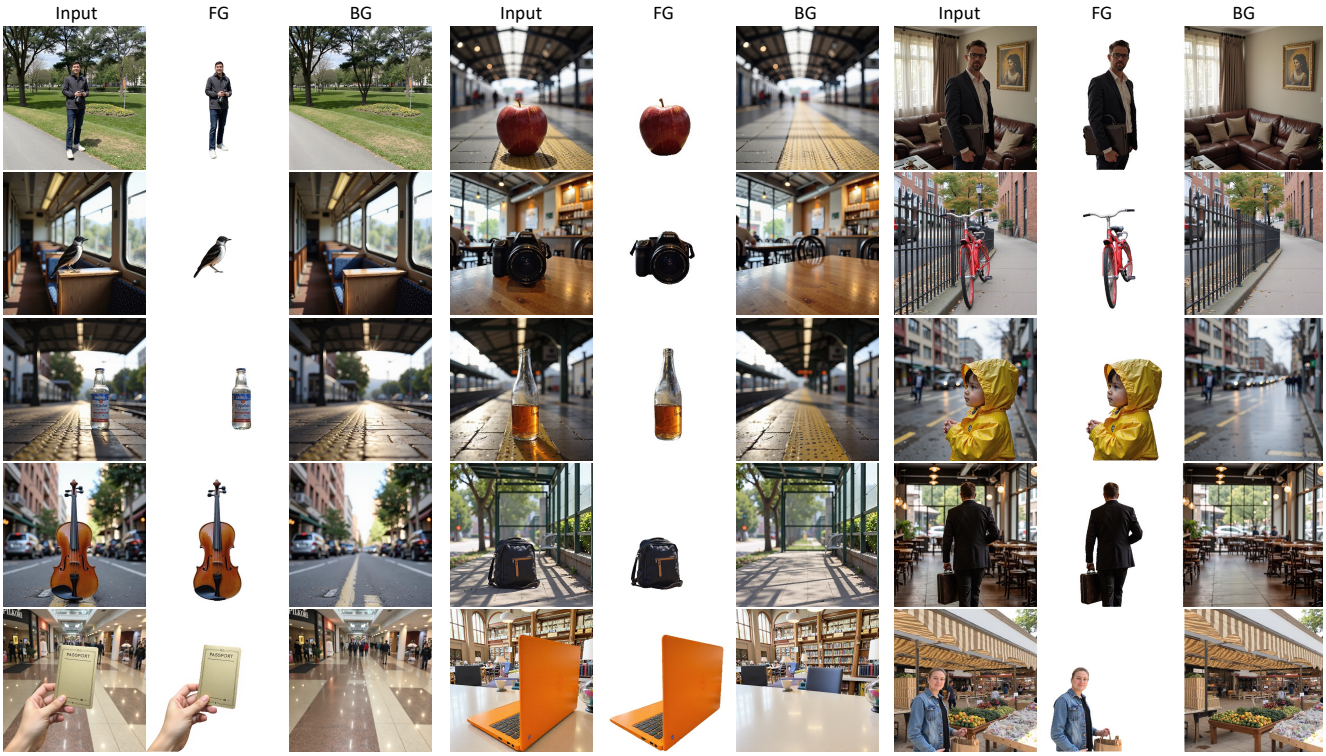

Figure S8. Additional results on foreground-background decomposition.