ShadowDraw: From Any Object to Shadow–Drawing Compositional Art

Supplementary Material

1. Supplementary Material Overview

In this supplementary material, we provide additional implementation details and present extended qualitative results. Check out our project page for more results and an end-to-end real-world demonstration of our pipeline!

2. Implementation Details

2.1. Training Data Construction

We construct a paired dataset of shadow contours and line drawings to train our line drawing generation model. The pipeline proceeds as follows. We first generate 100 line drawings using GPT-4o, prompting it with descriptions of everyday objects. We retain only those drawings that contain at least one closed region bounded by strokes. Next, we fine-tune FLUX-1-dev LoRA [21] on this filtered subset and employ it to synthesize an additional 10K line drawings from GPT-4o-generated prompts describing everyday subjects. Finally, we apply OpenCV's `FindContour` algorithm to extract closed regions from the synthesized drawings and use a greedy merging strategy that iteratively combines the two smallest connected regions until only four remain. The closed contours of these merged regions (and their union) serve as the shadow contour conditions for training.

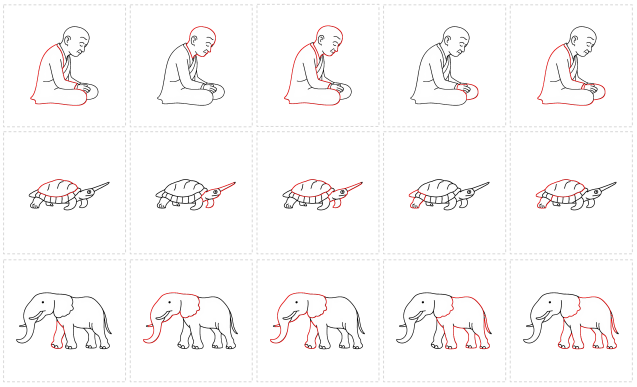

Figure 8. Examples of training data pairs. Each row shows a line drawing generated by our finetuned FLUX model, with different closed contours extracted from it. Each image forms a training pair, where the red contour is used as the condition and the full line drawing serves as the target.

Dataset Statistics. Our final line drawing dataset contains 10K image–prompt pairs spanning 648 distinct subjects. Owing to the diversity of the training distribution, the resulting model is not confined to a fixed taxonomy and generalizes to novel, out-of-distribution concepts. As one example, it can generate a line drawing of Pikachu conditioned only on the stroke contour of its head (Figure 9), despite Pikachu not appearing in the training set.

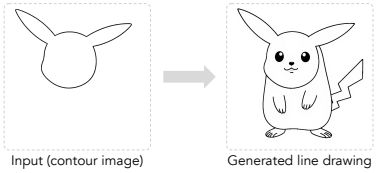

Figure 9. **Generalization of the line drawing generation model.** Although Pikachu is absent from training, our model can synthesize a coherent line drawing when conditioned on a partial contour (here, the head), demonstrating transfer to unseen subjects.

2.2. Visual Prompt Proposal

Vision–language models are highly sensitive to input formatting, and poorly structured inputs often result in uninformative or inconsistent outputs. To address this, we carefully design a system prompt that guides the VLM, specifically GPT-4.1, to generate detailed, semantically meaningful descriptions of the provided stroke and its role in the complete line drawing. The full system prompt is given below.

```
You are a skilled artist specializing in
    expressive, imaginative, and visually
    striking minimalist line drawings. You will
    be shown an image containing a contour. Your
    task is to interpret this contour and create
    a complete line drawing, using the provided
    contour as the core expressive element of
    your composition. The subject you draw should
     be a character, either a human, an animal,
    or a cartoon or anthropomorphic figure.

# Instructions

First, analyze the contour to identify its
    function in the drawing. Follow these steps:

1. Analyze the contour's geometry and position on
     the canvas.

2. Determine the subject of the line drawing and
    which major, prominent body part (such as
    body, head, face) or clothing (such as skirt
    or dress) the contour outlines. Do not
    describe small or less essential features,
    such as hands, tails, wings, or beaks.
    Specify the subject in precise terms:
For people, use identifiers (e.g., man, woman) or
     vocations (e.g., dancer, sailor, guitarist);
For animals, name the species (e.g., bird, fish,
    cat, dog);
For cartoon or anthropomorphic characters, name
    the type (e.g., ghost, robot, cookie
    character, book character).

3. Explain your reasoning in detail, including
```

```
    the stroke's shape, its position on the
    canvas, and why it is a good fit for the
    composition.

Next, write a description of the complete drawing
     without referencing the provided contour,
    following this structure:

1. Opening: A minimalist line drawing of a [
    character] [in a pose or with an expression],
     matching your earlier interpretation.

2. Physical description: The [character] has [
    facial feature or expression] and wears [
    clothing or accessories].

3. Object or motion (optional): The [character]
    is [doing something, holding something, or in
     motion].

4. Gesture or interaction (optional): Further
    describe the subject's gesture or interaction
     with their environment.

5. Conclude with a style remark: The style is [
    adjective(s)], [additional notes about
    technique or focus].

# Format requirement

1. Separate the two parts with a blank line.

2. Do not use numbering, bullet points, or extra
    formatting.

3. Strictly follow this structure without
    additional comments:

The provided contour shows an outline of the [
    specific body part or clothing] of a [
    character]. The reason is [contour geometry
    interpretation]. [Additional reasons for your
     interpretation].

A minimalist line drawing of a [character] [in a
    pose or with an expression]. The [character]
    has [facial feature or expression] and wears
    [clothing/accessories]. [Optional action or
    motion]. [Optional gesture or interaction]. [
    Artistic style remark].
```

Listing 1. System prompt for creating the textual description of the intended line drawing in the visual prompt proposal stage.

2.3. Baseline Execution Details

Here we describe how we use Gemini Flash 2.5 Image (a.k.a. nano banana) [13] to generate shadow–drawing art. In the *object–shadow* version, we provide the model with the object–shadow composite and the line drawing description produced by our approach, and ask it to directly generate a shadow–drawing composition. In the *shadow contour* version, we instead provide the shadow contour and the line drawing description, prompting the model to complete the drawing. As in our framework, we then remove the input shadow contour from the generated drawing, reinsert the 3D object, and render the final composition, where the cast shadow completes the drawing.

2.4. Discussion on the Evaluation Algorithm

Visual Illustration. Figure 10 illustrates how our evaluation and ranking algorithm selects high-quality shadow-drawing compositions. In the first stage, the VQA-based verification discards incoherent cases (e.g., when the shadow stroke does not correspond to the intended body part of the character). In the second stage, the shadow contribution assessment compares complete and contour-removed versions, ensuring that the shadow meaningfully enhances the drawing. As shown, this two-step process ranks plausible results higher while filtering out cases where the shadow plays only a minor or misleading role. Overall, the pipeline balances semantic alignment, structural coherence, and visual quality, producing consistent and interpretable rankings.

Analyses. Evaluating our generated shadow–drawing compositions is inherently challenging, as their abstract and artistic qualities often resist objective evaluation. To rigorously assess effectiveness, we designed two complementary user studies based on pairwise preference judgments. (1) For each object, we randomly select one result from the top-4 ranked outputs and another from the remaining, asking evaluators to choose their preferred composition; and (2) we randomly select two results among the top-3, where differences are more subtle. In both cases, evaluators may also indicate that neither is preferable. Altogether, we collected 2,000 preference pairs from 10 annotators for the first study, and 2 labels per top-3 pair across all 200 objects for the second study.

In the first study, our ranking algorithm achieves a strong alignment with human judgment, agreeing on 63.5% of pairs, disagreeing on only 11.0%, and with 25.5% of cases marked as no clear preference. These results indicate that our automated ranking provides a reliable proxy for subjective evaluation in the broader design space. In the second study, where all candidates are already of very high quality, the agreement rate with human preference naturally decreases to 39.8%, with another 24.3% of cases judged as indeterminate. Crucially, the agreement between two independent sets of human annotations is itself only 44.5%, underscoring the intrinsic subjectivity of evaluating artistic compositions. Taken together, these findings suggest that while perfect alignment is unattainable in such a subjective domain, our ranking system performs comparably to human consensus and thus provides a practical, scalable tool for curating shadow–drawing art.

2.5. Subject-Specified Generation

We also demonstrate that our pipeline supports user-specified subject control through prompt editing. In practice, the desired subject is directly specified in the system prompt used

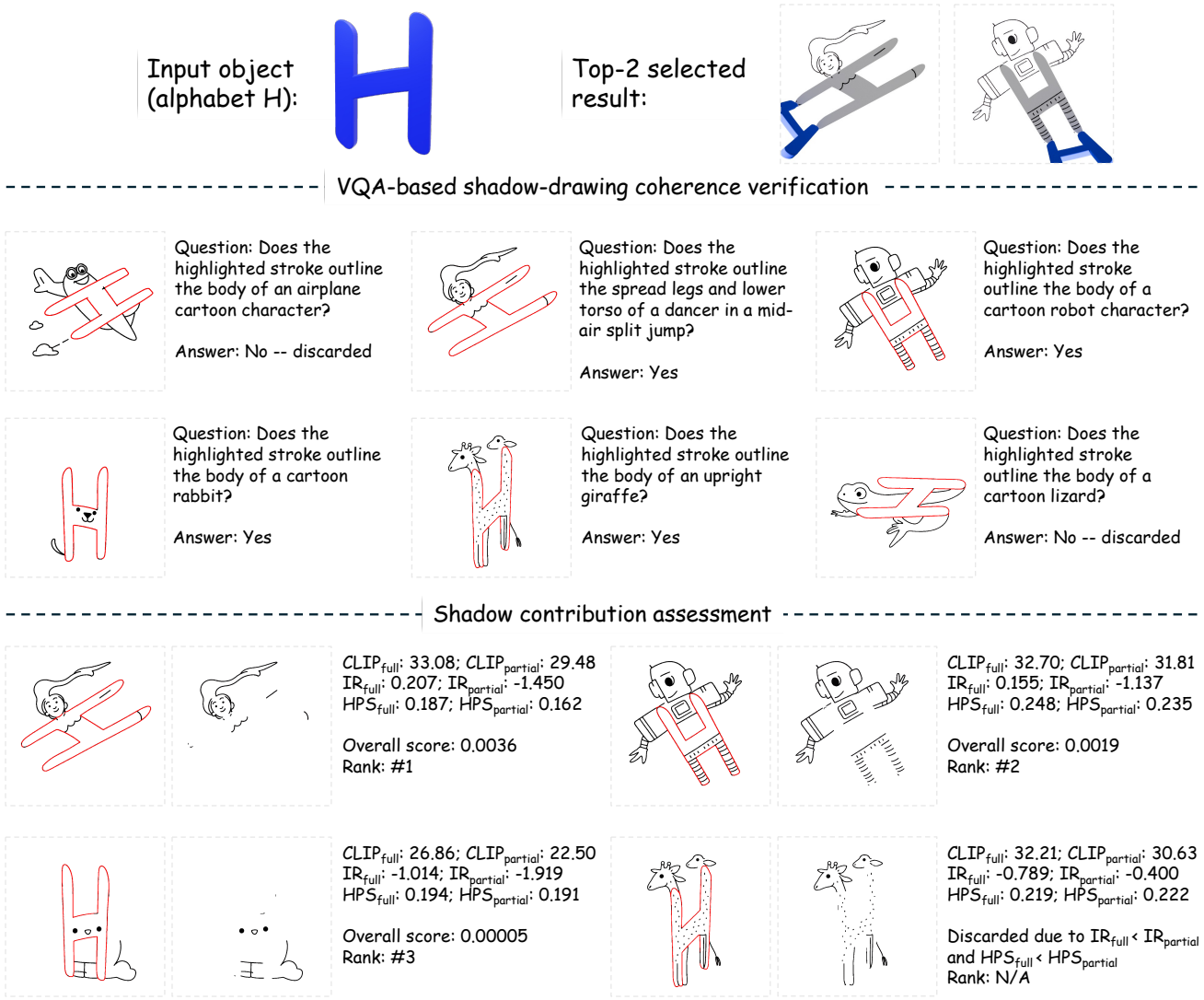

Figure 10. Illustration of the evaluation and ranking process, which discards incoherent cases and preserves only those where the shadow meaningfully contributes to the composition.

by the VLM during visual prompt proposal. However, we observe that certain object–subject combinations are inherently incompatible: the geometry of the 3D object may not afford a shadow stroke that can be meaningfully integrated into the specified concept, leading to unsatisfactory results. Examples are shown in Fig. 11.

2.6. Animated Shadow-drawing Art

As mentioned in Sec. 4.3, our framework supports animated objects without requiring additional training. We provide the implementation details as follows. For each candidate configuration, we render five keyframes of the animation and extract their shadow contours. To preserve temporal information, we overlay the strokes into a single composite image, assigning distinct colors to each frame so that the VLM can recognize temporal variation and generate a prompt.

A critical step in this pipeline is the construction of a binary mask to restrict where strokes may be placed. Without such a constraint, strokes might appear in regions where shadows change across frames, leading to incoherent results. To build the mask, we proceed as follows. First, we render the shadow silhouettes of all five keyframes. Pixels that are covered by shadows in every frame are designated as the *static region*, while pixels that are covered in at least one but not all frames are designated as the *dynamic region*. Next, for each pixel on the canvas, we compute its distance to both the static and dynamic regions. If a pixel is closer to the dynamic region than to the static region, we mask it out, prohibiting stroke placement in that location. Intuitively, this rule ensures that strokes are confined to stable shadow

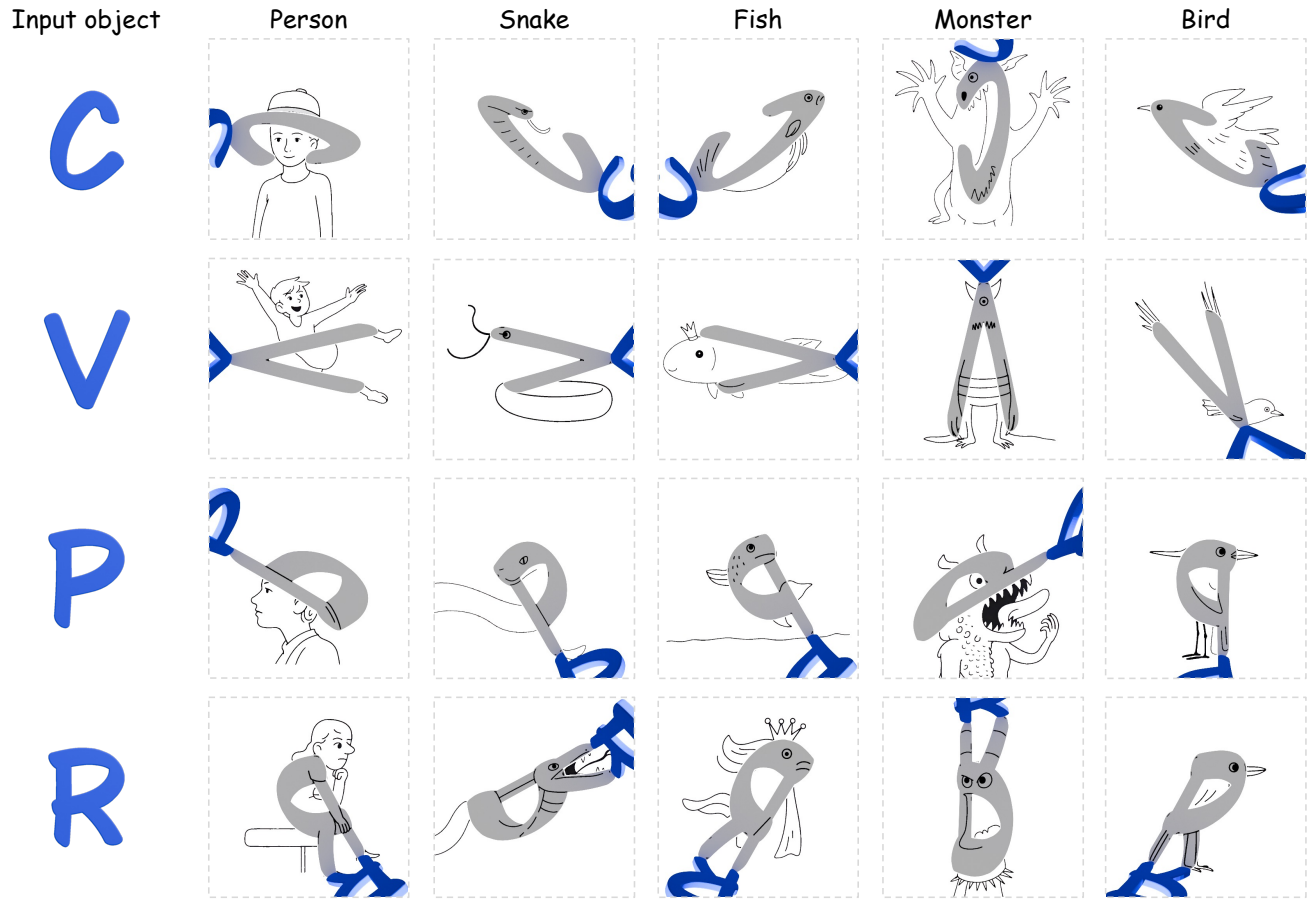

Figure 11. **Shadow–drawing compositions with user-specified subjects.** We show results for alphabet-shaped objects (C, V, P, R) conditioned on the subjects *ghost*, *fish*, *person*, and *bird*. While the pipeline supports flexible subject control through prompt editing, certain object geometries inherently limit the achievable compositions, leading to occasional unsatisfactory outcomes.

regions and avoid areas subject to temporal variation.

Once the mask is established, the generation process follows the same procedure as in the static-object setting. The shadow contour and the corresponding VLM-generated prompt are fed to the line drawing generator, and masked regions are excluded during synthesis. Finally, the animated object is reinserted into the scene and rendered across frames, with its shadow complementing the static line drawing to form a temporally consistent shadow–drawing composition.

We evaluate this pipeline using animated objects from Objaverse. Qualitative results are presented in Fig. 6(d) (main paper) and Fig. 2.6. As shown, our system successfully generates line drawings that remain coherent with dynamic shadows, demonstrating the ability of our approach to extend from static to temporally varying scenes.

2.7. Runtime Analysis

The only trainable component in our framework is the line drawing generation model based on FLUX.1-Canny [21]. Specifically, we train a DoRA [29] adapter on all queries, keys, values, and MLPs of the backbone diffusion transformer. We use the Adam optimizer with a constant learning rate of 10^{-4} and train for roughly 12 hours on 8 A6000 GPUs. At inference, the dominant cost arises from diffusion sampling, taking about 40 seconds per image with 30 steps. For a single object, generating line drawings for 48 sampled scene configurations requires approximately 30 minutes, and the full pipeline completes in about 35 minutes on a single A6000 GPU. Reducing the number of inference steps from 30 to 10 lowers latency to around 15 minutes with minimal quality degradation. Because the process is fully parallelizable, latency can be reduced to under 5 minutes on standard 8-GPU nodes. Further acceleration may be achieved by distilling the multi-step diffusion process into a one- or few-step generator.

3. Limitations

While our method enables diverse and visually engaging shadow–drawing compositions, several limitations remain. First, the quality of results is closely tied to the intrinsic shape of the object: some objects inherently produce shadows that are either visually uninteresting or too ambiguous

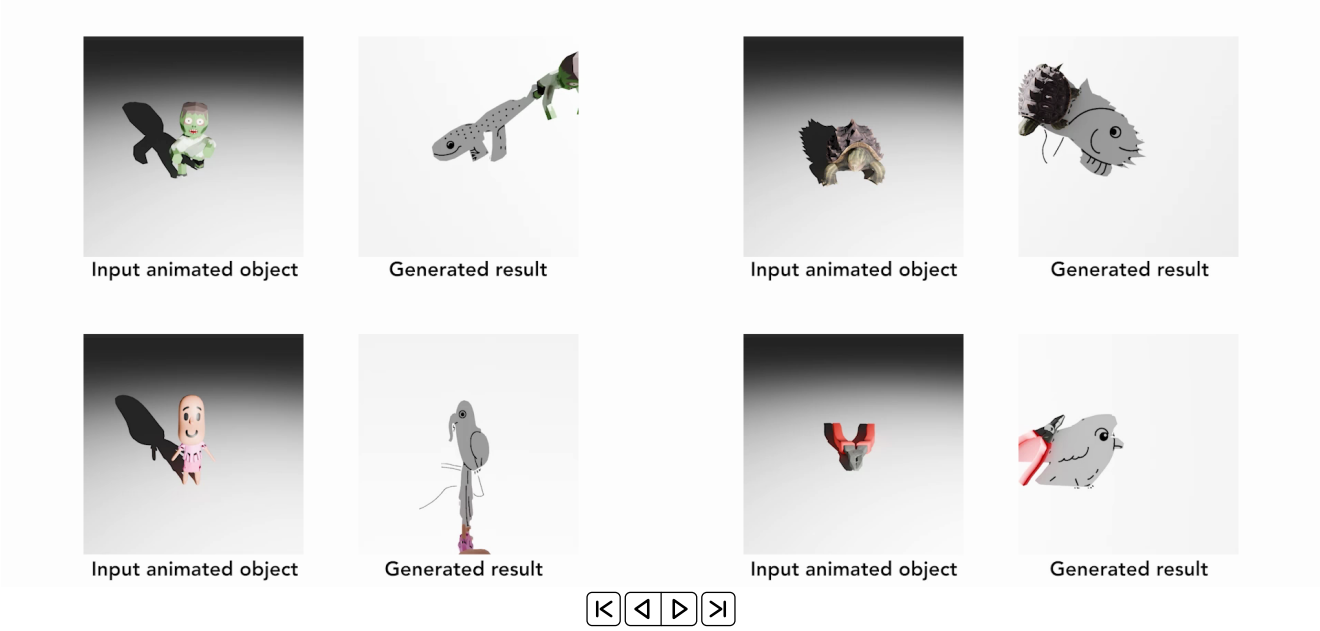

Figure 12. **Animated shadow–drawing art.** Line drawings generated with our pipeline remain coherent as dynamic shadows complete the composition across frames. *Best viewed in Adobe Acrobat Reader for the embedded animation.*

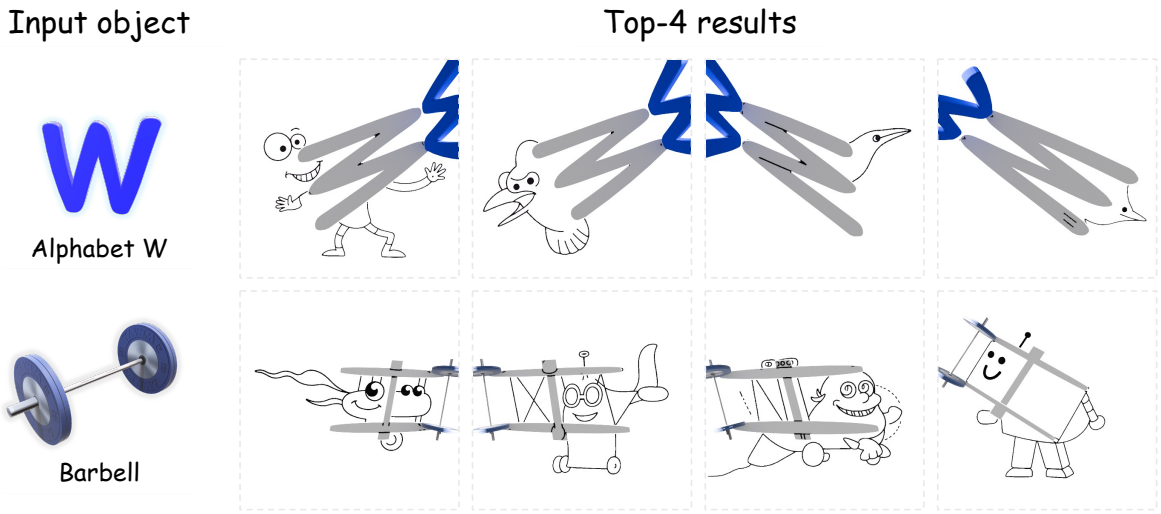

Figure 13. **Failure cases.** Some objects produce shadows that are ambiguous or uninformative, making it difficult for our system to produce meaningful shadow-drawing compositions.

to interpret, regardless of lighting or pose. As illustrated in Fig. 13, such cases often yield shadows that lack recognizable structure or fail to align meaningfully with the generated drawing. Second, the joint search and generation process over scene parameters introduces noticeable runtime overhead. Although this procedure is necessary to explore the large design space, generating results for a single object still takes a relatively long duration. Finally, while our ranking algorithm is generally effective at surfacing strong candidates, it is not flawless. In practice, users may still need to examine multiple outputs to identify the most compelling result. Addressing these limitations through richer shadow descriptors other than fractal dimension, more efficient search strategies, and refined ranking or user-in-the-loop mechanisms represents promising directions for future work.