VIRAL: VISUAL IN-CONTEXT REASONING VIA ANALOGY IN DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review

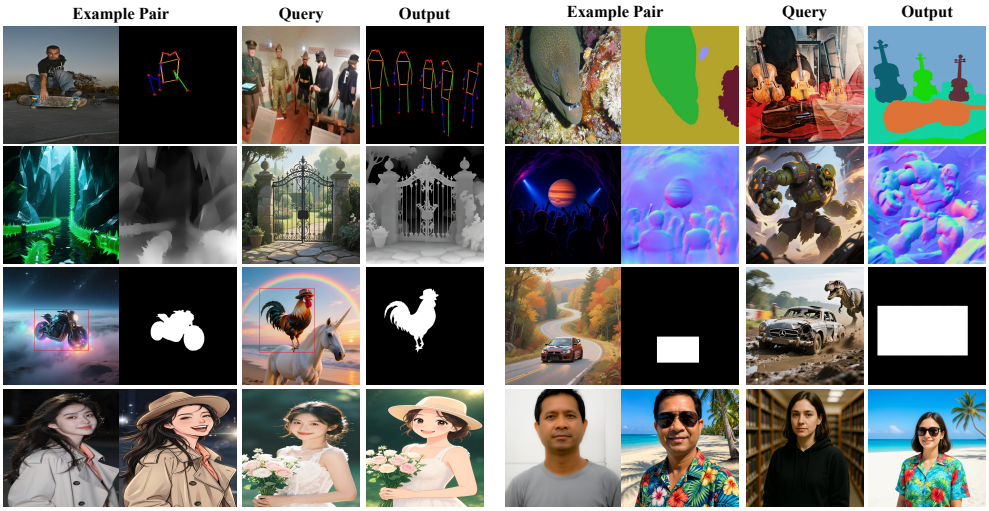

Figure 1: **Visual examples as instructions.** Given a source–target pair, **VIRAL** applies the demonstrated transformation to a new query image. The same model spans structured predictions and open-domain edits such as style and scene changes, without text instructions or test-time updates.

ABSTRACT

Visual in-context learning (V-ICL) uses a source–target image pair to specify a transformation for a new query. A central challenge is to support structured prediction, restoration, and open-domain editing through this interface, despite their different output structures and appearances. We present **VIRAL**, a V-ICL framework for pre-trained Diffusion Transformers (DiTs), instantiated with Qwen-Image-Edit. VIRAL encodes the exemplar and query as separate image sequences, distinguishes their roles through 3D-MSRoPE, and uses MoE-LoRA to accommodate heterogeneous transformations while keeping pre-trained weights frozen. We construct a dataset of transformation-consistent exemplar–query quadruplets through standard-task pairing and analogous edit mining and synthesis. The resulting model follows visual demonstrations without task-specific text instructions or test-time parameter updates. Across eight task groups, VIRAL outperforms four V-ICL baselines on all 14 reported metrics. It also performs open-domain editing and generalizes to held-out styles and lineart generation, which are excluded from VIRAL training. These results show that one pre-trained DiT can support a shared visual-demonstration interface across diverse tasks and edits.

1 INTRODUCTION

A before-and-after image pair can communicate a transformation that is difficult to capture with a task label or a short text instruction. Given such a pair, visual in-context learning (V-ICL) asks a model to apply the demonstrated change to a new query image without test-time parameter updates (Bar et al., 2022; Wang et al., 2023a; Xu et al., 2024). Tasks as different as segmentation, restoration, and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

stylistic editing can all be expressed as image-to-image mappings, yet they require different kinds of transformation. This breadth raises a central question: can a single model infer the intended mapping from a visual example and apply it to a new query? Figure 1 previews this range of tasks.

Many V-ICL systems place the exemplar pair, query image, and masked target on a shared canvas, then predict the missing region (Bar et al., 2022; Wang et al., 2023a; Xu et al., 2024). This grid formulation provides a common prediction format, but ties image roles to fixed canvas locations and divides the available resolution among its panels. Diffusion-based V-ICL methods have brought generative priors into this setting: Prompt Diffusion combines visual exemplars with text guidance (Wang et al., 2023c), whereas SD-VICL uses inference-time exemplar–query attention matching and is evaluated primarily on structured vision tasks (Oorloff et al., 2025). These approaches motivate a visual-example-driven V-ICL framework that can use a pre-trained DiT to span structured prediction, restoration, and open-domain editing within one generative formulation.

We introduce **VIRAL**, a V-ICL framework designed for pre-trained DiT backbones and instantiated with Qwen-Image-Edit (Wu et al., 2025). It formulates one-shot inference as the visual analogy $x_s : x_t :: x_q : y_q$, where a source–target example specifies the transformation to apply to a query. VIRAL encodes the three inputs as separate image token sequences and assigns the support source, support target, and query source distinct frame indices within 3D-MSRoPE. MoE-LoRA provides expert-structured adaptation for the heterogeneous transformations learned jointly while preserving the generative priors of the frozen backbone.

Learning to transfer a demonstrated change requires more than isolated input–output pairs: training must link two distinct images through the same transformation. But there is no off-the-shelf dataset directly provides these matched quadruplets across the standard visual tasks and open-domain edits studied here. We therefore construct a large dataset of transformation-consistent quadruplets (x_s, x_t, x_q, y_q) for both settings: standard-task examples draw on paired annotations and task-specific transformations, while open-domain editing examples are mined or synthesized to align editing intent across distinct scenes. This construction makes the relation between the support pair and query target an explicit training signal.

VIRAL performs strongly across this range of transformations. In the unified comparison, it outperforms four evaluated V-ICL baselines on all 14 reported metrics across eight task groups; interactive-segmentation IoU reaches 0.795, compared with 0.348 for the strongest baseline. Ablations show that distinct image-role indices and the evaluated MoE-LoRA configuration improve the multi-task model. Beyond the predefined tasks, VIRAL follows open-domain edits, transfers to test styles and tasks excluded from training. Together, these results show that a single pre-trained DiT can be adapted to follow visual demonstrations across diverse transformations. Our code is available at `https://anonymous.4open.science/r/VIRAL-744A`.

Our contributions are threefold:

- We develop an exemplar-driven V-ICL framework designed for pre-trained DiT backbones, organizing separate support and query images by their visual-analogy roles to follow the demonstrated transformation.
- We construct a large dataset of transformation-consistent exemplar–query quadruplets spanning standard visual tasks and open-domain editing.
- We evaluate the framework across heterogeneous visual tasks, analyze its role-indexing and adapter choices, and demonstrate transfer to open-domain edits and unseen tasks.

2 Related Work

Visual In-Context Learning. Visual Prompting casts an input–output example and a new query as a composite image whose missing target region is predicted by an inpainting model (Bar et al., 2022). IMProv extends image-based prompting with optional textual task descriptions (Xu et al., 2024). Painter trains on annotated pairs from multiple visual tasks and uses masked image modeling to predict outputs ranging from perceptual maps to restored images (Wang et al., 2023a); SegGPT develops an in-context coloring formulation for segmentation (Wang et al., 2023b). Together, these works establish visual examples as a way to specify task outputs. In grid-based methods, however, image roles depend on canvas placement and the query target is treated as a masked region of the

composite image. This makes it difficult to scale to high-resolution images and open-domain semantic edits. VIRAL instead represents the exemplar source, exemplar target, and query as separate image token sequences, explicitly distinguishes their roles, and generates the complete query target.

Diffusion-Based Visual In-Context Learning. Prompt Diffusion (Wang et al., 2023c) conditions a diffusion model on a visual example pair together with text guidance, so the example is not its only task signal. SD-VICL (Oorloff et al., 2025) takes a training-free route by recomputing self-attention between exemplar and query features in a pre-trained Stable Diffusion model; its six evaluated tasks center on structured prediction and colorization. These studies show how generative priors can support V-ICL through different conditioning mechanisms. VIRAL instead adapts a pre-trained DiT using exemplar–query quadruplets, with a single visual pair specifying transformations across structured prediction, restoration, and open-domain editing without task text.

3 METHOD

VIRAL learns to apply the transformation demonstrated by a source–target image pair to a new query. It combines role-aware conditioning with MoE-LoRA adaptation of a pre-trained DiT and is trained on transformation-consistent exemplar–query quadruplets. Figure 2 provides an overview.

3.1 VISUAL ANALOGY FORMULATION

We consider visual tasks whose inputs and outputs can be represented as RGB images, including visualizations of structured predictions, restored images, and edited images. Each training instance is a quadruplet (x_s, x_t, x_q, y_q): the support pair (x_s, x_t) demonstrates a transformation, and (x_q, y_q) applies the same transformation to different image content. We denote the dataset of these instances by $\mathcal{D}$. Its construction is described in Section 4.

Given a single support pair and a query image, VIRAL models the conditional distribution

$$p_{\theta_0,\phi}(y_q \mid x_s, x_t, x_q, I = \emptyset), \tag{1}$$

where θ_0 denotes the frozen pre-trained parameters and ϕ denotes the trainable adaptation parameters. The inference problem is expressed by the visual analogy $x_s : x_t :: x_q : \hat{y}_q$. The model receives no task-specific text instruction and uses the same generative output interface across tasks, without task-specific prediction heads or test-time parameter updates.

3.2 ROLE-AWARE MULTI-IMAGE CONDITIONING

We instantiate VIRAL with Qwen-Image-Edit-2511 (Wu et al., 2025). The three conditioning images provide the demonstrated input, demonstrated output, and query input, respectively. We keep the pre-trained backbone weights fixed and learn the adaptation parameters described below.

Image tokens. Let $\mathcal{E}$ denote the frozen VAE encoder followed by latent patchification. For $i \in \{s, t, q\}$, the resulting image tokens are

$$\mathbf{z}_i = \mathcal{E}(x_i) \in \mathbb{R}^{L_i \times d_z}, \qquad \mathbf{Z}_{\text{cond}} = \text{Concat}(\mathbf{z}_s, \mathbf{z}_t, \mathbf{z}_q) \in \mathbb{R}^{(L_s+L_t+L_q)\times d_z}. \tag{2}$$

Here, L_i is the token count and d_z is the packed latent dimension. The three images are concatenated in support-source, support-target, and query-source order. The DiT input projection maps these tokens to its hidden dimension D. The noisy target tokens $\mathbf{z}_{y,\tau}$ are processed jointly with $\mathbf{Z}_{\text{cond}}$ in the image stream, and the model predicts the velocity of the target tokens. When all four images have the same token count L, the conditioning sequence contains $3L$ tokens and the full image stream contains $4L$ tokens.

Image roles in 3D-MSRoPE. VIRAL uses 3D-MSRoPE to encode image identity together with spatial position. Qwen-Image's editing formulation already includes a frame coordinate alongside the two spatial coordinates (Wu et al., 2025). We assign the support source, support target, and query source distinct frame indices r_s, r_t, and r_q. Thus, a token at spatial location (h, w) in image i receives the coordinate

$$\text{pos}(i, h, w) = (r_i, h, w). \tag{3}$$

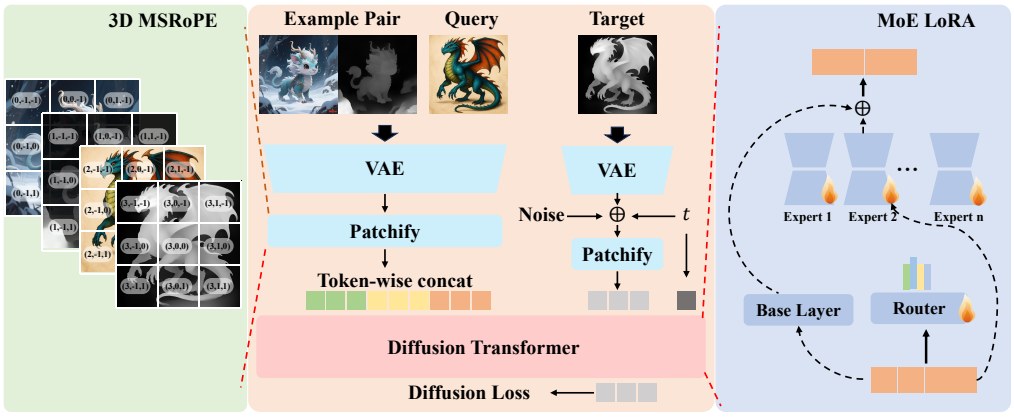

Figure 2: **Overview of VIRAL.** An exemplar pair specifies the transformation applied to a query image. Role-indexed 3D-MSRoPE distinguishes the image tokens, while MoE-LoRA adapts the DiT to generate the query target.

The target tokens use the coordinate (r_y, h, w), with the index assignment shown in Figure 2. The frame indices identify image roles rather than temporal order. This assignment distinguishes the three conditioning roles while retaining spatial coordinates within each image, so the DiT can associate the demonstrated input–output relation with the query.

3.3 MoE-LoRA Adaptation

The jointly learned tasks require different output structures and appearances. We use multiple low-rank experts to provide input-dependent adaptation capacity (Zhang et al., 2025; Jiang et al., 2024). As specified in our implementation, MoE-LoRA is applied to the output projections of the DiT feed-forward networks, while standard LoRA is used in the other selected projections.

For each token feature $\mathbf{u} \in \mathbb{R}^{d_{\text{in}}}$ processed by a MoE-LoRA layer, a learnable router produces probabilities over N experts and selects k active experts. Routing applies to all tokens in that layer, including the conditioning and noisy-target image tokens:

$$\mathbf{p}(\mathbf{u}) = \text{Softmax}(W_g \mathbf{u}), \qquad \mathcal{S}(\mathbf{u}) = \text{TopK}(\mathbf{p}(\mathbf{u}), k). \tag{4}$$

The adapted projection is

$$\mathbf{h} = W_0 \mathbf{u} + a \sum_{i \in \mathcal{S}(\mathbf{u})} g_i(\mathbf{u}) B_i A_i \mathbf{u}, \tag{5}$$

where $A_i \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B_i \in \mathbb{R}^{d_{\text{out}} \times r}$ are the low-rank factors, a is the LoRA scaling factor, and g_i is the weight of the selected expert. The selected probabilities are renormalized as

$$g_i(\mathbf{u}) = \frac{p_i(\mathbf{u})}{\sum_{j \in \mathcal{S}(\mathbf{u})} p_j(\mathbf{u})}, \qquad i \in \mathcal{S}(\mathbf{u}). \tag{6}$$

We use $N = 4$ experts, $k = 2$, and rank $r = 64$. The trainable parameters ϕ consist exclusively of the MoE-LoRA and standard LoRA parameters in the DiT, including the expert factors and routers. All other backbone parameters remain frozen.

3.4 Training and Inference

Flow-matching objective. For the Qwen-Image-Edit implementation, we formulate target generation with flow matching (Wu et al., 2025). Let $\mathbf{z}_{y,0} = \mathcal{E}(y_q)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Using a noise level $\tau \in [0, 1]$, we define

$$\mathbf{z}_{y,\tau} = (1 - \tau)\mathbf{z}_{y,0} + \tau \boldsymbol{\epsilon}, \qquad \mathbf{v}^* = \boldsymbol{\epsilon} - \mathbf{z}_{y,0}. \tag{7}$$

The DiT predicts the target velocity conditioned on the visual context:

$$\widehat{\mathbf{v}} = v_{\theta_0,\phi}(\mathbf{z}_{y,\tau}, \tau \mid \mathbf{Z}_{\text{cond}}). \tag{8}$$

We optimize the target-token loss

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{(x_s,x_t,x_q,y_q)\sim\mathcal{D},\tau\sim p_\tau,\boldsymbol{\epsilon}}\left[w(\tau)\,\|\widehat{\mathbf{v}}-\mathbf{v}^*\|_2^2\right], \tag{9}$$

where p_τ is the training noise-level distribution and $w(\tau)$ is the loss weighting.

Expert balancing. To discourage imbalanced expert utilization, we include a routing regularizer. For M routed tokens in a layer, define

$$f_i = \frac{1}{kM}\sum_{m=1}^{M}\mathbb{I}[i\in\mathcal{S}(\mathbf{u}_m)], \qquad \bar{p}_i = \frac{1}{M}\sum_{m=1}^{M}p_i(\mathbf{u}_m). \tag{10}$$

We average $N\sum_i f_i\bar{p}_i$ over the MoE layers to obtain $\mathcal{L}_{\mathrm{bal}}$. The total objective is

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{bal}}\mathcal{L}_{\mathrm{bal}}. \tag{11}$$

Inference. We encode the support pair and query once, initialize the target latent with Gaussian noise, and integrate the learned velocity field from $\tau = 1$ to $\tau = 0$ using the configured sampler. The conditioning image latents remain fixed as inputs across sampling steps; only the target latent is iteratively updated. Finally, the target tokens are unpacked and decoded by the frozen VAE decoder to obtain $\hat{y}_q$. More inference settings are provided in Appendix B.

4 Visual In-Context Dataset

Learning visual analogy requires two image pairs that share a transformation while depicting different content. The paired annotations and instruction-based editing corpora used here provide individual input–output examples, which we organize into exemplar–query quadruplets (x_s, x_t, x_q, y_q). We follow two construction routes: pairing examples under predefined task specifications for standard visual tasks, and matching editing intent across scenes for open-domain editing.

Standard Visual Tasks. We form standard-task quadruplets from two distinct image pairs that follow the same task specification. These pairs combine newly synthesized images with existing annotated datasets. For the synthetic component, we sample prompts from DiffusionDB (Wang et al., 2022) and generate RGB images with Qwen-Image (Wu et al., 2025).

For edge, depth, and surface normal estimation, we use ControlNetAux (Face, 2024) to obtain model-generated target maps . **For object detection**, Qwen2.5-VL-7B-Instruct produces category and bounding-box annotations. We represent category-specific localization as filled white rectangles on a black canvas and pair examples targeting the same category. **Human pose estimation** uses COCO-2017 (Lin et al., 2014) images and the provided keypoint annotations rendered as pose maps. **For interactive segmentation**, the input image contains a red box marking the selected object, and the target is a binary mask produced by prompting SAM (Kirillov et al., 2023) with the corresponding box. **For entity segmentation**, we use EntityV2 (Qi et al., 2023) annotations and render distinct entity regions in different colors. **For restoration and enhancement**, each pair maps a degraded image to its reference image. **Colorization** inputs are grayscale versions of RGB targets; **watermark-removal** inputs are constructed by overlaying CLWD logo templates (Liu et al., 2021) on clean images. **Deraining** and **low-light enhancement** use the paired images provided by Rain200L (Yang et al., 2017) and LOL-v2 (Yang et al., 2021), respectively.

Open-Domain Editing. Open-domain edits vary in both operation and image content, so quadruplet construction requires matching the intended change across scenes. Using GPT-Image-Edit-1.5M (Wang et al., 2025) and Pico-Banana-400k (Qian et al., 2025) as source corpora, we construct candidate quadruplets through complementary synthesis and retrieval procedures.

Analogous synthesis. Starting from an editing pair (x_s, x_t) and its instruction I, we use Qwen-3.6 to describe a different scene in which the same edit remains applicable. Qwen-Image generates the new source image x_q from this description, and Qwen-Image-Edit applies I to produce y_q. This yields a candidate quadruplet with a shared editing intent across different scenes. The caption-generation prompt is provided in Appendix D.3.

Transformation-based mining. For each existing editing pair (x, y), we compute a descriptor from the difference between its CLIP ViT-L/14 (Radford et al., 2021) image embeddings:

$$\mathbf{d}(x, y) = \mathcal{E}_{\text{CLIP}}(y) - \mathcal{E}_{\text{CLIP}}(x). \tag{12}$$

We cluster these descriptors using K-means and retrieve a different pair within each cluster by cosine similarity. Nearby descriptors identify candidate pairs that express a similar editing transformation. We further screen the candidates by editing-instruction similarity before assembling the quadruplets.

Quality Control and Dataset Composition. We filter invalid automatically generated annotations for standard tasks. For editing data, we additionally remove visually redundant candidate source pairs and manually inspect the quadruplets to remove inconsistent editing intent or poor visual quality. Construction instructions guide synthesis and curation; VIRAL learns from the resulting image quadruplets without receiving task text. Construction parameters and dataset examples are provided in Appendices D and D.4. A shared pool of 100K synthetic RGB images supports depth estimation, surface normal estimation, edge detection, colorization, and watermark removal. Additional task-specific data and their sizes are summarized in Appendix D. The open-domain component contains 40K editing quadruplets. Evaluation images are held out from VIRAL training. Lineart generation and the styles used in our style-transfer evaluation are excluded from the training transformations, enabling evaluation beyond the trained task and style set.

5 Experiments

5.1 Experimental Setup

We inject MoE-LoRA modules ($N = 4$, Top-2 routing) into the FFN layers of the DiT backbone, while standard LoRA is applied to other layers. The model is fine-tuned on our In-Context Dataset (Sec. 4) and all quantitative evaluations are conducted on a held-out test set unseen during training. During inference, we adopt a 1-shot setting by providing a single task-specific exemplar pair. Additional implementation details are in Appendix B

5.2 Comparisons on Downstream Visual Tasks

Comparison with V-ICL methods. We first quantitatively evaluate VIRAL against representative Visual In-Context Learning (V-ICL) baselines, including Painter (Wang et al., 2023a), IMProv (Xu et al., 2024), VisualPrompt (Bar et al., 2022), and PromptDiff (Wang et al., 2023c). Specifically, we follow the evaluation protocols of PromptDiff (Wang et al., 2023c) for edge detection, SD-VICL (Oorloff et al., 2025) for colorization, and adopt standard metrics from DepthAnything (Yang et al., 2024) and StableNormal (Ye et al., 2024) for geometric estimation (depth and normal). For image restoration, we employ the test pipelines from CSUD (Dong et al., 2025) and HVID (Yan et al., 2025) for deraining and low-light enhancement, respectively.

As shown in Table 1, VIRAL achieves the best result among the four evaluated V-ICL baselines on all reported metrics across eight task groups under our evaluation protocol. The improvement is particularly clear in interactive segmentation, where IoU increases from 0.348 for the strongest baseline, Painter, to 0.795 for VIRAL. On deraining, VIRAL improves PSNR from 19.13 for Painter to 29.67. Together, these results show that our method can perform strongly across both structured prediction and image restoration.These differences may partly reflect how prior methods represent visual context and the tasks on which they were trained. PromptDiff uses ControlNet-based conditioning and performs relatively well on edge estimation, but is weaker on restoration and object detection in our benchmark. Grid-based inpainting methods such as Painter, IMProv, and VisualPrompt place the exemplar and query on a shared canvas, which may constrain the detail available to each image. In contrast, by harnessing the generative priors of pre-trained image foundation models, VIRAL allowing the model to maintain high fidelity across the full spectrum of visual tasks. Figure 3 displays the visualization results of example-based rule interventions; modifying the transformations specified in the examples guides the model to produce correspondingly different outputs. Figure 5 presents qualitative visualization results comparing various V-ICL methods. VIRAL demonstrates remarkable robustness in complex, real-world scenarios where previous approaches often fail to capture the target mapping. It accurately preserves identity and fine-grained details while strictly adhering to the semantic transformation defined by the user-provided examples.

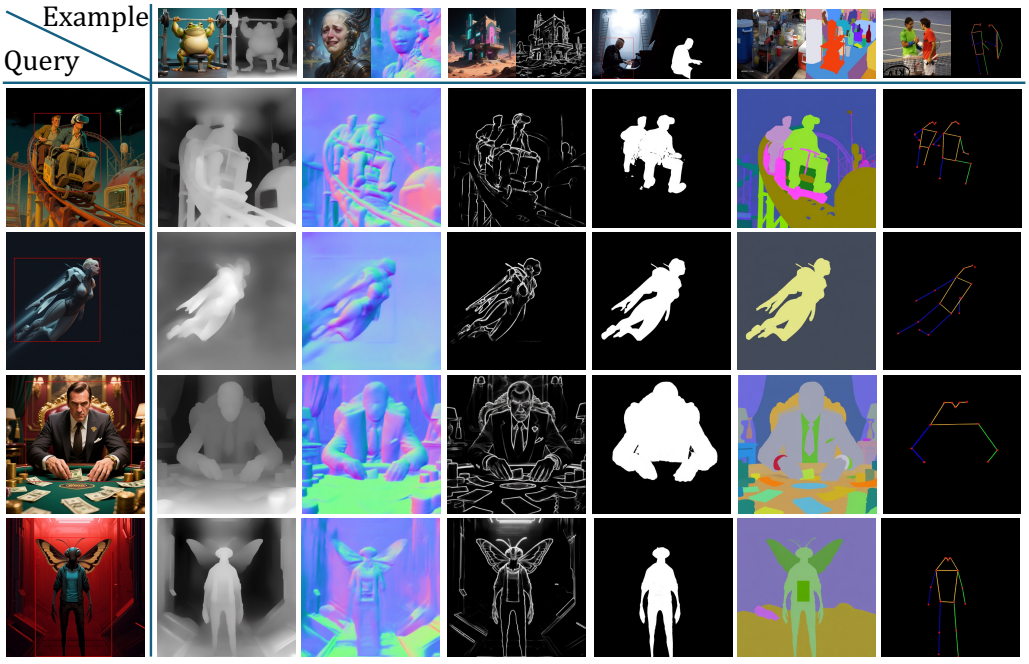

Figure 3: With the query fixed, changing the exemplar transformation changes the output accordingly, illustrating that the model follows the visual demonstration.

Table 1: Quantitative comparison with the V-ICL model. "Seg." refers to interactive segmentation.

Method	Seg.	Obj. Det.	Edge		Colorization		Depth		Normal		Deraining		Enhancement	
	IoU ↑	IoU ↑	RMSE ↓	LPIPS ↓	FID ↓	LPIPS ↓	AbsRel ↓	δ_1 ↑	Med ↓	Mean ↓	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
IMProv	0.178	0.337	99.36	0.682	210.89	0.714	0.175	0.711	56.08	52.77	15.29	0.330	15.14	0.353
VisualPrompt	0.191	0.347	63.23	0.585	214.90	0.707	0.173	0.732	48.52	46.03	15.08	0.329	15.30	0.444
PromptDiff	0.178	0.324	35.88	0.255	179.21	0.598	0.160	0.714	97.27	105.92	8.67	0.128	8.75	0.323
Painter	0.348	0.387	94.93	0.632	176.71	0.624	0.167	0.732	116.01	119.87	19.13	0.634	16.39	0.712
Ours	**0.795**	**0.562**	**28.59**	**0.133**	**43.75**	**0.126**	**0.138**	**0.812**	**17.582**	**20.187**	**29.67**	**0.889**	**25.24**	**0.878**

Comparison with Task-Specific Models. We benchmark VIRAL against leading domain-specific experts across five representative downstream tasks. Specifically, we compare against SLBR (Liang et al., 2021) for watermark removal, DepthAnything (Yang et al., 2024) for depth estimation, StableNormal (Ye et al., 2024) for surface normal estimation, CSUD (Dong et al., 2025) for deraining, and CHVI (Yan et al., 2025) for low-light enhancement. More details are in the Appendix C.2.

The quantitative results, detailed in Table 2 and Table 3, indicate that our generalist method achieves performance comparable to, and in many cases significantly surpassing, state-of-the-art specialized models. VIRAL improves over SLBR on watermark removal (36.958 versus 30.972 PSNR; 0.959 versus 0.934 SSIM). On low-light enhancement, it has higher PSNR than CHVI (25.248 versus 24.797) but lower SSIM (0.878 versus 0.919) .While there is a slight performance gap in the deraining task compared to CSUD, visual inspection reveals that our results remain perceptually indistinguishable from ground truth (see Appendix F.4). For geometric estimation, VIRAL obtains lower depth AbsRel than DepthAnything (0.138 versus 0.154) and lower normal median angular error than StableNormal (17.582 versus 21.352), as reported in Table 3. These results are encouraging for a model that also handles editing and restoration. The "Ours (single)" denotes a baseline trained exclusively on the corresponding individual task. The comparison reveals that our unified training strategy successfully circumvents the negative transfer often observed in multi-task learning, exhibiting no performance degradation compared to the single-task counterparts. More results are provided in Appendix E.1.

Overall, VIRAL provides a unified visual-demonstration interface while achieving competitive performance against selected task-specific models on several benchmarks. This breadth comes at an inference cost: the standard 40-step model takes 25.366 s per image at 512 resolution, whereas a separately distilled three-step variant reduces latency to 1.472 s with some accuracy loss (Appendix E.4).

Table 2: Comparison with specialized models on three image reconstruction tasks.

Task	Method	PSNR ↑	SSIM ↑
Derain	CSUD	**33.31**	**0.957**
	Ours	29.67	0.889
Light Enhance	CHVI	24.797	**0.919**
	Ours	**25.248**	0.878
Watermark Removal	SLBR	30.972	0.934
	Ours	**36.958**	**0.959**

Table 3: Comparison with specialized models on depth and surface normal estimation.

Task	Method	AbsRel ↓	δ_1 ↑
Depth	DepthAnything	0.154	0.775
	Ours(single)	0.145	0.796
	Ours	**0.138**	**0.812**
Task	**Method**	**Med ↓**	**Mean ↓**
Normal	StableNormal	21.352	25.845
	Ours(single)	18.295	21.913
	Ours	**17.582**	**20.187**

Table 4: Ablation study on adapter architectures. MoE-LoRA outperforms a larger standard LoRA.

Method	Param.	Depth		Normal		Seg.	Soft Edge		Obj. Det.
		AbsRel ↓	δ_1 ↑	Med ↓	Mean ↓	IoU ↑	RMSE ↓	LPIPS ↓	IoU ↑
LoRA ($r = 70$)	∼530M	0.141	0.805	17.876	21.461	0.721	38.494	0.224	0.367
MoE-LoRA ($r = 64$)	∼500M	**0.138**	**0.812**	**17.582**	**20.187**	**0.795**	**28.59**	**0.133**	**0.562**

5.3 Architectural Ablations

Mixture-of-experts adaptation. Table 4 compares the evaluated MoE-LoRA configuration with a larger standard-LoRA adapter. MoE-LoRA uses 412M trainable parameters versus 530M for standard LoRA and performs better on every metric shown; the largest differences include edge RMSE (28.59 versus 38.49) and object-detection IoU (0.562 versus 0.367). These results suggest that the expert-structured adapter better accommodates heterogeneous task mappings in our multi-task setting while using fewer trainable parameters.

Role-aware positional encoding. Qwen-Image's editing formulation already uses a frame coordinate to distinguish images; our question is how to assign these coordinates to the support source, support target, and query source in a visual-analogy input. Table 5 compares the full role-indexed configuration with the normal MSRoPE in which no role dimension. The full configuration improves interactive-segmentation IoU from 0.3501 to 0.795 and reduces edge RMSE from 82.83 to 28.59. These results highlight the benefit of assigning distinct positional indices to the demonstration and query in our V-ICL framework.

Further analysis regarding single-task learning, exemplar selection, few-shot scaling, time costs, and more can be found in the Appendix E.

5.4 Generalization and Versatility

Open-Domain In-Context Editing Ability. We emphasize that the in-context reasoning capability of VIRAL is generalizable beyond specific tasks. We evaluate VIRAL against instruction editing model, including Qwen-Image-Edit and ICEdit (Zhang et al., 2025). We employ Qwen2.5-VL-7B-Instruct (Bai et al., 2025) to transcribe the visual demonstrations into precise text instructions for these baselines. We specifically focus on Style Transfer, a representative editing task where textual descriptions are often insufficient to accurately capture the visual style. We evaluate performance using 20 diverse styles from OmniConsistency (Song et al., 2025). We employ CLIP (Radford et al., 2021), LPIPS (Zhang et al., 2018), and DINO (Oquab et al., 2024) to measure the semantic and perceptual similarity between the generated results and the ground truth target.

Table 6 shows that VIRAL outperforms the evaluated baselines across the reported metrics. Its lower LPIPS and higher DINO similarity indicate closer perceptual and semantic agreement with the target images. VIRAL also surpasses its Qwen-Image-Edit backbone, demonstrating the effectiveness of the full visual-demonstration framework for these edits. The gains are particularly relevant to style transfer, where an exemplar can convey appearance cues that are difficult to specify precisely in text. Figure 4 illustrates how VIRAL follows such example-defined transformations across diverse edits. These results suggest that our in-context fine-tuning stage does not merely exploit existing capabilities but effectively elicits dormant reasoning potentials within the pre-trained DiT.

Table 5: Ablation study on role-aware positional encoding.

Method	Depth		Seg.	Edge		Normal		Obj. Det.
	AbsRel ↓	δ_1 ↑	IoU ↑	RMSE ↓	LPIPS ↓	Med ↓	Mean ↓	IoU ↑
MSRoPE	0.169	0.728	0.350	82.827	0.432	36.064	39.644	0.359
3D-MSRoPE	**0.138**	**0.812**	**0.795**	**28.59**	**0.133**	**17.582**	**20.187**	**0.562**

Table 6: Comparison on open-domain editing and style transfer.

Task	Method	CLIP ↑	LPIPS ↓	DINO ↑
Open-domain Editing	qwen-edit	0.845	0.652	0.734
	ICEdit	0.823	0.621	0.717
	Ours	**0.880**	**0.517**	**0.798**
Style Transfer	qwen-edit	0.701	0.814	0.598
	ICEdit	0.710	0.813	0.653
	Ours	**0.832**	**0.718**	**0.757**

Table 7: Quantitative comparison of object detection and lineart estimation.

Method	Object Det.	Lineart	
	IoU ↑	RMSE ↓	LPIPS ↓
IMProv	0.251	80.25	0.726
VisualPrompt	0.318	50.81	0.645
PromptDiff	0.326	189.89	0.780
Painter	0.458	82.22	0.650
Ours	**0.721**	**34.82**	**0.339**

Task and Domain Generalization We further assess two forms of generalization: applying object detection to a separate dataset and performing lineart generation, a task absent from training. On our PASCAL-5i-derived object-detection benchmark (Shaban et al., 2017), VIRAL achieves 0.721 IoU, compared with 0.458 for the strongest V-ICL baseline (Table 7). This result shows that our visual-example interface remains effective on images from a different data source. Lineart generation requires clean, stylized contours distinct from the edge maps used during training. Without lineart training pairs, VIRAL achieves 34.82 RMSE and 0.339 LPIPS, improving over the strongest baseline on both metrics (50.81 and 0.645, respectively). Figure 7 illustrates how a single exemplar pair guides this unseen transformation on new queries. Additional cross-task examples are provided in Appendix F.2. Together, these results highlight VIRAL's strong generalization across datasets and tasks and the versatility of its visual-demonstration interface.

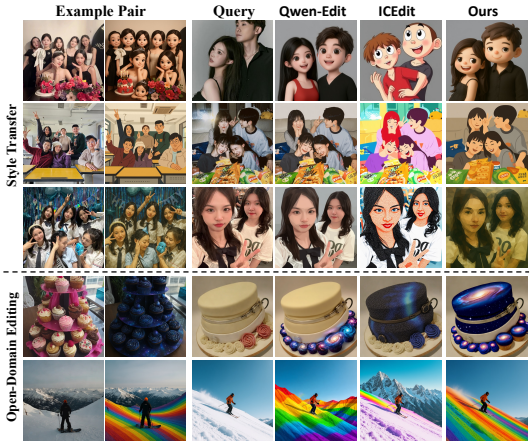

Figure 4: Qualitative comparison on open-domain editing and style transfer.

6 Conclusion

We presented VIRAL, a visual in-context learning framework that adapts pre-trained DiT backbones to follow transformations specified by a source–target exemplar pair. Through role-aware multi-image conditioning and MoE-LoRA adaptation, VIRAL supports perception, restoration, and open-domain editing within a single generative model. To support this capability, we constructed a dataset of transformation-consistent exemplar–query quadruplets spanning these task families. Experiments show that VIRAL outperforms the evaluated V-ICL baselines, achieves competitive performance against task-specific models, and transfers to styles and lineart generation excluded from training. These results demonstrate that a single model can use visual examples to perform diverse image transformations without task-specific prediction heads or test-time parameter updates, highlighting visual demonstrations as a flexible interface for general-purpose visual learning. We hope this work inspires further research into building universal visual generalists that flexibly adapt to user needs through visual demonstrations.

AI USE STATEMENT

We used an LLM-based assistant to help organize and improve the writing, and provide feedback on the presentation of our methodology, experiments, and results. Generative and vision-language models were also used in the dataset construction pipeline described in Section 4 and Appendix D, including the generation of synthetic images and analogous editing examples. The authors reviewed AI-assisted writing against the experimental records and cited sources, and screened candidate editing quadruplets for consistency and visual quality. We take responsibility for the final text, claims, data, and results reported in this paper.

ETHICS STATEMENT

This work uses existing image datasets and pre-trained generative models to study visual in-context learning; it does not involve a human-subject study. An editing-capable model can be misused to produce misleading images, and its outputs may reflect biases or sensitive content inherited from the source datasets and pre-trained models. Our evaluation measures task performance and does not constitute a comprehensive assessment of safety, fairness, or privacy. Any release or reuse of derived data should respect the terms of the source datasets and account for privacy and harmful-editing risks.

REPRODUCIBILITY STATEMENT

Section 3 describes the model, visual-analogy formulation, and training objective. Section 4 identifies the data sources and explains how exemplar–query quadruplets are constructed. The experimental section reports the comparison protocols and evaluation metrics, while Appendix B provides implementation and baseline details and Appendix D records data construction settings and corpus sizes. The prompts used for analogous data synthesis are included in Appendix D.3. We will provide an anonymous repository containing the implementation code for review.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A. Efros. Visual prompting via image inpainting. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Guanglu Dong, Tianheng Zheng, Yuanzhouhan Cao, Linbo Qing, and Chao Ren. Channel consistency prior and self-reconstruction strategy based unsupervised image deraining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 7469–7479. Computer Vision Foundation / IEEE, 2025.

Hugging Face. Controlnet auxiliary models. `https://github.com/huggingface/controlnet_aux`, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *CoRR*, abs/2401.04088, 2024.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Jing Liang, Li Niu, Fengjun Guo, Teng Long, and Liqing Zhang. Visible watermark removal via self-calibrated localization and background refinement. In Heng Tao Shen, Yueting Zhuang, John R. Smith, Yang Yang, Pablo César, Florian Metze, and Balakrishnan Prabhakaran (eds.), *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, pp. 4426–4434. ACM, 2021.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

Yang Liu, Zhen Zhu, and Xiang Bai. Wdnet: Watermark-decomposition network for visible watermark removal. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, pp. 3684–3692. IEEE, 2021.

Trevine Oorloff, Vishwanath Sindagi, Wele Gedara Chaminda Bandara, Ali Shafahi, Amin Ghiasi, Charan Prakash, and Reza Ardekani. Stable diffusion models are secretly good at visual in-context learning. *CoRR*, abs/2508.09949, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024.

Lu Qi, Jason Kuen, Tiancheng Shen, Jiuxiang Gu, Weidong Guo, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High-quality entity segmentation. In *ICCV*, 2023.

Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-banana-400k: A large-scale dataset for text-guided image editing. *CoRR*, abs/2510.19808, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press, 2017.

Yiren Song, Cheng Liu, and Mike Zheng Shou. Omniconsistency: Learning style-agnostic consistency from paired stylization data. 2025.

Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 6830–6839. IEEE, 2023a.

Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Towards segmenting everything in context. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 1130–1140. IEEE, 2023b.

Yuhan Wang, Siwei Yang, Bingchen Zhao, Letian Zhang, Qing Liu, Yuyin Zhou, and Cihang Xie. GPT-IMAGE-EDIT-1.5M: A million-scale, gpt-generated image dataset. *CoRR*, abs/2507.21033, 2025.

Zhendong Wang, Yifan Jiang, Yadong Lu, Yelong Shen, Pengcheng He, Weizhu Chen, Zhangyang (Atlas) Wang, and Mingyuan Zhou. In-context learning unlocked for diffusion models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023c.

Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv:2210.14896 [cs]*, 2022.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025.

Jiarui Xu, Yossi Gandelsman, Amir Bar, Jianwei Yang, Jianfeng Gao, Trevor Darrell, and Xiaolong Wang. Improv: Inpainting-based multimodal prompting for computer vision tasks. *Trans. Mach. Learn. Res.*, 2024, 2024.

Qingsen Yan, Yixu Feng, Cheng Zhang, Guansong Pang, Kangbiao Shi, Peng Wu, Wei Dong, Jinqiu Sun, and Yanning Zhang. HVI: A new color space for low-light image enhancement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 5678–5687. Computer Vision Foundation / IEEE, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 1685–1694. IEEE Computer Society, 2017.

Wenhan Yang, Wenjing Wang, Haofeng Huang, Shiqi Wang, and Jiaying Liu. Sparse gradient regularized deep retinex network for robust low-light image enhancement. *IEEE Trans. Image Process.*, 30:2072–2086, 2021.

Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. Stablenormal: Reducing diffusion variance for stable and sharp normal. *ACM Trans. Graph.*, 43(6):250:1–250:18, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 586–595. Computer Vision Foundation / IEEE Computer Society, 2018.

Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *CoRR*, abs/2504.20690, 2025.

APPENDIX CONTENTS

- Appendix A: Qualitative comparison with V-ICL methods 13
- Appendix B: Model and Training Details 13
- Appendix C: Evaluation Protocols 14
- Appendix D: Dataset Construction 16
- Appendix E: Additional Quantitative Results 18
- Appendix F: Generalization and Qualitative Results 20
- Appendix G: Limitations 25

A QUALITATIVE COMPARISON WITH V-ICL METHODS

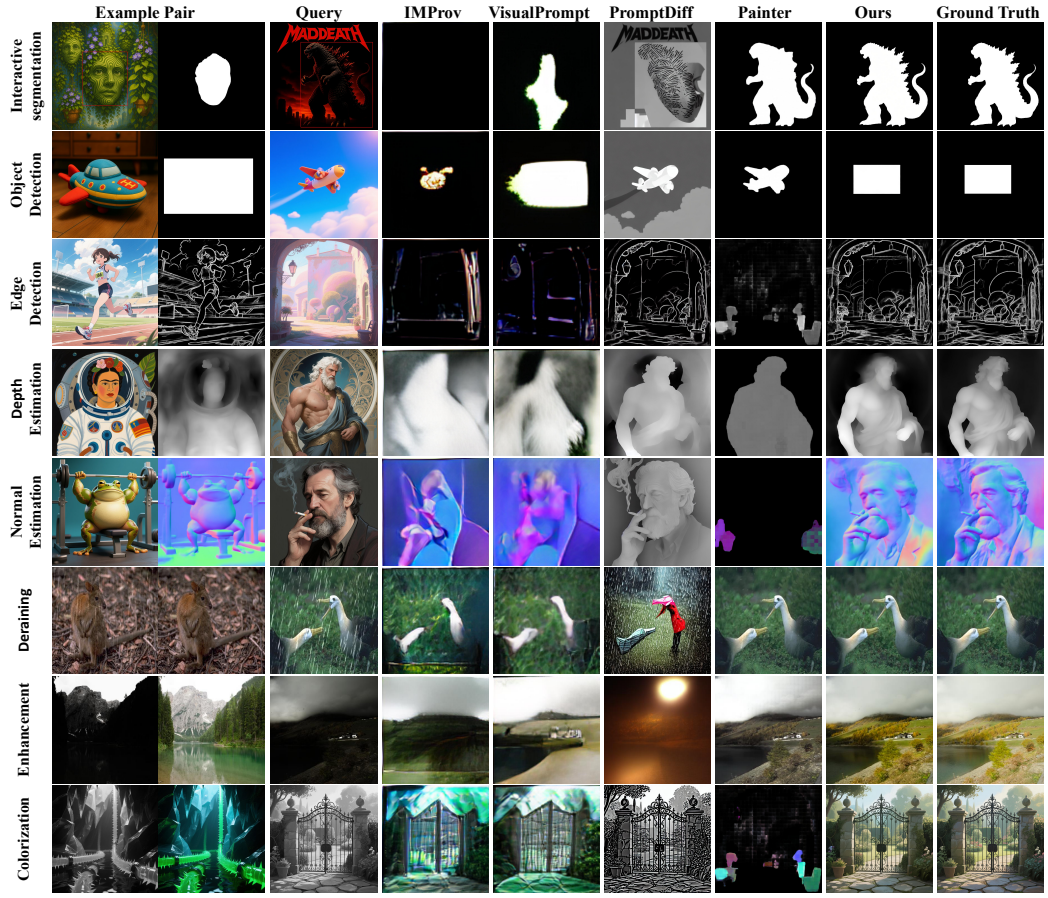

Figure 5: **Qualitative comparison.** While existing baselines either exhibit restricted task versatility or suffer from performance degradation when encountering complex scenarios due to their reliance on over-simplified training distributions, VIRAL achieves superior accuracy and visual fidelity.

B MODEL AND TRAINING DETAILS

B.1 BACKBONE AND TRAINABLE MODULES

VIRAL is instantiated with `Qwen-Image-Edit-2511` (Wu et al., 2025). We freeze the pre-trained backbone parameters and train only the MoE-LoRA and standard LoRA modules within the DiT. MoE-LoRA is applied to the feed-forward output projections, with four experts of rank 64 and Top-2 routing. The router operates on all tokens processed by each adapted layer, including conditioning and noisy-target tokens. The selected expert probabilities are renormalized to sum to one. Standard

LoRA of rank 64 is applied to the other selected projections, including attention query, key, value, and output projections. The exact module paths and scaling factors are specified in Table 8.

Table 8: Model adaptation and optimization settings.

Setting	Value
Backbone	Qwen-Image-Edit-2511
MoE-LoRA placement	FFN output projections
Experts / active experts / rank	4 / 2 / 64
Routing scope and weights	All tokens in each MoE-LoRA layer; renormalized Top-2 weights
Standard LoRA placement	Other selected projections
Standard LoRA rank	64
Optimizer / learning rate	AdamW / 10^{-4}, constant
Per-device batch size	1 quadruplet
Training hardware / duration	8 NVIDIA A800 GPUs / approximately 48 hours
Training resolution / preprocessing	512×512

B.2 Training Procedure

At each iteration, each GPU independently selects a task and samples an exemplar–query quadruplet. Task sampling probabilities are given in Table 8. This procedure permits multiple tasks within a distributed batch. The effective batch size per optimizer update is eight times the gradient-accumulation factor when each device processes one quadruplet per forward pass.

The support source, support target, and query source are concatenated in that order. The target latent is perturbed according to the flow-matching path in Section 3.4, and the velocity loss is evaluated on the target predictions. The three conditioning latents are supplied without target-style noise corruption. The actual noise-level distribution, loss weighting, and auxiliary-loss configuration are listed in Table 8. Frame-index assignments follow Figure 2.

B.3 Inference Settings

Unless stated otherwise, evaluation uses one exemplar pair, 40 sampling steps, and an image resolution of 512. The target latent is initialized from Gaussian noise. The conditioning image latents remain fixed as inputs across sampling steps, while the target latent is updated and then decoded by the frozen VAE. VIRAL receives no task-specific text instruction and performs no test-time parameter updates. Any fixed backbone prompt or auxiliary conditioning is specified below.

Table 9: Inference settings for the standard VIRAL model.

Setting	Value
Visual context	One source–target exemplar pair
Sampling steps	40
Guidance configuration	4.0
Resolution convention	512×512
number of outputs	1

C Evaluation Protocols

C.1 Test Splits and Exemplar Selection

All quantitative evaluations use images held out from VIRAL training. For standard tasks, exemplar pairs are sampled from the held-out pool of the corresponding task. The selected exemplar for each query is shared across V-ICL methods. We use reproducible seed settings for stochastic inference.

For category-specific localization, the exemplar and query refer to the same target category, and the target represents bounding boxes as filled white rectangles on a black canvas. For interactive segmentation, a red box on the query identifies the object to segment, and the target is a binary mask.

Table 10 records the evaluation sources, sample counts, and output conversion rules. The task-dependent conversion from generated RGB images to the quantities used by each metric is held fixed across the compared predictions.

Table 10: Evaluation datasets and metric protocols. Each task uses 500 held-out test samples, except deraining (Rain200L) and low-light enhancement (LOLv2), which use 100 samples each.

Task	count	Metrics and conversion details
Interactive segmentation	500 samples	IoU
Object detection	500 samples	Box-mask IoU;
Edge estimation	500 samples	RMSE and LPIPS;
Colorization	500 samples	FID and LPIPS;
Depth estimation	500 samples	AbsRel and δ_1;
Surface normals	500 samples	Median and mean angular errors;
Deraining	Rain200L; 100 samples	PSNR and SSIM;
Low-light enhancement	100 samples	PSNR and SSIM; same stated color-range and aggregation conventions
Watermark removal	500 samples	PSNR and SSIM; same stated color-range and aggregation conventions
Open-domain editing	500 samples	CLIP, LPIPS, DINO;
Style transfer	500 samples	CLIP, LPIPS, DINO;
Detection transfer	PASCAL-5i-derived; 500 samples	IoU; category and fold selection as specified
Lineart generation	500 samples	RMSE and LPIPS;

C.2 Baseline Implementations

We evaluate VisualPrompt, IMProv, Painter, and Prompt Diffusion using their released checkpoints. We retain each method's supported image layout and provide task descriptions to methods that use text. IMProv receives its task-description prompt, and Prompt Diffusion receives the corresponding task descriptor.

Table 11: Input configuration of the compared methods. Canvas resolution and per-image resolution are distinguished for grid-based methods.

Method	Input format	Resolution convention
VisualPrompt / IMProv	Composite image with exemplar, query, and target region	448
Painter	Vertically stacked sources; separately arranged target tensor	448 per image; 896×448 source tensor
Prompt Diffusion	Visual exemplar, query, and task descriptor	512
VIRAL	Separate exemplar and query images	512; Table 9
Qwen-Image-Edit / ICEdit	Query image and transcribed edit instruction	512

For Painter, each image is resized to 448×448. The exemplar source and query source are stacked vertically into a 896×448 input tensor. A separate target tensor contains the exemplar target in the upper half and the masked query-target region in the lower half; the predicted lower half provides the 448×448 query output.

Following inference, model-specific post-processing converts outputs to standard RGB images. All predictions are then resized to the original ground-truth resolution for metric computation. For the specialist comparisons in Tables 2 and 3, we use officially released checkpoints, including a checkpoint trained on LOLv2 for low-light enhancement.

C.3 Text Instructions for Editing Baselines

For the comparisons in Figure 4, Qwen2.5-VL-7B-Instruct converts each visual exemplar pair (x_s, x_t) into a text instruction. The instruction is then supplied with the query image to the text-based editing baselines. VIRAL uses the visual exemplar pair. The transcription prompt is given below, with example instructions in Table 12.

Prompt for Instruction Generation (Qwen2.5-VL)

"Compare Image A and Image B. Write a concise English editing instruction to transform Image A into Image B. Only return the string of the editing instructions; nothing else."

Table 12: Text instructions supplied to the editing baselines for the examples in Figure 4.

ID	Text Editing Instruction
1	Transformed into 3D Chibi Style, A digital illustration of a young boy and girl, both with big, expressive eyes and wide smiles, standing close together.
2	Transformed into Flat vector art style. A flat vector illustration of four friends hanging out indoors, sitting on a couch and surrounded by colorful snacks and drinks.
3	Transformed into low poly style, Photo of two women in a low-quality, pixelated style, with a geometric, faceted appearance. The woman on the left is wearing a black and white outfit with a black bow tie and a white shirt.
4	Replace the cupcakes with a galaxy theme.
5	Replace the snowy mountain landscape with a rainbow-colored mountain landscape.

D Dataset Construction

D.1 Sources and Supervision

The shared synthetic component contains 100K RGB source images generated with Qwen-Image from DiffusionDB prompts. These images support edge, depth, normal, colorization, and watermark-removal pairs. Table 13 reports the remaining task-specific data and the 40K open-domain editing quadruplets. The count stage, train/test allocation, and retained per-task pair counts are specified by table 13 and 10. Teacher-generated maps and masks are training annotations; existing datasets supply the annotations or paired references used for the other tasks.

D.2 Mining and Filtering

For each editing pair (x, y), we compute the difference between its CLIP ViT-L/14 image embeddings, as defined in Section 4. We cluster these descriptors with K-means and retrieve a different editing pair from the same cluster by cosine similarity. The number of clusters is

$$K = \begin{cases} 1500, & N < 25{,}000, \\ 3000, & N \geq 25{,}000, \end{cases} \tag{13}$$

where N is the number of editing pairs in the subset being clustered.

We remove candidate quadruplets when the cosine similarity between their source images exceeds 0.98. We also compare the editing instructions associated with the two pairs to screen for a shared edit intent. Finally, we manually discard quadruplets with inconsistent edits or poor visual quality. For standard visual tasks, we filter invalid automatically generated annotations.

Training and evaluation separation. Evaluation images are held out from VIRAL training. Lineart generation and the 20 styles used in the style-transfer evaluation are excluded from the training transformations.

Table 13: Sources and target representations in the V-ICL training corpus. The first three rows share one pool of 100K synthetic RGB source images generated from DiffusionDB prompts with Qwen-Image. The remaining standard-task counts are input–output pairs; the open-domain count is exemplar–query quadruplets.

Task	Source and target construction	Amount
Edge / depth / normal	Synthetic RGB → predictor-generated maps via ControlNetAux (Face, 2024)	Shared pool
Colorization	Grayscale synthetic image → RGB image	Shared pool
Watermark removal	Synthetic RGB with CLWD logos (Liu et al., 2021) → clean RGB image	Shared pool
Interactive segmentation	Box-marked RGB image → SAM (Kirillov et al., 2023) binary mask	100K pairs
Human pose	COCO-2017 (Lin et al., 2014) image → keypoint map	50K pairs
Entity segmentation	EntityV2 (Qi et al., 2023) image → colored entity map	20K pairs
Object detection	RGB image → category-specific box mask from Qwen2.5-VL annotations	8K pairs
Deraining	Rain200L (Yang et al., 2017) rainy image → clean image	2K pairs
Low-light enhancement	LOL-v2 (Yang et al., 2021) low-light image → reference image	2K pairs
Open-domain editing	GPT-Image-Edit-1.5M (Wang et al., 2025) and Pico-Banana-400k (Qian et al., 2025); analogous synthesis and pair mining	40K quadruplets

D.3 Prompt for Analogous Scene Synthesis

To facilitate **Generative Analogous Synthesis**, we employ the Qwen-Max API to generate descriptive captions for novel scenes. The objective is to synthesize a semantically distinct source description that maintains logical compatibility with the original editing instruction. The specific system prompt provided to the LLM is detailed below:

System Prompt for Analogous Caption Generation

The user will input an image description and an editing command.
Your task is to replace as many entities or attributes as possible in the image description to generate a new image description. However, the editing command must remain valid for the rewritten image.
You should first analyze the editing command to determine what modifications it made. Ensure that the rewritten image description can also be edited using this command. For example, if the command changes A to B, the rewritten description should explicitly include A. If the command adds A, then A should not appear in the rewritten description. Entities not mentioned in the instructions can be freely replaced with other entities.
Examples:

- **Image Description:** "A sleek black sports car with a closed top parked on the side of a mountain road at sunset."
- **Editing Instruction:** "Change the color of the sports car to red and make it a convertible."
- **Acceptable Rewrite:** "A glossy black coupe parked along a coastal cliff road during golden hour."

(Rationale: The scene remains a non-red, non-convertible sports car parked in a sunset-like setting—thus editable. Since the instruction modifies the car's color and type, the rewritten description must include the car but preserve its original attributes. The background, not targeted by the instruction, is altered from a mountain road to a coastal cliff.)
Note: The rewritten image description should be clearly different from the original input image description, but it shouldn't describe the edited scene. The rewritten caption cannot be the edited scene. The rewritten description should be enclosed in `<rw></rw>` tags.

The rewritten description is used to generate a new source image, and the editing model applies the original instruction to that image. The resulting pair is combined with the reference pair and passed through the quality-control procedure. The instruction is used for data construction and is not supplied as task text during VIRAL training.

D.4 Dataset Examples

Figure 6 shows task-specific image pairs and open-domain editing quadruplets. The upper rows illustrate the source images, annotation maps, and degraded images used for standard tasks. The bottom row shows editing pairs grouped by shared editing intent across different scenes.

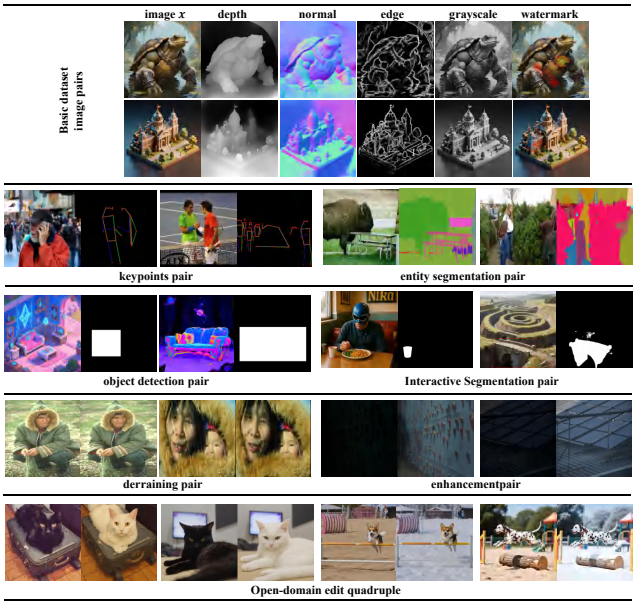

Figure 6: Examples of the constructed data. Standard-task image pairs are shown above; the bottom row presents open-domain editing quadruplets with matching edit intent across different scenes.

E Additional Quantitative Results

E.1 Single-Task and Joint Training

We compare the joint model with separately trained depth and surface-normal models in Table 3. Joint training improves depth AbsRel from 0.145 to 0.138 and δ_1 from 0.796 to 0.812. For surface normals, median and mean angular errors decrease from 18.295 and 21.913 to 17.582 and 20.187, respectively. These results show that the joint model retains strong performance on both geometric tasks while supporting the other transformations in the training mixture.

E.2 Sensitivity to Exemplar Selection

To evaluate stability against the inherent sensitivity of ICL, we compare a Curated Exemplar with Random Exemplars, where we report the mean and standard deviation derived from 5 randomly sampled pairs. As shown in Table 14, VIRAL exhibits minimal variance; the curated exemplar performs on par with the random average across tasks. This consistency indicates that our model successfully decouples transformation logic from incidental visual content, performing robust semantic analogy rather than relying on spurious pixel-level alignments. Consequently, VIRAL remains reliable even when user-provided demonstrations are varied or sub-optimal.

E.3 Few-shot Context Scaling

The main paper adopts a one-shot setting because a single exemplar pair is sufficient for strong performance and provides a simple evaluation protocol. To examine whether VIRAL can use additional context, we also train and evaluate a two-shot variant on representative standard tasks. In this setting, two support exemplar pairs are concatenated into the conditioning sequence and assigned distinct role-aware positional encodings.

Table 14: Within-task exemplar selection for depth and normal estimation. The random-exemplar column reports the recorded mean and standard deviation over five selections. Protocol and metric definitions are specified in the accompanying text.

Task	Metric	Fixed Exemplar	Random Exemplars
Depth	AbsRel ↓	0.1408	0.1409 ± 0.0004
	δ_1 ↑	0.7944	0.8085 ± 0.0012
Normal	Med ↓	20.643	20.925 ± 0.0496
	Mean ↓	18.028	18.195 ± 0.0202

Table 15 shows that adding a second exemplar yields small gains on most metrics, especially object detection and edge LPIPS, but increases inference time from 25s to 38s per image. This suggests that VIRAL can incorporate multiple exemplars, while the one-shot setting already captures most of the practical benefit at lower cost.

Table 15: One-shot and two-shot variants on the evaluated task subset. Two-shot conditioning improves several metrics and increases runtime; segmentation IoU is lower.

Setting	Time/img	Depth		Seg.	Edge		Normal		Obj. Det.
		AbsRel ↓	δ_1 ↑	IoU ↑	RMSE ↓	LPIPS ↓	Med ↓	Mean ↓	IoU ↑
One-shot	25s	0.1431	0.801	**0.738**	32.3817	0.1890	18.653	20.166	0.527
Two-shot	38s	**0.1430**	**0.802**	0.731	**31.3361**	**0.1796**	**18.224**	**20.012**	**0.626**

E.4 Resolution, Runtime, and Distillation

Resolution. We evaluate VIRAL under different inference resolutions in Table 16. The model remains effective at both 512 and 1024 resolution. Higher resolution slightly improves depth AbsRel and object detection IoU and improves edge LPIPS, but it also increases memory and runtime cost. We therefore use 512 resolution as the default trade-off in the main experiments.

Table 16: Performance at two inference resolutions. Increasing resolution produces metric-dependent changes.

Resolution	Depth		Edge		Obj. Det.
	AbsRel ↓	δ_1 ↑	RMSE ↓	LPIPS ↓	IoU ↑
512	0.138	**0.812**	**28.59**	0.133	0.562
1024	**0.137**	0.807	31.97	**0.120**	**0.565**

Runtime measurement. We also report inference cost in Table 17. VIRAL is slower than lightweight grid-based V-ICL baselines because it uses a stronger DiT editing backbone and processes multiple image token sequences. This is an expected cost of a flexible generative interface rather than a replacement for efficient specialist models.

Three-step distilled variant. To evaluate practical acceleration, we test a 3-step distilled variant. As shown in Table 18, it reduces inference time to 1.472s per image while retaining competitive performance, although with some degradation relative to the 40-step model.

E.5 Additional Comparison with SD-VICL

We include an additional comparison with SD-VICL (Oorloff et al., 2025), a recent training-free method based on Stable Diffusion attention manipulation. Since the official code was not released at the time of our experiments, we use an open-source unofficial reproduction and report the results as an informative comparison rather than a definitive reproduction of the original method.

Table 17: Recorded inference times and reported trainable-parameter counts. For Painter, 448 denotes each image; its source tensor is 896×448.

Method	Trainable Param.	Time/Image	Input Resolution
IMProv	443M	0.517s	226
VisualPrompt	89M	0.589s	226
PromptDiffusion	362M	3.83s	512
Painter	370M	1.413s	448
VIRAL	412M	25.366s	512
VIRAL (distilled)	412M	1.472s	512

Table 18: Accuracy of the standard 40-step model and the separately distilled three-step variant. The corresponding reported times are 25.366 and 1.472 seconds per image.

Method	Depth		Normal		Seg.	Obj. Det.	Edge	
	AbsRel ↓	δ_1 ↑	Med ↓	Mean ↓	IoU ↑	IoU ↑	RMSE ↓	LPIPS ↓
VIRAL	**0.138**	**0.812**	**17.582**	**20.187**	**0.795**	**0.562**	**28.59**	**0.133**
VIRAL (distilled)	0.1540	0.7761	18.132	22.784	0.7313	0.5271	32.381	0.178

SD-VICL transfers context by using the query image as attention queries and the exemplar pair as keys and values. This design implicitly benefits from selecting a reference exemplar that is visually similar to the query. In our benchmark, however, exemplar pairs are randomly sampled from the held-out test split to match the V-ICL protocol used for all methods. Under this random-exemplar setting, SD-VICL lags behind VIRAL across open-domain editing and structured perception tasks, as shown in Table 19. These results support the need for explicit role-aware multi-image conditioning and quadruplet training when the query and exemplar images are not near duplicates.

Table 19: Comparison with the evaluated unofficial SD-VICL implementation under the shared random-exemplar protocol.

Method	Open-domain Edit		Depth		Normal		Seg.	Edge		Obj. Det.
	CLIP ↑	LPIPS ↓	AbsRel ↓	δ_1 ↑	Med ↓	Mean ↓	IoU ↑	RMSE ↓	LPIPS ↓	IoU ↑
SD-VICL	0.524	0.841	0.168	0.728	41.764	42.400	0.1222	56.940	0.6337	0.2556
VIRAL	**0.880**	**0.517**	**0.138**	**0.812**	**17.582**	**20.187**	**0.795**	**28.59**	**0.133**	**0.562**

F Generalization and Qualitative Results

F.1 Lineart Generation

Lineart generation is excluded from VIRAL training. Figure 7 shows one-shot predictions guided by an exemplar pair on new query images. The examples complement the quantitative results in Table 7, where VIRAL obtains 34.82 RMSE and 0.339 LPIPS. The evaluation source, target-generation procedure, intensity convention, and metrics are specified in Table 10.

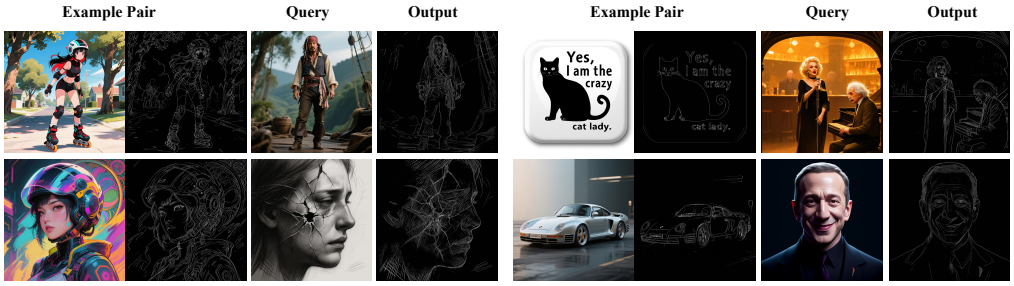

Figure 7: One-shot lineart generation on new queries. Each exemplar pair specifies the desired contour rendering; lineart pairs are excluded from VIRAL training.

F.2 Cross-Task Generalization and Domain Robustness

To investigate whether visual context effectively dictates task semantics against strong data-driven priors, we conduct a cross-domain inference experiment. We pair a Depth Estimation exemplar with query images sampled from distinct domains, specifically Deraining and Low-light Enhancement. As illustrated in Figure 8, VIRAL consistently executes the context-specified depth estimation rather than triggering the restoration tasks typically associated with such degraded inputs. The high fidelity of the generated depth maps demonstrates an emergent capability to perceive underlying geometry amidst severe visual corruption. These results indicate that the exemplar pair functions as a dominant non-parametric instruction, effectively overriding the inductive biases of the query domain. Furthermore, this demonstrates that our model successfully disentangles high-level task logic from low-level image statistics, establishing the framework as a genuine in-context reasoner capable of robust generalization in out-of-distribution

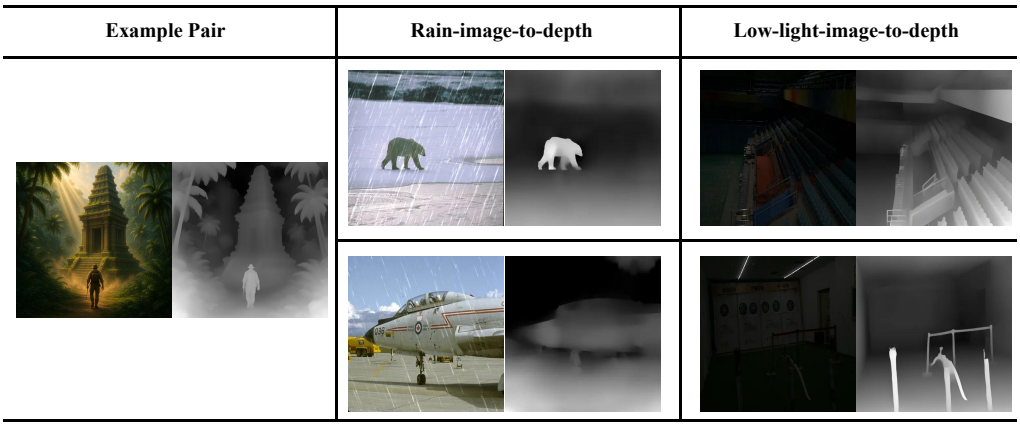

Figure 8: Depth prediction on rainy and low-light queries. A depth exemplar specifies the output representation for queries with different input appearances.

F.3 Generation from Geometric Conditions

Figure 9 presents the reverse direction of the geometric tasks: exemplar pairs map depth, edge, or surface-normal images to RGB images, and the query provides another geometric condition. The generated outputs illustrate the use of this visual interface for conditional image synthesis.

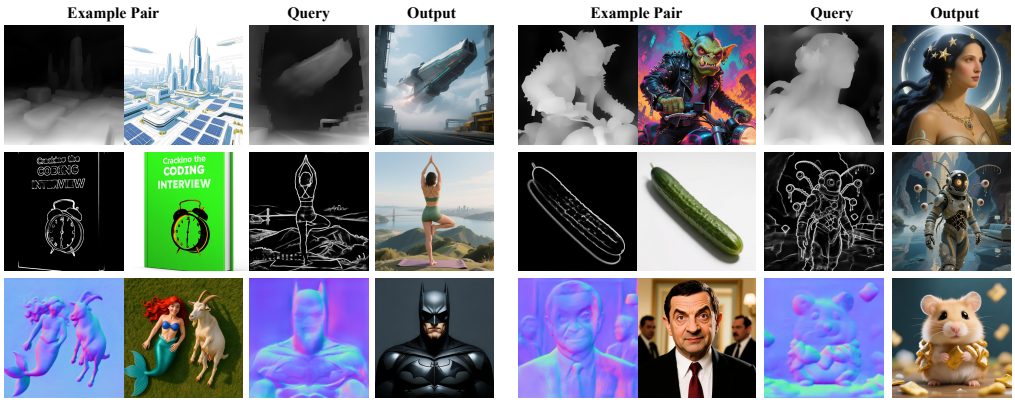

Figure 9: Generation from geometric conditions. Exemplar pairs map depth, edge, or surface-normal images to RGB outputs, and VIRAL applies the mapping to new conditions.

F.4 Image Deraining

Figure 10 compares VIRAL with CSUD (Dong et al., 2025) on selected queries. VIRAL removes visible rain streaks in these examples, with differences in local texture and appearance relative to the reference images. In the quantitative comparison in Table 2, CSUD achieves 33.31 PSNR and 0.957 SSIM, compared with 29.67 and 0.889 for VIRAL. The qualitative examples illustrate the outputs of the unified model alongside this specialist comparison.

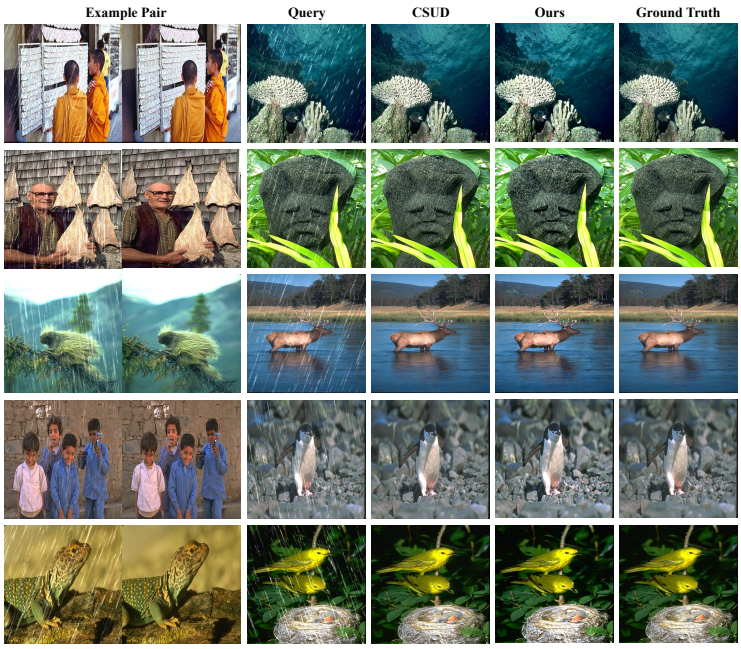

Figure 10: Qualitative deraining comparison. Each row shows the exemplar pair, rainy query, CSUD output, VIRAL output, and reference image.

F.5 Additional Task Examples

The following figures show the exemplar pair, query, and generated output for open-domain editing and individual visual tasks. They illustrate the transformations represented by the shared image interface.



Figure 11: Open-domain editing with visual exemplars. The examples include changes in appearance, material, and scene content.

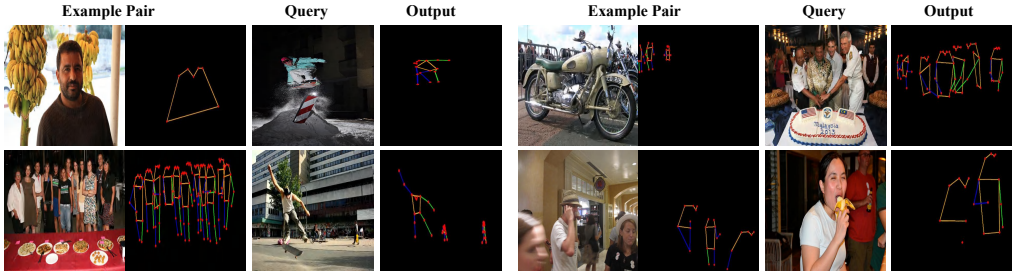

Figure 12: Human pose estimation from a visual exemplar. Each group shows the exemplar pair, query, and predicted pose map.

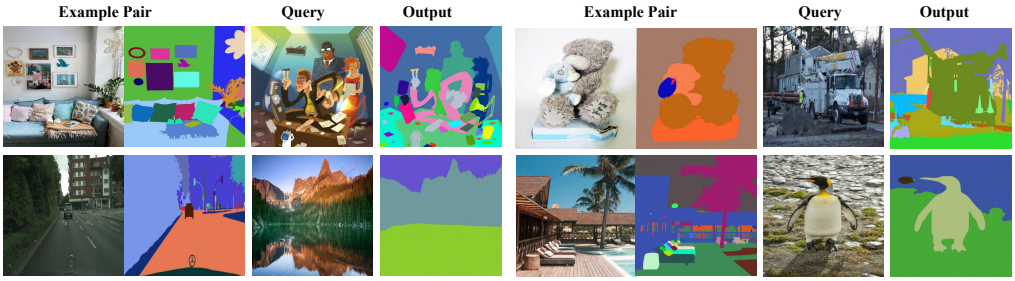

Figure 13: Entity segmentation from a visual exemplar. Predicted regions are represented by distinct colors.

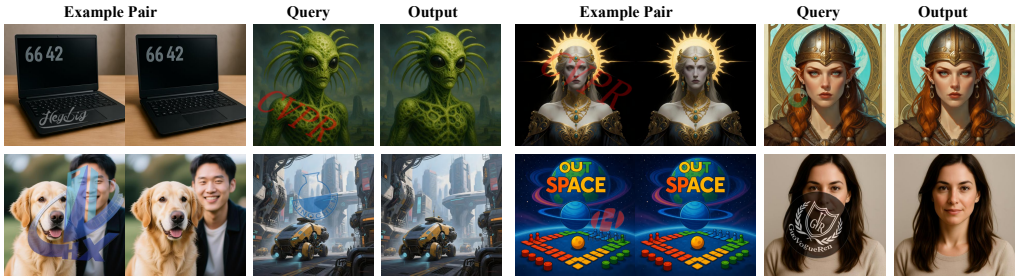

Figure 14: Watermark removal from a visual exemplar. Each group shows the exemplar pair, watermarked query, and generated output.

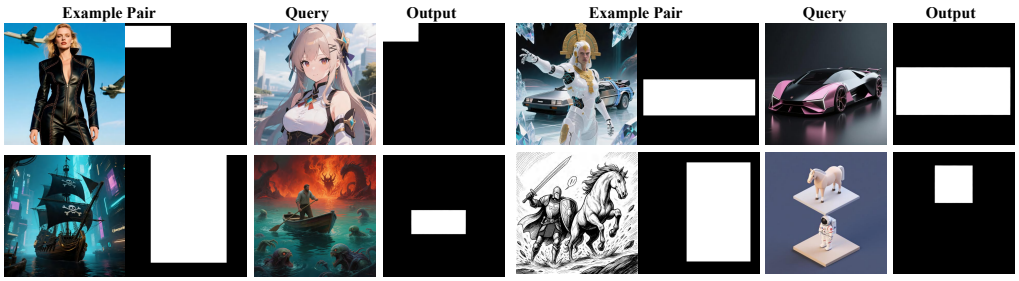

Figure 15: Category-specific localization. The exemplar specifies the target category, and the output represents predicted boxes as filled white regions.

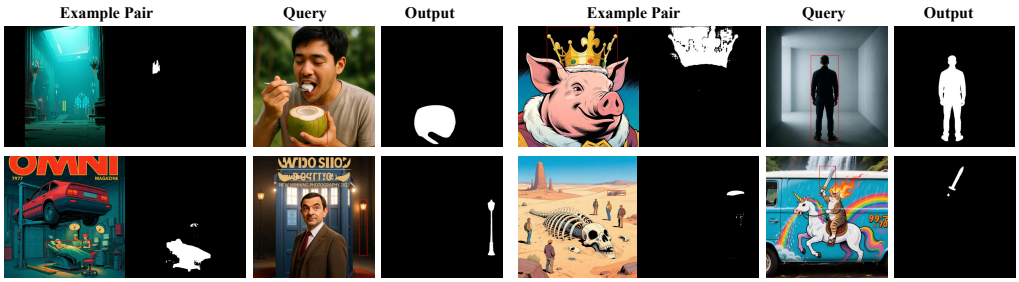

Figure 16: Interactive segmentation. A visible box identifies the query object, and VIRAL predicts its binary mask using the exemplar-defined output format.

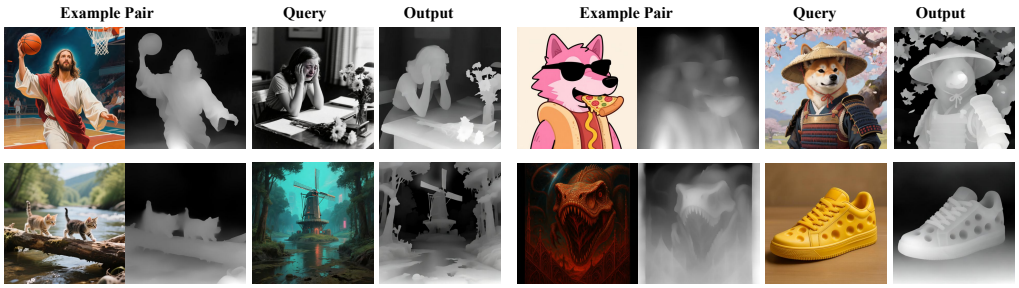

Figure 17: Depth estimation from a visual exemplar. Each group shows the exemplar pair, query, and predicted depth map.

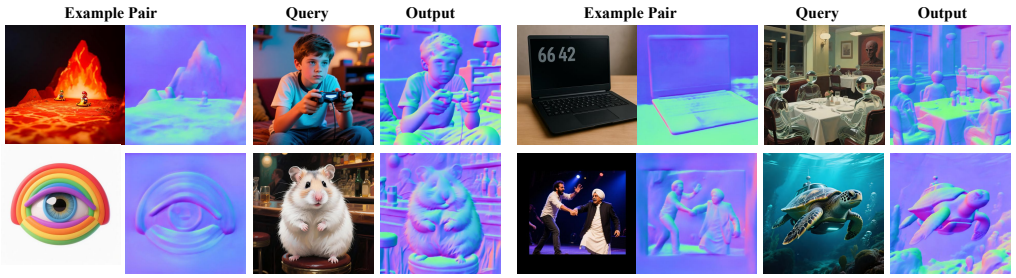

Figure 18: Surface-normal estimation from a visual exemplar. Each group shows the exemplar pair, query, and predicted normal map.

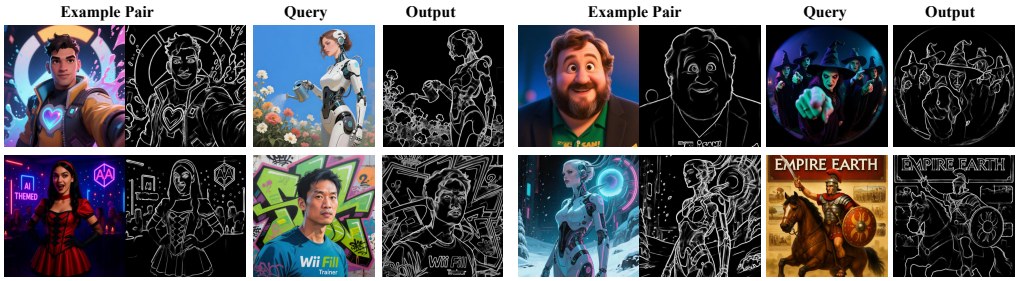

Figure 19: Edge estimation from a visual exemplar. Each group shows the exemplar pair, query, and predicted edge map.

G LIMITATIONS

VIRAL offers a common visual-demonstration interface, but its 40-step generative inference has substantial latency. The three-step distilled variant reduces this cost with accuracy losses on the measured tasks. Performance against task-specific models also varies: the deraining specialist obtains higher PSNR and SSIM in our comparison.

The framework represents task outputs as images. Structured quantities such as depth and surface normals therefore require an explicit encoding and metric-conversion protocol. Some training targets are generated by pre-trained predictors or editing models, so annotation errors can influence learned transformations despite filtering and manual inspection.

Generalization is evaluated through held-out styles, lineart generation, a separate detection dataset, and selected qualitative examples. Broader coverage of editing combinations and stronger guarantees of fidelity remain directions for further study. The current implementation uses one pre-trained DiT backbone; evaluating the framework across additional backbones would clarify how its behavior depends on the underlying generative model.