QUO VADIS, VISUAL IN-CONTEXT LEARNING? A UNIFIED BENCHMARK ACROSS DOMAINS AND TASKS

Anonymous authors
Paper under double-blind review



Figure 1: Overview of our Visual In-Context BEnchmark (VIBE) with diverse tasks and domains.

ABSTRACT

Visual in-context learning has been proposed as a pathway towards dynamic models that can generate predictions based on a provided context and thereby can adapt to new vision tasks at test-time. Yet, the evaluation of the *adaptation capabilities* of these models has been limited to narrow setups that mainly mirror tasks or image domains from pre-training for which real adaptation is not required. We address this gap by constructing a broad Visual In-Context BEnchmark (VIBE) with a focus on diverse imaging domains and a wide range of tasks. With this, we are able to get a much clearer picture of the adaptive capabilities of visual in-context models when faced with new image- and task distributions. We stress test seven models on 14 datasets and 12 tasks (in total, we explore 106 dataset-task combinations) and compare them under a unified, reproducible evaluation protocol, in an one-shot setting. Our evaluation uncovers key insights on the state of visual in-context learning, including limitations, systematic failure modes and promising directions. To foster broader evaluation, we openly release our VIBE toolkit upon acceptance.

1 INTRODUCTION

In the bygone decade, much of the success in computer vision research centered around using deep neural networks (Krizhevsky et al., 2012; Girshick, 2015; Ronneberger et al., 2015) to address new tasks through supervised training on task-specific curated datasets. While offering a generic pathway towards vision systems for different applications, a fundamental prerequisite for this paradigm is the access to large, labeled datasets (Everingham et al., 2010; Deng et al., 2009; Lin et al., 2014).

Advancing vision models towards requiring fewer labeled examples for adapting to new tasks, similar to humans (Walker & Gopnik, 2014), is still a fundamental open question. Different prominent research strains have tried to address this, *e.g.*, by self-supervised representation learning (Vincent et al., 2008; Chen et al., 2020) and successive transfer learning with a small set of labeled examples (Thrun, 1995; Caruana, 1997; Donahue et al., 2014), semi-supervised learning where few labeled examples

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.

are used in conjunction with a large set of unlabeled samples (Lee et al., 2013; Arazo et al., 2020; Sohn et al., 2020), or, few-shot learning (Snell et al., 2017; Finn et al., 2017; Ravi & Larochelle, 2017), where models converge with few examples, *e.g.*, through meta-learning (Schmidhuber, 1987).

Recently, visual in-context learning (VICL) has been explored as a different pathway towards data-efficient adaptation (Bar et al., 2022). VICL models are given a few examples for a new task (the context), as input to the model at test-time as well as a query image for which the new task shall be solved. Thereby, the model needs to infer the task from the given context first and successively apply it to the query image to produce a fitting output. This setup can be seen as solving a 'visual analogy' (Hertzmann et al., 2001) in the form:

$$\begin{array}{c} C_{\text{in}} \rightarrow C_{\text{out}} \\ Q \rightarrow \ ? \end{array} \tag{1}$$

where the context pair (C_{in}, C_{out}) demonstrates a transformation, which then has to be applied to the query image Q in order to fill the placeholder ? with a corresponding output image. This is a significant shift as it views models as inherently contextual and thereby turning them into dynamic analogy-making systems rather than static, task-specific models. As visual analogies can already be defined with one example pair of images, data-efficient adaptation is at the center of this learning paradigm. Interestingly, this view coincides with findings in human cognition, where analogies have been argued to be a driver of conceptualizations of wholly new situations (Mitchell, 2021) and may even lie at the core of human thought (Hofstadter et al., 2001).

Early on, image analogies were explored for texture synthesis (Hertzmann et al., 2001). More recently, research has turned to deep learning models for solving visual analogies, *e.g.*, Sadeghi et al. (2015) trained a model to score candidate images and select the one best completing the analogy. Similarly, visual analogies were explored in terms of Raven-style Progressive Matrices (Raven, 1941), *i.e.*, simple visual IQ tests (Barrett et al., 2018; Hill et al., 2019). While candidate scoring remains an important way to probe the contextual capabilities of vision models (Born et al., 2026; Bitton et al., 2023), the next step was generating the output image itself, as Webb et al. (2020) did for simple geometric patterns by conditioning generation on an encoding of the context.

The recent culmination of vision foundation models (Rombach et al., 2022; Dosovitskiy et al., 2021; He et al., 2022) with rich image representations and observations on few-shot prompting in natural language processing (Brown et al., 2020), have sparked new interest in visual analogies under the term visual in-context learning (Bar et al., 2022) and have created space for imagination to capture more complex tasks. Rather than operating on image textures (Hertzmann et al., 2001), geometric patterns (Barrett et al., 2018) or line drawings (Lu et al., 2019), the models tackle vision tasks like segmentation, colorization or denoising, which are rearranged into visual analogies.

In this new attempt at training data-efficient, adaptive vision models, the focus has mostly been on architectural designs for conditioning the models on context and query images, on the training strategy and the data for training these models. Now that the community trained a wide bouquet of models (Bar et al., 2022; Bai et al., 2024; Wang et al., 2023a; Oorloff et al., 2025; Wang et al., 2023b; Hao et al., 2024; Xu et al., 2024; Reiß et al., 2025; Czolbe & Dalca, 2023; Schmidt & Reiß, 2026; Negrini & Reiß, 2025), what remains unexplored is an in-depth evaluation and analysis of their capabilities, limitations and failure modes. So far, evaluation has been limited to image distributions present in training and simplified task evaluation settings that conceal how far models predict contextually rather than taking shortcuts.

We address this gap by evaluating the most prominent visual in-context learning models on multi-domain visual analogies, which has been identified as important factor for making progress in the field (Mitchell, 2021). Specifically, we stress test VICL models on our Visual In-Context BEnchmark (VIBE), on out-of-domain images from medicine, underwater- and aquatic environments, on images from art and aesthetics, wildlife as well as object- and human-centric imagery in natural- and infrared imagery. Further, we include a wide range of vision tasks, including perception and localization tasks, restoration and image enhancement tasks, generative image manipulation tasks and we also probe model's capability with geometric transformations, and subsequently provide a detailed analysis.

Our contributions are as follows. (1) We introduce the VIBE toolkit with a new evaluation protocol that includes deterministic context sampling, unified task encodings and unified post-processing to obtain metric scores. (2) We evaluate and stress test the seven most prominent VICL models on

the currently broadest set of 12 tasks, 14 datasets across 6 domains in 106 one-shot task-dataset pairings. (3) We derive key insights of the current state of visual in-context learning, including model properties and failure modes in this broad evaluation, starting to isolate effects of design choices made with respect to model architectures, training data and training strategies.

2 Related work

Visual in-context learning In natural language processing (NLP), specifically in the GPT-3 model (Brown et al., 2020), notable few-shot learning capabilities were observed, without weight updates, but by merely providing the model with few-shot examples of the task to solve in the input. The first approach to adapt this setup to the vision domain is MAE-VQGAN (Bar et al., 2022), which designed the conditional input as an image grid, where the context input-output examples are placed at the top next to each other, while the query image is placed in the bottom left. The model inpaints the masked bottom right region by predicting tokens from a visual codebook (Esser et al., 2021). Following a similar design, but moving away from uncurated Computer Vision Figures as training data and towards a collection of curated vision datasets is the Painter model (Wang et al., 2023a). There, the training process is designed as pixel-wise regression utilizing a Vision Transformer (ViT) (Dosovitskiy et al., 2021) with a masked image modeling strategy (He et al., 2022). The IMProv model (Xu et al., 2024) added a way to integrate textual task prompts into the grid-based design for multi-modal in-context prediction. The Visual Token Matching model (Kim et al., 2023) also uses a ViT, but re-designs the self-attention mechanism to hard-wire contextual prediction on indoor data.

To closely mirror the training recipe from NLP, the Large Vision Model (LVM) (Bai et al., 2024) is a 7 billion parameter model trained on a large amount of vision datasets and formulates VICL as sequence prediction. Further, an image quantization model (Esser et al., 2021) is utilized to operate on a visual vocabulary rather than pixels, mirroring language models. In an effort to cope with the data intensive compute of the LVM training recipe, DeLVM (Hao et al., 2024) was introduced with a focus on data-efficient training, utilizing a much smaller amount of datasets and leveraging the idea of knowledge distillation (Hinton et al., 2015) to distill smaller, parameter-efficient model variants.

Diffusion models (Rombach et al., 2022) have seen some application in VICL through the approach PromptDiffusion (Wang et al., 2023b) which is trained on the synthetic InstructPix2Pix dataset (Brooks et al., 2023) and uses a ControlNet (Zhang et al., 2023) to integrate visual context. Analogist (Gu et al., 2024) also utilizes diffusion models, but is training-free. Similar to MAE-VQGAN, it builds an image grid as input and re-designs the self-attention mechanism towards contextual prediction. It further extracts a text prompt from the visual analogy to guide the prediction. SD-VICL (Oorloff et al., 2025) is a diffusion model-based VICL approach that is training-free and does not use text. It expands on ideas from image editing (Alaluf et al., 2024) and attention re-wiring (Kim et al., 2023).

Cross-domain benchmarks Benchmarks containing imaging data from diverse domains have been proposed in adjacent fields, *e.g.*, few-shot classification (Guo et al., 2020), few-shot object detection (Qiu et al., 2026), or, open vocabulary segmentation (Blumenstiel et al., 2023). For visual in-context classification (Chan et al., 2022), few-shot protocols on individual existing datasets (Lake et al., 2019) are often used. Zong et al. (2024) took a step further and propose a multi-modal benchmark on in-context classification with vision language models. While not multi-modal, the Visual analogy dataset (Yiu et al., 2024) matured visual analogies towards more diverse image types with focus on task diversity. Xia et al. (2025) propose a real-world dense prediction benchmark to explore fine-tuning with few labeled examples for semantic segmentation and depth estimation. Our efforts also consider dense prediction but on a much broader set of tasks and imaging domains, and, we are set in VICL, *i.e.*, prediction without any parameter optimization.

3 VIBE: Visual In-Context BEnchmark

The goal of our VIBE toolkit is to provide a unified, reproducible and extensible evaluation protocol for VICL across heterogeneous tasks and diverse imaging domains, which we introduce next.

3.1 PRELIMINARIES

Notation A dataset with N samples can be defined as $\mathcal{D} = \{(q_i, y_i, c_i)\}_{i=1}^{N}$, where $q_i \in \mathbb{R}^{3\times W\times H}$ is an input image, $y_i \in \mathbb{R}^{3\times W\times H}$ its encoded target image (*e.g.*, a segmentation map, depth map, *etc.*) and c_i a deterministically selected, fixed context for each query q_i. For a query image q_i, a VICL model $f_\theta(\cdot,\cdot)$ receives a context pair $c_i = (c_{\text{in}}, c_{\text{out}}) \in \mathbb{R}^{2\times 3\times W\times H}$ based on which it needs to infer the task and apply it to q_i to produce $\hat{y}_i = f_\theta(q_i, c_i)$. The goal is to predict $\hat{y}_i$ as close to y_i as possible, which is assessed with task-specific metrics. The process for selecting the context c_i for a query q_i is task-dependent, *i.e.*, for tasks with semantic classes (*e.g.*, semantic segmentation, object detection), the context is chosen via a class-aware greedy policy, to make sure the context contains as many classes in the query as possible, all remaining tasks follow fixed-seed random selection.

Pre-processing for visual tasks As all tasks are formulated as image-to-image transformations to provide a unified interface for processing them with a VICL model f_θ, tasks which are not naturally image-to-image tasks are encoded as RGB images. These task-encodings are designed to preserve the structure of the original annotations to retain the nature of the task. Segmentation masks are rendered with a color palette where each class is assigned a fixed color, depth maps are rendered with a color mapping that transfers the continuous distance values to colors, and, for edge detection, black and white images are used with white indicating the presence of an edge. The encoding for keypoint detection renders red circle markers as indication for the presence of a keypoint on a black image. For object detection the bounding boxes are rectangles filled with the value 1 in the red channel, and, in case of overlapping boxes that indicate different instances, we use the green and blue channels to encode them, allowing the representation of up to three instances at the same location.

3.2 VISION TASKS AND IMAGING DOMAINS IN THE VIBE TOOLKIT

Our VIBE toolkit comprises 14 datasets, 6 different domains and 12 tasks. We group the tasks into three categories based on their prediction objective. **Perception & Localization** covers tasks that require spatial understanding of images, namely, *semantic segmentation*, *edge detection*, *depth estimation*, *keypoint-* and *object detection*. **Restoration & Enhancement** includes tasks in which degraded or incomplete visual content has to be recovered, *i.e.*, *inpainting*, *denoising*, *colorization* and *artifact removal*. **Manipulation & Transformation** contains tasks where the model has to infer and apply a systematic modification to the input, covering *modality transfer*, *geometric transformations* and *object removal*. These categories allow us to evaluate the adaptive visual reasoning capabilities across the different models. Figure 2 shows the encoding for each task on an exemplary dataset.

Further, the benchmark covers six imaging domains to evaluate generalization beyond natural images. **Medical & biomedical** imaging includes scans of diseases and anatomy – challenging out-of-domain images for VICL models. We include Duke (Srinivasan et al., 2014), retinal optical coherence tomographies with diseased regions and layer annotations, and IXI (IXI Project), brain magnetic resonance images in two modalities. **Underwater & aquatic** environments introduce low-light and color distortions through optical properties of water. Datasets in this domain are ATLANTIS (Erfani et al., 2022), for water body segmentation, SUIM (Islam et al., 2020), for underwater segmentation and

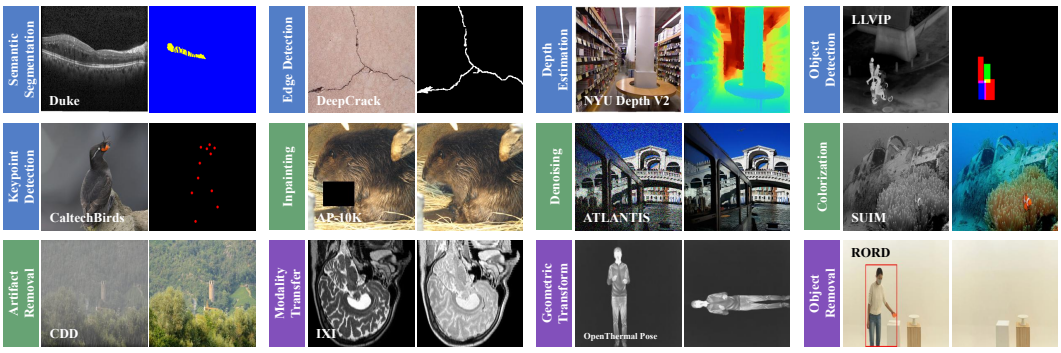


Figure 2: Visualization of task encodings in the VIBE toolkit. **Perception & Localization**, **Restoration & Enhancement**, and **Manipulation & Transformation** tasks tinted in their respective colors.

SeaThru (Akkaynak & Treibitz, 2019), a collection of underwater scenes with depth maps. **Indoor & structural** environments, represented by NYU Depth V2 (Silberman et al., 2012), a RGB-Depth dataset with densely annotated indoor scenes and DeepCrack (Liu et al., 2019), for crack segmentation. This domain challenges VICL models on geometry-driven prediction and detection of fine structures. **Arts, aesthetics and photography** covers images with artistic character, leading to unusual texture and color patterns, and scenic photography images. It includes DRAM (Cohen et al., 2022), a stylized painting dataset and CDD (Guo et al., 2024), an image restoration dataset with diverse degradations. **Object- and human-centric scenes** are centered on people or distinguished objects, including low-light, infrared and thermal imagery. We include OpenThermalPose (Kuzdeuov et al., 2024; 2025), a thermal human pose dataset, LLVIP (Jia et al., 2021), a low-light visible-infrared dataset and RORD (Sagong et al., 2022), an object removal dataset. **Wildlife and animal-centric** imagery focuses on animals in nature and their variation in appearance and pose. We use CaltechBirds (Wah et al., 2011), a bird species-, and AP-10K (Yu et al., 2021), an animal pose dataset.

4 Evaluation

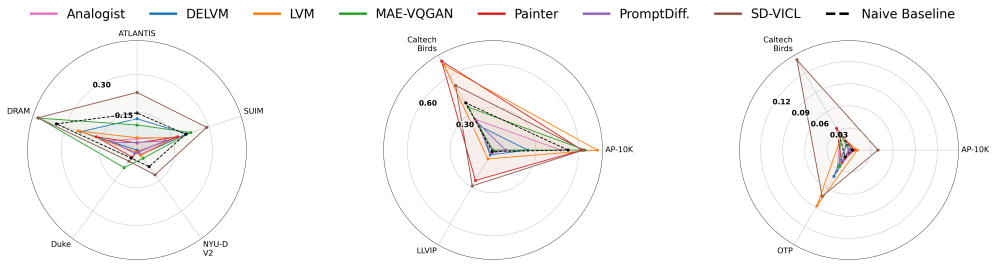
Next, we introduce the VICL models we stress test on VIBE, we mainly emphasize their differences. **MAE-VQGAN** (Bar et al., 2022) trains a ViT on an uncurated dataset of computer vision figures with masked image modeling (He et al., 2022) to predict tokens from a visual codebook (Esser et al., 2021). **Painter** (Wang et al., 2023a) uses a ViT and masked image modeling as well, but it is trained on a highly curated set of vision tasks that are put into a grid. Further, it directly regresses RGB values for the masked out regions. **LVM** (Bai et al., 2024) also mainly uses curated image data, but from 50 computer vision datasets, which are put into sequential form. The approach uses a codebook to shorten these sequences and trains a 7B parameter LLamA model (Touvron et al., 2023) with next token prediction (Radford et al., 2019). **DeLVM** (Hao et al., 2024) follows the training recipe of LVM on a set of five datasets and adds data augmentation to train a 1B parameter model. **PromptDiffusion** (Wang et al., 2023b) makes use of a pre-trained stable diffusion model (Rombach et al., 2022) for prediction. It builds upon ControlNet (Zhang et al., 2023) to condition the generation on the context and is trained on synthetic data (Brooks et al., 2023). **SD-VICL** (Oorloff et al., 2025) also uses stable diffusion but is a training-free approach. It expands on methods from style-transfer (Alaluf et al., 2024) to condition the image generation on the visual analogy by reformulating self-attention at inference. **Analogist** (Gu et al., 2024) is also training-free and based on stable diffusion with an adapted self-attention mechanism, but frames the prediction as grid inpainting similar to MAE-VQGAN and Painter. It also uses a VLM (we use Qwen3-VL-8B-Instruct (Bai et al., 2025)) to extract a text description from the image grid as additional input. **Copy target** is a naive baseline, that takes a target image from the context set as output.

4.1 Quantitative comparison

Next, we show the results for these seven VICL models on VIBE, which includes evaluation on 12 tasks across 14 datasets in a total of 106 task-dataset pairings. All computation is done on NVIDIA A100 GPUs in the one-shot setting introduced in Section 3.1 with one context pair defining the task.

4.1.1 Perception and localization tasks

We present our results either in numerical form through tables, or in task-specific spider-plots showing results on different datasets. We start with Figure 3, the dashed lines always indicate the Copy target baseline. For Figure 3a, we see the results for semantic segmentation on five datasets. The segmentation performance as measured in color-aware IoU for VICL models is generally low, only MAE-VQGAN and SD-VICL surpass the baseline, whereas the latter consistently exceeds it. One important factor for the low scores is that apart from these two models, no other model is able to follow the segmentation class-colors as defined in the given context examples, failing to shift away from their color-maps from pre-training. On the detection task, in Figure 3b, three models are able to solve the task consistently better than the baseline: Painter, LVM and SD-VICL. The lowest mAP scores are achieved for the LLVIP dataset, which requires the detection of multiple instances in images, making the task more challenging, while the other two datasets always contain one instance to localize. Keypoint detection is not well addressed by any model with low F1-Scores throughout. Concluding the results of the perception and localization tasks are edge detection and



(a) Semantic segmentation (IoU ↑) (b) Object detection (mAP@50% ↑) (c) Keypoint detection (F1-Score ↑)

Figure 3: Quantitative results for semantic segmentation in color-aware mIoU (for DUKE foreground class IoU is shown), object detection in mAP@50% and keypoint detection in F1-Score.

depth estimation in Table 1 and Table 2, respectively. For edge detection we see a similar pattern as before, where SD-VICL leads, followed by MAE-VQGAN. Interestingly, this is one of the few tasks, where PromptDiffusion works quite well, which can be attributed to its set of pre-training tasks including edge detection. The results on depth estimation indicate, that none of the models is able to coherently solve the task with RMSE values close or higher than Copy target, indicating that dense regression is a hard class of tasks in VICL. Mostly, models tend to produce segmentation maps or depth maps with different depth color coding as defined in the context. On the NYU depth indoor dataset, Analogist performs poorly, as it tends to generate dark images of rooms instead of depth maps, which likely stems from the VLM-extracted text prompts that commonly contain keywords such as *bedroom scene* or *dark room silhouette*. MAE-VQGAN and SD-VICL match the colors of the depth maps from the context, but do not correctly predict the depth.

4.1.2 Image restoration and enhancement tasks

In Figure 4, we consider tasks where a clean image needs to be restored from a corrupted image. Inpainting (Figure 4a) is done best by LVM, with the exception of the IXI dataset, where SD-VICL yields the highest PSNR. This is unsurprising, as LVM is trained on next-token prediction, which naturally aligns with predicting masked out regions. Similarly, MAE-VQGAN was pre-trained on masked image modeling but produces lower results, which can be attributed to the dated codebook used by the model, limiting the generated level of detail. Interestingly, Painter, also trained with a masking objective, is not able to coherently solve the task, just passing along the masked query.

The front runners for image denoising (Figure 4b) are Painter, SD-VICL and LVM. SD-VICL is the best model on the medical domain, where it even removes the original image noise in the vitreous body on the retina scans of DUKE. This may be due to the use of a quantized codebook (Rombach et al., 2022), which does not capture noise patterns well (Reiß et al., 2025). LVM also uses such a codebook while Painter does not, which aligns with the PSNR scores. While Painter performs well overall for denoising, on which it was pretrained, its results drop on medical scans and on Seathru. This is due to the low brightness in images of these datasets, where Painter incorrectly infers the low-light enhancement task from its pre-training, leading to bright instead of denoised predictions.

Table 1: Edge detection results on DUKE and DeepCrack.

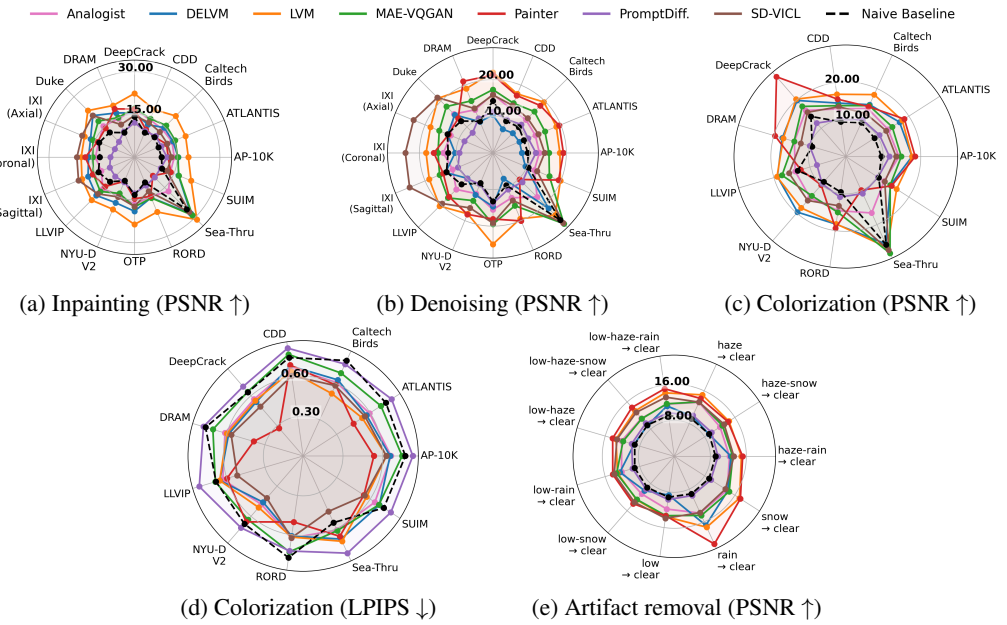
Method	DeepCrack			DUKE		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Copy target	0.05	0.05	0.04	0.05	0.06	0.06
Analogist	0.03	0.62	0.05	0.03	0.15	0.05
MAE-VQGAN	0.35	0.29	0.29	0.01	0.03	0.02
DeLVM	0.03	0.01	0.00	0.00	0.00	0.00
LVM	0.16	0.13	0.08	0.00	0.14	0.01
Painter	0.02	0.04	0.02	0.02	**0.73**	0.03
PromptDiffusion	0.20	0.29	0.20	0.01	0.32	0.01
SD-VICL	**0.72**	**0.69**	**0.62**	**0.55**	0.54	**0.54**

Table 2: Depth estimation (RMSE) on NYU and Seathru.

Method	NYU-D v2	Sea-thru
Copy target	1.60	**1.64**
Analogist	3.15	3.17
MAE-VQGAN	1.75	2.16
DeLVM	3.10	2.87
LVM	2.73	4.34
Painter	2.70	3.58
PromptDiffusion	2.76	2.16
SD-VICL	**1.48**	2.28

Table 3: Object removal results on RORD-mini in PSNR, SSIM and LPIPS.

Method	PSNR	SSIM	LPIPS
Copy target	9.58	0.22	0.77
Analogist	13.47	0.38	0.63
MAE-VQGAN	14.23	0.38	0.70
DeLVM	14.62	0.41	0.60
LVM	**15.84**	0.42	**0.57**
Painter	13.51	**0.49**	0.58
PromptDiffusion	10.68	0.34	0.75
SD-VICL	12.19	0.36	0.64



(a) Inpainting (PSNR ↑) (b) Denoising (PSNR ↑) (c) Colorization (PSNR ↑)

(d) Colorization (LPIPS ↓) (e) Artifact removal (PSNR ↑)

Figure 4: Quantitative results for image restoration tasks. We report PSNR for inpainting, denoising and artifact removal, for colorization PSNR and LPIPS. Artifact removal is done on the CDD dataset.

This pattern can also be seen on colorization (Figure 4c). There, LVM and DeLVM perform most robust in terms of PSNR. Yet, DeLVM and Painter merely predict the grayscale query image as 'colorized image', indicating, that in conjunction to PSNR, a perceptual metric like LPIPS (Figure 4d) needs to be considered for this task. Still, passing along the query image leads to perceptually sound images, which lead to low LPIPS scores in the case of Painter as opposed to, *e.g.*, LVM and SD-VICL, where the codebooks limit the generated level of detail, but which meaningfully colorize images.

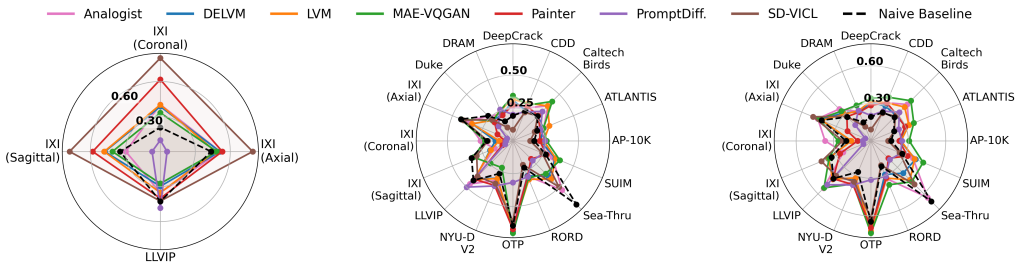
Lastly, in Figure 4e, we evaluate the models on artifact removal, *i.e.*, transferring snowy, rainy, hazy and low-light, to clear images. Painter, which was pre-trained on image de-raining, performs well on rain → clear, while SD-VICL and LVM mostly match it's performance, even with their limitation to not reconstruct fine details. PromptDiffusion performs worst producing no coherent image content.

4.1.3 Image manipulation and transformation tasks

As the last task group, we consider image manipulations and transformations. First, we start with the task of object removal, in Table 3, where models have to remove objects that are marked with red bounding boxes. While none of the models shine on the task, the only models that start to address it are LVM, which sometimes removes highly dynamic objects, like a moving person, which might be due to its pretraining on image sequences from video, and, SD-VICL, which consistently removes the red boxes and the area within it, but tends to produce artifact riddled reconstructions of the scene.

In Figure 5a, we show the performance of VICL models on transferring medical imaging modalities from T2 weighted to PD MRI images on the IXI dataset, and, RGB images to infrared on LLVIP. On MRI modality transfer SD-VICL works best with a substantial margin in PSNR. Due to the homogeneous target domain, the AdaIN (Huang & Belongie, 2017) mechanism in SD-VICL may help here. On LLVIP, all models cluster around the baseline, showing no meaningful task adaptation.

In our final task-specific evaluation in Figure 5b and Figure 5c, we test, whether models can cope with simple geometric transformations. Specifically, we look into whether VICL models can solve the task of rotating an image by 90° or flipping it horizontally. We see, that no model is able to accomplish this, as performance is mostly on par or worse than the baseline in SSIM, which is consistent with insights from Vision-Language models (Anis et al., 2025). There is just one notable exception, which is MAE-VQGAN, which exceeds the baseline consistently on the flip task, falling short only once.



(a) Modality transfer (PSNR ↑) (b) Rotation (SSIM ↑) (c) Flip (SSIM ↑)

Figure 5: Quantitative results for modality transfer in PSNR, and rotation and flipping in SSIM.

4.1.4 Ranking visual in-context learners

To get a clearer picture of which model performs best and adapts to the most datasets and tasks we cover in VIBE, we compute a ranking in Table 4, which is based on all datasets in each task. Then we take the average of the ranks for a final ranking (rightmost column). We see a clear winning model, which is the training-free SD-VICL approach, ahead of LVM, Painter, MAE-VQGAN, Analogist and DeLVM. PromptDiffusion ranks below the Copy baseline, indicating no real capabilities for adaptive task solving. Whereas, Analogist is the only model using a VLM-generated text prompt describing the visual analogy, but still does not outperform the best purely visual approaches.

	Sem. seg.	*Edge det.*	*Depth est.*	*Obj. det.*	*Keyp. det.*	*Inpaint*	*Denoise*	*Colorize*	*Artif. rem.*	*Mod. trans.*	*Rot.*	*Flip*	*Obj. rem.*	*Total*
SD-VICL	1	1	2	2	1	2	1	3	2	1	5	7	6	1
LVM	6	5	8	3	2	1	3	2	3	3	5	3	1	2
Painter	7	6	6	1	4	4	2	1	1	2	7	4	2	3
MAE-VQGAN	2	2	3	4	6	5	4	6	5	5	2	1	5	4
Analogist	5	4	7	6	3	5	5	5	4	7	4	2	4	5
DeLVM	4	8	5	7	5	3	7	4	6	4	3	4	3	6
Copy target	3	6	1	5	7	7	6	7	8	6	1	6	8	7
PromptDiffusion	8	3	4	8	8	8	8	8	7	8	8	8	7	8

Table 4: Ranking of the seven VICL models and baseline across all tasks and datasets, as well as average total rank.

4.2 Qualitative comparison

In Figure 6, we show the predictions of the seven models, which highlights failure cases, such as predicting task encodings from pretraining, handing through query images and failing to adapt to new tasks as well as notable skills, *e.g.*, SD-VICL's capabilities on edge detection and modality transfer, MAE-VQGAN on flipping an image or successful object removal by both (rightmost column). Analogist, led by the VLM-generated textual prompt, for some tasks seems to hallucinate image patterns, objects and scenes, *e.g.*, in column one a fish or in column three an indoor scene.

5 Discussion and conclusion

Most models are multi-task models, not in-context learners. With our experiments, it becomes evident, that most VICL models are heavily limited by their pre-training tasks. Their predictions are biased towards tasks seen in training and prompting them with new context sets still leads to outputs aligned with seen tasks. This shows that rather than being adaptive models, they seem to learn a *switch* based on the context that merely steers the model to an appropriate task from pre-training. The most notable exceptions are SD-VICL and MAE-VQGAN, which draw contextual predictions. Analogist can also predict contextually, yet, the use of text seems to nudge the model towards more text-aligned outputs, reducing context-dependence, which may cause hallucinations as seen in Figure 6.

Correspondence is key. Contextual prediction, which is fundamental for VICL, requires models to find relevant correspondences between the query image and the context. This is possible by hard-wiring such correspondences into the prediction process, as, *e.g.*, SD-VICL does with its altered self-attention mechanism, or, Analogist with its self-attention cloning. An alternative to this is providing models with an incentive to learn correspondences. MAE-VQGAN is an example where correspondences need to be learned due to the training data and objective. It is trained on *unstructured*

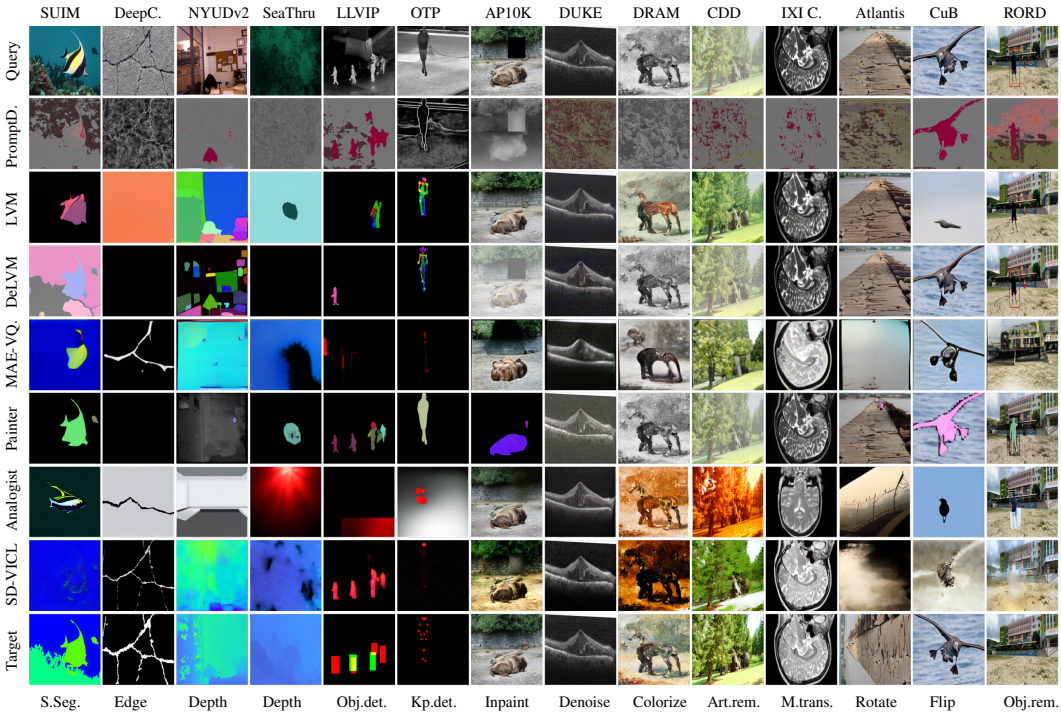


Figure 6: Qualitative results for subset of 14 task-dataset pairs in VIBE, showing skills and failures of models. Top row shows the dataset- and bottom row task abbreviations (best viewed zoomed in).

data from figures of computer vision papers. To solve masked image modeling on this data, MAE-VQGAN needs to learn correspondences within these diverse figures. Models like LVM, DeLVM, Painter and PromptDiffusion have no incentive to learn correspondences, as the context is always provided in a *structured* way, *i.e.*, through regular grids, visual sentences or ControlNet, leading them to take the shortcut of learning a *switch* based on the image statistics in the task-specific contexts of the pre-training tasks, degenerating them to multi-task models. Research on Visual in-Context Classification (Chan et al., 2022; Bratulić et al., 2025) also points to the central role of skewed distributions in the contexts in training as important factor to encourage learning correspondences.

Inherently contextual vision tasks. To a computer vision researcher, it is clear what is meant by a task like *image colorization*: sensibly fill in a grayscale image with colors that are plausible. Yet, this notion is vastly different from the contextual colorization task in VICL, where the context defines the task to carry out on a query image. Take a context of a wolf in a snowy forest, and a grayscale image of a bison in a forest as query image, as in Table 5. The ground-truth and metrics suggest that LVM's prediction is more accurate, although, based on the context, one can sensibly make the case that SD-VICL's prediction is far more fitting. This exemplifies the vastly different view on what a task is in the VICL setting, *i.e.*, inherently contextual, and profoundly ill-posed. To address this limitation, in evaluation context-sensitive metrics or ground-truths that vary with different contexts could be considered in the future. Altering reference-based perceptual metrics (Fu et al., 2023), *e.g.*, through using context sensitive image representations (Born et al., 2026), or designing human evaluation protocols, could be promising directions for better measuring adaptive capabilities of VICL models.



Table 5: Example of the ill-posedness in VICL showing the need for context sensitive metrics.

AI USE STATEMENT

In this work, we used generative AI tools as coding assistants. We have not used generative AI tools for writing in the main manuscript. We did not use AI tools for idea generation. In the supplemental material, we used AI tools to formulate text from bullet-points for coherent, fluent benchmark task descriptions. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including code artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We introduce a benchmark for evaluating visual in-context learning models and do not introduce a newly trained model. All datasets and models we use are publicly available and used in the form they were released with licenses listed in Table 9 and Table 10.

Datasets. Several datasets do involve human subjects, such as medical scans (IXI (IXI Project) and Duke (Srinivasan et al., 2014)) and images of people (LLVIP (Jia et al., 2021), OpenThermalPose (Kuzdeuov et al., 2024; 2025), RORD (Sagong et al., 2022) and ATLANTIS (Erfani et al., 2022)). We collect no new data nor annotate beyond the re-encoding of it. Our use is purely to analyse the data already released. On this basis, we do not require an additional ethical review.

Models. The models we benchmark inherit the biases of their pre-training data, which we do not specifically explore in our one-shot experiments. Generative models such as the openly accessible stable diffusion checkpoints used by some VICL approaches can be fine-tuned for misuse to generate deep fakes and illegal content (Foundation, 2026). Due to the nature of VICL – quick adaptation to new tasks – misusing the models is easier for a layperson as merely a malicious image analogy (an image pair and a query image) needs to be supplied and no re-training is involved. Yet, this also goes the other way, VICL models can be key enablers for addressing a vision task of interest to a layperson in a data-efficient way, *e.g.*, a researcher without programming skills and no budget for data annotation, a private citizen or small businesses. Our research assesses to what extent a broad diversity of productive vision tasks can be addressed by current VICL models.

REPRODUCIBILITY STATEMENT

We release the full implementation of the VIBE toolkit and all configuration files required to reproduce the results in this paper under an open-source license upon acceptance.

Further, we provide necessary information to reproduce our results in the paper and its appendix. The image-to-image task formulation of our benchmark, including specific task encodings along with the details on post-processing applied before evaluation, are described in the appendix Section A.3. Information about the benchmark protocol, as well as the different implementation steps are described in the appendix Section A.4. Sources and licenses for all used datasets and models are provided in the appendix (Table 9 and Table 10). Considering datasets used, we also describe in detail how they are used in VIBE in Table 6. Especially, for the models used in VIBE, the appendix reports their architectures in Table 8 and the exact implementations we build on in Table 10. Hardware used and an upper bound for the runtime needed for all experiments are reported in Section A.6 and Table 7. Pre-processing, task encoding formulations and the post-processing for evaluation are performed automatically by our toolkit. This means no manual data preparation is needed.

REFERENCES

Derya Akkaynak and Tali Treibitz. Sea-thru: A method for removing water from underwater images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1682–1691, 2019.

Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–12, 2024.

Ahmad Mustafa Anis, Hasnain Ali, and Saquib Sarfraz. On the limitations of vision-language models in understanding image transforms. *arXiv preprint arXiv:2503.09837*, 2025.

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22861–22872, 2024.

Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *Advances in neural information processing systems*, 35:25005–25017, 2022.

David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. Measuring abstract reasoning in neural networks. In *International conference on machine learning*, pp. 511–520. PMLR, 2018.

Yonatan Bitton, Ron Yosef, Eliyahu Strugo, Dafna Shahaf, Roy Schwartz, and Gabriel Stanovsky. Vasr: Visual analogies of situation recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 241–249, 2023.

Benedikt Blumenstiel, Johannes Jakubik, Hilde Kühne, and Michael Vössing. What a mess: Multi-domain evaluation of zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 36:73299–73311, 2023.

Frieda Born, Tom Neuhäuser, Lukas Muttenthaler, Brett D Roads, Bernhard Spitzer, Andrew K Lampinen, Matt Jones, Klaus-Robert Müller, and Michael C Mozer. Context sensitivity improves human-machine visual alignment. *arXiv preprint arXiv:2604.13883*, 2026.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Jelena Bratulić, Sudhanshu Mittal, David T Hoffmann, Samuel Böhm, Robin Tibor Schirrmeister, Tonio Ball, Christian Rupprecht, and Thomas Brox. Unlocking in-context learning for natural datasets beyond language modelling. In *DAGM German Conference on Pattern Recognition*, pp. 303–319. Springer, 2025.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. *Advances in neural information processing systems*, 35:18878–18891, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Nadav Cohen, Yael Newman, and Ariel Shamir. Semantic segmentation in art paintings. In *Computer graphics forum*, volume 41, pp. 261–275. Wiley Online Library, 2022.

Andrew Collette. *Python and HDF5*. O'Reilly, 2013.

Steffen Czolbe and Adrian V Dalca. Neuralizer: General neuroimage analysis without re-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6217–6230, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pp. 647–655. PMLR, 2014.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Seyed Mohammad Hassan Erfani, Zhenyao Wu, Xinyi Wu, Song Wang, and Erfan Goharian. Atlantis: A benchmark for semantic segmentation of waterbody images. *Environmental Modelling & Software*, 149:105333, 2022.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Internet Watch Foundation. Ai-generated child sexual abuse: 2026 report on trends, data & human impact. https://www.iwf.org.uk/about-us/why-we-exist/our-research/how-ai-is-being-abused-to-create-child-sexual-abuse-imagery/, 2026. Accessed: 2026-05-05.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Zheng Gu, Shiyuan Yang, Jing Liao, Jing Huo, and Yang Gao. Analogist: Out-of-the-box visual in-context learning with image diffusion model. *ACM Transactions on Graphics (TOG)*, 43(4): 1–15, 2024.

Yu Guo, Yuan Gao, Yuxu Lu, Huilin Zhu, Ryan Wen Liu, and Shengfeng He. Onerestore: A universal restoration framework for composite degradation. In *European conference on computer vision*, pp. 255–272. Springer, 2024.

Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *European conference on computer vision*, pp. 124–141. Springer, 2020.

Zhiwei Hao, Jianyuan Guo, Chengcheng Wang, Yehui Tang, Han Wu, Han Hu, Kai Han, and Chang Xu. Data-efficient large vision models through sequential autoregression. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17572–17596. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/hao24b.html.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pp. 327–340, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 158113374X. doi: 10.1145/383259.383295. URL https://doi.org/10.1145/383259.383295.

Felix Hill, Adam Santoro, David GT Barrett, Ari S Morcos, and Timothy Lillicrap. Learning to make analogies by contrasting abstract relational structure. *arXiv preprint arXiv:1902.00120*, 2019.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, 2015. URL http://arxiv.org/abs/1503.02531.

Douglas R Hofstadter et al. Analogy as the core of cognition. *The analogical mind: Perspectives from cognitive science*, pp. 499–538, 2001.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510, 2017.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3): 90–95, 2007. doi: 10.1109/MCSE.2007.55.

Md Jahidul Islam, Chelsey Edge, Yuyang Xiao, Peigen Luo, Muntaqim Mehtaz, Christopher Morse, Sadman Sakib Enan, and Junaed Sattar. Semantic segmentation of underwater imagery: Dataset and benchmark. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 1769–1776. IEEE, 2020.

IXI Project. Ixi dataset: Information extraction from images. https://brain-development.org/ixi-dataset/. URL https://brain-development.org/ixi-dataset/.

Xinyu Jia, Chuang Zhu, Minzhen Li, Wenqi Tang, and Wenli Zhou. Llvip: A visible-infrared paired dataset for low-light vision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3496–3504, 2021.

Donggyun Kim, Jinwoo Kim, Seongwoong Cho, Chong Luo, and Seunghoon Hong. Universal few-shot learning of dense prediction tasks with visual token matching. *arXiv preprint arXiv:2303.14969*, 2023.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Askat Kuzdeuov, Darya Taratynova, Alim Tleuliyev, and Huseyin Atakan Varol. Openthermalpose: An open-source annotated thermal human pose dataset and initial yolov8-pose baselines. In *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–8. IEEE, 2024.

Askat Kuzdeuov, Miras Zakaryanov, Alim Tleuliyev, and Huseyin Atakan Varol. Openthermalpose2: Extending the open-source annotated thermal human pose dataset with more data, subjects, and poses. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. The omniglot challenge: a 3-year progress report. *Current Opinion in Behavioral Sciences*, 29:97–104, 2019.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896. Atlanta, 2013.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 2019.

Hongjing Lu, Qing Liu, Nicholas Ichien, Alan L Yuille, and Keith J Holyoak. Seeing the meaning: Vision meets semanticsin solving pictorial analogy problems. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 41, 2019.

TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. `https://github.com/pytorch/vision`, 2016.

Christopher J. Markiewicz, Matthew Brett, Michael Hanke, Marc-Alexandre Côté, Ben Cipollini, Paul McCarthy, Dimitri Papadopoulos Orfanos, Dorota Jarecka, Christopher P. Cheng, Eric Larson, Yaroslav O. Halchenko, Michiel Cottaar, Satrajit Ghosh, Demian Wassermann, Stephan Gerhard, Gregory R. Lee, Zvi Baratz, Brendan Moloney, Hao-Ting Wang, Erik Kastman, Jakub Kaczmarzyk, Roberto Guidotti, Jonathan Daniel, Or Duek, Ariel Rokem, Mathieu Scheltienne, Cindee Madison, Anibal Sólon, Félix C. Morency, Mathias Goncalves, Ross Markello, Cameron Riddell, Christopher Burns, Jarrod Millman, Alexandre Gramfort, Jaakko Leppäkangas, Jasper J.F. van den Bosch, Jon Haitz Legarreta, Robert D. Vincent, Henry Braun, Krish Subramaniam, Andrew Van, Krzysztof J. Gorgolewski, Pradeep Reddy Raamana, Julian Klug, Reinder Vos de Wael, B. Nolan Nichols, Eric M. Baker, Serge Koudoro, Soichi Hayashi, Basile Pinsard, Christian Haselgrove, Mark Hymers, Oscar Esteban, Fernando Pérez-García, Guillaume Becq, Jérôme Dockès, Nikolaas N. Oosterhof, Bago Amirbekian, Benjamin Thyreau, Horea Christian, Ian Nimmo-Smith, Ly Nguyen, Peter Suter, Samir Reddigari, Samuel St-Jean, Alex Daniel, Egor Panfilov, Eleftherios Garyfallidis, Gael Varoquaux, Joshua Newton, Kevin S. Hahn, Lea Waller, Leonardo Ayala, Oliver P. Hinds, Sandro, Bennet Fauber, Blake Dewey, Fabian Perez, Jacob Roberts, Jean-Baptiste Poline, Jon Stutters, Kesshi Jordan, Matthew Cieslak, Miguel Estevan Moreno, Tomáš Hrnčiar, Valentin Haenel, Yannick Schwartz, Benjamin A. Beasley, Benjamin C Darwin, Bertrand Thirion, Carl Gauthier, Clorindo Notte, Igor Solovey, Ivan Gonzalez, Jath Palasubramaniam, Jon Cluce, Justin Lecher, Katrin Leinweber, Konstantinos Raktivan, Markéta Calábková, Peter Fischer, Philippe Gervais, Rémi Gau, Syam Gadde, Thomas Ballinger, Thomas Roos, Venkateswara Reddy Reddam, and freec84. nipy/nibabel: 5.4.1, March 2026. URL `https://doi.org/10.5281/zenodo.18944560`.

Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman (eds.), *Proceedings of the 9th Python in Science Conference*, Proceedings of the Python in Science Conference, pp. 56–61. SciPy, 2010. doi: 10.25080/Majora-92bf1922-00a.

Melanie Mitchell. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101, 2021.

Alessio Negrini and Simon Reiß. Conquering the retina: Bringing visual in-context learning to oct. In *International Workshop on Efficient Medical Artificial Intelligence*, pp. 21–30. Springer, 2025.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23604–23613, 2025.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Xingyu Qiu, Yuqian Fu, Jiawei Geng, Bin Ren, Jiancheng Pan, Zongwei Wu, Hao Tang, Yanwei Fu, Radu Timofte, Nicu Sebe, et al. The second challenge on cross-domain few-shot object detection at ntire 2026: Methods and results. *arXiv preprint arXiv:2604.11998*, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

John C Raven. Standardization of progressive matrices, 1938. *British Journal of Medical Psychology*, 1941.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International conference on learning representations*, 2017.

Simon Reiß, Zdravko Marinov, Alexander Jaus, Constantin Seibold, M Saquib Sarfraz, Erik Rodner, and Rainer Stiefelhagen. Is visual in-context learning for compositional medical tasks within reach? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2642–2652, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Fereshteh Sadeghi, C Lawrence Zitnick, and Ali Farhadi. Visalogy: Answering visual analogy questions. *Advances in Neural Information Processing Systems*, 28, 2015.

Min-Cheol Sagong, Yoon-Jae Yeo, Seung-Won Jung, and Sung-Jea Ko. Rord: A real-world object removal dataset. In *BMVC*, pp. 542, 2022.

Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

Carlos Schmidt and Simon Reiß. From static to interactive: Adapting visual in-context learners for user-driven tasks. *arXiv preprint arXiv:2604.06748*, 2026.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pp. 746–760. Springer, 2012.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Pratul P Srinivasan, Leo A Kim, Priyatham S Mettu, Scott W Cousins, Grant M Comer, Joseph A Izatt, and Sina Farsiu. Fully automated detection of diabetic macular edema and dry age-related macular degeneration from optical coherence tomography images. *Biomedical optics express*, 5 (10):3568–3577, 2014.

Sebastian Thrun. Is learning the n-th thing any easier than learning the first? *Advances in neural information processing systems*, 8, 1995.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Caren M Walker and Alison Gopnik. Toddlers infer higher-order relational principles in causal learning. *Psychological science*, 25(1):161–169, 2014.

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

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*, 2023b. URL http://papers.nips.cc/paper_files/paper/2023/hash/1b3750390ca8b931fb9ca988647940cb-Abstract-Conference.html.

Taylor Webb, Zachary Dulberg, Steven Frankland, Alexander Petrov, Randall O'Reilly, and Jonathan Cohen. Learning representations that support extrapolation. In *International conference on machine learning*, pp. 10136–10146. PMLR, 2020.

Changliang Xia, Chengyou Jia, Zhuohang Dang, Minnan Luo, Zhihui Li, and Xiaojun Chang. From ideal to real: Unified and data-efficient dense prediction for real-world scenarios. *arXiv preprint arXiv:2506.20279*, 2025.

Jiahao Xie, Alessio Tonioni, Nathalie Rauschmayr, Federico Tombari, and Bernt Schiele. Test-time visual in-context tuning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19996–20005, 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. URL https://openreview.net/forum?id=qBTgnk2HAf.

Eunice Yiu, Maan Qraitem, Anisa Noor Majhi, Charlie Wong, Yutong Bai, Shiry Ginosar, Alison Gopnik, and Kate Saenko. Kiva: Kid-inspired visual analogies for testing large multimodal models. *arXiv preprint arXiv:2407.17773*, 2024.

Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. Ap-10k: A benchmark for animal pose estimation in the wild. *arXiv preprint arXiv:2108.12617*, 2021.

Guanhua Zhang and Moritz Hardt. Inherent trade-offs between diversity and stability in multi-task benchmarks. *arXiv preprint arXiv:2405.01719*, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Yongshuo Zong, Ondrej Bohdal, and Timothy Hospedales. Vl-icl bench: The devil in the details of multimodal in-context learning. *arXiv preprint arXiv:2403.13164*, 2024.

A Technical Appendices and Supplementary Material

A.1 Limitations

Next, we discuss some limitations of our work and give insight into our assessment of them.

While extensive, VIBE contains some datasets which are rather small, particularly in the medical domain. While this is a notoriously data-scarce imaging domain, reflecting real-world settings, additional larger medical datasets would be interesting to explore. Yet, of course, there has to be a limit, which led to the medical domain being represented by two datasets that are on the smaller side.

As we discuss in Section 5, existing metrics, even perceptual metrics like LPIPS not always reflect the task adaptive performance faithfully. This is especially the case for models that can pixel-wise hand-through the query, which leads to good results on some metric-task combinations, even though the model did not really address the task. Similarly, some models can segment images, but their segmentation follows the learned color map from pre-training rather than the colors provided in the context. By reporting results in mIoU with Hungarian matching instead of color-aware mIoU, such models would achieve better segmentation performance. Yet, we argue, that for evaluation of task adaptation, *i.e.*, whether a model predicts contextually, models need to base their predictions on the context and not on their color palette from pre-training.

In Table 4, we aggregate results into rankings. Inherently, these rankings are biased by factors such as metrics used for the tasks, the number of datasets in the different domains, or, the selection of vision tasks in the benchmark. To address these effects, task taxonomies could be explored to attribute the same weight to distinct task categories. As stated in Section 5, we take a more *task instance* based view, considering each context set as defining a own task, which avoids the notion of task categories. Yet, of course, there are highly correlated and less correlated task instances, which affects the ranking.

Our idea in the VIBE toolkit is to use task-encodings which are as simple as possible. The models we benchmark often define such tasks encodings differently, and with their task encodings the results of these models would likely be better. As we want to measure task adaptiveness we take the view that models should be able to adapt to new task encodings, which is the premise of VICL. Yet, this may lead to some models having better aligned task encodings that they were pre-trained on than other models, an advantage for the former. As one of our insights is that most models that are trained on structured inputs tend to collapse to multi-task models, in practice, this effect does not take shape.

Similarly, there are overlaps between training tasks of the VICL models and our tasks in VIBE. This is unavoidable, as there is no standardized set of training and evaluation tasks in literature, and the different works make use of many different publicly available image datasets. Even though, some models may have been pre-trained on datasets we use for evaluation, through the different task encodings, trained models generally still fail to adapt. A particularly noteworthy example is Painter on NYU-Depth V2, a dataset on which it was trained to estimate depth maps color coded in gray. In Figure 7, rather than adapting to the tasks of colorization on NYU-Depth V2 as defined in the context (top row of image grids), Painter resorts to always produce gray depth maps (bottom right quadrant), as the pre-training bias is that strong, hindering adaptation. This is also a hint that a model seems to not necessarily require task outputs in the context to learn a *switch* as we reason in Section 5, it can also be learned solely based on dataset statistics of the input images.

A shortcoming in this work – which we deem to be out of scope given our extensive experiments – is considering inference compute as part of the evaluation metrics. Qualifying the performance metrics with compute cost gives credit to efficient models and incentivizes research into these models. Given the vastly different inference costs of the models this is an important factor to explore in the future.

A.2 Benchmark datasets

In Table 6, we provide an overview of the datasets we use in VIBE, indicating the abbreviated domain name and the tasks for which the datasets provide annotations for. Aside to these tasks, we evaluate datasets where possible on the tasks which do not specifically require annotations, such as denoising, colorization, *etc.*.

Next, we describe dataset-specific filtering and which tasks each dataset is evaluated on in VIBE.



Figure 7: Qualitative examples for Painter on NYU-Depth V2, prompted with the colorization task.

Table 6: Summary of the datasets included in VIBE. Sample counts refer to the benchmark subsets after dataset-specific filtering and preprocessing. For Duke, the two numbers correspond to the cyst segmentation and the layer edge-detection subsets, respectively.

Dataset	Task(s)	Benchmark subset / remarks	Samples
		Medical & Biomedical Imaging	
Duke	Semantic segmentation, edge detection	Filtered scans with fluid cysts/boundaries	78 / 110
IXI	Modality transfer	Preprocessed into 2D slices	578
		Underwater & Aquatic environments	
ATLANTIS	Sem. segmentation	Waterbody segmentation dataset	5,195
SeaThru	Image restoration	Subset with paired depth info; low-light	1,205
SUIM	Semantic segmentation	Underwater segmentation benchmark subset	1,635
		Indoor & Structural Environments	
NYU Depth V2	Depth, sem. seg.	Standard labeled benchmark subset	1449
DeepCrack	Edge detection	Edge targets derived from crack masks	537
		Arts, Aesthetics & Photography	
CDD	Image restoration	Benchmark uses the test split	200
DRAM	Semantic segmentation	Only the labeled subset is used	718
		Object- and Human-Centric Scenes	
OpenThermalPose	Keypoint, detection	Filtered to single-instance images	1,950
RORD	Object removal	Benchmark-specific validation subset	343
LLVIP	Detection, Modality transfer	Filtered to max 3 objects; every 25th frame	608
		Wildlife and Animal-Centric Imagery	
AP-10K	Keypoint/obj. detection	Filtered to remove multi-object images	5,961
CaltechBirds	Keypoint/obj. detection	Randomly filtered to 30 images per class	6,000

ATLANTIS (Erfani et al., 2022) We use the entire dataset as is.

Tasks in VIBE: Semantic segmentation, colorization, denoising, inpainting, rotation, flip.

Caltech Birds (Wah et al., 2011) To have a manageable size of the dataset for benchmarking, we filter the dataset and obtain 30 images for each of the 200 bird classes by random selection.

Tasks in VIBE: Object detection, keypoint detection (based on landmarks on birds), colorization, denoising, inpainting, rotation, flip.

CDD (Guo et al., 2024) Due to the large size of the original dataset and the many different artifact types, we utilize 200 images from the official test split for each of the artifact removal tasks.

Tasks in VIBE: Artifact removal (haze → clear, haze+rain → clear, haze+snow → clear, low+haze+rain → clear, low+haze+snow → clear, low+haze → clear, low+rain → clear, low → clear, rain → clear, snow → clear, low+snow → clear), colorization, denoising, inpainting, rotation, flip.

DeepCrack (Liu et al., 2019) We use the all images in the dataset.

Tasks in VIBE: Edge detection, colorization, denoising, inpainting, rotation, flip.

DRAM (Cohen et al., 2022) We use all images with associated segmentation annotations.

Tasks in VIBE: Semantic segmentation, colorization, denoising, inpainting, rotation, flip.

DUKE For the edge task, we only use images in DUKE, which have annotated retinal layers and render the topmost and lowest layers in our encoding. For the segmentation task, we only use images which have fluid / cyst annotations.

Tasks in VIBE: Edge detection, semantic segmentation, denoising, inpainting, rotation, flip.

IXI (IXI Project) For each patient 3D volume, the most prominent slice is selected by means of highest intensity among pixels in 2D slices. We repeat this process for the volumes from the T2 weighted and PD MRI modalities in the three dimensions, *i.e.*, sagittal, lateral, coronal, yielding the three dataset variants in the main paper.

Tasks in VIBE (for sagittal, lateral, coronal) : Modality transfer (T2 weighted MRI to PD), denoising, inpainting, rotation, flip.

LLVIP (Jia et al., 2021) In this dataset, to obtain a manageable size, we use every 25th frame based on the provided video data. Further, filter out images, where the bounding boxes of more than three persons overlap, in order to simplify the task encoding for multi-object detection, where overlapping boxes are represented in different color channels of the RGB image.

Tasks in VIBE: Object detection, modality transfer (RGB to infrared), colorization, denoising, inpainting, rotation, flip.

NYU Depth V2 (Silberman et al., 2012) We use the whole dataset, for segmentation, use the 40 classes provided.

Tasks in VIBE: Depth estimation, semantic segmentation, colorization, denoising, inpainting, rotation, flip.

OpenThermalPose (Kuzdeuov et al., 2024; 2025) We filter the dataset to only contain images with one human, which eases the process of encoding keypoints based on human poses, as they are always associated to the single human in the scene.

Tasks in VIBE: Keypoint detection (from human poses), denoising, inpainting, rotation, flip.

RORD mini (Sagong et al., 2022) As RORD is a very large dataset, we use the provided RORD mini subset (the val-343 split) to have a manageable size.

Tasks in VIBE: Object removal (based on red bounding boxes), colorization, denoising, inpainting, rotation, flip.

SeaThru (Akkaynak & Treibitz, 2019) We use all images in the dataset with the associated depth maps.

Tasks in VIBE: Depth estimation, colorization, denoising, inpainting, rotation, flip.

SUIM (Islam et al., 2020) We use all images in the dataset with their associated segmentation annotations.

Tasks in VIBE: Semantic segmentation, colorization, denoising, inpainting, rotation, flip.

AP-10K (Yu et al., 2021) As for OpenThermalPose, we filter out images with multiple instances and images with less than 7 keypoints annotated which indicates an occlusion.

Tasks in VIBE: Object detection, keypoint detection (based on animal poses), colorization, denoising, inpainting, rotation, flip.

A.3 BENCHMARK TASKS

Here, we describe selection criteria, encoding and post-processing for each task in the VIBE toolkit.

Segmentation

- Selection criteria: Semantic segmentation is included as a dense pixel-wise classification task in the benchmark. It requires structured understanding of scene content by assigning a discrete semantic class to every pixel. This directly tests whether VICL models can infer region-level semantic mappings from a single context example. It covers a broad range of domains in the benchmark, including artistic paintings (DRAM (Cohen et al., 2022)), underwater and aquatic environment scenes (SUIM (Islam et al., 2020), ATLANTIS (Erfani et al., 2022)), medical imaging (DUKE (Srinivasan et al., 2014)) and indoor scenes (NYU Depth V2 (Silberman et al., 2012)), making it a central task for evaluating cross-domain adaptation.
- Encoding process: Raw integer-valued masks in $H \times W$ are converted into 3-channel image encodings. In our benchmark, we use a color palette that maps each class index to a color. We use the pre-defined color map of the ADE20K dataset (Zhou et al., 2017). Each class index is mapped to a predefined RGB value from this palette. From this color list, we select C visually distinct colors by using a greedy spread sampling strategy. We start from the darkest color, while each subsequent color is chosen to maximize its distance to the already selected colors in RGB space. This resulting mapping is then used to map it to the 3-channel target in our benchmark. See Figure 8 for an example visual analogy.
- Post-processing: The model's prediction is decoded into class indices by assigning each pixel in the prediction to the nearest color from the palette among the colors shown in the context examples presented to the model. The resulting discrete label map is resized with nearest-neighbor interpolation to target resolution and is then compared against the ground truth mask using *mIoU* (this way of evaluating can be referred to as color-aware mIoU (Bar et al., 2022)).

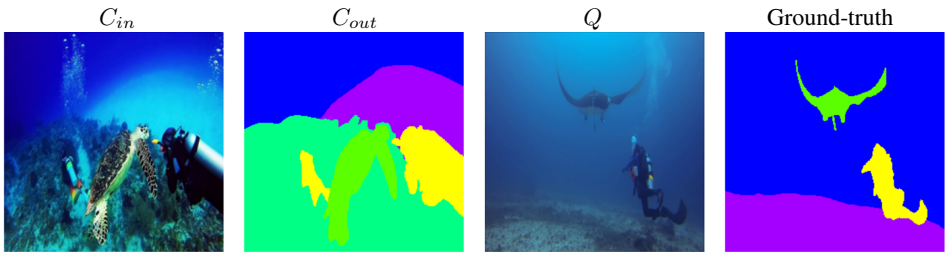


Figure 8: Example visual analogy in the SUIM dataset for semantic segmentation encoding in VIBE.

Depth Estimation

- Selection criteria: Depth estimation is included as a representative continuous dense regression task. Unlike categorical dense tasks, it requires the model to infer a per-pixel quantity from the input image. This task evaluates whether VICL models can adapt not only to discrete classes but to continuous values for spatial prediction. Within the benchmark, it is instantiated on NYU Depth V2 (Silberman et al., 2012), which captures cluttered indoor scenes, and on SeaThru (Akkaynak

& Treibitz, 2019) to introduce underwater and low-light conditions with scattering effects to test robustness under domain-specific visual distortions.

- Encoding process: Raw target depth maps are stored as numeric arrays and dynamically transformed into 3-channel image representations. Continuous depth values are quantized into 256 bins and each of the bins is mapped to an RGB value using a Turbo color map[1]. This converts the scalar depth map into a 3-channel image while preserving a reversible mapping to an approximate depth range. NaN values are mapped directly to the invalid value -1.0. See Figure 9 for an example visual analogy.
- Post-processing: Predicted RGB images are decoded by finding the nearest color for each pixel in the Turbo color map lookup table. The corresponding bin index is then mapped back to a depth value (bin center). The reconstructed depth map is then compared to the depth ground truth.

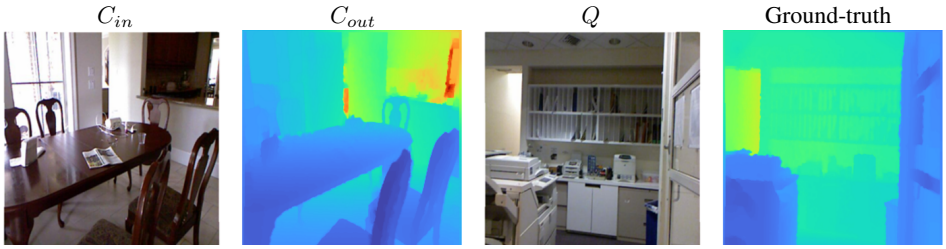


Figure 9: Example visual analogy in the NYU-D V2 dataset for depth estimation encoding in VIBE.

Edge Detection

- Selection criteria: Edge detection is included as a sparse binary dense prediction task that emphasizes fine geometric structures and thin boundaries. It evaluates whether VICL models can achieve high spatial precision. We evaluate edge detection on structurally demanding datasets such as DeepCrack (Liu et al., 2019) and Duke (Srinivasan et al., 2014), to cover the structural and the medical domain.
- Encoding process: Edge annotations are represented as binary image masks with background encoded as 0 and edge pixels as 255. In order to get a 3-channel target image, the grayscale mask is represented as a 3-channel image by duplicating the same edge mask across channels. See Figure 10 for an example visual analogy.
- Post-processing: The predicted edge images are converted into edge maps by using intensity-based thresholding. The decoded edge map is then compared with the ground truth edge annotation.

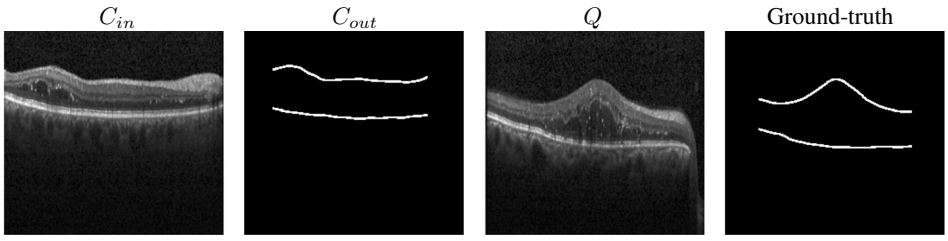


Figure 10: Example visual analogy in the DUKE dataset for edge detection encoding in VIBE.

Artifact Removal

- Selection criteria: Artifact removal is included as a compound degradation restoration task. Unlike denoising, which applies a single controlled noise type, it targets real-world degradations that occur in combination rather than in isolation. The task is instantiated on CDD (Guo et al., 2024), a dataset

[1] https://research.google/blog/turbo-an-improved-rainbow-colormap-for-visualization/

covering 11 degradation types across composite corruption settings. We evaluate heterogeneous VICL models under diverse degradation combinations to assess their robustness and capabilities to remove artifacts.

- Encoding process: The degraded image serves as the model input, while the corresponding clean restored image is retained as the target for evaluation. See Figure 11 for an example visual analogy.
- Post-processing: The predicted output is resized to the spatial resolution of the target and is evaluated using standard image quality metrics PSNR, SSIM and LPIPS.



Figure 11: Example visual analogy in the CDD dataset for artifact removal encoding in VIBE.

Object Removal

- Selection criteria: Object removal is included as a spatially conditioned image generation task that requires the model to remove a designated object while preserving scene structure. It is the most demanding manipulation task in the benchmark, as it requires inferring from the bounding box that the enclosed region corresponds to an object to be removed, suppressing it and synthesizing plausible background content to fill the resulting gap. The task is instantiated on a subset of RORD (Sagong et al., 2022), a large-scale real-world dataset with paired before-and-after images and precise annotations.
- Encoding process: The query is the original RGB image annotated with a red bounding box indicating the object region to be removed, while its corresponding target is the same image with the specific object removed. This preserves the full scene while providing an explicit spatial cue for the removal target. The same construction of the bounding box surrounding the target is applied to C_{in}. See Figure 12 for an example visual analogy.
- Post-processing: The predicted output is resized to the spatial resolution of the target and is evaluated using standard image quality metrics PSNR, SSIM and LPIPS.

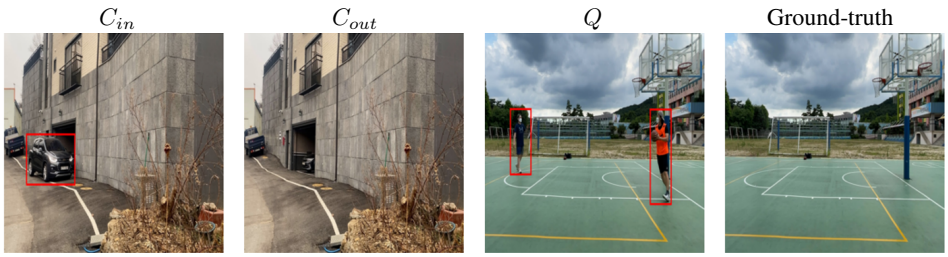


Figure 12: Example visual analogy in the RORD dataset for object removal encoding in VIBE.

Keypoint Detection

- Selection criteria: Keypoint detection is included as a sparse localization task in the benchmark. It requires predicting a small set of precise point locations rather than dense per-pixel labels, which tests a different form of structured spatial inference. It is instantiated across both human-centric (OpenThermalPose (Kuzdeuov et al., 2024; 2025)) and animal-centric (CaltechBirds (Wah et al., 2011), AP-10K (Yu et al., 2021)) datasets, which cover humans and different animals with substantial pose- and species variation. This breadth makes it a strong test case of whether VICL models can adapt to fine-grained localization in different domains from a single context example.

- Encoding process: Keypoint annotations are originally provided as sparse coordinate locations and are encoded as 3-channel target images to be compatible for prompting VICL approaches. Visible keypoints are clipped to the valid image region and are rendered as filled red circles with a fixed radius of 5 pixels on a black background. All keypoints use the same red marker instead of category-specific colors, which keeps the target representation simple and avoids any ambiguities from overlapping visual encodings. The original coordinates are still retained for postprocessing. See Figure 13 for an example visual analogy.
- Post-processing: The predicted keypoint prediction is decoded using a redness score map which is calculated by $S = \text{clip}(R - \max(G, B), 0, 1)$. The score map is thresholded to obtain a binary keypoint mask. For point-level evaluation, connected components are extracted and each component is reduced to the pixel with the highest redness score. Predicted points are then matched to visible ground truth keypoints based on spatial proximity using a threshold $\tau = 2r$, where r is the radius of the rendered ground truth keypoints.

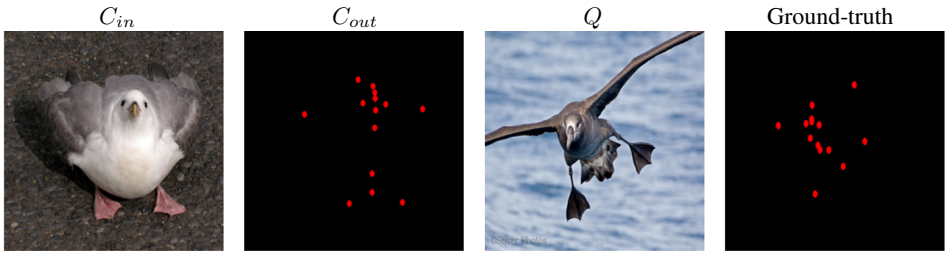


Figure 13: Example visual analogy in the Caltech Birds dataset for keypoint detection encoding in VIBE.

Modality Transfer

- Selection criteria: Modality transfer is included as it focuses on cross-modality image-to-image mappings within a unified framework, requiring VICL models to infer an appearance transformation rule rather than a fixed pattern from a single context pair. The task is instantiated on infrared-to-RGB transfer (LLVIP (Jia et al., 2021)) and T2-weighted to proton density (PD) MRI transfer (IXI (IXI Project)), covering object- and human-centric as well as medical imaging domains.
- Encoding process: Each sample consists of an input-output pair, which is derived from a source and a corresponding target domain. See Figure 14 for an example visual analogy.
- Post-processing: The post-processing stage is restricted to quantitative evaluation of the generated outputs. Model predictions are compared against the corresponding ground-truth target images using PSNR, SSIM and LPIPS for evaluation.

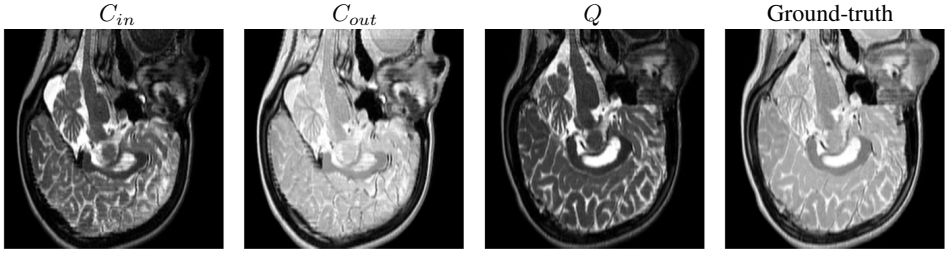


Figure 14: Example visual analogy in the IXI dataset for modality transfer encoding in VIBE.

Object Detection

- Selection criteria: Object detection is included as a structured spatial localization task which requires models to identify and localize a single- or multiple objects. The task is instantiated on LLVIP (Jia et al., 2021), which introduces pedestrian detection under low-light and infrared

conditions, AP-10K (Yu et al., 2021) and CaltechBirds (Wah et al., 2011), which both provide animal-centric annotations.

- Encoding process: Bounding box annotations are rendered into 3-channel image targets. Each object is represented as a bounding box drawn as a filled colored rectangle on black background. To preserve overlapping objects, boxes are assigned to separate RGB channels: red by default, green if they overlap existing red boxes and blue if they overlap both red and green boxes. Samples with more than three overlapping boxes are excluded. See Figure 15 for an example visual analogy.
- Post-processing: Predicted RGB outputs are thresholded separately per channel and connected-component analysis is used to extract object regions. Each connected component is then converted into a bounding box, aligned to the image axis, with using its mean intensity as a confidence score for mAP ranking. Boxes are matched to ground truth using IoU and evaluated with mAP@0.5.

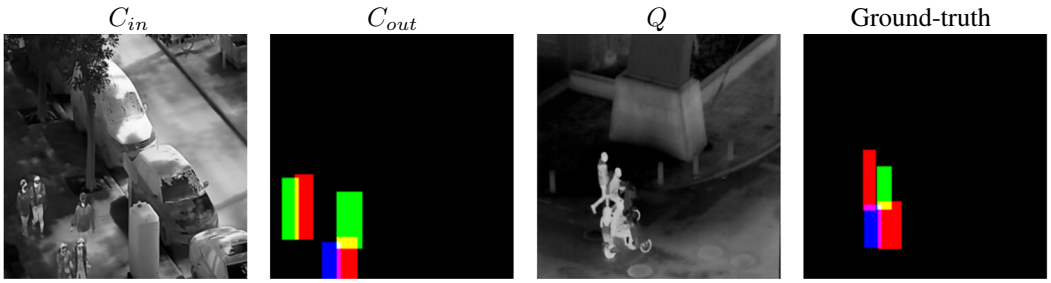


Figure 15: Example visual analogy in the LLVIP dataset for object detection encoding in VIBE.

Inpainting

- Selection criteria: Inpainting is included as the representative region-level reconstruction task. It is used to test whether VICL models can complete missing image regions by inferring plausible content from surrounding visual context and whether they can infer the reconstruction pattern demonstrated in the prompt. The task is applied across all datasets and domains in the benchmark.
- Encoding process: For each sample, a binary mask is generated and applied to the original RGB image. The masked pixels are replaced by a fixed fill value (black, *i.e.*, pixel value 0) while unmasked regions remain unchanged. The resulting masked image serves as the query input, while the original uncorrupted RGB image is retained as the target for evaluation. The same binary mask is applied to C_{in}. See Figure 16 for an example visual analogy.
- Post-processing: The predicted output is resized to the spatial resolution of the target. Evaluation is executed by using the standard image quality metrics PSNR, SSIM and LPIPS.



Figure 16: Example visual analogy in the DRAM dataset for inpainting encoding in VIBE.

Denoising

- Selection criteria: Denoising is included as a pixel-level signal recovery task. Because the corruption is applied using a fixed, deterministically generated noise configuration, it provides a fully controlled and reproducible low-level restoration challenge. It tests whether VICL models can infer the noise-removal objective from a noisy-to-clean context pair and apply it faithfully to an unseen query.

- Encoding process: Additive Gaussian noise with a fixed standard deviation of $\sigma = 0.15$ is applied to the original clean RGB image to produce the noisy model input. The original image is retained as the ground-truth target for evaluation. The same noise construction is also applied to C_{in}. The noise realization for each sample is generated deterministically from the sample index, ensuring reproducibility while producing non-identical corruptions across samples. See Figure 17 for an example visual analogy.
- Post-processing: The predicted output is resized to the spatial resolution of the target and is evaluated using standard image quality metrics PSNR, SSIM and LPIPS.

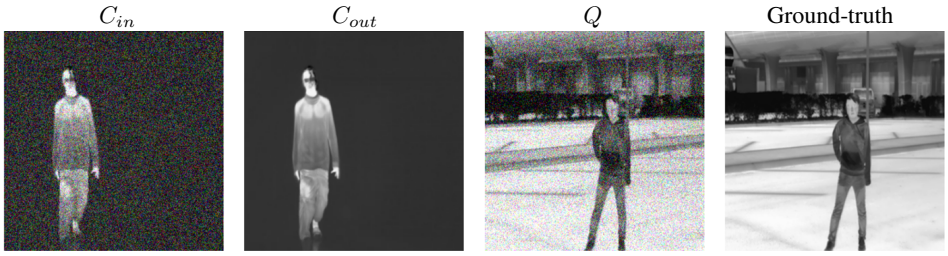


Figure 17: Example visual analogy in the OTP dataset for denoising encoding in VIBE.

Image Colorization

- Selection criteria: Colorization is included as a perceptual inference task within the Image Restoration. Unlike denoising or inpainting, it requires the model to infer plausible color information from a grayscale input. This task evaluates whether VICL models can learn an appearance mapping from a single context example rather than simply recovering a known corruption.
- Encoding process: The original RGB image is converted to a grayscale representation using the standard luminance transform GRAY $= 0.299\,R + 0.587\,G + 0.114\,B$. The resulting single channel is repeated across three channels so that the model receives a 3-channel input. For evaluation, the original RGB image is retained as the target. See Figure 18 for an example visual analogy.
- Post-processing: The predicted output is resized to the spatial resolution of the target and is evaluated using standard image quality metrics PSNR, SSIM and LPIPS.

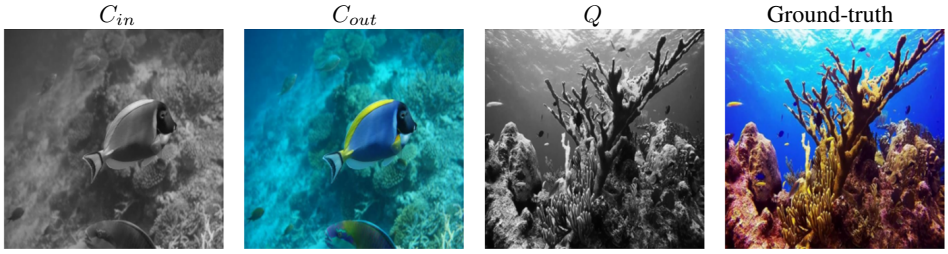


Figure 18: Example visual analogy in the SUIM dataset for image colorization encoding in VIBE.

Geometric Transformations

- Selection criteria: With geometric tasks, we probe, whether VICL models can handle simple transformations like rotating or flipping an image, where the task does not involve altering the image content but geometrically transforming it.
- Encoding process: The original RGB image serves as the model input. The target is obtained by applying the respective geometric transformation directly to that image. We include a $90°$-rotated- and a horizontally flipped version. See Figure 19 for an example visual analogy for rotation.
- Post-processing: The predicted output is resized to the spatial resolution of the ground-truth target. This is followed by using the shared image-to-image protocol using PSNR, SSIM and LPIPS for evaluation.



Figure 19: Example visual analogy in the ATLANTIS dataset for geometric transformations encoding of Rotation with 90° in VIBE.

A.4 Benchmark implementation pipelines

Next, we briefly describe how we have considered the notions of extensibility and reproducibility in VIBE. Implementation details are described below and the full toolkit will be made available upon acceptance. Our benchmark is structured into two stages, the first for dataset download and pre-processing, the second for model inference.

Stage 1: Offline dataset preparation. In the first stage, the datasets are downloaded automatically and further converted into a canonical format and put into a unified folder structure. In this pre-processing, each dataset is stored in a separate folder with subfolders for different annotation types. During this process, each image is assigned an ID. We then deterministically pre-compute for each image ID a list of context samples for later prompt construction and store it into a CSV file. This ensures, reproducible runs with fixed context sets for each query image. We extract a list of context samples to enable future exploration beyond one-shot, towards few-shot prompting experiments. New datasets and tasks can be added to the benchmark by following the steps described in the source code, making extensibility a core principle of the VIBE toolkit.

Stage 2: Online inference and evaluation. In the second stage, the prepared data can be loaded by specifying input and output folders based on the previously described folder structure. Further, a model can be specified, which determines how the context set and query image is provided to the model at inference time. Then, the CSV file for prompt construction is loaded and inference is done for each entry in the file using the specified contexts. For less disk space requirements, some task targets are computed at runtime (*e.g.*, inpainting, denoising, *etc.*). The predictions of the specified model are then decoded and evaluated with a set of task-specific metrics. The results are persistently logged into CSV files. Additionally, it can be specified to log qualitative results during inference. Here, the VIBE toolkit can also be extended towards including new models and prompt strategies.

A small note: In the source code of our VIBE toolkit, we also provide train / validation / test splits (25%/25%/50%), even though in the experiments of the main paper we do not train models, and therefore just evaluate on the test split. We do this with extensibility in mind, *e.g.*, to enable future development of visual in-context learners through hold-tasks-out cross-validation or to allow for development of test-time fine-tuning methods tailored for VICL (Xie et al., 2025).

A.5 Rank aggregation

To compare models across tasks with different metrics and value ranges, we report a rank-aggregated benchmark score in Table 4. We compute this ranking by aggregating performance the level of tasks to avoid averaging different metrics and value ranges, such as mIoU, PSNR, LPIPS or RMSE.

For each task $t \in T$, we first compute the mean score of each model m for every metric k across all datasets $d \in D_t$ that belong to this task: $\bar{\mu}^{(t)}_{m,k} = \frac{1}{|D_t|} \sum_{d \in D_t} \mu_{m,d,k}$. This gives each dataset the same weight within a task, independent of the number of samples it contains. We then rank the models separately for each metric. For error metrics, lower values receive better ranks, while for performance metrics, higher values receive better ranks. This yields metric-specific ranks $r^{(t)}_{m,k}$, where rank 1 corresponds to the best model.

The final rank for a task is obtained by averaging over all metrics K_t used for this task: $\bar{r}_m^{(t)} = \frac{1}{|K_t|} \sum_{k \in K_t} r_{m,k}^{(t)}$. The overall benchmark rank is then computed by averaging these task-level ranks over all tasks:

$$\bar{r}_m = \frac{1}{|T|} \sum_{t \in T} \bar{r}_m^{(t)}.$$

Lower values therefore indicate stronger overall performance across the benchmark.

This rank aggregation avoids comparing raw metric values across incompatible scales and instead measures how consistently a model performs relative to the other methods. It also prevents individual datasets or metrics from dominating the final ranking. Additionally, in Table 4, we show the task-level ranks $\bar{r}_m^{(t)}$ to indicate where each model performs well or poorly before computing the final average shown on the far right. Such rankings on multi-task benchmarks are very useful to get a clearer view on how models compare to each other, especially on such a diverse evaluation as ours, yet, naturally nuance can be lost (Zhang & Hardt, 2024) when viewing them in isolation.

A.6 Compute utilized for all experiments

All computation in this benchmarking effort was carried out on NVIDIA A100 GPUs. Considering all experiments, we utilized computation for a wallclock time of 160.875 days with a total energy consumption of 16,805.07 Megajoule (4,668.08 Kilowatt-hours). The average memory utilized was 8.71 GB although this varies with the models tested and the batch size which was used in inference: MAE-VQGAN (12.05 GB, batch size 20), Painter (6.39 GB, batch size 4), DeLVM (5.20 GB), LVM (15.40 GB), PromptDiffusion (10.23 GB), SD-VICL (4.93 GB), Analogist (8.02 GB). Unlike the other methods, Analogist requires a text prompt extraction step with a vision-language model (Qwen3-VL-8B (Bai et al., 2025) served via vLLM (Kwon et al., 2023)) before diffusion inference. Thus, the reported numbers include both stages of the pipeline. Table 7 summarizes the wall-clock time per model. While this includes re-runs and is mainly an indication of the compute we used to collect the results in this paper, it can be seen, that SD-VICL comes with significantly increased inference time as compared to the other models with more than half the total wall-clock time.

Table 7: Computation statistics for carrying out the experiments as in the paper. These numbers do not indicate the precise efficiency of each model as this includes some computation jobs that failed, or which had to be re-run, rather it is an indication of how much compute was expended per model.

Method	Total Memory (GB)	Total Wallclock time (h:m:s)
MAE-VQGAN	843.49	12 : 06 : 12
Painter	767.20	21 : 36 : 58
DeLVM	1,216.98	258 : 31 : 52
LVM	3,603.21	350 : 41 : 20
PromptDiffusion	2,372.44	995 : 48 : 19
SD-VICL	1,113.55	2,163 : 23 : 17
Analogist	2,309.80	108 : 31 : 30

A.7 Model architecture overview

In Table 8, we compare the architectures of the VICL models that we compare in our experiments.

A.8 Licenses of datasets and used codebases

In Table 9 and Table 10, we provide meta information on datasets and code we utilize in VIBE.

We would like to acknowledge the following python packages in particular, as we make substantial use of them in the VIBE toolkit: Pytorch (Paszke et al., 2019), Numpy (Harris et al., 2020), Pandas (McKinney, 2010), nibabel (Markiewicz et al., 2026), Torchvision (maintainers & contributors, 2016), scikit-learn (Pedregosa et al., 2011), SciPy (Virtanen et al., 2020), OpenCV (Bradski, 2000), matplotlib (Hunter, 2007) and h5py (Collette, 2013).

Table 8: Overview of the compared VICL approaches and their main architectural properties.
* Painter parameter count refers to the reported ViT-L backbone setting.
** MAE-VQGAN parameter count refers to the ViT-L-based inpainting model.
*** Combined parameters of frozen Stable Diffusion and trainable ControlNet.

Model	Backbone	Parameters	Codebook (size)	Resolution
Painter	ViT-L (Dosovitskiy et al., 2021)	307M*	–	448×448
MAE-VQGAN	ViT-MAE (He et al., 2022)	307M**	VQGAN (1024)	224×224
LVM	LLaMA (Touvron et al., 2023)	7B	VQGAN (8192)	256×256
DeLVM	LLaMA (Touvron et al., 2023)	1B	VQGAN (8192)	256×256
PromptDiffusion	Stable Diffusion (Rombach et al., 2022)	1.2B***	VAE-KL	512×512
SD-VICL	Stable Diffusion (Rombach et al., 2022)	943M	VAE-KL	512×512
Analogist	Stable Diffusion (Rombach et al., 2022)	1.07B	VAE-KL	512×512

Table 9: Meta information on the datasets we utilize in VIBE.

Dataset name	License	Link
DUKE	–	https://people.duke.edu/~sf59/Chiu_BOE_2014_dataset.htm
IXI	CC BY-SA 3.0	https://brain-development.org/ixi-dataset/
ATLANTIS	Custom	https://github.com/smhassanerfani/atlantis
SUIM	MIT License	https://irvlab.cs.umn.edu/resources/suim-dataset
Sea-thru	CC BY-NC-SA 4.0	https://www.kaggle.com/datasets/colorlabeilat/seathru-dataset
NYU Depth v2	Citation	https://cs.nyu.edu/~fergus/datasets/nyu_depth_v2.html
DeepCrack	Custom	https://github.com/qinnzou/DeepCrack
DRAM	Citation	https://nadavc220.github.io/ssiap/
CDD	–	https://github.com/gy65896/OneRestore
OTP	MIT License	https://github.com/IS2AI/OpenThermalPose
LLVIP	Custom License	https://github.com/bupt-ai-cz/LLVIP
RORD	MIT License	https://github.com/IIDA-AILab/RORD
Caltech Birds	Citation	https://www.vision.caltech.edu/datasets/cub_200_2011/
AP-10K	CC BY 4.0	https://github.com/AlexTheBad/AP-10K

Table 10: Meta information on the code we utilize in VIBE.

Code	License	Link
MAE-VQGAN	–	https://github.com/amirbar/visual_prompting
Painter	MIT License	https://github.com/baaivision/Painter
DeLVM	MIT License	https://github.com/ggjy/DeLVM/tree/main
LVM	Apache License 2.0	https://github.com/ytongbai/LVM
PromptDiffusion	Apache License 2.0	https://github.com/Zhendong-Wang/Prompt-Diffusion
Analogist	MIT License	https://github.com/edward3862/Analogist
SD-VICL re-impl. based on (Alaluf et al., 2024)	MIT License	github.com/garibida/cross-image-attention

A.9 NUMERICAL RESULTS AS TABLES

In this section, we provide the numerical results of our experiments. The mean and standard deviation were computed along each prediction in the dataset-task setting of a model, *e.g.*, in Table 11, on the ATLANTIS dataset, for all predictions of DeLVM, the mIoU across the classes is calculated and for these values mean and standard deviation are reported.

A.9.1 PERCEPTION & LOCALIZATION

Table 11: Segmentation results by dataset (mIoU ↑). Mean ± standard deviation.

Model	ATLANTIS	DRAM	Duke	NYU-D V2	SUIM
Analogist	0.045 ± 0.068	0.118 ± 0.118	0.016 ± 0.033	0.021 ± 0.022	0.165 ± 0.125
DELVM	0.125 ± 0.133	0.236 ± 0.211	0.000 ± 0.002	0.025 ± 0.028	0.208 ± 0.173
LVM	0.049 ± 0.072	0.247 ± 0.232	0.009 ± 0.032	0.026 ± 0.029	0.156 ± 0.152
MAE-VQGAN	0.100 ± 0.106	0.411 ± 0.161	**0.087 ± 0.132**	0.041 ± 0.033	0.225 ± 0.152
Painter	0.028 ± 0.064	0.171 ± 0.223	0.035 ± 0.058	0.008 ± 0.013	0.171 ± 0.183
PromptDiff.	0.030 ± 0.050	0.097 ± 0.105	0.015 ± 0.023	0.010 ± 0.015	0.119 ± 0.094
SD-VICL	**0.229 ± 0.151**	**0.415 ± 0.177**	0.055 ± 0.108	**0.122 ± 0.069**	**0.292 ± 0.192**
Copy target	0.148 ± 0.101	0.337 ± 0.116	0.039 ± 0.087	0.082 ± 0.041	0.203 ± 0.127

Table 12: Depth estimation results by dataset (RMSE ↓). Mean ± standard deviation.

Model	NYU-D V2	Sea-Thru
Analogist	3.150 ± 1.159	3.167 ± 2.143
DELVM	3.096 ± 0.907	2.872 ± 2.665
LVM	2.727 ± 0.884	4.341 ± 4.311
MAE-VQGAN	1.747 ± 0.767	2.158 ± 1.646
Painter	2.704 ± 0.658	3.583 ± 1.793
PromptDiff.	2.760 ± 0.782	**2.157 ± 2.023**
SD-VICL	**1.478 ± 0.689**	2.283 ± 1.486
Copy target	1.596 ± 0.712	1.637 ± 1.243

Table 13: Edge detection results by dataset (Edge F1 ↑). Mean ± standard deviation.

Model	DeepCrack	Duke
Analogist	0.049 ± 0.046	0.052 ± 0.046
DELVM	0.003 ± 0.028	0.000 ± 0.000
LVM	0.082 ± 0.171	0.008 ± 0.020
MAE-VQGAN	0.289 ± 0.211	0.019 ± 0.029
Painter	0.019 ± 0.116	0.030 ± 0.019
PromptDiff.	0.195 ± 0.141	0.011 ± 0.007
SD-VICL	**0.625 ± 0.225**	**0.538 ± 0.120**
Copy target	0.037 ± 0.052	0.056 ± 0.075

Table 14: Object detection results by dataset (mAP@0.5 ↑). Mean ± standard deviation.

Model	AP-10K	Caltech Birds	LLVIP
Analogist	0.477 ± 0.479	0.249 ± 0.421	0.004 ± 0.036
DELVM	0.244 ± 0.382	0.197 ± 0.367	0.039 ± 0.109
LVM	**0.731 ± 0.438**	0.688 ± 0.455	0.072 ± 0.180
MAE-VQGAN	0.646 ± 0.478	0.348 ± 0.476	0.000 ± 0.000
Painter	0.624 ± 0.448	**0.721 ± 0.435**	0.246 ± 0.314
PromptDiff.	0.089 ± 0.244	0.232 ± 0.407	0.029 ± 0.111
SD-VICL	0.635 ± 0.480	0.523 ± 0.497	**0.290 ± 0.351**
Copy target	0.526 ± 0.499	0.383 ± 0.486	0.012 ± 0.058

Table 15: Keypoint detection results by dataset (F1 ↑). Mean ± standard deviation.

Model	AP-10K	Caltech Birds	OTP
Analogist	0.005 ± 0.010	0.024 ± 0.034	0.033 ± 0.041
DELVM	0.002 ± 0.011	0.007 ± 0.031	0.041 ± 0.059
LVM	0.011 ± 0.023	0.021 ± 0.045	**0.087 ± 0.073**
MAE-VQGAN	0.000 ± 0.006	0.023 ± 0.042	0.026 ± 0.038
Painter	0.007 ± 0.027	0.034 ± 0.062	0.019 ± 0.049
PromptDiff.	0.001 ± 0.007	0.001 ± 0.009	0.016 ± 0.022
SD-VICL	**0.039 ± 0.063**	**0.142 ± 0.088**	0.073 ± 0.116
Copy target	0.004 ± 0.011	0.027 ± 0.035	0.011 ± 0.021

A.9.2 IMAGE RESTORATION AND ENHANCEMENT

Table 16: Artifact removal results by dataset (PSNR ↑). Mean ± standard deviation.

Model	CDD
Analogist	12.325 ± 2.467
DELVM	11.112 ± 4.608
LVM	14.502 ± 2.560
MAE-VQGAN	12.985 ± 2.099
Painter	**15.313 ± 3.663**
PromptDiff.	9.904 ± 1.437
SD-VICL	13.470 ± 2.454
Copy target	9.378 ± 1.623

Table 17: Artifact removal results by dataset (SSIM ↑). Mean ± standard deviation.

Model	CDD
Analogist	0.360 ± 0.155
DELVM	0.327 ± 0.188
LVM	0.386 ± 0.160
MAE-VQGAN	0.358 ± 0.157
Painter	**0.517 ± 0.148**
PromptDiff.	0.272 ± 0.118
SD-VICL	0.398 ± 0.130
Copy target	0.246 ± 0.129

Table 18: Artifact removal results by dataset (LPIPS ↓). Mean ± standard deviation.

Model	CDD
Analogist	0.649 ± 0.095
DELVM	0.676 ± 0.126
LVM	0.654 ± 0.080
MAE-VQGAN	0.721 ± 0.082
Painter	**0.590 ± 0.120**
PromptDiff.	0.842 ± 0.069
SD-VICL	0.611 ± 0.099
Copy target	0.747 ± 0.081

Table 19: Denoising results by dataset (PSNR ↑) . Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	12.660 ± 2.596	12.522 ± 2.402	12.735 ± 2.628	12.935 ± 2.408	16.056 ± 2.989	12.745 ± 2.734	15.006 ± 1.521	12.434 ± 1.687
DELVM	7.908 ± 2.974	7.831 ± 3.481	7.649 ± 3.363	7.012 ± 3.318	10.350 ± 4.249	9.226 ± 4.522	14.669 ± 6.548	13.381 ± 3.335
LVM	18.252 ± 2.858	18.693 ± 3.079	**19.636 ± 3.276**	**17.426 ± 2.790**	**21.984 ± 3.596**	19.006 ± 2.588	21.093 ± 0.895	18.405 ± 0.890
MAE-VQGAN	15.273 ± 1.776	15.337 ± 1.853	15.953 ± 1.984	14.679 ± 1.806	17.284 ± 1.852	15.382 ± 1.723	17.918 ± 1.117	15.837 ± 0.943
Painter	**19.213 ± 3.471**	**19.560 ± 4.061**	18.266 ± 4.707	16.960 ± 4.976	21.269 ± 2.302	**21.154 ± 2.973**	13.368 ± 1.597	13.072 ± 1.264
PromptDiff.	11.725 ± 1.999	10.403 ± 1.672	10.959 ± 2.131	9.857 ± 1.518	11.233 ± 2.936	10.179 ± 1.529	7.818 ± 0.482	8.381 ± 0.412
SD-VICL	14.103 ± 2.650	13.389 ± 2.672	13.385 ± 3.313	14.012 ± 2.537	15.830 ± 3.519	13.128 ± 3.105	**21.384 ± 2.454**	**23.416 ± 2.387**
Copy target	9.509 ± 1.575	9.117 ± 1.627	9.958 ± 2.123	9.378 ± 1.630	14.206 ± 2.747	9.347 ± 1.761	12.301 ± 2.110	13.417 ± 1.120

Table 20: Denoising results by dataset (PSNR ↑). Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	13.492 ± 1.034	11.887 ± 1.029	14.288 ± 1.990	12.019 ± 2.387	15.499 ± 2.533	13.269 ± 3.050	17.016 ± 1.159	12.224 ± 2.329
DELVM	10.315 ± 2.947	13.966 ± 4.203	9.842 ± 2.898	12.405 ± 4.872	14.659 ± 5.876	7.623 ± 3.003	21.427 ± 5.142	9.581 ± 4.219
LVM	18.215 ± 0.694	18.772 ± 0.895	**20.528 ± 2.389**	18.002 ± 2.518	**24.943 ± 2.384**	19.442 ± 3.281	24.013 ± 1.390	**19.896 ± 3.495**
MAE-VQGAN	14.958 ± 0.687	15.730 ± 0.823	17.357 ± 1.375	14.098 ± 1.274	19.487 ± 2.116	15.621 ± 1.741	**27.346 ± 2.979**	16.658 ± 2.152
Painter	16.091 ± 0.742	14.648 ± 0.704	16.900 ± 2.171	**18.157 ± 4.152**	18.048 ± 1.798	**20.097 ± 3.882**	10.275 ± 2.066	19.374 ± 4.685
PromptDiff.	9.118 ± 0.386	10.321 ± 0.247	9.926 ± 1.269	9.267 ± 1.493	13.821 ± 2.185	10.783 ± 1.768	11.295 ± 0.394	10.511 ± 1.643
SD-VICL	**23.851 ± 1.813**	**24.560 ± 1.903**	19.552 ± 3.291	16.266 ± 3.587	19.171 ± 6.509	14.048 ± 2.981	27.050 ± 2.855	13.062 ± 2.510
Copy target	12.199 ± 0.674	13.692 ± 0.825	12.150 ± 2.102	8.930 ± 1.420	13.224 ± 4.048	9.576 ± 1.834	25.947 ± 2.798	10.467 ± 1.964

Table 21: Denoising results by dataset (SSIM ↑) . Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	0.261 ± 0.122	0.338 ± 0.150	0.415 ± 0.175	0.350 ± 0.150	0.273 ± 0.151	0.281 ± 0.104	0.337 ± 0.059	0.443 ± 0.103
DELVM	0.114 ± 0.127	0.167 ± 0.177	0.149 ± 0.207	0.207 ± 0.190	0.222 ± 0.191	0.139 ± 0.125	0.196 ± 0.092	0.331 ± 0.113
LVM	0.324 ± 0.163	0.405 ± 0.187	0.511 ± 0.201	0.374 ± 0.174	0.360 ± 0.212	0.355 ± 0.132	0.396 ± 0.059	0.267 ± 0.054
MAE-VQGAN	0.306 ± 0.135	0.360 ± 0.158	0.410 ± 0.169	0.369 ± 0.157	0.294 ± 0.161	0.292 ± 0.104	0.374 ± 0.069	0.472 ± 0.045
Painter	**0.577 ± 0.131**	**0.627 ± 0.128**	**0.707 ± 0.160**	**0.495 ± 0.153**	**0.659 ± 0.130**	**0.692 ± 0.073**	0.368 ± 0.040	0.375 ± 0.039
PromptDiff.	0.227 ± 0.094	0.246 ± 0.120	0.301 ± 0.146	0.255 ± 0.118	0.205 ± 0.108	0.255 ± 0.090	0.170 ± 0.018	0.026 ± 0.021
SD-VICL	0.360 ± 0.116	0.418 ± 0.131	0.544 ± 0.140	0.399 ± 0.131	0.358 ± 0.142	0.412 ± 0.098	**0.443 ± 0.054**	**0.796 ± 0.044**
Copy target	0.162 ± 0.082	0.200 ± 0.112	0.266 ± 0.136	0.246 ± 0.129	0.191 ± 0.120	0.160 ± 0.064	0.269 ± 0.055	0.429 ± 0.052

Table 22: Denoising results by dataset (SSIM ↑) . Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	0.425 ± 0.069	0.286 ± 0.067	0.528 ± 0.136	0.381 ± 0.102	0.720 ± 0.200	0.311 ± 0.181	**0.672 ± 0.106**	0.355 ± 0.168
DELVM	0.196 ± 0.081	0.275 ± 0.056	0.237 ± 0.173	0.319 ± 0.084	0.531 ± 0.399	0.164 ± 0.185	0.371 ± 0.167	0.212 ± 0.205
LVM	0.384 ± 0.038	0.339 ± 0.042	0.578 ± 0.147	0.484 ± 0.132	0.809 ± 0.136	0.399 ± 0.233	0.542 ± 0.064	0.460 ± 0.207
MAE-VQGAN	0.320 ± 0.045	0.392 ± 0.040	0.582 ± 0.112	0.426 ± 0.090	0.741 ± 0.172	0.375 ± 0.205	0.601 ± 0.121	0.437 ± 0.175
Painter	0.643 ± 0.041	0.504 ± 0.032	0.670 ± 0.097	**0.657 ± 0.129**	**0.828 ± 0.046**	**0.580 ± 0.170**	0.218 ± 0.054	**0.618 ± 0.170**
PromptDiff.	0.020 ± 0.032	0.063 ± 0.013	0.469 ± 0.103	0.389 ± 0.104	0.412 ± 0.097	0.334 ± 0.184	0.215 ± 0.043	0.224 ± 0.095
SD-VICL	**0.860 ± 0.029**	**0.806 ± 0.041**	**0.697 ± 0.122**	0.616 ± 0.131	0.768 ± 0.221	0.389 ± 0.175	0.517 ± 0.110	0.420 ± 0.131
Copy target	0.199 ± 0.042	0.342 ± 0.039	0.427 ± 0.134	0.270 ± 0.068	0.647 ± 0.240	0.221 ± 0.139	0.686 ± 0.088	0.270 ± 0.143

Table 23: Denoising results by dataset (LPIPS ↓) . Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	0.685 ± 0.080	0.639 ± 0.079	0.641 ± 0.119	0.643 ± 0.083	0.615 ± 0.055	0.647 ± 0.082	0.564 ± 0.023	0.412 ± 0.060
DELVM	0.747 ± 0.087	0.749 ± 0.090	0.757 ± 0.100	0.764 ± 0.093	0.741 ± 0.147	0.705 ± 0.105	0.628 ± 0.071	0.480 ± 0.044
LVM	0.614 ± 0.056	0.564 ± 0.069	0.510 ± 0.093	0.631 ± 0.056	0.583 ± 0.064	0.562 ± 0.042	0.593 ± 0.022	0.393 ± 0.036
MAE-VQGAN	0.701 ± 0.054	0.659 ± 0.065	0.615 ± 0.060	0.722 ± 0.074	0.643 ± 0.061	0.647 ± 0.040	0.633 ± 0.022	0.371 ± 0.033
Painter	**0.502 ± 0.095**	**0.475 ± 0.101**	**0.438 ± 0.124**	0.659 ± 0.139	**0.451 ± 0.070**	**0.377 ± 0.067**	0.657 ± 0.030	0.447 ± 0.021
PromptDiff.	0.818 ± 0.055	0.738 ± 0.060	0.790 ± 0.104	0.850 ± 0.069	0.701 ± 0.043	0.803 ± 0.078	0.682 ± 0.024	0.668 ± 0.012
SD-VICL	0.638 ± 0.081	0.589 ± 0.080	0.599 ± 0.124	**0.615 ± 0.098**	0.557 ± 0.077	0.573 ± 0.074	**0.422 ± 0.017**	**0.218 ± 0.029**
Copy target	0.770 ± 0.059	0.740 ± 0.057	0.789 ± 0.084	0.747 ± 0.082	0.635 ± 0.057	0.766 ± 0.057	0.550 ± 0.050	0.370 ± 0.039

Table 24: Denoising results by dataset (LPIPS ↓) . Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	0.395 ± 0.035	0.419 ± 0.040	0.568 ± 0.052	0.589 ± 0.081	0.441 ± 0.112	0.689 ± 0.087	0.622 ± 0.047	0.641 ± 0.110
DELVM	0.619 ± 0.103	0.535 ± 0.079	0.689 ± 0.076	0.610 ± 0.077	0.461 ± 0.208	0.781 ± 0.091	0.678 ± 0.072	0.705 ± 0.120
LVM	0.377 ± 0.034	0.399 ± 0.030	0.592 ± 0.057	0.516 ± 0.058	0.396 ± 0.067	0.603 ± 0.066	0.634 ± 0.043	0.514 ± 0.078
MAE-VQGAN	0.419 ± 0.024	0.414 ± 0.023	0.628 ± 0.048	0.601 ± 0.037	0.422 ± 0.132	0.699 ± 0.075	0.607 ± 0.059	0.594 ± 0.077
Painter	0.263 ± 0.023	0.358 ± 0.020	0.542 ± 0.037	0.448 ± 0.076	0.381 ± 0.040	**0.534 ± 0.086**	0.700 ± 0.049	**0.441 ± 0.093**
PromptDiff.	0.677 ± 0.017	0.739 ± 0.016	0.621 ± 0.035	0.760 ± 0.066	0.761 ± 0.052	0.750 ± 0.090	0.817 ± 0.025	0.793 ± 0.062
SD-VICL	**0.165 ± 0.025**	**0.221 ± 0.027**	**0.429 ± 0.046**	**0.445 ± 0.066**	**0.351 ± 0.165**	0.651 ± 0.096	**0.546 ± 0.045**	0.592 ± 0.108
Copy target	0.416 ± 0.038	0.372 ± 0.032	0.615 ± 0.093	0.676 ± 0.043	0.441 ± 0.194	0.773 ± 0.071	0.551 ± 0.032	0.722 ± 0.093

Table 25: Inpainting results by dataset (PSNR ↑) . Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	13.994 ± 2.544	14.269 ± 2.510	13.879 ± 2.863	14.257 ± 2.845	16.798 ± 3.433	14.625 ± 2.734	15.467 ± 1.572	13.633 ± 1.103
DELVM	15.336 ± 3.290	14.630 ± 3.379	14.708 ± 4.140	12.364 ± 3.590	15.617 ± 2.047	16.814 ± 2.411	20.654 ± 2.885	16.690 ± 2.034
LVM	**18.930 ± 2.882**	**19.428 ± 3.182**	**20.490 ± 3.060**	**18.099 ± 2.871**	**22.372 ± 3.947**	**19.643 ± 2.300**	**23.144 ± 1.167**	18.980 ± 1.290
MAE-VQGAN	15.184 ± 1.966	15.465 ± 2.176	15.770 ± 2.217	14.634 ± 1.976	17.218 ± 2.662	15.033 ± 1.916	17.957 ± 1.158	15.771 ± 1.097
Painter	11.883 ± 5.117	12.164 ± 4.561	10.004 ± 4.631	10.505 ± 3.222	16.932 ± 1.609	18.359 ± 2.921	14.355 ± 1.327	14.048 ± 1.150
PromptDiff.	11.442 ± 1.802	10.197 ± 1.680	10.553 ± 2.041	9.907 ± 1.466	12.147 ± 3.218	9.754 ± 1.393	7.728 ± 0.492	7.498 ± 0.371
SD-VICL	13.059 ± 2.277	12.478 ± 2.363	12.549 ± 3.006	13.224 ± 2.312	14.622 ± 3.109	12.419 ± 2.795	20.721 ± 1.970	**19.919 ± 1.696**
Copy target	9.509 ± 1.575	9.117 ± 1.627	9.958 ± 2.123	9.378 ± 1.630	14.206 ± 2.747	9.347 ± 1.761	12.301 ± 2.110	13.417 ± 1.120

Table 26: Inpainting results by dataset (PSNR ↑) . Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	13.683 ± 0.790	12.596 ± 0.898	15.669 ± 1.949	14.375 ± 2.631	16.525 ± 1.861	14.925 ± 2.874	16.901 ± 0.876	13.652 ± 2.899
DELVM	16.122 ± 2.117	17.646 ± 1.791	18.958 ± 2.505	17.275 ± 2.649	18.955 ± 3.445	15.357 ± 3.064	27.992 ± 4.916	16.748 ± 3.404
LVM	18.716 ± 1.061	19.937 ± 0.988	**21.292 ± 2.868**	**19.785 ± 2.579**	**23.601 ± 4.427**	**20.781 ± 4.259**	**31.000 ± 2.564**	**21.547 ± 4.429**
MAE-VQGAN	14.628 ± 0.714	15.611 ± 0.903	16.962 ± 1.805	13.752 ± 1.386	17.439 ± 2.307	15.494 ± 2.147	28.766 ± 3.018	16.878 ± 2.665
Painter	17.278 ± 1.516	15.836 ± 1.293	14.509 ± 2.673	9.141 ± 1.559	14.566 ± 4.141	14.827 ± 3.937	9.161 ± 2.947	13.745 ± 5.656
PromptDiff.	8.421 ± 0.368	9.414 ± 0.241	9.538 ± 1.291	9.428 ± 1.561	12.682 ± 1.881	10.701 ± 1.645	10.022 ± 0.662	9.986 ± 1.520
SD-VICL	**20.208 ± 1.722**	**21.270 ± 1.743**	18.193 ± 2.769	15.068 ± 3.061	17.279 ± 5.678	12.948 ± 2.643	27.177 ± 3.039	12.363 ± 2.304
Copy target	12.199 ± 0.674	13.692 ± 0.825	12.150 ± 2.102	8.930 ± 1.420	13.224 ± 4.048	9.576 ± 1.834	25.947 ± 2.798	10.467 ± 1.964

Table 27: Inpainting results by dataset (SSIM ↑) . Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	0.333 ± 0.156	0.410 ± 0.164	0.484 ± 0.184	0.385 ± 0.161	0.337 ± 0.199	0.381 ± 0.136	0.358 ± 0.070	0.556 ± 0.082
DELVM	0.377 ± 0.172	0.438 ± 0.182	0.528 ± 0.231	0.372 ± 0.183	0.353 ± 0.192	0.433 ± 0.123	0.401 ± 0.067	0.579 ± 0.089
LVM	0.396 ± 0.169	**0.466 ± 0.178**	**0.607 ± 0.182**	**0.417 ± 0.163**	0.368 ± 0.205	0.430 ± 0.128	0.423 ± 0.066	0.586 ± 0.067
MAE-VQGAN	0.328 ± 0.154	0.387 ± 0.173	0.469 ± 0.196	0.379 ± 0.161	0.325 ± 0.196	0.321 ± 0.122	0.381 ± 0.067	0.462 ± 0.044
Painter	**0.399 ± 0.301**	0.459 ± 0.272	0.336 ± 0.328	0.380 ± 0.196	**0.797 ± 0.118**	**0.836 ± 0.083**	**0.516 ± 0.051**	0.413 ± 0.079
PromptDiff.	0.266 ± 0.120	0.282 ± 0.125	0.346 ± 0.139	0.289 ± 0.116	0.236 ± 0.139	0.279 ± 0.095	0.187 ± 0.028	0.042 ± 0.019
SD-VICL	0.360 ± 0.109	0.420 ± 0.120	0.532 ± 0.125	0.405 ± 0.127	0.379 ± 0.130	0.421 ± 0.097	0.476 ± 0.050	**0.808 ± 0.039**
Copy target	0.162 ± 0.082	0.200 ± 0.112	0.266 ± 0.136	0.246 ± 0.129	0.191 ± 0.120	0.160 ± 0.064	0.269 ± 0.055	0.429 ± 0.052

Table 28: Inpainting results by dataset (SSIM ↑) . Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	0.455 ± 0.079	0.337 ± 0.084	0.620 ± 0.142	0.501 ± 0.120	0.786 ± 0.152	0.416 ± 0.216	0.684 ± 0.134	0.426 ± 0.193
DELVM	0.552 ± 0.134	0.608 ± 0.089	0.660 ± 0.136	0.604 ± 0.085	0.813 ± 0.135	0.454 ± 0.235	0.631 ± 0.187	0.518 ± 0.197
LVM	0.591 ± 0.048	0.623 ± 0.045	0.684 ± 0.140	0.604 ± 0.099	**0.850 ± 0.122**	0.483 ± 0.239	**0.741 ± 0.077**	**0.546 ± 0.203**
MAE-VQGAN	0.313 ± 0.044	0.393 ± 0.035	0.594 ± 0.133	0.446 ± 0.097	0.738 ± 0.168	0.396 ± 0.220	0.705 ± 0.095	0.476 ± 0.202
Painter	0.696 ± 0.056	0.550 ± 0.058	**0.715 ± 0.125**	0.382 ± 0.084	0.689 ± 0.262	**0.582 ± 0.233**	0.212 ± 0.070	0.511 ± 0.348
PromptDiff.	0.061 ± 0.015	0.067 ± 0.013	0.510 ± 0.104	0.401 ± 0.104	0.333 ± 0.075	0.323 ± 0.165	0.210 ± 0.040	0.306 ± 0.119
SD-VICL	**0.845 ± 0.028**	**0.820 ± 0.037**	0.699 ± 0.114	**0.618 ± 0.127**	0.743 ± 0.213	0.390 ± 0.159	0.536 ± 0.113	0.426 ± 0.116
Copy target	0.199 ± 0.042	0.342 ± 0.039	0.427 ± 0.134	0.270 ± 0.068	0.647 ± 0.240	0.221 ± 0.139	0.686 ± 0.088	0.270 ± 0.143

Table 29: Inpainting results by dataset (LPIPS ↓) . Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	0.595 ± 0.098	0.526 ± 0.094	0.544 ± 0.143	**0.560 ± 0.084**	0.575 ± 0.064	0.520 ± 0.101	0.539 ± 0.027	0.342 ± 0.045
DELVM	0.560 ± 0.087	0.490 ± 0.115	0.470 ± 0.160	0.588 ± 0.125	0.503 ± 0.074	0.468 ± 0.049	0.526 ± 0.031	0.276 ± 0.069
LVM	**0.543 ± 0.076**	**0.451 ± 0.088**	**0.387 ± 0.087**	0.578 ± 0.073	0.480 ± 0.057	0.467 ± 0.048	0.515 ± 0.024	0.245 ± 0.027
MAE-VQGAN	0.693 ± 0.063	0.644 ± 0.079	0.585 ± 0.080	0.709 ± 0.078	0.633 ± 0.062	0.643 ± 0.047	0.625 ± 0.022	0.371 ± 0.033
Painter	0.588 ± 0.213	0.541 ± 0.190	0.621 ± 0.208	0.756 ± 0.137	**0.284 ± 0.099**	**0.213 ± 0.082**	0.430 ± 0.020	0.248 ± 0.031
PromptDiff.	0.840 ± 0.069	0.739 ± 0.072	0.772 ± 0.107	0.844 ± 0.078	0.719 ± 0.057	0.825 ± 0.076	0.703 ± 0.028	0.655 ± 0.034
SD-VICL	0.601 ± 0.081	0.544 ± 0.080	0.562 ± 0.121	0.590 ± 0.095	0.541 ± 0.080	0.528 ± 0.074	**0.399 ± 0.023**	**0.194 ± 0.023**
Copy target	0.770 ± 0.059	0.740 ± 0.057	0.789 ± 0.084	0.747 ± 0.082	0.635 ± 0.057	0.766 ± 0.057	0.550 ± 0.050	0.370 ± 0.039

Table 30: Inpainting results by dataset (LPIPS ↓) . Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	0.375 ± 0.041	0.394 ± 0.046	0.481 ± 0.066	0.444 ± 0.105	0.369 ± 0.076	0.566 ± 0.103	0.589 ± 0.057	0.559 ± 0.133
DELVM	0.300 ± 0.082	0.293 ± 0.058	0.455 ± 0.058	**0.380 ± 0.039**	0.290 ± 0.084	0.538 ± 0.117	0.587 ± 0.067	0.443 ± 0.100
LVM	0.257 ± 0.024	0.258 ± 0.020	0.479 ± 0.060	0.385 ± 0.053	**0.261 ± 0.086**	0.511 ± 0.089	0.563 ± 0.054	**0.420 ± 0.097**
MAE-VQGAN	0.411 ± 0.024	0.410 ± 0.024	0.595 ± 0.060	0.589 ± 0.043	0.387 ± 0.121	0.679 ± 0.093	0.614 ± 0.068	0.566 ± 0.105
Painter	**0.150 ± 0.022**	0.197 ± 0.018	**0.327 ± 0.062**	0.733 ± 0.066	0.313 ± 0.171	**0.483 ± 0.151**	0.521 ± 0.066	0.457 ± 0.234
PromptDiff.	0.701 ± 0.022	0.746 ± 0.032	0.600 ± 0.041	0.711 ± 0.073	0.784 ± 0.041	0.742 ± 0.095	0.831 ± 0.048	0.809 ± 0.065
SD-VICL	0.157 ± 0.024	**0.186 ± 0.026**	0.350 ± 0.051	0.401 ± 0.062	0.328 ± 0.142	0.609 ± 0.096	**0.470 ± 0.060**	0.545 ± 0.109
Copy target	0.416 ± 0.038	0.372 ± 0.032	0.615 ± 0.093	0.676 ± 0.043	0.441 ± 0.194	0.773 ± 0.071	0.551 ± 0.032	0.722 ± 0.093

Table 31: Colorization results by dataset (PSNR ↑) . Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	LLVIP	NYU-D V2
Analogist	14.160 ± 2.561	13.423 ± 2.568	14.115 ± 2.700	12.882 ± 2.121	16.330 ± 4.042	14.297 ± 2.287	14.080 ± 2.517	14.557 ± 2.289
DELVM	17.659 ± 3.672	17.039 ± 4.410	15.268 ± 5.966	14.450 ± 4.369	19.847 ± 6.660	17.669 ± 2.567	19.026 ± 4.037	**19.684 ± 2.208**
LVM	17.682 ± 2.546	17.903 ± 2.812	**18.218 ± 2.931**	**16.832 ± 2.625**	20.312 ± 3.902	17.372 ± 2.409	**19.145 ± 3.908**	18.172 ± 2.402
MAE-VQGAN	14.814 ± 1.810	14.708 ± 1.837	14.979 ± 2.031	13.758 ± 1.674	17.746 ± 2.143	14.383 ± 1.676	17.746 ± 3.823	13.859 ± 1.257
Painter	**18.510 ± 5.418**	**18.580 ± 5.222**	14.678 ± 6.234	15.510 ± 3.984	**28.247 ± 4.721**	**19.681 ± 3.377**	12.515 ± 2.510	10.059 ± 2.503
PromptDiff.	11.646 ± 1.882	10.664 ± 1.839	10.776 ± 2.139	10.045 ± 1.504	11.836 ± 3.280	10.060 ± 1.496	8.023 ± 1.332	9.658 ± 1.620
SD-VICL	13.409 ± 2.422	12.780 ± 2.495	12.690 ± 3.080	13.449 ± 2.447	15.661 ± 3.451	12.510 ± 2.912	15.797 ± 4.405	15.547 ± 3.240
Copy target	9.509 ± 1.575	9.117 ± 1.627	9.958 ± 2.123	9.378 ± 1.630	14.206 ± 2.747	9.347 ± 1.761	13.397 ± 4.185	8.930 ± 1.420

Table 32: Colorization results by dataset (PSNR ↑) . Mean ± standard deviation.

Model	RORD	Sea-Thru	SUIM
Analogist	14.610 ± 3.658	16.588 ± 1.266	12.173 ± 2.442
DELVM	18.272 ± 4.866	25.669 ± 4.102	14.130 ± 3.608
LVM	18.189 ± 3.184	26.776 ± 3.091	**16.112 ± 3.028**
MAE-VQGAN	15.060 ± 1.937	**28.418 ± 2.745**	14.078 ± 2.164
Painter	**19.363 ± 4.847**	9.873 ± 2.469	14.509 ± 4.537
PromptDiff.	10.946 ± 1.703	10.876 ± 0.500	10.227 ± 1.528
SD-VICL	13.293 ± 2.820	27.495 ± 3.116	12.369 ± 2.382
Copy target	9.576 ± 1.834	25.947 ± 2.798	10.467 ± 1.964

Table 33: Colorization results by dataset (SSIM ↑) . Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	LLVIP	NYU-D V2
Analogist	0.350 ± 0.158	0.407 ± 0.164	0.535 ± 0.167	0.383 ± 0.157	0.337 ± 0.196	0.378 ± 0.123	0.459 ± 0.178	0.512 ± 0.106
DELVM	0.395 ± 0.178	0.450 ± 0.190	0.490 ± 0.269	0.383 ± 0.185	0.339 ± 0.213	0.449 ± 0.138	0.510 ± 0.200	0.633 ± 0.094
LVM	0.382 ± 0.163	0.453 ± 0.175	0.583 ± 0.179	0.409 ± 0.167	0.362 ± 0.201	0.410 ± 0.126	**0.522 ± 0.181**	0.578 ± 0.100
MAE-VQGAN	0.322 ± 0.149	0.376 ± 0.166	0.463 ± 0.190	0.372 ± 0.159	0.329 ± 0.195	0.308 ± 0.118	0.498 ± 0.170	0.444 ± 0.096
Painter	**0.662 ± 0.217**	**0.703 ± 0.189**	**0.598 ± 0.319**	**0.546 ± 0.171**	**0.855 ± 0.113**	**0.840 ± 0.078**	0.411 ± 0.174	0.451 ± 0.120
PromptDiff.	0.262 ± 0.116	0.259 ± 0.101	0.338 ± 0.144	0.265 ± 0.103	0.212 ± 0.124	0.277 ± 0.093	0.237 ± 0.074	0.405 ± 0.107
SD-VICL	0.385 ± 0.114	0.445 ± 0.123	0.562 ± 0.128	0.421 ± 0.124	0.450 ± 0.152	0.449 ± 0.102	0.399 ± 0.173	**0.635 ± 0.132**
Copy target	0.162 ± 0.082	0.200 ± 0.112	0.266 ± 0.136	0.246 ± 0.129	0.191 ± 0.120	0.160 ± 0.064	0.348 ± 0.181	0.270 ± 0.068

Table 34: Colorization results by dataset (SSIM ↑). Mean ± standard deviation.

Model	RORD	Sea-Thru	SUIM
Analogist	0.423 ± 0.229	0.650 ± 0.134	0.407 ± 0.177
DELVM	0.474 ± 0.248	0.532 ± 0.192	0.454 ± 0.196
LVM	0.463 ± 0.233	0.630 ± 0.087	0.473 ± 0.179
MAE-VQGAN	0.392 ± 0.217	**0.698 ± 0.084**	0.419 ± 0.178
Painter	**0.671 ± 0.190**	0.238 ± 0.076	**0.657 ± 0.264**
PromptDiff.	0.317 ± 0.163	0.217 ± 0.040	0.289 ± 0.117
SD-VICL	0.412 ± 0.165	0.542 ± 0.114	0.447 ± 0.121
Copy target	0.221 ± 0.139	0.686 ± 0.088	0.270 ± 0.143

Table 35: Colorization results by dataset (LPIPS ↓) . Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	LLVIP	NYU-D V2
Analogist	0.662 ± 0.104	0.594 ± 0.096	0.607 ± 0.131	0.670 ± 0.094	0.565 ± 0.081	0.614 ± 0.095	0.631 ± 0.092	0.481 ± 0.083
DELVM	0.655 ± 0.109	0.567 ± 0.119	0.629 ± 0.170	0.685 ± 0.107	0.539 ± 0.131	0.585 ± 0.080	0.654 ± 0.084	0.461 ± 0.070
LVM	0.627 ± 0.088	0.529 ± 0.096	**0.517 ± 0.122**	0.643 ± 0.084	0.550 ± 0.103	0.599 ± 0.077	0.655 ± 0.088	0.512 ± 0.082
MAE-VQGAN	0.745 ± 0.072	0.696 ± 0.073	0.686 ± 0.100	0.772 ± 0.089	0.639 ± 0.062	0.709 ± 0.061	0.690 ± 0.096	0.634 ± 0.054
Painter	**0.533 ± 0.157**	**0.451 ± 0.145**	0.587 ± 0.191	0.691 ± 0.142	**0.276 ± 0.093**	**0.388 ± 0.098**	0.598 ± 0.092	0.656 ± 0.096
PromptDiff.	0.826 ± 0.074	0.791 ± 0.061	0.760 ± 0.111	0.819 ± 0.076	0.691 ± 0.049	0.783 ± 0.074	0.816 ± 0.074	0.715 ± 0.078
SD-VICL	0.627 ± 0.083	0.553 ± 0.083	0.582 ± 0.119	**0.609 ± 0.099**	0.492 ± 0.077	0.565 ± 0.078	**0.518 ± 0.095**	**0.418 ± 0.067**
Copy target	0.770 ± 0.059	0.740 ± 0.057	0.789 ± 0.084	0.747 ± 0.082	0.635 ± 0.057	0.766 ± 0.057	0.684 ± 0.115	0.676 ± 0.043

Table 36: Colorization results by dataset (LPIPS ↓) . Mean ± standard deviation.

Model	RORD	Sea-Thru	SUIM
Analogist	0.624 ± 0.117	0.612 ± 0.074	0.642 ± 0.125
DELVM	0.608 ± 0.127	0.687 ± 0.088	0.682 ± 0.097
LVM	0.616 ± 0.091	0.706 ± 0.108	0.568 ± 0.133
MAE-VQGAN	0.729 ± 0.095	0.618 ± 0.069	0.679 ± 0.106
Painter	**0.502 ± 0.124**	0.664 ± 0.081	**0.544 ± 0.134**
PromptDiff.	0.722 ± 0.091	0.804 ± 0.037	0.785 ± 0.064
SD-VICL	0.622 ± 0.102	**0.459 ± 0.061**	0.551 ± 0.113
Copy target	0.773 ± 0.071	0.551 ± 0.032	0.722 ± 0.093

A.9.3 IMAGE MANIPULATION & TRANSFORMATION

Table 37: Style Transfer results by dataset (PSNR ↑). Mean ± standard deviation.

Model	IXI (Axial)	IXI (Coronal)	IXI (Sagittal)	LLVIP
Analogist	12.349 ± 1.281	12.393 ± 1.082	11.420 ± 0.970	10.598 ± 2.093
DELVM	13.425 ± 2.515	11.043 ± 2.033	11.937 ± 2.112	10.389 ± 2.147
LVM	14.060 ± 2.079	11.741 ± 1.559	12.648 ± 1.638	10.817 ± 2.020
MAE-VQGAN	14.897 ± 1.164	13.577 ± 0.696	14.598 ± 0.875	10.177 ± 2.299
Painter	19.791 ± 1.552	14.008 ± 1.676	16.153 ± 1.428	11.518 ± 1.105
PromptDiff.	8.268 ± 0.471	9.869 ± 0.385	9.175 ± 0.248	9.828 ± 1.146
SD-VICL	**21.038 ± 2.316**	**19.262 ± 2.033**	**20.495 ± 1.941**	**11.976 ± 2.301**
Copy target	12.446 ± 1.188	11.010 ± 0.725	12.721 ± 0.975	12.150 ± 2.102

Table 38: Style Transfer results by dataset (SSIM ↑). Mean ± standard deviation.

Model	IXI (Axial)	IXI (Coronal)	IXI (Sagittal)	LLVIP
Analogist	0.513 ± 0.073	0.403 ± 0.057	0.296 ± 0.055	0.352 ± 0.177
DELVM	0.511 ± 0.101	0.386 ± 0.136	0.476 ± 0.119	0.298 ± 0.162
LVM	0.473 ± 0.099	0.401 ± 0.105	0.481 ± 0.085	0.352 ± 0.124
MAE-VQGAN	0.485 ± 0.053	0.332 ± 0.045	0.411 ± 0.038	0.274 ± 0.164
Painter	0.531 ± 0.080	0.617 ± 0.028	0.580 ± 0.027	0.401 ± 0.102
PromptDiff.	0.058 ± 0.025	0.098 ± 0.025	0.071 ± 0.011	**0.483 ± 0.103**
SD-VICL	**0.792 ± 0.053**	**0.801 ± 0.048**	**0.778 ± 0.048**	0.425 ± 0.127
Copy target	0.436 ± 0.054	0.207 ± 0.043	0.345 ± 0.043	0.427 ± 0.134

Table 39: Style Transfer results by dataset (LPIPS ↓). Mean ± standard deviation.

Model	IXI (Axial)	IXI (Coronal)	IXI (Sagittal)	LLVIP
Analogist	0.373 ± 0.052	0.404 ± 0.037	0.407 ± 0.038	0.636 ± 0.067
DELVM	0.293 ± 0.079	0.350 ± 0.119	0.336 ± 0.103	0.728 ± 0.081
LVM	0.303 ± 0.051	0.321 ± 0.043	0.317 ± 0.047	0.737 ± 0.089
MAE-VQGAN	0.354 ± 0.034	0.403 ± 0.026	0.400 ± 0.023	0.692 ± 0.048
Painter	0.237 ± 0.029	0.193 ± 0.020	**0.211 ± 0.015**	0.801 ± 0.089
PromptDiff.	0.661 ± 0.028	0.689 ± 0.028	0.743 ± 0.020	0.604 ± 0.036
SD-VICL	**0.200 ± 0.029**	**0.185 ± 0.031**	0.225 ± 0.030	**0.585 ± 0.059**
Copy target	0.360 ± 0.042	0.418 ± 0.040	0.367 ± 0.033	0.615 ± 0.093

Table 52: Object Removal results by dataset (PSNR ↑). Mean ± standard deviation.

Model	RORD
Analogist	13.472 ± 3.002
DELVM	14.625 ± 3.988
LVM	**15.838 ± 2.503**
MAE-VQGAN	14.230 ± 2.058
Painter	13.506 ± 4.755
PromptDiff.	10.675 ± 1.664
SD-VICL	12.187 ± 2.499
Copy target	9.576 ± 1.834

Table 53: Object Removal results by dataset (SSIM ↑). Mean ± standard deviation.

Model	RORD
Analogist	0.377 ± 0.209
DELVM	0.406 ± 0.230
LVM	0.423 ± 0.220
MAE-VQGAN	0.376 ± 0.211
Painter	**0.487 ± 0.246**
PromptDiff.	0.336 ± 0.179
SD-VICL	0.357 ± 0.158
Copy target	0.221 ± 0.139

Table 54: Object Removal results by dataset (LPIPS ↓). Mean ± standard deviation.

Model	RORD
Analogist	0.629 ± 0.110
DELVM	0.598 ± 0.126
LVM	**0.574 ± 0.087**
MAE-VQGAN	0.701 ± 0.089
Painter	0.578 ± 0.159
PromptDiff.	0.748 ± 0.092
SD-VICL	0.643 ± 0.098
Copy target	0.773 ± 0.071

Table 40: Flip results by dataset (PSNR ↑). Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	10.910 ± 2.733	11.816 ± 3.001	12.116 ± 3.170	11.486 ± 2.880	16.169 ± 3.713	11.093 ± 2.816	**15.210 ± 1.874**	12.466 ± 1.137
DELVM	10.780 ± 2.930	10.546 ± 3.216	9.614 ± 3.580	8.944 ± 2.997	15.612 ± 4.431	12.384 ± 2.303	14.351 ± 2.836	14.914 ± 2.037
LVM	10.287 ± 2.550	12.241 ± 2.896	11.314 ± 2.938	11.206 ± 2.659	16.396 ± 4.171	12.209 ± 2.316	13.305 ± 3.090	14.724 ± 1.553
MAE-VQGAN	**12.886 ± 2.094**	**13.552 ± 2.403**	**13.575 ± 2.505**	**12.738 ± 2.010**	15.930 ± 2.815	**13.244 ± 2.041**	14.889 ± 1.663	15.159 ± 1.282
Painter	10.840 ± 2.778	11.415 ± 3.142	10.431 ± 3.598	10.625 ± 2.693	**17.533 ± 3.579**	12.230 ± 2.141	13.512 ± 1.742	13.832 ± 1.209
PromptDiff.	11.468 ± 1.794	10.434 ± 1.525	10.875 ± 1.966	10.105 ± 1.308	11.834 ± 3.253	9.859 ± 1.387	7.981 ± 0.462	7.550 ± 0.396
SD-VICL	9.184 ± 1.528	9.231 ± 1.729	9.905 ± 2.254	9.421 ± 1.601	12.126 ± 2.372	8.975 ± 1.744	13.985 ± 2.365	**16.257 ± 1.436**
Copy target	9.509 ± 1.575	9.117 ± 1.627	9.958 ± 2.123	9.378 ± 1.630	14.206 ± 2.747	9.347 ± 1.761	12.301 ± 2.110	13.417 ± 1.120

Table 41: Flip results by dataset (PSNR ↑) . Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	12.713 ± 0.857	11.778 ± 0.975	13.564 ± 2.532	10.629 ± 2.490	14.314 ± 2.298	11.741 ± 3.266	16.707 ± 1.186	11.759 ± 2.893
DELVM	11.157 ± 1.454	14.748 ± 2.447	11.964 ± 2.214	10.497 ± 2.397	15.189 ± 2.298	9.889 ± 2.878	20.055 ± 6.667	12.416 ± 3.870
LVM	11.751 ± 0.940	15.061 ± 1.841	12.272 ± 2.040	9.598 ± 2.405	14.782 ± 1.987	11.731 ± 3.028	23.867 ± 4.376	12.249 ± 3.640
MAE-VQGAN	**13.467 ± 0.775**	14.996 ± 0.952	**14.765 ± 1.826**	**11.178 ± 1.901**	**15.399 ± 2.183**	**13.552 ± 2.390**	**25.367 ± 3.302**	**14.576 ± 2.829**
Painter	11.804 ± 0.667	15.065 ± 1.075	10.805 ± 1.672	9.159 ± 1.559	13.885 ± 2.352	11.717 ± 2.803	9.535 ± 2.083	12.292 ± 4.020
PromptDiff.	8.593 ± 0.431	9.442 ± 0.169	9.670 ± 1.293	9.464 ± 1.385	13.034 ± 1.959	11.124 ± 1.507	10.721 ± 0.533	10.212 ± 1.506
SD-VICL	13.288 ± 0.799	**16.554 ± 1.167**	11.793 ± 1.894	10.003 ± 1.746	12.892 ± 3.842	9.199 ± 1.874	25.095 ± 2.945	9.780 ± 1.846
Copy target	12.199 ± 0.674	13.692 ± 0.825	12.150 ± 2.102	8.930 ± 1.420	13.224 ± 4.048	9.576 ± 1.834	25.947 ± 2.798	10.467 ± 1.964

Table 42: Flip results by dataset (SSIM ↑). Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	0.242 ± 0.141	0.329 ± 0.177	0.411 ± 0.193	0.330 ± 0.163	0.302 ± 0.201	0.253 ± 0.124	**0.360 ± 0.068**	0.461 ± 0.067
DELVM	0.234 ± 0.146	0.300 ± 0.182	0.286 ± 0.228	0.301 ± 0.184	0.307 ± 0.207	0.251 ± 0.114	0.315 ± 0.075	0.440 ± 0.078
LVM	0.241 ± 0.135	0.329 ± 0.176	0.382 ± 0.191	0.329 ± 0.164	0.319 ± 0.209	0.260 ± 0.114	0.302 ± 0.087	0.426 ± 0.069
MAE-VQGAN	**0.307 ± 0.145**	**0.372 ± 0.172**	**0.453 ± 0.193**	**0.377 ± 0.161**	**0.341 ± 0.203**	**0.307 ± 0.122**	0.335 ± 0.075	0.453 ± 0.050
Painter	0.227 ± 0.139	0.297 ± 0.180	0.293 ± 0.215	0.331 ± 0.167	0.283 ± 0.200	0.248 ± 0.114	0.273 ± 0.041	0.205 ± 0.057
PromptDiff.	0.249 ± 0.115	0.241 ± 0.112	0.322 ± 0.135	0.257 ± 0.109	0.202 ± 0.120	0.263 ± 0.092	0.179 ± 0.023	0.051 ± 0.013
SD-VICL	0.139 ± 0.087	0.202 ± 0.132	0.278 ± 0.158	0.240 ± 0.144	0.090 ± 0.083	0.122 ± 0.068	0.277 ± 0.074	**0.503 ± 0.067**
Copy target	0.162 ± 0.082	0.200 ± 0.112	0.266 ± 0.136	0.246 ± 0.129	0.191 ± 0.120	0.160 ± 0.064	0.269 ± 0.055	0.429 ± 0.052

Table 43: Flip results by dataset (SSIM ↑). Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	**0.368 ± 0.056**	0.240 ± 0.074	**0.537 ± 0.163**	0.356 ± 0.111	0.734 ± 0.209	0.327 ± 0.215	**0.642 ± 0.162**	0.365 ± 0.205
DELVM	0.163 ± 0.050	0.372 ± 0.100	0.412 ± 0.166	0.343 ± 0.105	0.728 ± 0.235	0.291 ± 0.214	0.366 ± 0.196	0.387 ± 0.218
LVM	0.157 ± 0.049	0.368 ± 0.072	0.435 ± 0.129	0.321 ± 0.110	0.721 ± 0.207	0.323 ± 0.221	0.442 ± 0.178	0.379 ± 0.209
MAE-VQGAN	0.262 ± 0.048	0.364 ± 0.043	0.532 ± 0.141	0.374 ± 0.104	**0.739 ± 0.170**	**0.381 ± 0.219**	0.463 ± 0.158	**0.453 ± 0.199**
Painter	0.107 ± 0.023	0.270 ± 0.065	0.429 ± 0.116	0.379 ± 0.087	0.696 ± 0.214	0.319 ± 0.224	0.198 ± 0.057	0.317 ± 0.222
PromptDiff.	0.067 ± 0.014	0.066 ± 0.009	0.497 ± 0.105	**0.383 ± 0.101**	0.312 ± 0.062	0.308 ± 0.159	0.214 ± 0.040	0.259 ± 0.113
SD-VICL	0.226 ± 0.044	**0.430 ± 0.066**	0.395 ± 0.131	0.354 ± 0.086	0.618 ± 0.208	0.211 ± 0.155	0.493 ± 0.106	0.249 ± 0.143
Copy target	0.199 ± 0.042	0.342 ± 0.039	0.427 ± 0.134	0.270 ± 0.068	0.647 ± 0.240	0.221 ± 0.139	0.686 ± 0.088	0.270 ± 0.143

Table 44: Flip results by dataset (LPIPS ↓). Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	0.711 ± 0.091	**0.624 ± 0.108**	0.649 ± 0.141	**0.648 ± 0.087**	0.611 ± 0.068	0.692 ± 0.105	**0.540 ± 0.036**	0.406 ± 0.050
DELVM	0.714 ± 0.094	0.681 ± 0.125	0.741 ± 0.136	0.728 ± 0.110	0.657 ± 0.126	0.646 ± 0.060	0.595 ± 0.047	**0.321 ± 0.071**
LVM	0.772 ± 0.098	0.625 ± 0.107	0.723 ± 0.137	0.724 ± 0.082	0.638 ± 0.094	0.677 ± 0.085	0.599 ± 0.039	0.342 ± 0.049
MAE-VQGAN	0.731 ± 0.064	0.679 ± 0.085	**0.647 ± 0.090**	0.732 ± 0.087	0.680 ± 0.068	0.685 ± 0.054	0.637 ± 0.028	0.378 ± 0.035
Painter	**0.708 ± 0.091**	0.644 ± 0.117	0.701 ± 0.106	0.795 ± 0.117	**0.554 ± 0.042**	**0.637 ± 0.060**	0.590 ± 0.025	0.330 ± 0.033
PromptDiff.	0.844 ± 0.068	0.747 ± 0.066	0.806 ± 0.106	0.853 ± 0.072	0.704 ± 0.049	0.833 ± 0.076	0.690 ± 0.022	0.673 ± 0.030
SD-VICL	0.765 ± 0.064	0.724 ± 0.060	0.791 ± 0.090	0.740 ± 0.079	0.680 ± 0.056	0.768 ± 0.060	0.555 ± 0.027	0.347 ± 0.039
Copy target	0.770 ± 0.059	0.740 ± 0.057	0.789 ± 0.084	0.748 ± 0.081	0.635 ± 0.057	0.766 ± 0.057	0.549 ± 0.050	0.370 ± 0.039

Table 45: Flip results by dataset (LPIPS ↓). Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	0.438 ± 0.041	0.498 ± 0.072	**0.576 ± 0.086**	0.613 ± 0.073	0.424 ± 0.114	0.674 ± 0.096	0.610 ± 0.064	0.636 ± 0.134
DELVM	0.514 ± 0.090	0.375 ± 0.102	0.619 ± 0.045	**0.610 ± 0.050**	**0.363 ± 0.140**	0.755 ± 0.124	0.679 ± 0.101	0.614 ± 0.136
LVM	0.477 ± 0.053	0.361 ± 0.057	0.625 ± 0.046	0.671 ± 0.095	0.387 ± 0.149	0.681 ± 0.095	0.625 ± 0.081	0.635 ± 0.138
MAE-VQGAN	**0.431 ± 0.027**	0.420 ± 0.026	0.609 ± 0.059	0.631 ± 0.046	0.405 ± 0.132	0.710 ± 0.098	0.613 ± 0.078	**0.596 ± 0.115**
Painter	0.457 ± 0.032	**0.309 ± 0.031**	0.628 ± 0.029	0.756 ± 0.066	0.379 ± 0.138	**0.668 ± 0.090**	0.632 ± 0.033	0.614 ± 0.121
PromptDiff.	0.703 ± 0.027	0.753 ± 0.021	0.624 ± 0.034	0.754 ± 0.067	0.795 ± 0.035	0.769 ± 0.083	0.815 ± 0.037	0.814 ± 0.060
SD-VICL	0.520 ± 0.028	0.391 ± 0.038	0.635 ± 0.048	0.675 ± 0.052	0.515 ± 0.122	0.770 ± 0.072	**0.609 ± 0.044**	0.728 ± 0.094
Copy target	0.416 ± 0.038	0.373 ± 0.032	0.614 ± 0.093	0.675 ± 0.043	0.441 ± 0.194	0.773 ± 0.071	0.551 ± 0.032	0.722 ± 0.093

Table 46: Rotation results by dataset (PSNR ↑). Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	9.639 ± 1.940	8.779 ± 1.854	11.075 ± 2.474	8.664 ± 1.939	14.353 ± 3.527	9.469 ± 1.877	11.250 ± 1.647	11.169 ± 1.396
DELVM	9.835 ± 2.583	8.655 ± 2.264	9.271 ± 3.384	7.357 ± 2.219	15.209 ± 4.627	**10.939 ± 2.155**	9.944 ± 2.850	12.413 ± 1.330
LVM	9.297 ± 1.963	9.422 ± 1.875	10.341 ± 2.554	9.025 ± 2.017	14.976 ± 3.917	9.802 ± 2.054	9.304 ± 3.145	12.674 ± 1.576
MAE-VQGAN	9.745 ± 2.382	8.271 ± 2.163	**11.493 ± 2.650**	7.947 ± 2.231	13.807 ± 2.714	9.785 ± 2.121	10.340 ± 1.774	13.400 ± 1.484
Painter	9.998 ± 2.488	9.022 ± 2.251	10.000 ± 3.411	9.143 ± 2.496	**17.210 ± 3.633**	10.925 ± 1.940	9.646 ± 1.123	11.312 ± 1.065
PromptDiff.	**11.449 ± 1.778**	**9.828 ± 1.519**	10.811 ± 1.962	**9.442 ± 1.446**	11.786 ± 3.198	9.693 ± 1.373	7.499 ± 0.636	7.180 ± 0.324
SD-VICL	8.995 ± 1.519	8.766 ± 1.625	9.816 ± 2.217	9.022 ± 1.446	12.121 ± 2.401	8.733 ± 1.748	**12.723 ± 1.564**	**14.216 ± 1.260**
Copy target	9.509 ± 1.575	9.117 ± 1.627	9.958 ± 2.123	9.378 ± 1.630	14.206 ± 2.747	9.347 ± 1.761	12.301 ± 2.110	13.417 ± 1.120

Table 47: Rotation results by dataset (PSNR ↑). Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	11.817 ± 0.952	10.962 ± 0.774	11.745 ± 1.691	9.354 ± 2.037	12.569 ± 2.711	9.835 ± 2.446	16.602 ± 1.448	10.386 ± 2.224
DELVM	10.246 ± 1.078	11.157 ± 1.260	11.640 ± 2.122	9.527 ± 2.203	**14.200 ± 3.177**	9.190 ± 2.575	19.984 ± 6.653	11.003 ± 3.151
LVM	10.671 ± 1.481	11.422 ± 1.591	**11.761 ± 1.940**	8.416 ± 1.600	13.722 ± 3.085	9.702 ± 2.117	20.969 ± 6.656	10.352 ± 2.631
MAE-VQGAN	11.773 ± 0.883	**13.635 ± 1.028**	10.902 ± 2.044	7.637 ± 2.155	13.624 ± 3.475	9.026 ± 2.245	24.213 ± 3.054	**11.247 ± 2.749**
Painter	10.790 ± 0.706	10.785 ± 0.780	10.486 ± 1.655	8.874 ± 1.567	13.088 ± 2.580	10.632 ± 2.584	9.494 ± 2.154	10.914 ± 3.303
PromptDiff.	7.890 ± 0.541	9.413 ± 0.168	10.000 ± 1.325	9.210 ± 1.359	13.099 ± 1.842	**10.754 ± 1.443**	10.728 ± 0.543	9.932 ± 1.640
SD-VICL	**12.056 ± 0.804**	12.999 ± 1.043	11.706 ± 1.800	**9.914 ± 1.750**	13.039 ± 4.241	9.009 ± 1.694	**25.326 ± 2.928**	9.664 ± 1.861
Copy target	12.199 ± 0.674	13.692 ± 0.825	12.150 ± 2.102	8.930 ± 1.420	13.224 ± 4.048	9.576 ± 1.834	25.947 ± 2.798	10.467 ± 1.964

Table 48: Rotation results by dataset (SSIM ↑). Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	0.214 ± 0.121	0.255 ± 0.149	0.378 ± 0.184	0.271 ± 0.141	0.262 ± 0.176	0.220 ± 0.101	**0.266 ± 0.060**	0.382 ± 0.061
DELVM	0.210 ± 0.139	0.251 ± 0.166	0.267 ± 0.223	0.250 ± 0.155	0.310 ± 0.211	0.213 ± 0.101	0.224 ± 0.075	0.337 ± 0.077
LVM	0.235 ± 0.126	**0.301 ± 0.150**	0.375 ± 0.184	**0.311 ± 0.147**	0.314 ± 0.197	0.252 ± 0.109	0.221 ± 0.076	0.339 ± 0.084
MAE-VQGAN	**0.257 ± 0.135**	0.252 ± 0.148	**0.424 ± 0.191**	0.294 ± 0.142	**0.344 ± 0.202**	0.251 ± 0.112	0.218 ± 0.065	**0.429 ± 0.051**
Painter	0.204 ± 0.129	0.224 ± 0.152	0.272 ± 0.208	0.286 ± 0.159	0.274 ± 0.197	0.211 ± 0.102	0.198 ± 0.026	0.129 ± 0.034
PromptDiff.	0.245 ± 0.114	0.229 ± 0.107	0.317 ± 0.133	0.239 ± 0.098	0.200 ± 0.121	**0.258 ± 0.090**	0.157 ± 0.016	0.049 ± 0.011
SD-VICL	0.132 ± 0.085	0.180 ± 0.118	0.268 ± 0.157	0.234 ± 0.135	0.086 ± 0.073	0.111 ± 0.063	0.229 ± 0.045	0.410 ± 0.051
Copy target	0.162 ± 0.082	0.200 ± 0.112	0.266 ± 0.136	0.246 ± 0.129	0.191 ± 0.120	0.160 ± 0.064	0.269 ± 0.055	0.429 ± 0.052

Table 49: Rotation results by dataset (SSIM ↑). Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	**0.247 ± 0.059**	0.152 ± 0.048	0.466 ± 0.137	0.326 ± 0.104	0.646 ± 0.258	0.291 ± 0.206	**0.530 ± 0.171**	0.327 ± 0.189
DELVM	0.135 ± 0.053	0.189 ± 0.059	0.411 ± 0.159	0.308 ± 0.101	0.691 ± 0.277	0.288 ± 0.222	0.385 ± 0.189	0.351 ± 0.216
LVM	0.106 ± 0.042	0.183 ± 0.057	0.422 ± 0.124	0.310 ± 0.095	0.675 ± 0.255	**0.318 ± 0.192**	0.411 ± 0.159	0.356 ± 0.180
MAE-VQGAN	0.214 ± 0.040	**0.326 ± 0.042**	0.243 ± 0.169	0.221 ± 0.087	**0.704 ± 0.217**	0.307 ± 0.206	0.354 ± 0.141	**0.389 ± 0.187**
Painter	0.060 ± 0.016	0.088 ± 0.014	0.415 ± 0.118	0.367 ± 0.085	0.674 ± 0.240	0.293 ± 0.219	0.194 ± 0.059	0.278 ± 0.210
PromptDiff.	0.054 ± 0.012	0.065 ± 0.010	**0.499 ± 0.104**	**0.373 ± 0.098**	0.318 ± 0.061	0.297 ± 0.153	0.213 ± 0.041	0.250 ± 0.107
SD-VICL	0.193 ± 0.034	0.300 ± 0.038	0.383 ± 0.132	0.353 ± 0.086	0.615 ± 0.235	0.197 ± 0.153	0.497 ± 0.108	0.241 ± 0.141
Copy target	0.199 ± 0.042	0.342 ± 0.039	0.427 ± 0.134	0.270 ± 0.068	0.647 ± 0.240	0.221 ± 0.139	0.686 ± 0.088	0.270 ± 0.143

Table 50: Rotation results by dataset (LPIPS ↓). Mean ± standard deviation.

Model	AP-10K	ATLANTIS	Caltech Birds	CDD	DeepCrack	DRAM	Duke	IXI (Axial)
Analogist	0.765 ± 0.064	0.742 ± 0.067	**0.718 ± 0.111**	**0.758 ± 0.064**	0.653 ± 0.080	0.761 ± 0.070	0.606 ± 0.027	0.495 ± 0.040
DELVM	0.744 ± 0.077	0.749 ± 0.087	0.760 ± 0.119	0.779 ± 0.082	0.676 ± 0.120	0.697 ± 0.059	0.643 ± 0.016	0.436 ± 0.046
LVM	0.819 ± 0.075	0.777 ± 0.080	0.772 ± 0.125	0.820 ± 0.074	0.685 ± 0.097	0.812 ± 0.079	0.711 ± 0.077	0.450 ± 0.046
MAE-VQGAN	0.792 ± 0.067	0.764 ± 0.073	0.719 ± 0.100	0.794 ± 0.072	0.712 ± 0.064	0.784 ± 0.066	0.643 ± 0.029	**0.433 ± 0.044**
Painter	**0.742 ± 0.075**	**0.729 ± 0.073**	0.728 ± 0.089	0.830 ± 0.081	**0.581 ± 0.041**	**0.696 ± 0.058**	0.666 ± 0.009	0.486 ± 0.030
PromptDiff.	0.845 ± 0.066	0.760 ± 0.059	0.810 ± 0.101	0.850 ± 0.067	0.702 ± 0.050	0.837 ± 0.075	0.691 ± 0.019	0.712 ± 0.024
SD-VICL	0.770 ± 0.062	0.740 ± 0.054	0.795 ± 0.089	0.762 ± 0.067	0.692 ± 0.057	0.774 ± 0.058	**0.581 ± 0.015**	0.452 ± 0.032
Copy target	0.771 ± 0.059	0.735 ± 0.056	0.788 ± 0.083	0.744 ± 0.079	0.635 ± 0.055	0.764 ± 0.056	0.533 ± 0.042	0.372 ± 0.040

Table 51: Rotation results by dataset (LPIPS ↓). Mean ± standard deviation.

Model	IXI (Coronal)	IXI (Sagittal)	LLVIP	NYU-D V2	OTP	RORD	Sea-Thru	SUIM
Analogist	0.549 ± 0.036	0.550 ± 0.045	0.627 ± 0.053	0.662 ± 0.057	0.477 ± 0.129	0.735 ± 0.086	0.643 ± 0.072	0.679 ± 0.120
DELVM	0.576 ± 0.049	0.569 ± 0.040	**0.621 ± 0.041**	**0.641 ± 0.045**	**0.393 ± 0.168**	0.779 ± 0.115	0.673 ± 0.104	0.659 ± 0.133
LVM	0.575 ± 0.058	0.575 ± 0.051	0.640 ± 0.062	0.769 ± 0.077	0.443 ± 0.195	0.797 ± 0.098	0.648 ± 0.098	0.725 ± 0.125
MAE-VQGAN	**0.509 ± 0.035**	**0.480 ± 0.030**	0.666 ± 0.052	0.682 ± 0.042	0.441 ± 0.170	0.779 ± 0.087	0.634 ± 0.083	**0.645 ± 0.136**
Painter	0.548 ± 0.017	0.585 ± 0.020	0.636 ± 0.028	0.760 ± 0.065	0.408 ± 0.155	**0.711 ± 0.078**	0.645 ± 0.030	0.660 ± 0.104
PromptDiff.	0.699 ± 0.020	0.783 ± 0.024	0.623 ± 0.035	0.756 ± 0.064	0.796 ± 0.035	0.774 ± 0.079	0.812 ± 0.035	0.813 ± 0.060
SD-VICL	0.551 ± 0.021	0.546 ± 0.021	0.636 ± 0.049	0.680 ± 0.052	0.528 ± 0.125	0.773 ± 0.076	**0.599 ± 0.044**	0.733 ± 0.091
Copy target	0.425 ± 0.040	0.383 ± 0.033	0.606 ± 0.093	0.676 ± 0.043	0.442 ± 0.187	0.769 ± 0.073	0.554 ± 0.032	0.721 ± 0.093

A.10 QUALITATIVE RESULTS

We show randomly selected qualitative results across tasks and datasets for all the models in VIBE.

A.10.1 PERCEPTION & LOCALIZATION TASKS

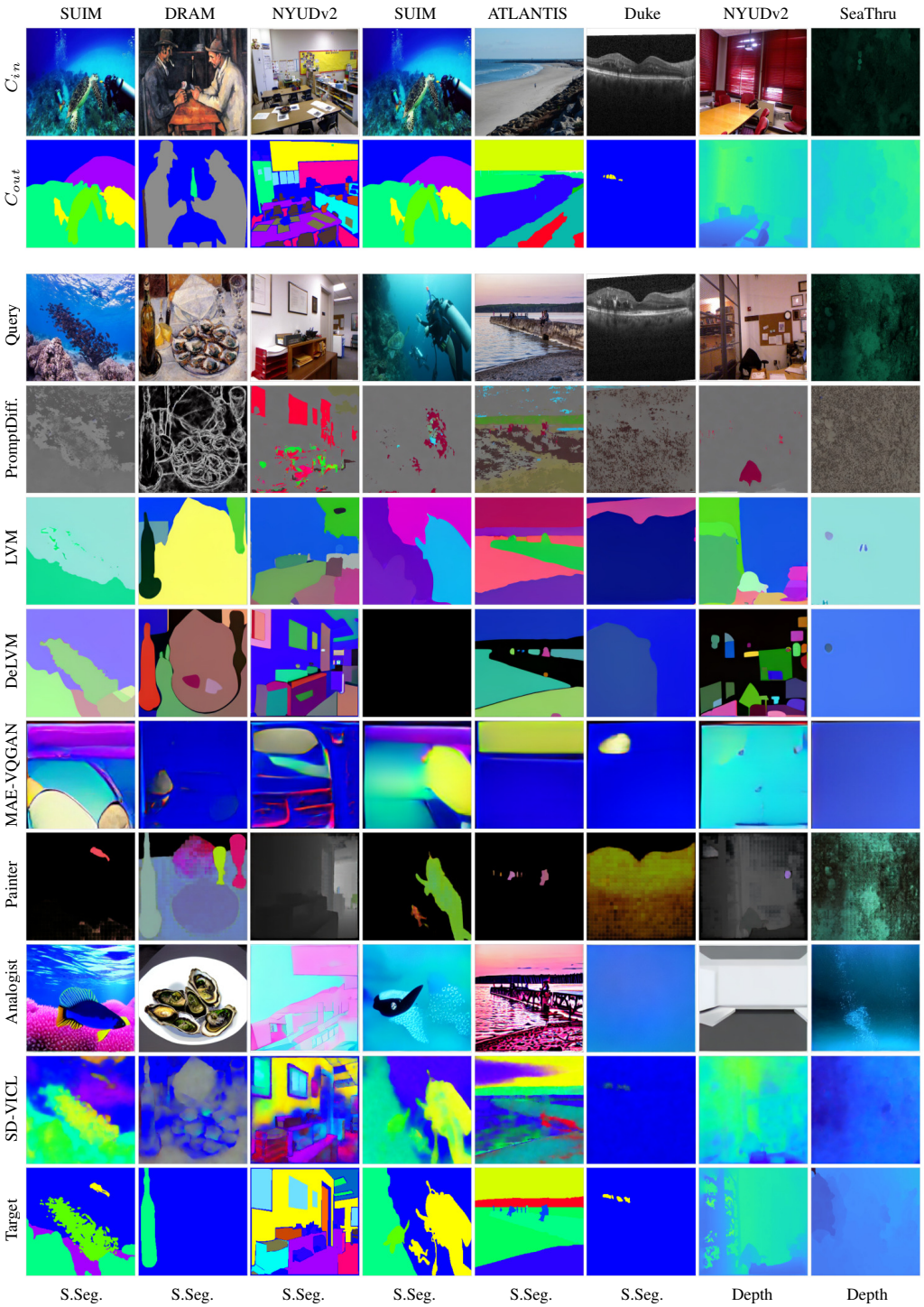


Figure 20: Qualitative samples for tasks segmentation and depth estimation.

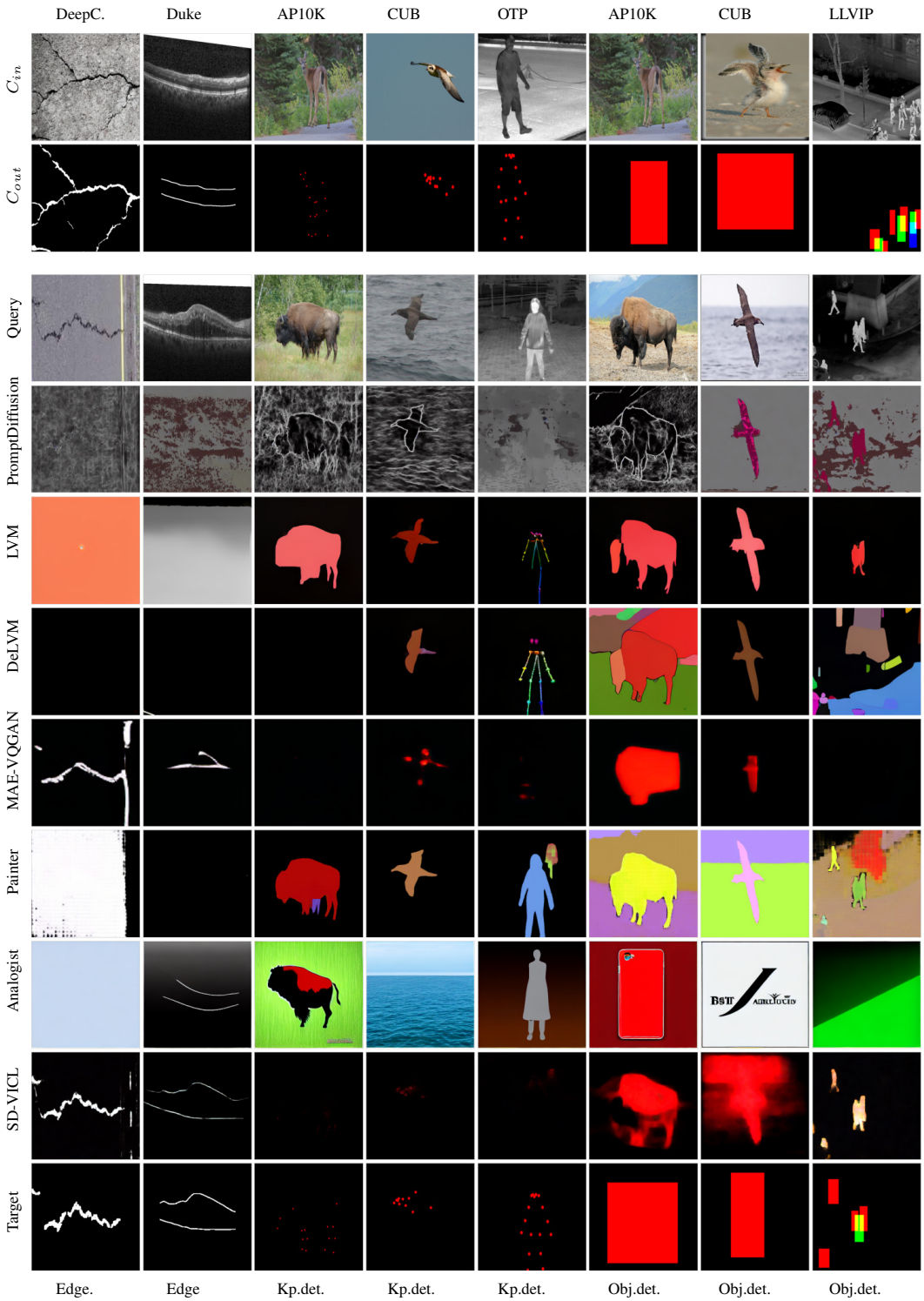


Figure 21: Qualitative samples for edge detection, object detection and keypoint detection.

A.10.2 IMAGE RESTORATION & ENHANCEMENT TASKS

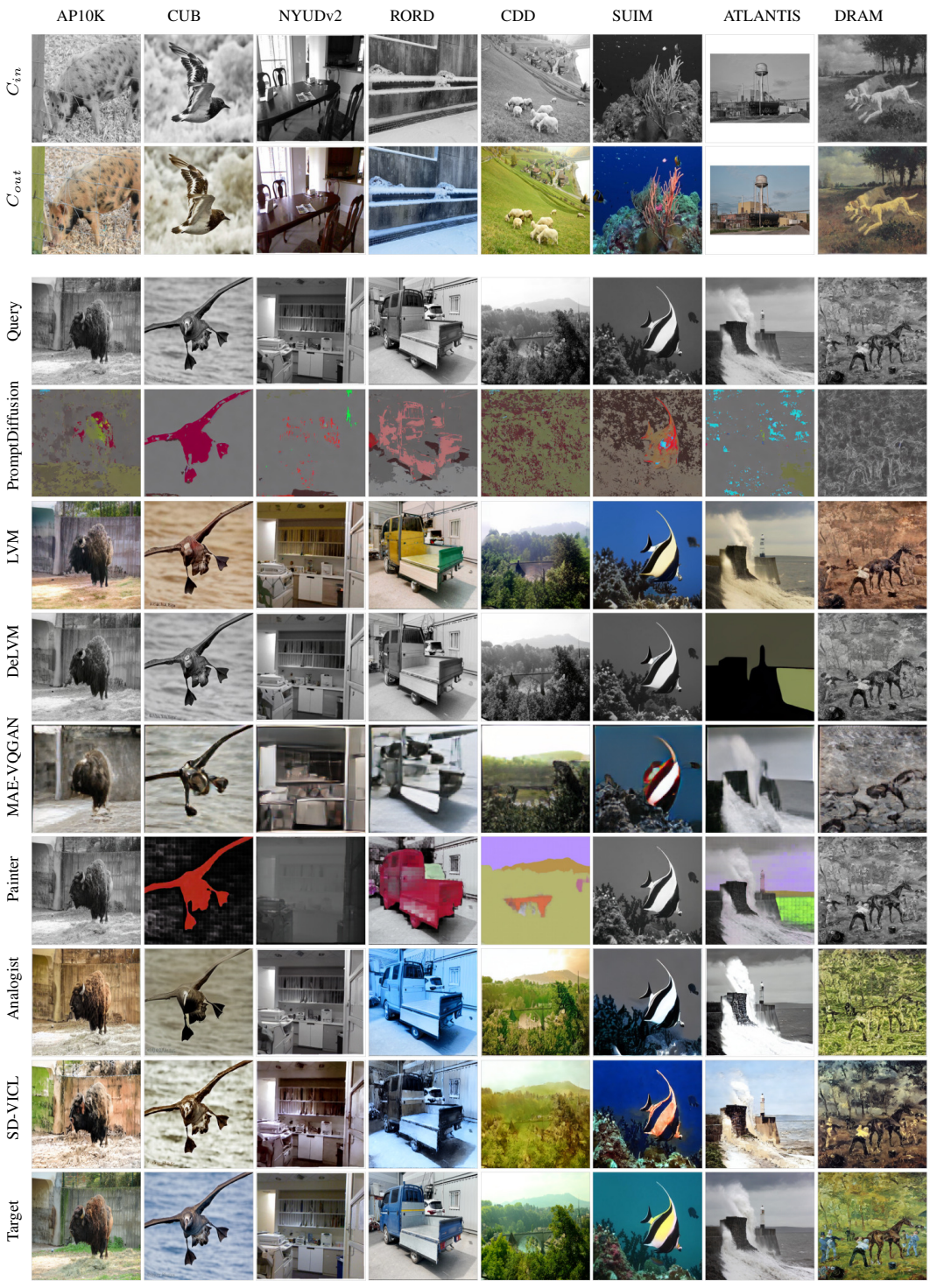


Figure 22: Qualitative samples for the colorization task covering different domains.

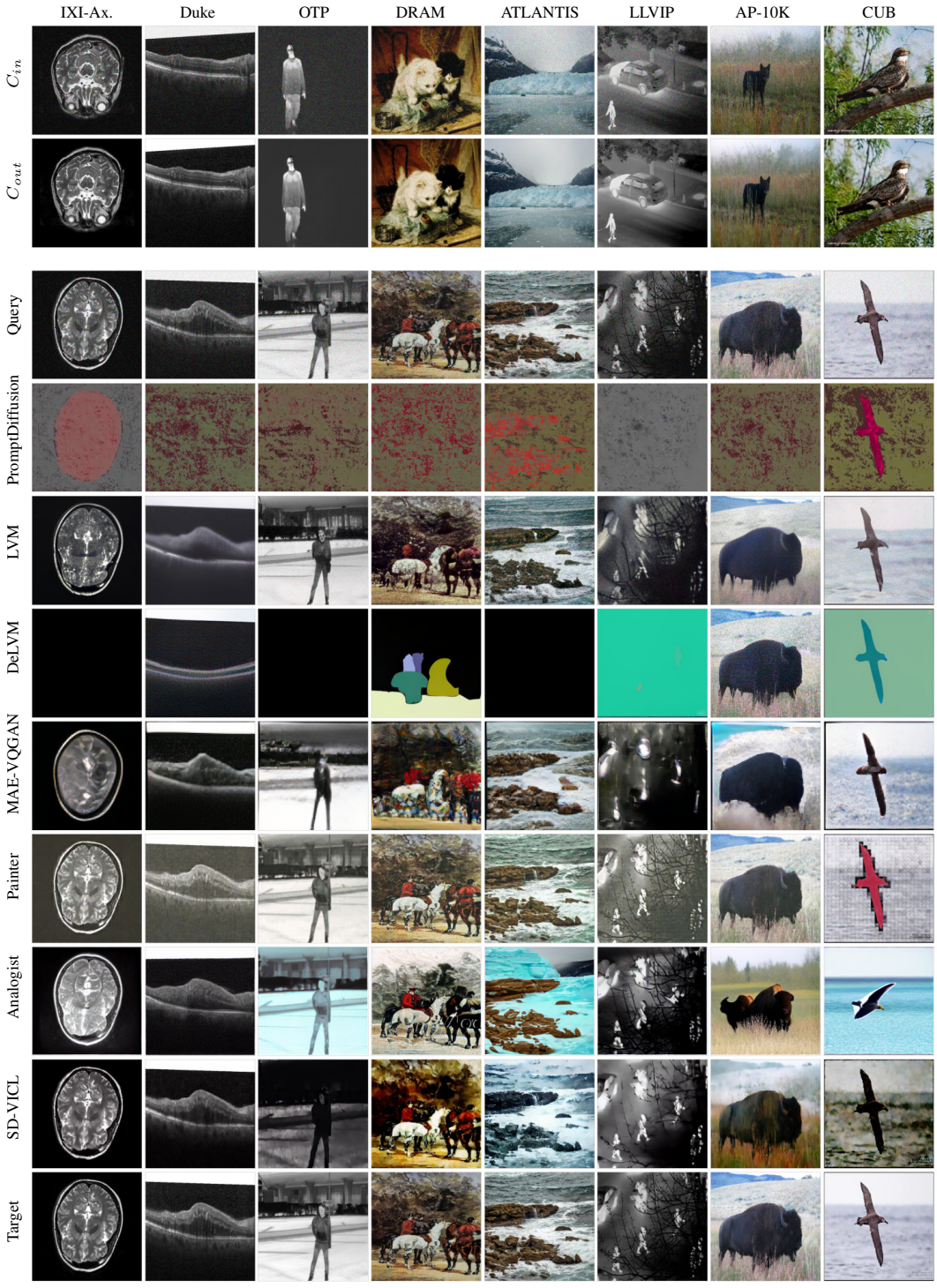


Figure 23: Qualitative samples for the denoising task covering different domains.

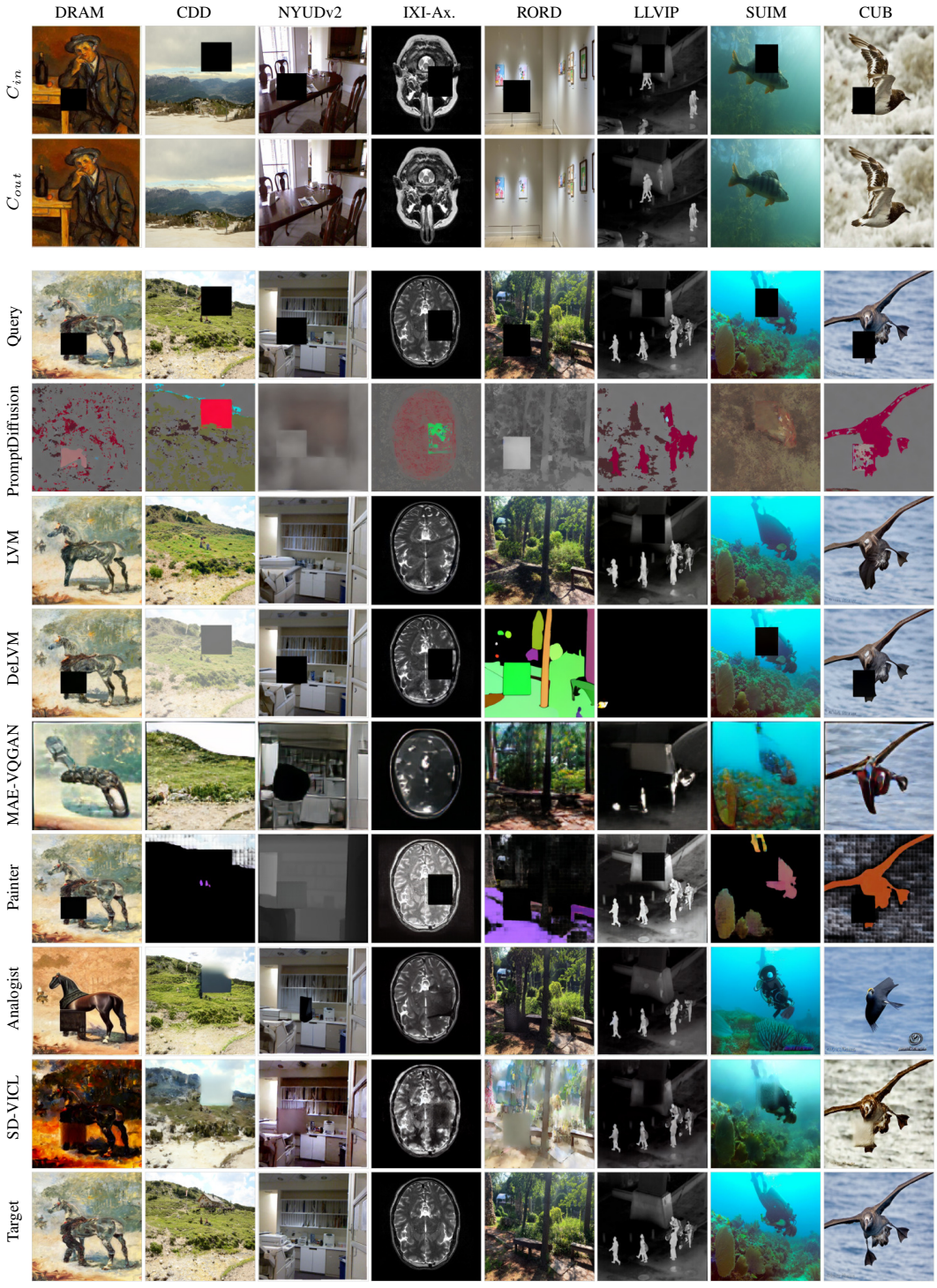


Figure 24: Qualitative samples for the inpainting task covering different domains.

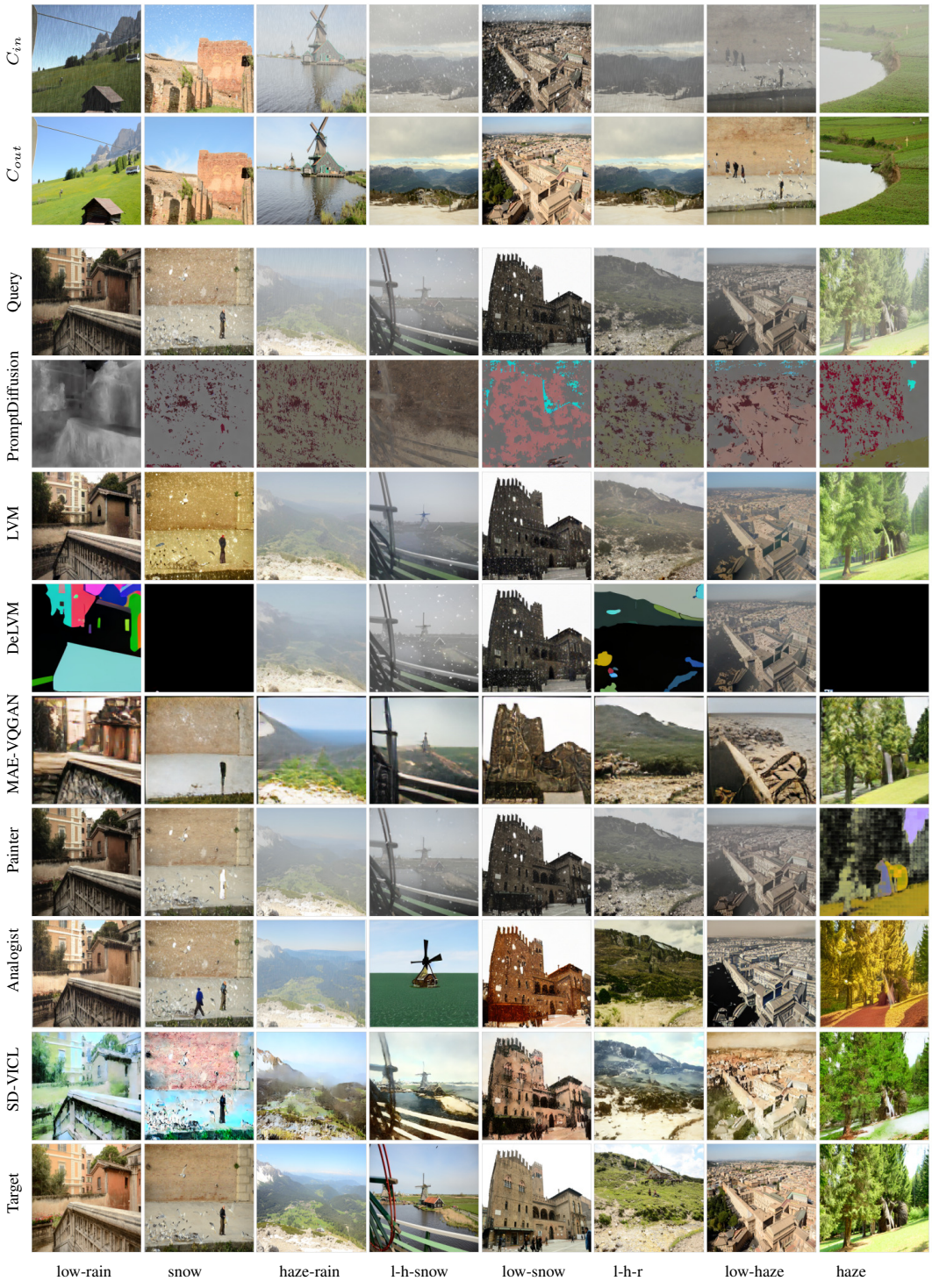


Figure 25: Qualitative samples for different artifact removal modes in CDD (Guo et al., 2024).

A.10.3 IMAGE MANIPULATION & TRANSFORMATION TASKS



Figure 26: Qualitative samples for modality transfer, rotation, flip and object removal.