# LaVin-DiT: Large Vision Diffusion Transformer

Zhaoqing Wang[1,4]    Xiaobo Xia[2]    Runnan Chen[1]    Dongdong Yu[4]

Changhu Wang[4*]    Mingming Gong[3*]    Tongliang Liu[1*]

[1]The University of Sydney    [2]National University of Singapore    [3]The University of Melbourne    [4]AIsphere

wangchanghu@aishi.ai, mingming.gong@unimelb.edu.au, tongliang.liu@sydney.edu.au
https://derrickwang005.github.io/LaVin-DiT/

## Abstract

*This paper presents the Large Vision Diffusion Transformer (LaVin-DiT), a scalable and unified foundation model designed to tackle over 20 computer vision tasks in a generative framework. Unlike existing large vision models directly adapted from natural language processing architectures, which rely on less efficient autoregressive techniques and disrupt spatial relationships essential for vision data, LaVin-DiT introduces key innovations to optimize generative performance for vision tasks. First, to address the high dimensionality of visual data, we incorporate a spatial-temporal variational autoencoder that encodes data into a continuous latent space. Second, for generative modeling, we develop a joint diffusion transformer that progressively produces vision outputs. Third, for unified multi-task training, in-context learning is implemented. Input-target pairs serve as task context, which guides the diffusion transformer to align outputs with specific tasks within the latent space. During inference, a task-specific context set and test data as queries allow LaVin-DiT to generalize across tasks without fine-tuning. Trained on extensive vision datasets, the model is scaled from 0.1B to 3.4B parameters, demonstrating substantial scalability and state-of-the-art performance across diverse vision tasks. This work introduces a novel pathway for large vision foundation models, underscoring the promising potential of diffusion transformers. The code and models will be open-sourced.*

## 1. Introduction

Large language models (LLMs) like GPT [10] and LLaMA [58] have rapidly gained widespread attention and transformed the field, demonstrating the strong capability to handle a wide range of language tasks within a unified
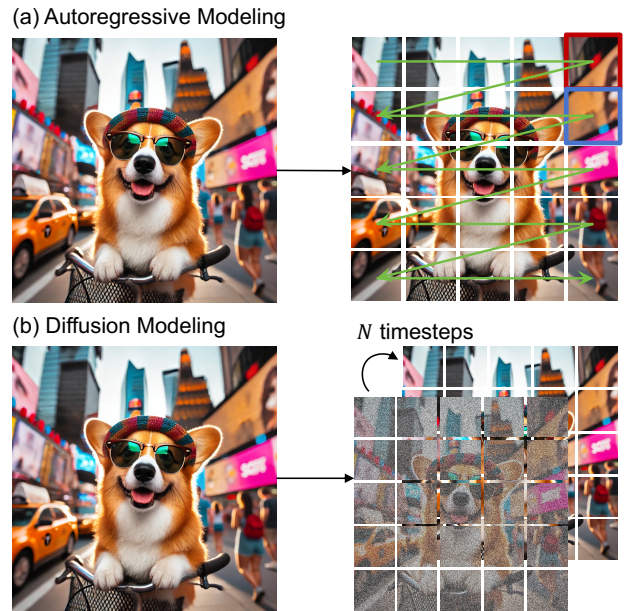


Figure 1. **Comparison of autoregressive and diffusion modeling.** (a) In **autoregressive modeling**, visual data is divided into a sequence of patches and transformed into a one-dimensional sequence. The model then predicts each token sequentially from left to right and top to bottom, which is computationally intensive for high-dimensional visual data. Besides, tokens marked in red and blue illustrate disrupted spatial dependencies, highlighting the limitations of preserving spatial coherence. (b) In contrast, **diffusion modeling** denoises all tokens in parallel across $N$ timesteps, significantly improving computational efficiency and preserving essential spatial structures crucial for high-performance vision tasks.

framework [1]. This breakthrough of integrating diverse language tasks into a single large model has sparked momentum to develop similar large models for computer vision. The potential to create large vision models (LVMs) capable of generalizing across multiple vision tasks repre-

*Corresponding author.

1

sents a promising step toward a more versatile, scalable, and efficient approach to vision-based AI [5, 8, 39, 63].

However, constructing LVMs presents greater complexity than LLMs due to the inherently diverse and high-dimensional nature of vision data, as well as the need to handle variations in scale, perspective, and lighting across tasks [31, 48, 52, 70]. To handle the problem, recent work [5] has developed a sequential modeling method that learns from purely vision data by representing images, videos, and annotations in a unified "visual sentence" format. This method enables the model to predict sequential vision tokens from a vast dataset, entirely independent of language-based inputs (see Figure 1(a)). Although this method has shown promising results in diverse vision tasks, it faces two primary challenges. Specifically, the first issue concerns the efficiency limitations inherent in autoregressive sequence modeling [55], as it demands token-by-token prediction, which is computationally intensive for high-dimensional vision data [49]. The second issue is the disruption of spatial coherence when converting vision data into a sequential format, which compromises the preservation of spatial dependencies crucial for performance in vision tasks [77].

In this paper, we introduce a large vision diffusion transformer (LaVin-DiT) to advance the development of next-generation LVMs. LaVin-DiT enjoys better computational efficiency and effectively preserves spatial relationships within vision data, thereby achieving superior performance across diverse vision tasks (see Figure 1(b)). Technically, to tackle the high-dimensional nature of vision data, we introduce a spatial-temporal variational autoencoder [30] that encodes data (*i.e.*, image and video) into a continuous latent space, allowing compact representation while preserving essential spatial and temporal features. This reduces computational demands and improves efficiency without sacrificing the model's ability to capture complex patterns. Besides, for generative modeling, we augment an existing diffusion transformer and propose a joint diffusion transformer with full-sequence joint attention. This module synthesizes visual outputs through parallel denoising steps, effectively reducing sequential dependencies to enhance processing efficiency while maintaining the spatial coherence essential for vision tasks. Moreover, to support unified multi-task training [61], we incorporate in-context learning [10, 19, 65, 73], where input-target pairs guide the diffusion transformer in aligning outputs with specific tasks. During inference, LaVin-DiT leverages task-specific context sets and test data as queries to adapt to various tasks without fine-tuning. This capability enables LaVin-DiT to achieve robust generalization across diverse tasks, leading to a versatile solution for complex vision applications.

We conduct comprehensive experiments to demonstrate the superiority of LaVin-DiT. Results show that LaVin-DiT significantly outperforms the strongest baseline LVM [5] on various vision benchmarks. For instance, it achieves a 24 lower AbsRel in NYU-v2 depth estimation [52]. Besides, LaVin-DiT offers $1.7 \sim 2.3\times$ faster inference speeds than LVM [5] across resolutions ranging from $256 \times 256$ to $512 \times 512$. Evaluations across different model sizes showcase the scalability and fast convergence of LaVin-DiT across multiple complex vision tasks. Finally, we observe that increasing the task context length consistently enhances performance across a diverse array of tasks. These promising results establish LaVin-DiT as a highly scalable, efficient, and versatile model, showing a new pathway for large vision foundation models.

## 2. Related Work

**Large vision model.** Developing a universal framework for diverse tasks across information sources is a longstanding goal in deep learning [39]. Natural language processing has achieved this with ChatGPT[1] that demonstrates versatility across numerous language tasks, *e.g.*, summarization, reasoning, and translation. In contrast, computer vision is relatively lacking in universal frameworks, largely due to the complexity and diversity of visual data and tasks. Existing methods of universal vision frameworks generally follow two main pathways: image-resembling generation [8, 14, 63] and sequential modeling [5, 33].

The image-resembling generation methods reformulate visual tasks as image generation problems, which allows models to handle dense visual predictions through inpainting and reconstruction tasks [8]. For instance, Painter [63] formulates dense prediction tasks as masked image inpainting, demonstrating in-context capabilities across multiple vision tasks. By leveraging pre-trained diffusion models [49], several methods [24, 43, 60, 64] utilize visual or textual instruction to guide generation and enhance adaptability across various tasks. The sequential modeling methods are largely inspired by breakthroughs in large language models and apply the sequence-to-sequence framework to visual data [58]. For these methods, visual data is typically quantized into sequences of discrete tokens [59]. The model is optimized through next-token prediction [10]. Recently, Bai et al. [5] introduce a framework that extends this concept to vision without relying on linguistic data, which treats visual data as a "visual sentence". By representing images and videos as one-dimensional sequences, this method [5] enables a unified transformer that can tackle image and video tasks within a single framework, expanding the scope of sequential modeling in computer vision.

In this paper, from the respective of image-resembling generation, we propose a universal diffusion framework with a transformer architecture tailored for visual data,

---

[1] https://openai.com/index/chatgpt/

which preserves spatial-temporal structure and minimizes information loss. Trained exclusively on visual data, our flexible framework unifies image and video tasks, advancing toward a generalist model in computer vision.

**Diffusion transformer.** By resorting to vision transformer (ViT) [21, 32, 37], recent advancements [6, 7, 13, 17, 23, 41, 75] in generative modeling achieves significant improvements in scalability and performance for both image [12, 20, 45, 57, 77] and video generation [26, 40, 42, 68]. Among these advancements, U-ViT [6] treats all inputs as tokens by combining transformer blocks with a U-net architecture. DiT [41] employs a straightforward and non-hierarchical transformer structure, showcasing the scalability and versatility of diffusion transformers. MDT [23] and MaskDiT [75] enhance the training efficiency of DiT by using a masking strategy [27]. Subsequently, Stable Diffusion 3 [20] introduces a novel transformer-based architecture for text-to-image generation, which enables bidirectional interaction between image and text. Furthermore, diffusion transformers demonstrate robust capabilities for spatial-temporal modeling in video generation [9]. Previous methods [13, 40] utilize separate spatial and temporal attention mechanisms to reduce intensive computational costs. Besides, recent works [26, 42, 68] have proposed using 3D full attention to capture spatial-temporal information, ensuring consistency for large-moving objects. While diffusion transformers have shown impressive potential in visual content generation, their capability to serve as a large vision model unifying multiple vision tasks remains underexplored. In this paper, we introduce a new joint diffusion transformer with full-sequence joint attention that effectively integrates diverse vision tasks into a cohesive framework, elevating diffusion transformers to a new level of unified understanding and generation.

**In-context learning.** In-context learning is initially conceptualized with GPT-3 [10]. It has revolutionized the approach to task-specific model training by allowing models to infer and execute tasks based directly on contextual examples provided in prompts [72]. This paradigm shift enables models to perform complex reasoning and novel pattern recognition without direct training on those specific tasks. Extending beyond text, Flamingo [3] incorporates visual inputs and broadens in-context learning to multi-modal tasks such as image captioning, visual question answering, and optical character recognition. This demonstrates the model's ability to integrate and interpret both textual and visual data, enhancing its application across different domains. In the realm of computer vision, the concept of in-context learning is explored through methods such as visual prompting [8], which infers tasks directly from concatenated image examples and queries. In this paper, we build on this idea. A set of examples are sampled as task definitions and concatenated with the input query for the

model, to obtain predictions accordingly.

# 3. Method

**Problem setup.** Computer vision includes a series of tasks like object detection [11, 35, 48] and panoptic segmentation [15, 31, 66], which are typically handled by specialized models designed for specific input-target mappings [22]. While effective for single tasks, this specialization restricts model adaptability and scalability across multiple tasks or diverse visual data. To overcome this limitation, we aim to design a *conditional generative framework* that unifies multiple vision tasks within a single cohesive model. Specifically, given a query $x$ (*e.g.*, an image or a video), the framework produces the corresponding prediction $\hat{y}$ to approximate the target $y$ conditioned on a set of input-target pairs $s$. These conditioning pairs provide task definitions and guidance, enabling the model to flexibly adapt to different tasks according to the supplied examples. Formally, the objective is to model the conditional distribution $p(y|x, s)$.

**Framework overview.** As shown in Figure 2(a), the proposed Large Vision Diffusion Transformer (**LaVin-DiT**) framework integrates a spatial-temporal variational autoencoder (ST-VAE) with a joint diffusion transformer to unify multiple vision tasks. Given a vision task, *e.g.*, panoptic segmentation, we first sample a set of input-target pairs as the task definition. Afterward, the set and other visual examples are fed into ST-VAE, which are encoded into latent representations. Subsequently, the encoded representations are patchified and unfolded into a sequential format. The set and input visual data form the conditional latent presentation $z_c$, while the target is perturbed with random Gaussian noise, yielding a noisy latent representation $z_t$. Both $z_c$ and $z_t$ are then put into the joint diffusion transformer (J-DiT), which denoises $z_t$ to recover a clean latent representation within the shared latent space. Lastly, the recovered latent representation is passed through the ST-VAE decoder to reconstruct the target in raw pixel space. Below we provide a detailed technical exposition of ST-VAE and J-DiT.

## 3.1. LaVin-DiT Modules

### 3.1.1. ST-VAE

It is computationally demanding to process visual data in raw pixel space [49]. To address this, we propose to use a spatial-temporal variational autoencoder (ST-VAE) [9, 62, 74]. ST-VAE can efficiently compress spatial and temporal information, and encode them from pixel space into compact latent space. As illustrated in Figure 2(b), ST-VAE uses causal 3D convolutions and deconvolutions to compress and reconstruct visual data. It overall includes an encoder, a decoder, and a latent regularization layer. These components are structured into four symmetric stages with alternating $2\times$ downsampling and upsampling. The first two stages
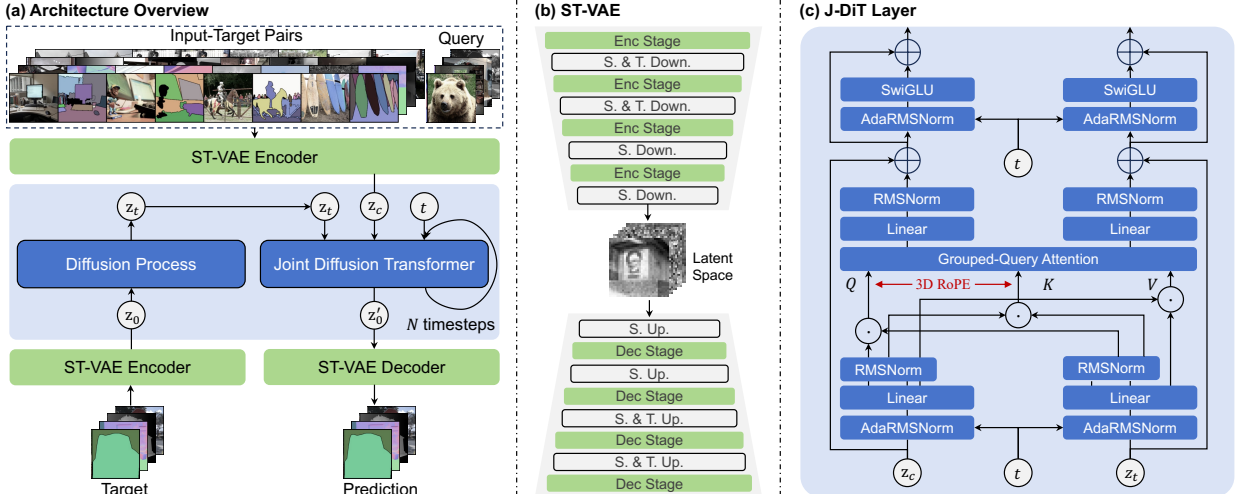
Figure 2. **Overview of Large Vision Diffusion Model (LaVin-DiT).** As shown in panel (a), the model initially compresses input visual data from the pixel space into a latent space, where multiple input-target pairs serve as the task context. A target is perturbed with Gaussian noise through a diffusion process. Guided by the task context and query, the Joint Diffusion Transformer (J-DiT) iteratively denoises this noisy target over $N$ timesteps to recover a clean latent representation. The prediction is then generated via the ST-VAE decoder. Panels (b) and (c) provide architectural details of the ST-VAE and J-DiT, respectively. "Down." and "Up." indicate the downsampling and upsampling, respectively. Concatenation is represented by $\odot$.

operate on both spatial and temporal dimensions, while the last stage affects only the spatial dimension, achieving an effective $4 \times 8 \times 8$ compression and reducing computational load. Besides, we apply a Kullback-Leibler (KL) constraint to regularize the Gaussian latent space.

To prevent future information leakage and its adverse effect on temporal predictions, we pad all locations at the start of the temporal convolution space. Additionally, to support both image and video processing, we treat the first frame of an input video independently, compressing it only spatially to maintain temporal independence. Subsequent frames are compressed along both spatial and temporal dimensions. The encoder of ST-VAE compresses the input to a lower-dimensional latent space, and the reconstruction is achieved through a decoding process. Training the ST-VAE occurs in two stages: we first train on images alone, then jointly on both images and videos. During each stage, we optimize the model using a combination of the mean squared error, perceptual loss [49, 71], and adversarial loss [49].

### 3.1.2. J-DiT

Diffusion transformers (DiT) [41] have emerged as a powerful method for generative modeling. Our joint diffusion transformer (J-DiT) builds upon DiT but introduces modifications to support the task-conditioned generation. A key distinction from the original DiT is our consideration of two conceptually different latent representations. The condition latent representation is clean, while the target latent representation is perturbed by Gaussian noise, resulting in poten-

tially distinct value ranges for the two. To handle the difference and improve alignment between task-specific and visual information, we construct separate patch embeddings for the condition and target latents. Each embedding layer uses a patch size of $2 \times 2$, which allows for tailoring the representations for each latent type. As shown in Figure 2, the sampled timestep $t$, along with the condition and target sequences, is fed into a series of diffusion transformer layers. Building on the MM-DiT [20] architecture, we introduce condition- and target-specific adaptive RMS normalization (AdaRN) to modulate each representation space independently. This is achieved through distinct timestep embeddings for the condition and target within AdaRN layers.

**Full-sequence joint attention.** Full-sequence joint attention is key in our transformer layers, which processes condition and noisy target sequences together to enhance task-specific alignment. As shown in Figure 2(c), the condition and target sequences are linearly projected, concatenated, and processed by a bidirectional attention module, allowing each to operate in its own space while considering the other. To improve speed and memory efficiency, we replace multi-head attention with grouped-query attention [2], which groups query heads to share a single set of key-value heads. This approach reduces parameters while retaining expressiveness, closely matching standard multi-head attention performance. Besides, to stabilize training with larger models and longer sequences, we add QK-Norm before query-key dot products to control attention entropy growth. Following [56], we also apply sandwich normal-

4

ization after each attention and FFN layer to maintain activation magnitudes amid residual connections.

**3D rotary position encoding.** Unlike [5], we argue that it is sub-optimal to model visual data as a one-dimensional sequence, because 1D positional embedding is limited in capturing precise spatial-temporal positions. Instead, by treating multiple image-annotation pairs or video clips as a single continuous sequence, we can use 3D Rotary Position Encoding (3D RoPE) [54] to represent spatial-temporal relationships concisely. Then, each location in a video can be expressed by a 3D coordinate. With the introduction of 3D RoPE, we provide a unified and accurate spatial-temporal representation of positional encoding for various vision tasks.

**Training procedure of J-DiT.** We train J-DiT using flow matching [36] in the latent space. Specifically, given a representation $z_0$ and noise $z_1 \sim \mathcal{N}(0,1)$, flow matching defines a linear interpolation based forward process: $z_t = tz_0 + (1-t)z_1$, where the timestep $t \in [0,1]$. This forward process induces a time-dependent velocity field $v(z_t, t)$ that drives the flow along the linear path in the direction of $(z_0 - z_1)$. The velocity field defines an ordinary differential equation (ODE): $dz_t = v(z_t, t)dt$. We employ J-DiT that is parameterized by $\theta$, to predict the velocity field that transforms noise into a clean latent representation. The training objective of flow matching is to directly regress the target velocity field, leading to the Conditional Flow Matching (CFM) loss [36]:

$$\ell_{\text{CFM}} = \int_0^1 \mathbb{E}[|v_{\theta}(z_t, t) - (z_0 - z_1)|_2^2]dt. \quad (1)$$

**Generation procedure of J-DiT.** Upon completion of J-DiT training, we use it to generate new representations by integrating from the noise distribution toward representation distribution. Specifically, starting from noise $z_1' \sim \mathcal{N}(0,1)$ at $t = 1$, we integrate the learned J-DiT backward to $t = 0$ to obtain a representation $z_0'$. For instance, using the Euler method, we discretize the time interval [0,1] to $N$ steps with a negative step size $\Delta t = -1/N$ to indicate backward integration in time. At each step $k = 0, 1/N, \dots, (N-1)/N$, we update the time and generated representation as follows:

$$t^{(k+1/N)} = t^{(k)} + \Delta t, \quad (2)$$

$$z^{(k+1/N)} = z^{(k)} + v_{\theta}(z^{(k)}, t^{(k)})\Delta t, \quad (3)$$

where $t^{(0)} = 1$, $t^{(1)} = 0$, $z^{(0)} = z_1'$, and $z^{(1)} = z_0'$. By iteratively applying these updates, we obtain a new presentation for the following decoding process of ST-VAE.

### 3.2. LaVin-DiT Inference

After completing the training of LaVin-DiT, the model becomes versatile and is ready to be applied across a range of downstream tasks. Specifically, when given a query (*e.g.*, an image or a video) for any chosen task, we randomly sample a set of input-target pairs that define the task. These pairs, alongside the visual input and a Gaussian noise component, are then fed into the Joint Diffusion Transformer (J-DiT). Within J-DiT, these elements are processed to generate a latent representation. Finally, this latent representation is passed through the ST-VAE decoder, which transforms it into the raw pixel space to produce the desired prediction. To better understand this inference procedure, please refer to Figure 2(a).

## 4. Experiments

### 4.1. Setup

**Training data.** To unify multiple computer vision tasks, we construct a large-scale multi-task dataset that encompasses indoor and outdoor environments, spanning real-world and synthetic domains. This dataset comprises approximately 3.2 million unique images [16, 18, 34, 51, 76] and 0.6 million unique videos [25, 28, 53], covering over 20 tasks:

- *Image-based tasks*: object detection, instance segmentation, panoptic segmentation, pose estimation, edge extraction, depth estimation, surface normal estimation, inpainting, colorization, image restoration tasks (*e.g.*, deraining, de-glass blur, and de-motion blur), depth-to-image, and normal-to-image generation.
- *Video-based tasks*: frame prediction, video depth estimation, video surface normal estimation, video optical flow estimation, video instance segmentation, depth-to-video, and normal-to-video generation.

To overcome the limitations of large-scale annotations for depth and surface normal estimation, we generate pseudo depth and normal maps on ImageNet-1K [18] by utilizing Depth-anything V2 [67] and Stable-Normal (turbo) [69], respectively.

**Implementation details.** We conduct training in two stages, progressively increasing the image resolution. In the first stage, we train at a $256 \times 256$ resolution for 100,000 steps, leveraging DeepSpeed ZeRO-2 [44] optimization and gradient checkpointing to manage memory and computational efficiency. We employ a global batch size of 640 and use an AdamW optimizer [38] with a learning rate of 0.0001, betas set to 0.9 and 0.95, and weight decay of 0.01. This setup provides stable training across configurations without the need for a warmup or additional regularization techniques. In the second stage, we upscale the resolution to $512 \times 512$ and continue training for an additional 20,000 steps, while the learning rate is adjusted to 0.00005. Other hyperparameters are retained from the first stage. This two-stage strategy enables efficient scaling, ensuring optimal performance across resolutions. By default, we utilize 20 timesteps ($N = 20$) during inference. All ex-

periments are conducted on 64×NVIDIA A100-80G GPUs.

**Evaluation protocols.** We assess our model on a comprehensive range of computer vision tasks spanning both image and video domains. Following established protocols, we report standard metrics for each task.

## 4.2. Main Results

**Quantitative analysis.** To assess the effectiveness of our proposed method, we conduct extensive experiments across a broad range of computer vision tasks and report results of the 3.4B model by default, as summarized in Tables 1 and 2. Our method consistently outperforms existing baselines across multiple tasks, including challenging cases such as unseen foreground segmentation and single-object detection, demonstrating superior generalization and adaptability across diverse scenarios. Note that unless otherwise specified, we report LaVin-Dit (3.4B) performance.

As shown in Table 1, we report the performance on foreground segmentation and single object detection across different splits. Our LaVin-DiT achieves significant improvements over baseline methods in all splits. Specifically, in the foreground segmentation task, we attain mIoUs of 67.87%, 75.80%, 66.98%, and 66.90% across four splits, consistently outperforming previous methods such as LVM [5] and MAE-VQGAN [8] by a substantial margin. Additionally, for single object detection, our model demonstrates strong performance, achieving top results in all splits. Notably, we achieve a mIoU of 68.88% in Split 4, which is a considerable margin of 19.96% over the best-performing baseline LVM. These significant gains highlight our model's ability to effectively segment and detect objects across a range of scenarios, even when faced with tasks unseen during training. Following prior work [5, 8], we further evaluate our model in the colorization task, where lower LPIPS and MSE values indicate superior performance. As shown in Table 1, our method achieves an LPIPS of 0.26 and an MSE of 0.24, significantly outperforming all baselines. These results underscore our model's capability to generate realistic and natural colors from grayscale images, which is essential in restoration and artistic fields.

To validate the ability of our model to understand the geometric structure of 3D scenes, we evaluate it on NYU-v2 depth estimation and surface normal estimation tasks [52], as shown in Table 2. As Bai et al. [5] do not report related results in their paper, we conduct evaluations using their official 7B model[2]. For depth estimation, our model achieves an AbsRel of 6.2 and a threshold accuracy $\delta_1$ of 96.1%, demonstrating competitive performance compared to expert models such as Marigold [29] and DPT [47]. In the surface normal estimation task, our method achieves an MAE of 15.901 and accuracy within a $< 11.25°$ threshold

---

[2]https://huggingface.co/Emma02/LVM_ckpts

of 58.382, surpassing the powerful expert model StableNormal [69]. This performance underscores our model's proficiency in estimating surface orientations accurately, enhancing its applicability in tasks requiring precise geometrical understanding, such as augmented reality and 3D reconstruction. These results reflect our model's capability to comprehend the geometric structure of 3D scenes with precision, even in complex environments, which is crucial for real-world applications like 3D scene reconstruction and spatial perception. Furthermore, we compare our LaVin-DiT to LVM on the inpainting task. Using 2,500 randomly selected images from the ImageNet-1K validation set, our model achieves an FID of 1.65, which greatly improves over the FID of 4.05 obtained by LVM.

**Qualitative analysis.** As shown in Figures 3, we present qualitative results in a wide variety of image-based and video-based tasks. Our model consistently follows task contexts and precisely generates the corresponding predictions. Furthermore, given sequential frames with task contexts, our model generates predictions for the subsequent 12 frames, which exhibits its ability to handle temporal consistency and scene dynamics effectively.

## 4.3. Scalability

To investigate the scalability of the proposed LaVin-DiT, we conduct experiments with three model sizes, *i.e.*, 0.1B, 1.0B, and 3.4B parameters. We train the three models for 100,000 steps. Figure 4 illustrates the training loss curves, which shows that larger models consistently achieve lower loss values. Additionally, the 3.4B model converges more rapidly, reaching smaller loss values in fewer training steps. This accelerated convergence suggests that larger models are better equipped to capture complex data patterns, leading to improved learning efficiency. The observed training dynamics underscore the advantages of scaling up model capacity for complex vision tasks, where larger models can more effectively capture diverse data characteristics.

Beyond training dynamics, the model size also has a substantial impact on downstream task performance. This is evident in colorization and depth estimation tasks, which were selected for their distinct requirements in capturing color fidelity and spatial structure. As seen in Figure 5, model performance improves consistently as its scale increases. Specifically, for colorization, the 3.4B model achieves an MSE of 0.273, significantly outperforming the 1.0B and 0.1B models that achieve MSEs of 0.311 and 0.609, respectively. Similarly, in depth estimation, the 3.4B model attains an AbsRel of 6.2, compared to 6.5 and 7.6 for the 1.0B and 0.1B models. These results demonstrate that larger models indeed deliver enhanced performance across multiple tasks, affirming LaVin-DiT as a scalable and adaptable framework for high-performance vision applications.

Table 1. **Comparison on foreground segmentation, single object detection, and colorization.** For foreground segmentation and single object detection, we report "mIoU" (higher is better). For colorization, we report "LPIPS" [71] and "MSE" (lower is better). Note that foreground segmentation and single object detection are *unseen* tasks during our training.

| Method | Foreground Segmentation (mIoU ↑) | | | | Single Object Detection (mIoU ↑) | | | | Colorization ↓ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Split 1 | Split 2 | Split 3 | Split 4 | Split 1 | Split 2 | Split 3 | Split 4 | MSE | LPIPS |
| MAE [8] | 17.42 | 25.70 | 18.64 | 16.53 | 5.49 | 4.98 | 5.24 | 5.84 | 0.43 | 0.55 |
| MAE-VQGAN [8] | 27.83 | 30.44 | 26.15 | 24.25 | 24.19 | 25.20 | 25.36 | 25.23 | 0.67 | 0.40 |
| LVM [5] | 48.94 | 51.29 | 47.66 | 50.82 | 48.25 | 49.60 | 50.08 | 48.92 | 0.51 | 0.46 |
| LaVin-DiT | **67.87** | **75.80** | **66.98** | **66.90** | **67.85** | **69.32** | **68.76** | **68.88** | **0.24** | **0.26** |



■ **Task Context**   ■ **Query**   ■ **Prediction**

Figure 3. **Qualitative results on diverse image and video-based tasks.** The first ten rows show image-based tasks, where each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). The last four rows show video-based tasks, where each row includes a video sequence with a series of target frames as task context, followed by a query frame. A set of frames in the red box indicates the model's predictions. *Best viewed in color.*

## 4.4. Inference Latency Analysis

As demonstrated in Figure 6, we compare the inference latency of LaVin-DiT and LVM (both 7B models) across in-creasing resolutions, demonstrating that our method is consistently more efficient. At a resolution of 256, LaVin-DiT requires only 4.67 seconds per example, while LVM takes 8.1 seconds, with this efficiency gap widening at higher res-
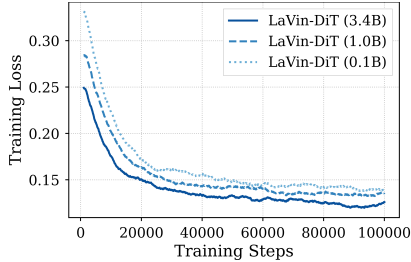
Figure 4. **Training loss curves for LaVin-DiT of varying model sizes.** The 3.4B model demonstrates faster convergence, achieving lower training losses than smaller models as training progresses.
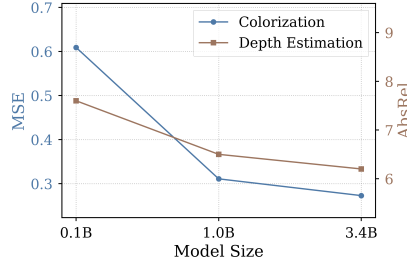
Figure 5. **Performance for LaVin-DiT of varying sizes.** Comparison of LaVin-DiT with different parameters on colorization (MSE) and depth estimation (AbsRel). Lower values indicate better performance.
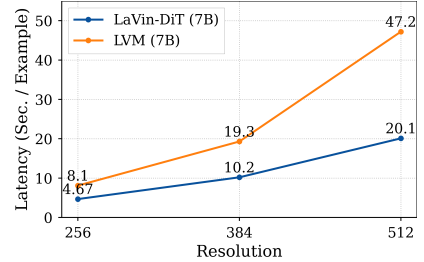
Figure 6. **Inference latency comparison.** LaVin-DiT consistently achieves lower latency than LVM [5] across different resolutions, as tested on an A100-80G GPU with 8 input-target pairs.



Figure 7. **Effect of task context length.** Longer task context can consistently improve the performance of downstream tasks.

Table 2. **Comparison on NYU-v2 depth estimation, surface normal estimation and ImageNet inpainting** [18, 52]. For depth estimation, we report absolute relative difference (AbsRel) and threshold accuracy ($\delta_1$). For surface normal estimation, we report mean angular error (MAE) and angle accuracy within a threshold ($< 11.25°$). We report FID for inpainting. † denotes evaluations on the official 7B model released by [5].

| Method | Depth Estimation | | Normal Estimation | | Inpainting |
|---|---|---|---|---|---|
| | AbsRel ($\downarrow$) | $\delta_1$ ($\uparrow$) | MAE ($\downarrow$) | $< 11.25°$ ($\uparrow$) | FID ($\downarrow$) |
| DPT [47] | 9.8 | 90.3 | - | - | - |
| StableNormal [69] | - | - | 19.707 | 53.042 | - |
| Marigold [29] | 6.0 | 95.9 | 20.864 | 50.457 | - |
| LVM† [5] | 30.2 | 52.3 | 23.433 | 44.836 | 4.05 |
| LaVin-DiT | **6.2** | **96.1** | **15.901** | **58.382** | **1.65** |

olutions (*e.g.*, 20.1 seconds *v.s.* 47.2 seconds at 512). This difference underscores a key advantage of diffusion models for vision tasks: unlike autoregressive models that process tokens sequentially and become increasingly time-intensive with larger inputs, diffusion models process tokens in parallel, allowing them to scale more effectively. This parallelism makes our LaVin-DiT a more suitable choice for large-scale vision applications.

## 4.5. Effect of Task Context Length

In-context learning enables the model to adapt to new tasks using a few examples, with performance generally improving as more examples are provided. We investigate this by assessing the effect of task context length across ten downstream tasks. As shown in Figure 7, the model consistently benefits from longer task contexts, achieving notable performance gains. For instance, with more input-target pairs, LaVin-DiT achieves lower FID in depth-to-image generation and higher PSNR in de-motion blur tasks. These results demonstrate that LaVin-DiT effectively leverages extended task context, highlighting its capacity to utilize additional information for enhanced task adaptation and accuracy.

## 5. Conclusion

We present LaVin-DiT, a scalable and unified foundation model for computer vision that integrates a spatial-temporal variational autoencoder and a diffusion transformer to efficiently process high-dimensional vision data while preserving spatial and visual coherence. Through in-context learning, LaVin-DiT adapts effectively to a wide range of tasks without fine-tuning, which shows remarkable versatility and adaptability. Extensive experiments validate LaVin-DiT's

scalability and performance, positioning it as a promising framework for developing generalist vision models.

**Limitations.** Despite its advantages, LaVin-DiT is limited by current constraints in large-scale training data, diverse task annotations, and computational resources, especially in comparison to large language models. While our model achieves strong results on seen tasks and related unseen tasks, it struggles with generalization when task definitions deviate significantly from the training distribution. This limitation highlights a key challenge in developing vision models that can generalize effectively to entirely new tasks defined solely by task context.

**Future work.** Future research should explore scaling LaVin-DiT further in terms of model capacity, dataset diversity, and task complexity to push the boundaries of vision generalization. We anticipate that as these elements expand, LaVin-DiT and similar models may gain the ability to handle arbitrary (out-of-training) vision tasks, guided only by a few input-target pairs. Additionally, investigating methods to select optimal task context automatically could provide a rapid and effective pathway to enhance model performance, ensuring that it leverages the most relevant examples for each task. These directions will drive further advances in developing robust, adaptable, and highly generalized foundation models for computer vision.

# References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *EMNLP*, 2023. 4

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, pages 23716–23736, 2022. 3

[4] Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *CVPR*, pages 9535–9545, 2024. 13

[5] 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 *CVPR*, pages 22861–22872, 2024. 2, 5, 6, 7, 8

[6] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, pages 22669–22679, 2023. 3

[7] Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu,

Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *ICML*, pages 1692–1717, 2023. 3

[8] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. In *NeurIPS*, pages 25005–25017, 2022. 2, 3, 6, 7, 13

[9] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 3

[10] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1, 2, 3

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 3

[12] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 3

[13] Shoufa Chen, Mengmeng Xu, Jiawei Ren, Yuren Cong, Sen He, Yanping Xie, Animesh Sinha, Ping Luo, Tao Xiang, and Juan-Manuel Perez-Rua. Gentron: Diffusion transformers for image and video generation. In *CVPR*, pages 6441–6451, 2024. 3

[14] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021. 2

[15] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, pages 12475–12485, 2020. 3

[16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 5, 14

[17] Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *ICML*, 2024. 3

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5, 8, 13, 14

[19] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022. 2

[20] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 3, 4

[21] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, pages 6824–6835, 2021. 3

[22] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *CVPR*, pages 2427–2436, 2019. 3

[23] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *ICCV*, pages 23164–23173, 2023. 3

[24] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *CVPR*, pages 12709–12720, 2024. 2

[25] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *CVPR*, pages 3749–3761, 2022. 5, 14

[26] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2023. 3

[27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 3

[28] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5, 14

[29] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, pages 9492–9502, 2024. 6, 8

[30] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[31] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, pages 9404–9413, 2019. 2, 3

[32] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[33] Alexander Kolesnikov, André Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. Uvim: A unified modeling approach for vision with learned guiding codes. In *NeurIPS*, pages 26295–26308, 2022. 2

[34] 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 *ECCV*, pages 740–755, 2014. 5, 14

[35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 3

[36] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023. 5, 12

[37] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 3

[38] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[39] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *ICLR*, 2022. 2

[40] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024. 3

[41] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 3, 4

[42] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 3

[43] Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, et al. Unicontrol: A unified diffusion model for controllable visual generation in the wild. In *NeurIPS*, 2024. 2

[44] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16, 2020. 5

[45] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

[46] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 13

[47] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, pages 12179–12188, 2021. 6, 8

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2016. 2, 3

[49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2, 3, 4

[50] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017. 13

[51] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, pages 8430–8439, 2019. 5, 14

[52] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012. 2, 6, 8, 13

[53] K Soomro. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5, 14

[54] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 5

[55] I Sutskever. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014. 2

[56] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024. 4

[57] Yuchuan Tian, Zhijun Tu, Hanting Chen, Jie Hu, Chao Xu, and Yunhe Wang. U-dits: Downsample tokens in u-shaped diffusion transformers. *arXiv preprint arXiv:2405.02730*, 2024. 3

[58] 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. 1, 2

[59] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017. 2

[60] Wouter Van Gansbeke and Bert De Brabandere. A simple latent diffusion approach for panoptic segmentation and mask inpainting. *arXiv preprint arXiv:2401.10227*, 2024. 2

[61] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3614–3633, 2021. 2

[62] Junke Wang, Yi Jiang, Zehuan Yuan, Binyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *arXiv preprint arXiv:2406.09399*, 2024. 3

[63] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, pages 6830–6839, 2023. 2

[64] Zhendong Wang, Yifan Jiang, Yadong Lu, Pengcheng He, Weizhu Chen, Zhangyang Wang, Mingyuan Zhou, et al. In-context learning unlocked for diffusion models. In *NeurIPS*, pages 8542–8562, 2023. 2

[65] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *ICLR*, 2022. 2

[66] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, pages 8818–8826, 2019. 3

[67] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 5

[68] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 3

[69] Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. Stablenormal: Reducing diffusion variance for stable and sharp normal. *arXiv preprint arXiv:2406.16864*, 2024. 5, 6, 8

[70] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666, 2016. 2

[71] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4, 7, 13

[72] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024. 3

[73] Shaokun Zhang, Xiaobo Xia, Zhaoqing Wang, Ling-Hao Chen, Jiale Liu, Qingyun Wu, and Tongliang Liu. Ideal: Influence-driven selective annotations empower in-context learners in large language models. In *ICLR*, 2024. 2

[74] Sijie Zhao, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Muyao Niu, Xiaoyu Li, Wenbo Hu, and Ying Shan. Cv-vae: A compatible video vae for latent generative video models. *arXiv preprint arXiv:2405.20279*, 2024. 3

[75] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *arXiv preprint arXiv:2306.09305*, 2023. 3

[76] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 5, 14

[77] Le Zhuo, Ruoyi Du, Han Xiao, Yangguang Li, Dongyang Liu, Rongjie Huang, Wenze Liu, Lirui Zhao, Fu-Yun Wang, Zhanyu Ma, et al. Lumina-next: Making lumina-t2x stronger and faster with next-dit. *arXiv preprint arXiv:2406.18583*, 2024. 2, 3

11

# LaVin-DiT: Large Vision Diffusion Transformer

## Supplementary Material

## A. More Technical Details of LaVin-DiT

### A.1. Details of 3D RoPE

Recall that we represent task context and query as a unified sequence of frames, which is a 3D representation. Afterward, we extend RoPE from 1D to 3D format to capture the essential structure of visual data. Specifically, each token in an input sequence is associated with a 3D coordinate $(t, x, y)$, representing its position in temporal and spatial dimensions. The 3D RoPE encodes positional information by decomposing it into three separate 1D RoPEs along the temporal and spatial axes, allowing the model to capture relative positional relationships across all dimensions inherently.

Technically, for each axis $a \in \{t, x, y\}$, we define a rotation matrix $R_p^{(a)}$ that operates on a dedicated subspace of an embedding vector $z$. The embedding vector is partitioned accordingly: $z = [z^{(t)}, \; z^{(x)}, \; z^{(y)}]$, where each subvector $z^{(a)} \in \mathbb{R}^{d_a}$ corresponds to axis $a$ and $d = d_t + d_x + d_y$. The rotation matrix $R_p^{(a)}$ is constructed in a block-wise manner, rotating each pair of dimensions $(2i, 2i + 1)$ by an angle $\theta_{p,i}^{(a)} = p^{(a)} \cdot \omega_i^{(a)}$, where $\omega_i^{(a)} = \omega_{\text{base}}^{-2i/d_a}$ and $\omega_{\text{base}}$ is a predefined constant:

$$R_p^{(a)} = \begin{bmatrix} R_p^{(a,0)} & & \\ & \ddots & \\ & & R_p^{(a,d_a/2-1)} \end{bmatrix}, \quad \text{where} \quad (4)$$

$$R_p^{(a,i)} = \begin{bmatrix} \cos\left(\theta_{p,i}^{(a)}\right) & -\sin\left(\theta_{p,i}^{(a)}\right) \\ \sin\left(\theta_{p,i}^{(a)}\right) & \cos\left(\theta_{p,i}^{(a)}\right) \end{bmatrix}. \quad (5)$$

When computing self-attention, the rotated query $q$ and key $k$ are obtained by applying the rotation matrices: $q'^{(a)} = R_p^{(a)} q^{(a)}$ and $k'^{(a)} = R_p^{(a)} k^{(a)}$. The full rotated query and key are then $q' = [q'^{(t)}, \; q'^{(x)}, \; q'^{(y)}]$ and $k' = [k'^{(t)}, \; k'^{(x)}, \; k'^{(y)}]$. When computing the attention between tokens at positions $j$ and $k$, the dot product incorporates the rotations from all axes:

$$(q_j'^{\top})k_k' = \sum_{a \in \{t,x,y\}} \left(q^{(a)}\right)^{\top} R_j^{(a)\top} R_k^{(a)} k^{(a)}. \quad (6)$$

The key property of rotation matrices is that the product of two rotation matrices corresponds to a rotation by the difference of their angles:

$$R_j^{(a)\top} R_k^{(a)} = R_{j-k}^{(a)}, \quad (7)$$

where $R_{p-q}^{(a)}$ is the rotation matrix for the relative position $j^{(a)} - k^{(a)}$, constructed as:

$$R_{j-k}^{(a)} = \begin{bmatrix} R_{j-k}^{(a,0)} & & \\ & \ddots & \\ & & R_{j-k}^{(a,N_a-1)} \end{bmatrix}, \quad \text{where} \quad (8)$$

$$R_{j-k}^{(a,i)} = \begin{bmatrix} \cos\left(\Delta_{jk}^{(a)} \omega_i^{(a)}\right) & -\sin\left(\Delta_{jk}^{(a)} \omega_i^{(a)}\right) \\ \sin\left(\Delta_{jk}^{(a)} \omega_i^{(a)}\right) & \cos\left(\Delta_{jk}^{(a)} \omega_i^{(a)}\right) \end{bmatrix}, \quad (9)$$

$$\Delta_{jk}^{(a)} = j^{(a)} - k^{(a)}. \quad (10)$$

This block-wise matrix format explicitly shows that the attention score depends on the relative positions $j^{(a)} - k^{(a)}$ along each axis $a$.

### A.2. Algorithm Flows of LaVin-DiT

In this section, we present algorithm flows of the proposed LaVin-DiT. It is built upon the flow matching framework [36]. The training and inference procedures are provided in Algorithm 1 and Algorithm 2, respectively.

---

**Algorithm 1** LaVin-DiT Training Procedure

---

**Require:** ST-VAE encoder $\text{Enc}(\cdot)$, dataset $\mathcal{D} = \{x_i\}_{i=1}^K$, initialized parameters $\theta$ of vector field $v_\theta(z, t)$, total iterations $T$, learning rate $\eta$.

1: **for** $n = 1$ to $T$ **do**
2:      Sample $x \sim \mathcal{D}, \; c \sim \mathcal{D}$
3:      Compute latents: $z_0 \leftarrow \text{Enc}(x), \; z_c \leftarrow \text{Enc}(c)$
4:      Initialize random latent: $z_1 \sim \mathcal{N}(0, 1)$
5:      Sample time step: $t \sim \text{LogitNormal}(0, 1)$
6:      Interpolate: $z_t \leftarrow (1 - t)z_1 + tz_0$
7:      Target vector: $u \leftarrow z_0 - z_1$
8:      Predicted vector: $v \leftarrow v_\theta(z_t, z_c, t)$
9:      Compute loss: $\mathcal{L} \leftarrow \mathbb{E}[\|v - u\|_2^2]$
10:      Update parameters: $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}$
11: **end for**

---

**Training procedure.** As illustrated in Algorithm 1, the primary goal is to learn a vector field $v_\theta(z, t)$ that maps the latent space dynamics conditioned on the target latent $z_0$, the task context latent $z_c$, and a time step $t$. The training process iteratively refines the parameters $\theta$ to minimize the discrepancy between the predicted and ground-truth latent trajectories.

**Inference procedure.** This process, described in Algorithm 2, employs the learned vector field $v_\theta$ to sample in the

latent space. Starting with an initial latent $z_1 \sim \mathcal{N}(0,1)$, the method denoises iteratively using the Euler method.

---

**Algorithm 2** LaVin-DiT Inference Procedure

---

**Require:** Trained vector field $v_\theta(z,t)$, ST-VAE encoder $\text{Enc}(\cdot)$, ST-VAE decoder $\text{Dec}(\cdot)$, timesteps $N$, dataset $\mathcal{D} = \{x_i\}_{i=1}^K$.
1: Set step size $\Delta t \leftarrow \frac{1}{N}$, initialize $t^{(N)} \leftarrow 1$
2: Sample initial latent: $z_1 \sim \mathcal{N}(0,1)$
3: Encode condition: $z_c \leftarrow \text{Enc}(c)$, $c \sim \mathcal{D}$
4: **for** $k = N$ down to 1 **do**
5:     Update time: $t^{(k-1)} \leftarrow t^{(k)} - \Delta t$
6:     Compute vector field: $v^{(k)} \leftarrow v_\theta(z^{(k)}, z_c, t^{(k)})$
7:     Update latent: $z^{(k-1)} \leftarrow z^{(k)} - \Delta t \cdot v^{(k)}$
8: **end for**
9: Decode sample: $\hat{y} \leftarrow \text{Dec}(z_0)$

---

Table 3. Configurations of LaVin-DiT with different numbers of parameters.

|  | **LaVin-DiT** | | |
|---|---|---|---|
|  | **0.1B** | **1.0B** | **3.4B** |
| **Latent channels** | 16 | 16 | 16 |
| **Patch size** | $2 \times 2$ | $2 \times 2$ | $2 \times 2$ |
| **Hidden channels** | 512 | 1024 | 2304 |
| **Num. layers** | 12 | 28 | 22 |
| **Num. heads** | 8 | 16 | 32 |
| **K.V. groups** | - | - | 4 |
| **Drop path** | 0.0 | 0.1 | 0.1 |
| **Uncond. ratio** | 0.1 | 0.1 | 0.1 |
| **Grad. clip** | 1.0 | 1.0 | 1.0 |
| **EMA moment.** | 0.9999 | 0.9999 | 0.9999 |
| **Extra norm.** | - | S-Norm. | S-Norm. |
| **Position embed.** | 3D-RoPE | 3D-RoPE | 3D-RoPE |

## B. Supplementary Experimental Settings

### B.1. Large-Scale Multi-Task Dataset Composition

Recall that we build a large-scale multi-task dataset to unify diverse computer vision tasks. We integrate multiple public image-level and video-level task benchmarks into a large-scale dataset for training. Details are listed in Table 4.

### B.2. Evaluation Metrics

In this work, we provide quantitative results for 10 tasks (The others are presented with visualization results). Here we introduce the evaluation metrics for these 10 tasks.

**Colorization.** We randomly sample 1,000 images from ImageNet-1K validation set [18] and convert them into grayscale. We adopt LPIPS [71] and mean squared error (MSE) as metrics.

**Inpainting.** We randomly sample 1,000 images from ImageNet-1K validation set [18] and mask out a $128 \times 128$ region for each image. We adopt the LPIPS [71] and Frechet Inception Distance (FID) as metrics.

**Depth Estimation.** We evaluate our model on NYUv2 test set [52], including 654 images. Following the protocol of affine-invariant depth evaluation [46], we first align the prediction to the ground truth with the least squares fitting. Afterwards, we adopt Absolute Mean Relative Error (AbsRel) and Mean Squared Error (MSE) as metrics.

**Surface Normal Estimation.** We evaluate our model on NYUv2 test set [52]. Following the protocol used in [4], we calculate the angular error between the prediction and the ground-truth normal maps and use the mean angular error as the metric.

**Depth-to-Image Generation.** We adopt all samples in the NYUv2 dataset [52], including 1,449 images. Given the pseudo label generated via Depth-anything V2 or Stable-Normal (turbo), we generate the corresponding RGB image and use the LPIPS [71] and Frechet Inception Distance (FID) as metrics.

**Normal-to-Image Generation.** The metrics are the same those in Depth-to-Image Generation.

**Single Object Detection.** We evaluate the model on the Pascal-5i dataset [50] and adopt the mean intersection-over-union (mIoU) as the metric.

**Foreground Segmentation.** We evaluate our model on the Pascal-5i dataset [50], including 4 different test splits. Following the protocol in [8], we extract binary masks from our predictions and report the mIoU.

**Deraining.** We randomly sample 1,000 images from ImageNet-1K validation set [18] and apply the raining filter on them. We adopt the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) as metrics.

**De-motion Blur.** We randomly sample 1,000 images from the ImageNet-1K validation set [18] and apply motion blur on these images. We adopt the PSNR and SSIM as metrics.

### B.3. Architecture Details of LaVin-DiT

Here we detail the architecture of the LaVin-DiT models. Table 3 outlines the configurations for three parameter scales: 0.1B, 1.0B, and 3.4B. Each configuration is characterized by key architectural hyperparameters, including the number of latent channels, patch size, hidden channels, and the number of layers. Additionally, the configurations specify the number of attention heads, key-value groups, drop path rates, and unconditional ratios. To further enhance model training, we incorporate advanced techniques such as gradient clipping and the Exponential Moving Average (EMA). All models utilize 3D-RoPE to ensure consistent spatial and temporal encoding across scales. For large mod-

Table 4. Summary of the large-scale multi-task dataset used in LaVin-DiT, including the number of examples and annotation types for each component dataset. Tasks range from visual understanding and generation.

| Task | Dataset | Number of Samples | Annotation Type |
|---|---|---|---|
| Single Object Detection | COCO 2017 train [34] | 117,266 | Ground Truth |
| | Object365 train [51] | 1,728,778 | Ground Truth |
| Instance Segmentation | COCO 2017 train [34] | 117,266 | Ground Truth |
| | ADE20K train+val [76] | 19,020 | Ground Truth |
| | Cityscapes train+val [16] | 3,457 | Ground Truth |
| Panoptic Segmentation | COCO 2017 train [34] | 117,266 | Ground Truth |
| | ADE20K train+val [76] | 19,020 | Ground Truth |
| | Cityscapes train+val [16] | 3,457 | Ground Truth |
| Pose Estimation | COCO 2017 train [34] | 64,115 | Ground Truth |
| Pose-to-Image Generation | COCO 2017 train [34] | 64,115 | Ground Truth |
| Depth Estimation | ImageNet1K train [18] | 1,281,167 | Depth-anything V2 |
| Depth-to-Image Generation | ImageNet1K train [18] | 1,281,167 | Depth-anything V2 |
| Surface Normal Estimation | COCO 2017 train [34] | 117,266 | Stable-Normal (turbo) |
| | ADE20K train+val [76] | 19,020 | Stable-Normal (turbo) |
| | Cityscapes train+val [16] | 3,457 | Stable-Normal (turbo) |
| Normal-to-Image Generation | COCO 2017 train [34] | 117,266 | Stable-Normal (turbo) |
| | ADE20K train+val [76] | 19,020 | Stable-Normal (turbo) |
| | Cityscapes train+val [16] | 3,457 | Stable-Normal (turbo) |
| Edge Detection | ImageNet1K [18] train | 1,281,167 | Canny (OpenCV) |
| | COCO 2017 train [34] | 117,266 | Canny (OpenCV) |
| Inpainting | ImageNet1K train [18] | 1,281,167 | Crop (OpenCV) |
| | COCO 2017 train [34] | 117,266 | Crop (OpenCV) |
| Colorization | ImageNet1K train [18] | 1,281,167 | Grayscale (OpenCV) |
| | COCO 2017 train [34] | 117,266 | Grayscale (OpenCV) |
| De-glass Blur | ImageNet1K train [18] | 1,281,167 | Albumentations |
| | COCO 2017 train [34] | 117,266 | Albumentations |
| De-motion Blur | ImageNet1K train [18] | 1,281,167 | Albumentations |
| | COCO 2017 train [34] | 117,266 | Albumentations |
| De-raining | ImageNet1K train [18] | 1,281,167 | Albumentations |
| | COCO 2017 train [34] | 117,266 | Albumentations |
| Frame Prediction | UCF101 train [53] | 7,629 | N/A |
| | Kinetic 700 train+val [28] | 570,465 | N/A |
| | Kubric train [25] | 48,689 | N/A |
| Video Depth Estimation | Kubric train [25] | 48,689 | Ground Truth |
| Depth-to-Video Generation | Kubric train [25] | 48,689 | Ground Truth |
| Video Surface Normal Estimation | Kubric train [25] | 48,689 | Ground Truth |
| Normal-to-Video Generation | Kubric train [25] | 48,689 | Ground Truth |
| Video Optical Flow Estimation | Kubric train [25] | 48,689 | Ground Truth |
| Video Instance Segmentation | Kubric train [25] | 48,689 | Ground Truth |

els, we employ sandwich normalization to improve training stability.

## C. Supplementary Qualitative Results

We show more visualization results for each task, including object detection (Figure 8), foreground segmentation (Figure 9), panoptic segmentation (Figure 10), pose estimation (Figure 11), pose-to-image generation (Figure 12), depth estimation (Figure 13), depth-to-image generation (Figure 14), surface normal estimation (Figure 15), normal-to-image generation (Figure 16), edge detection (Figure 17), inpainting (Figure 18), colorization (Figure 19), de-glass blur (Figure 20), de-motion blur (Figure 21), de-raining (Figure 22), frame prediction (Figure 23), video depth estimation (Figure 24), depth-to-video generation (Figure 25),
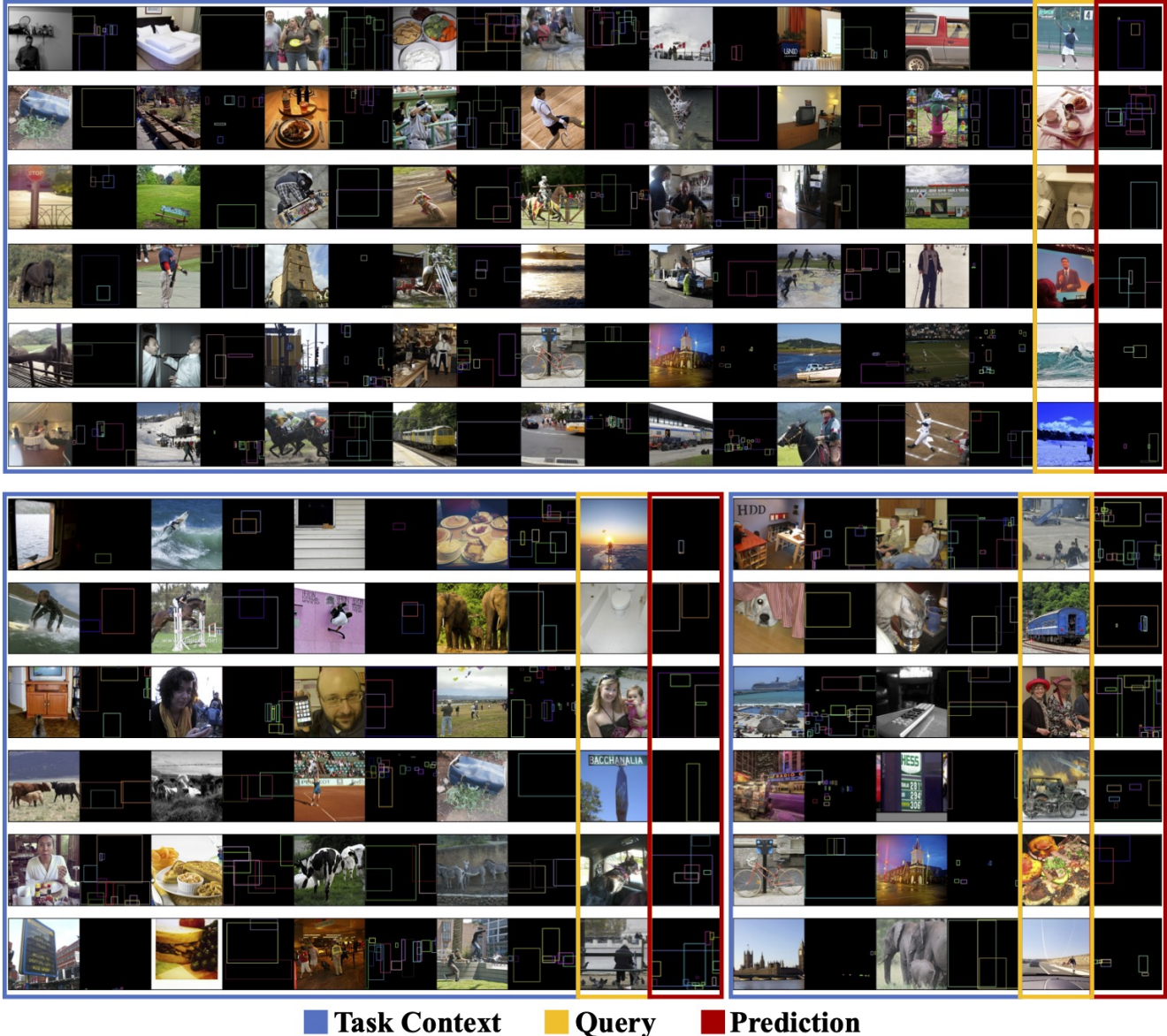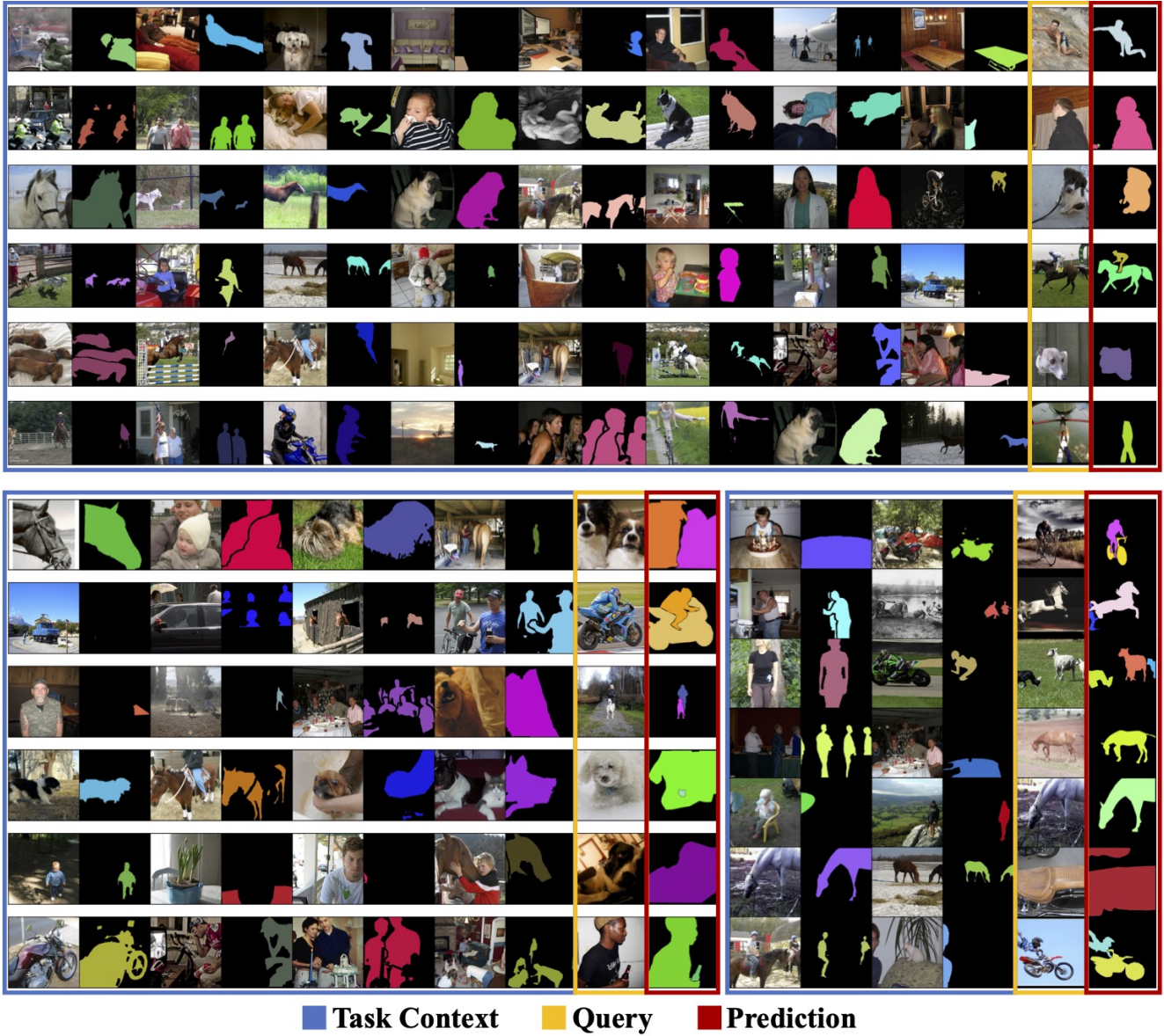
**Figure 8. Qualitative results on object detection.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*
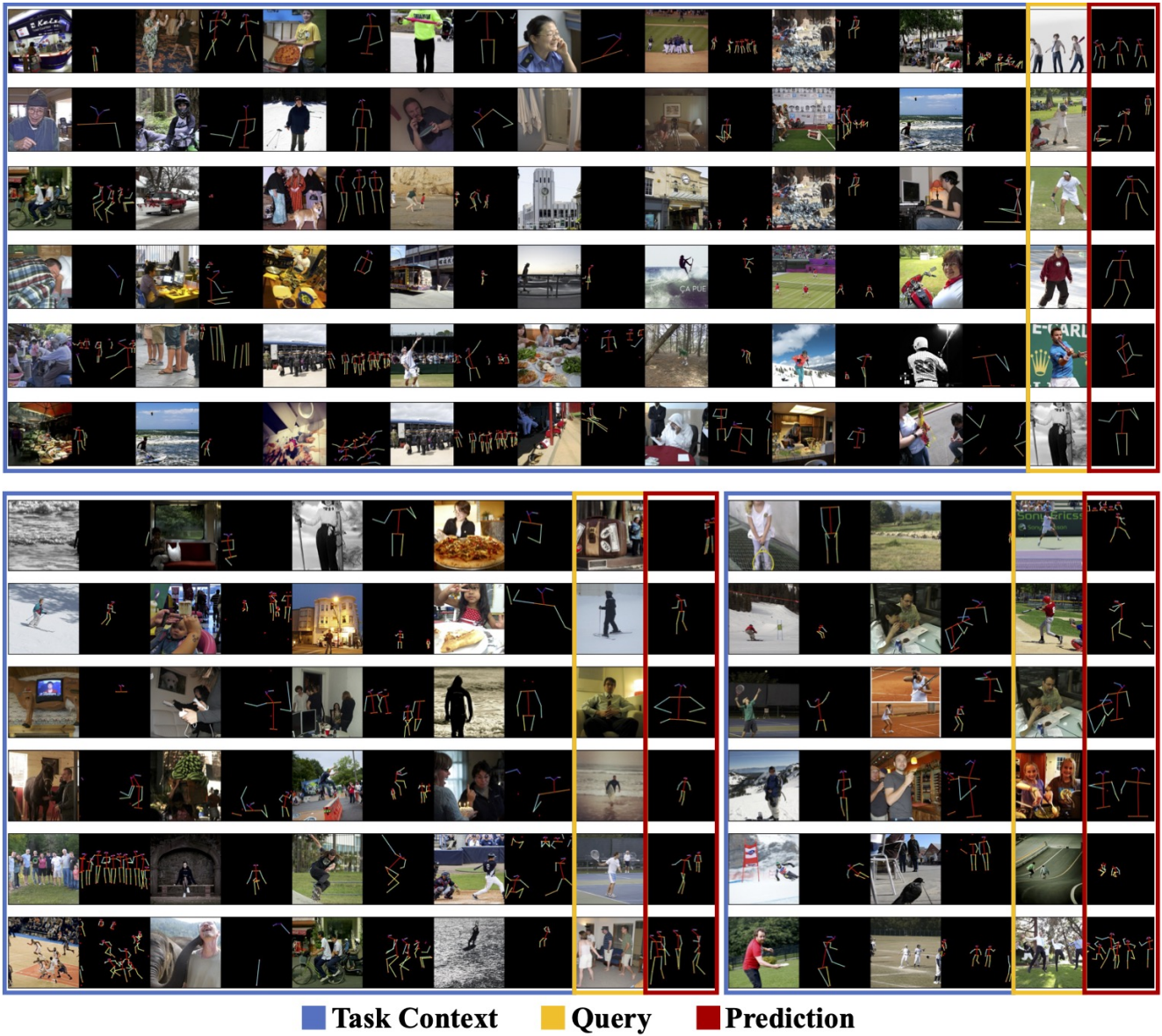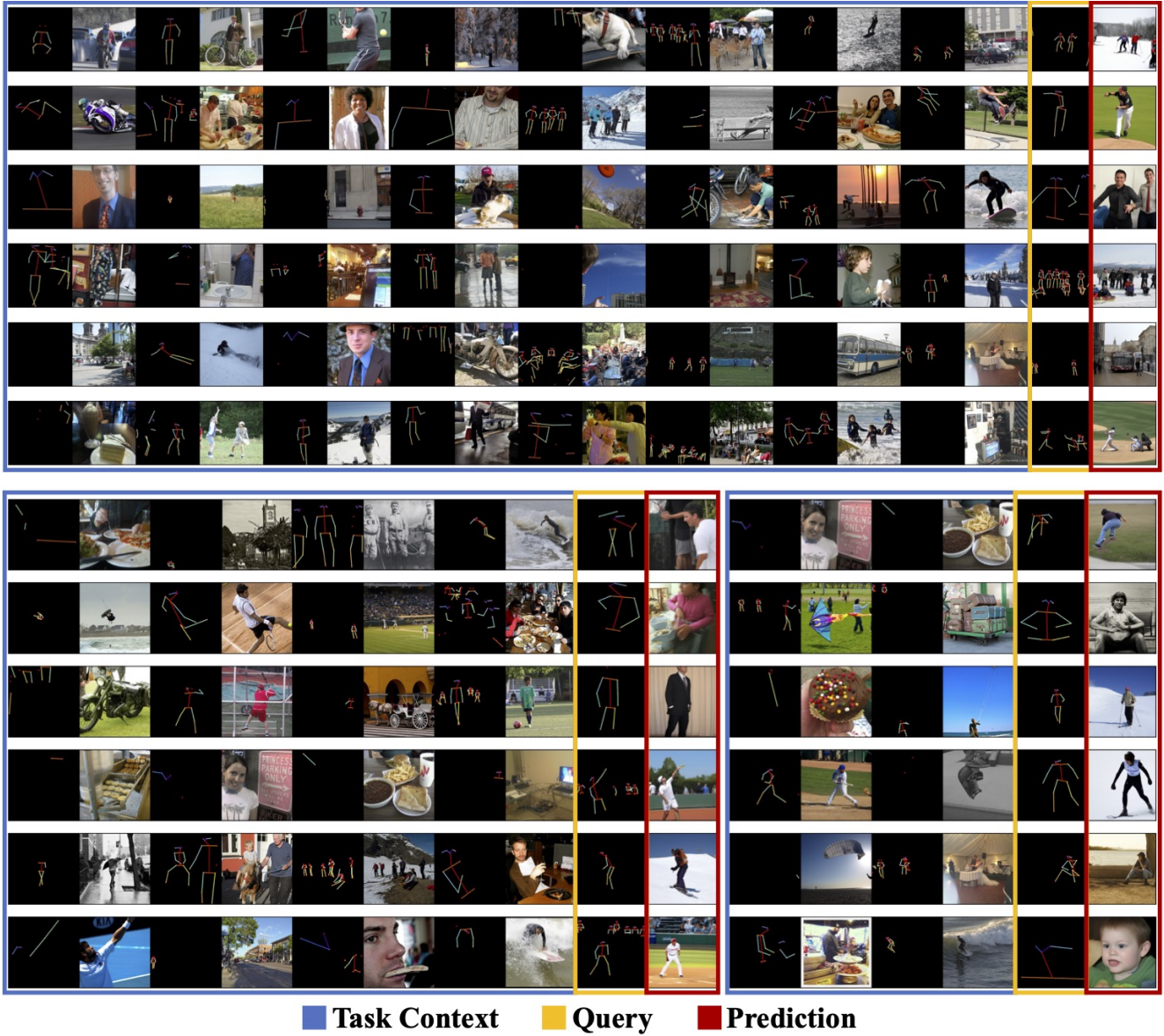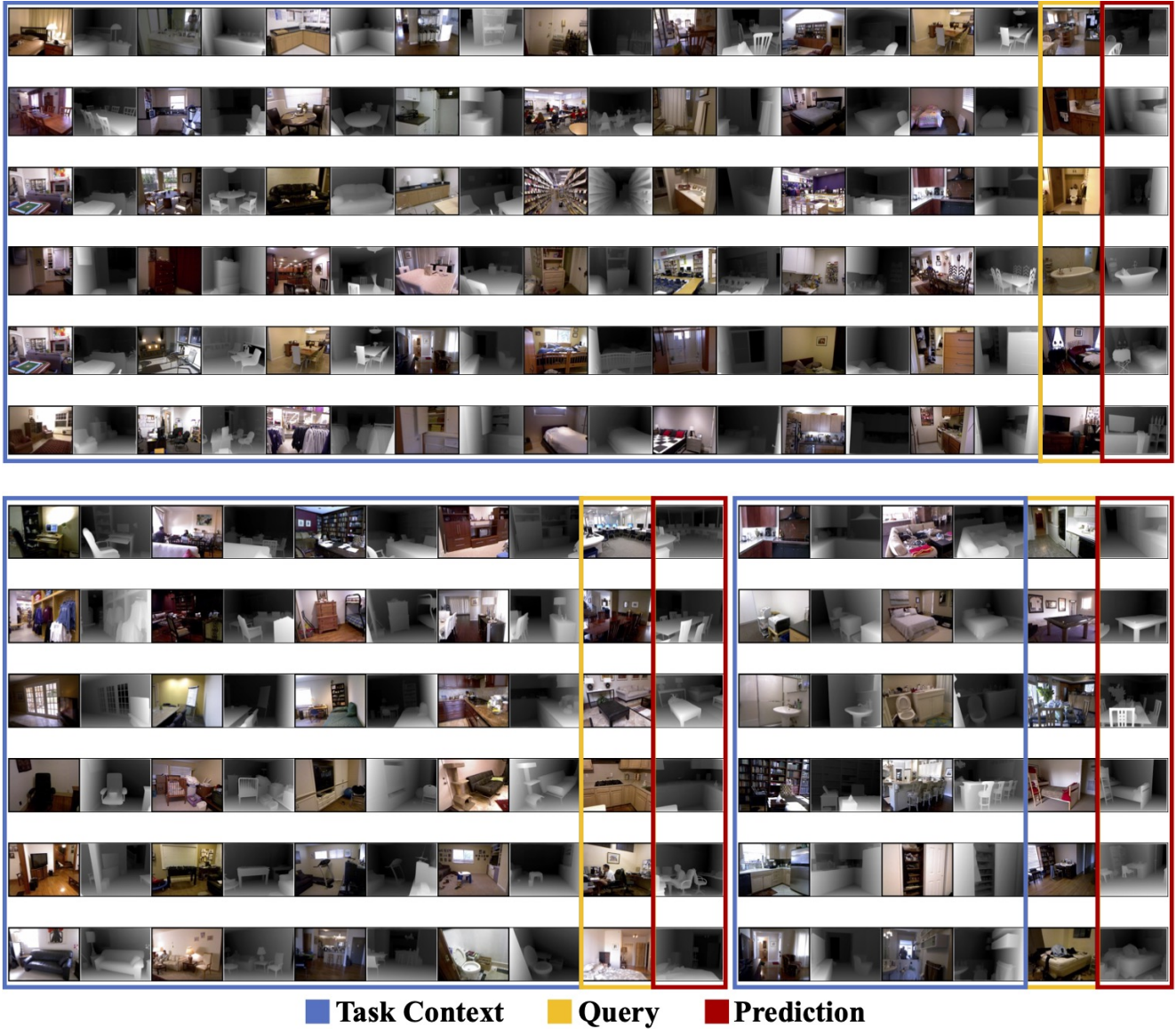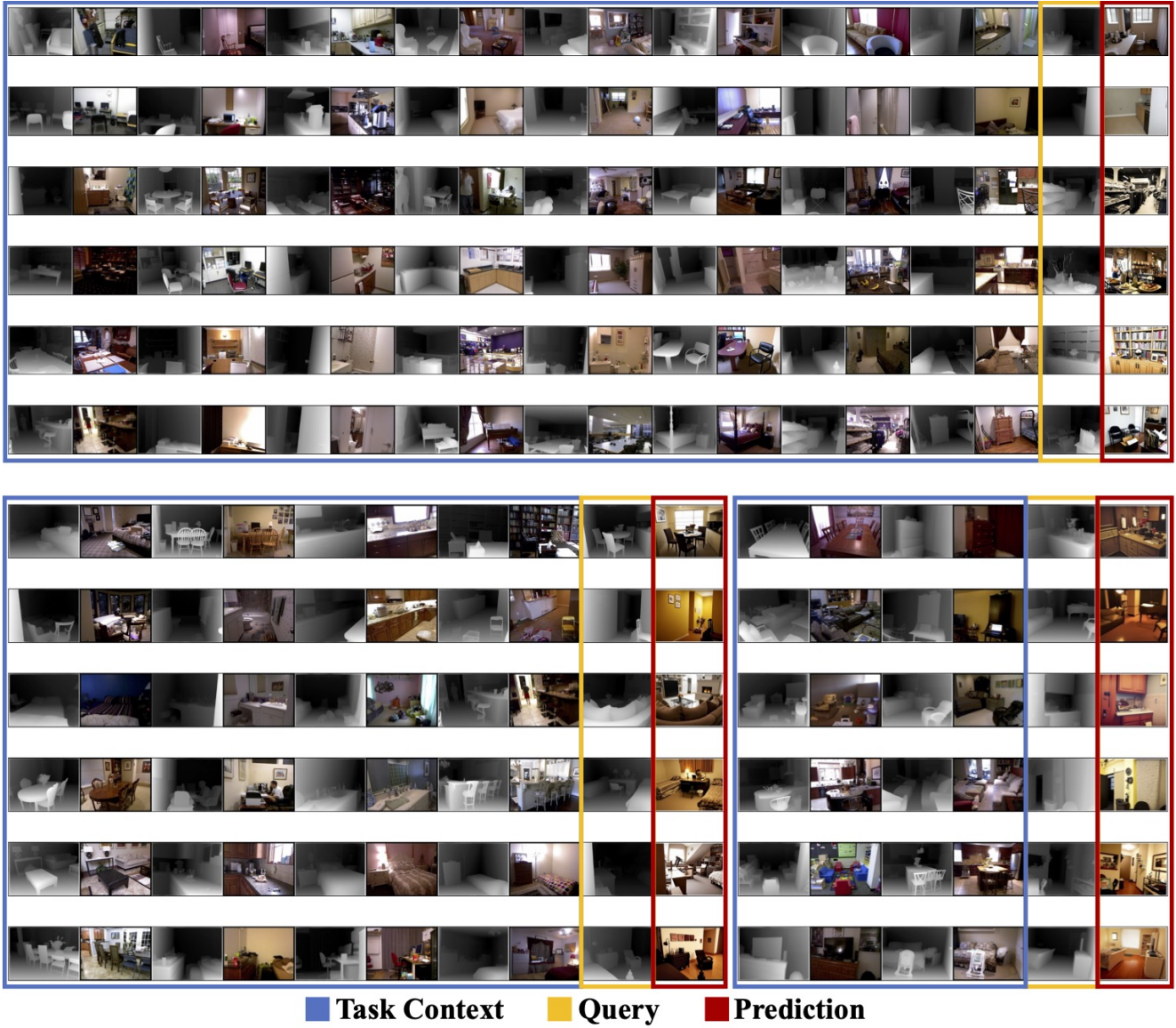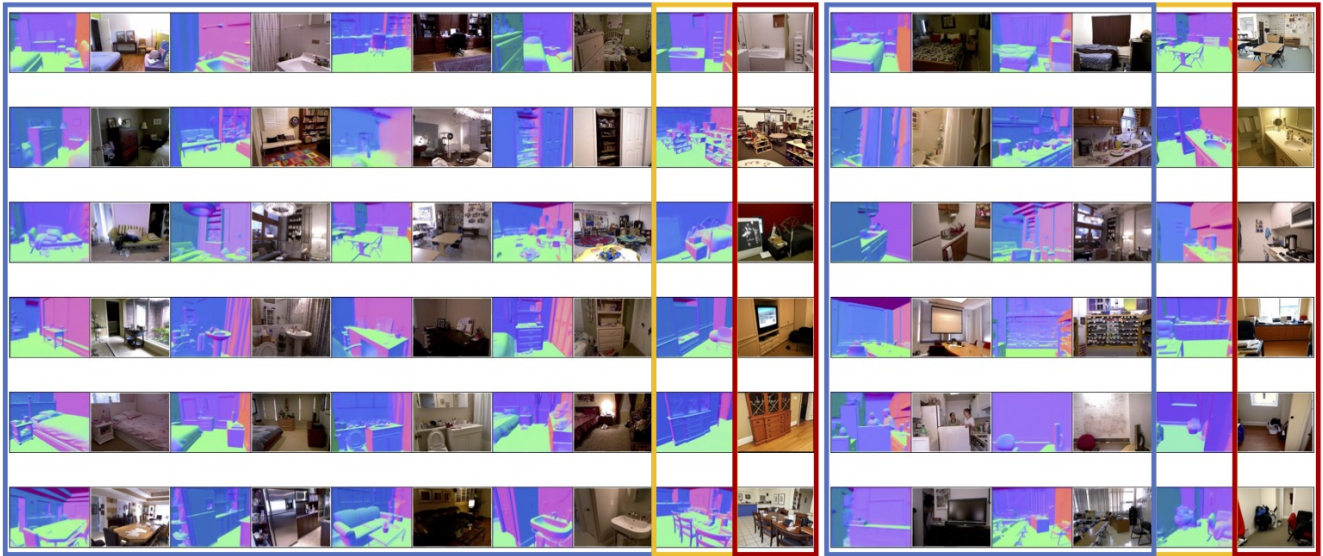
video surface normal estimation (Figure 26), normal-to-video generation (Figure 27), video optical flow estimation (Figure 28), and video instance segmentation (Figure 29).

## D. Potential Applications

LaVin-DiT opens transformative possibilities for tackling open-world computer vision challenges by unifying diverse vision tasks within a single generative framework. For instance, it can seamlessly generalize across tasks such as text-to-image generation, text-to-video generation, video understanding, 3D reconstruction (Figure 30), and 2D/3D visual editing without supervised fine-tuning. By leveraging its spatial-temporal variational autoencoder and joint diffusion transformer, LaVin-DiT excels at capturing the complexity of high-dimensional visual data while maintaining task-specific alignment through in-context learning. This capability positions LaVin-DiT as a foundation model capable of addressing dynamic realistic vision problems, including autonomous driving perception, robotic scene understanding, and interactive AI systems in mixed-reality environments, significantly advancing the frontier of adaptable and scalable AI systems.

Figure 9. **Qualitative results on foreground segmentation.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*
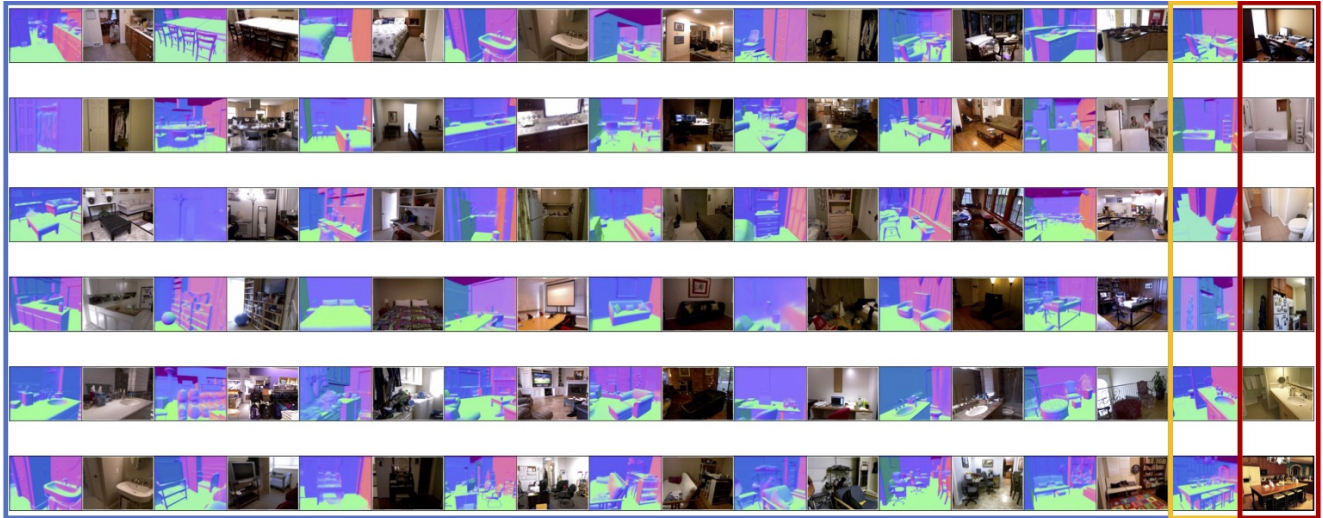
Figure 10. **Qualitative results on panoptic segmentation.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*
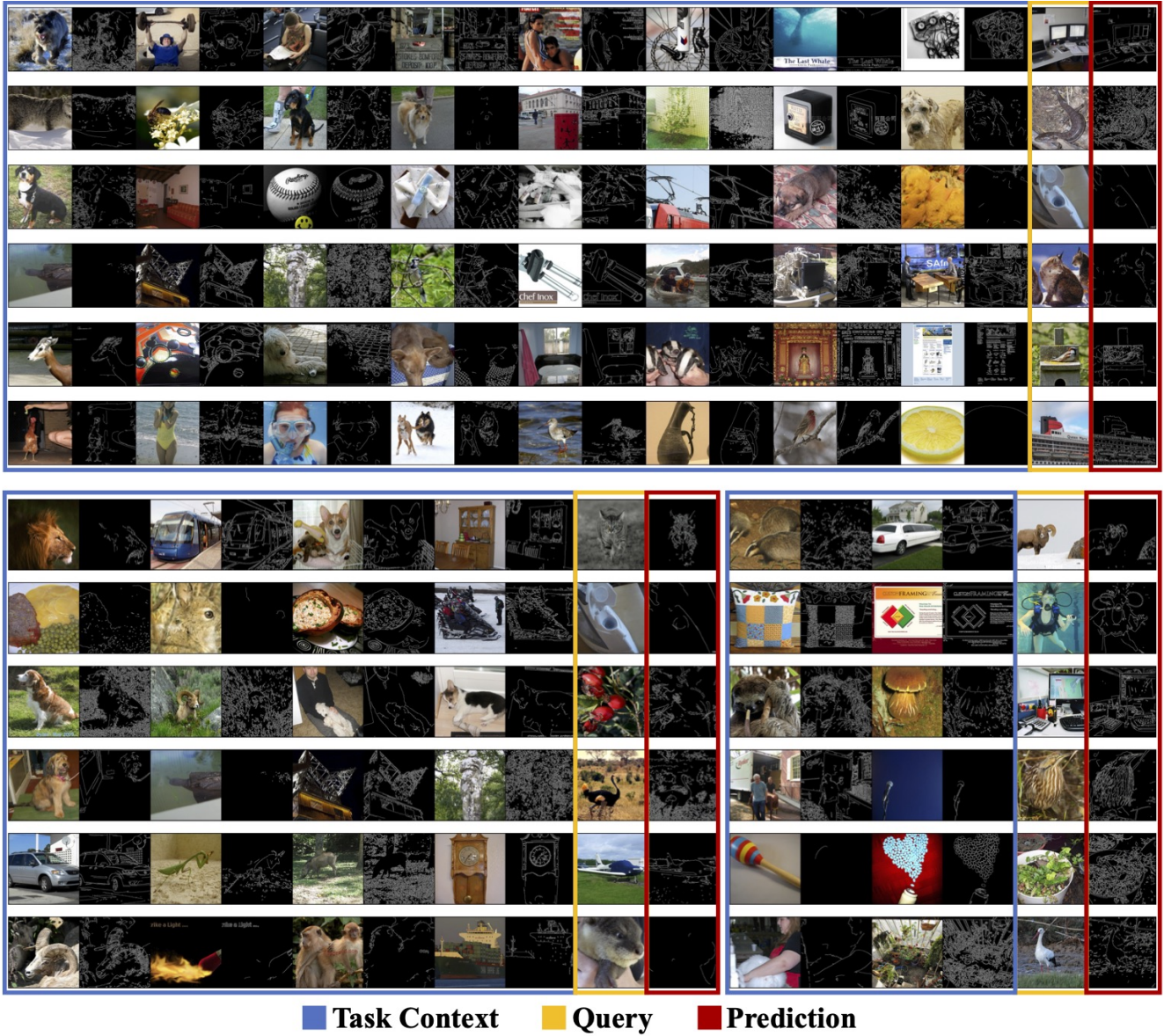
Figure 11. **Qualitative results on pose estimation.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*

Figure 12. **Qualitative results on pose-to-image generation.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*
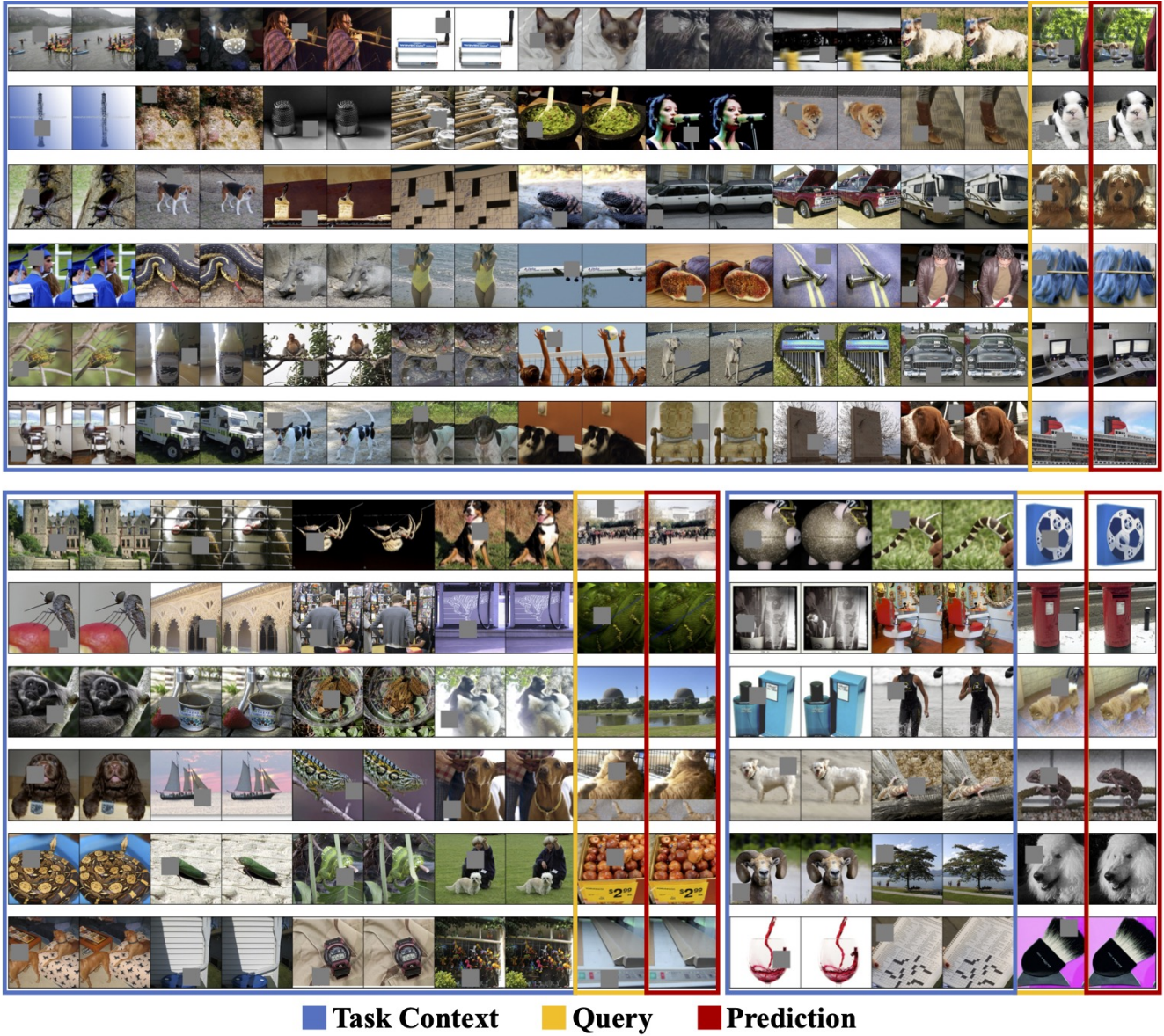
**Task Context** ■ **Query** ■ **Prediction** ■

Figure 13. **Qualitative results on depth estimation.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*
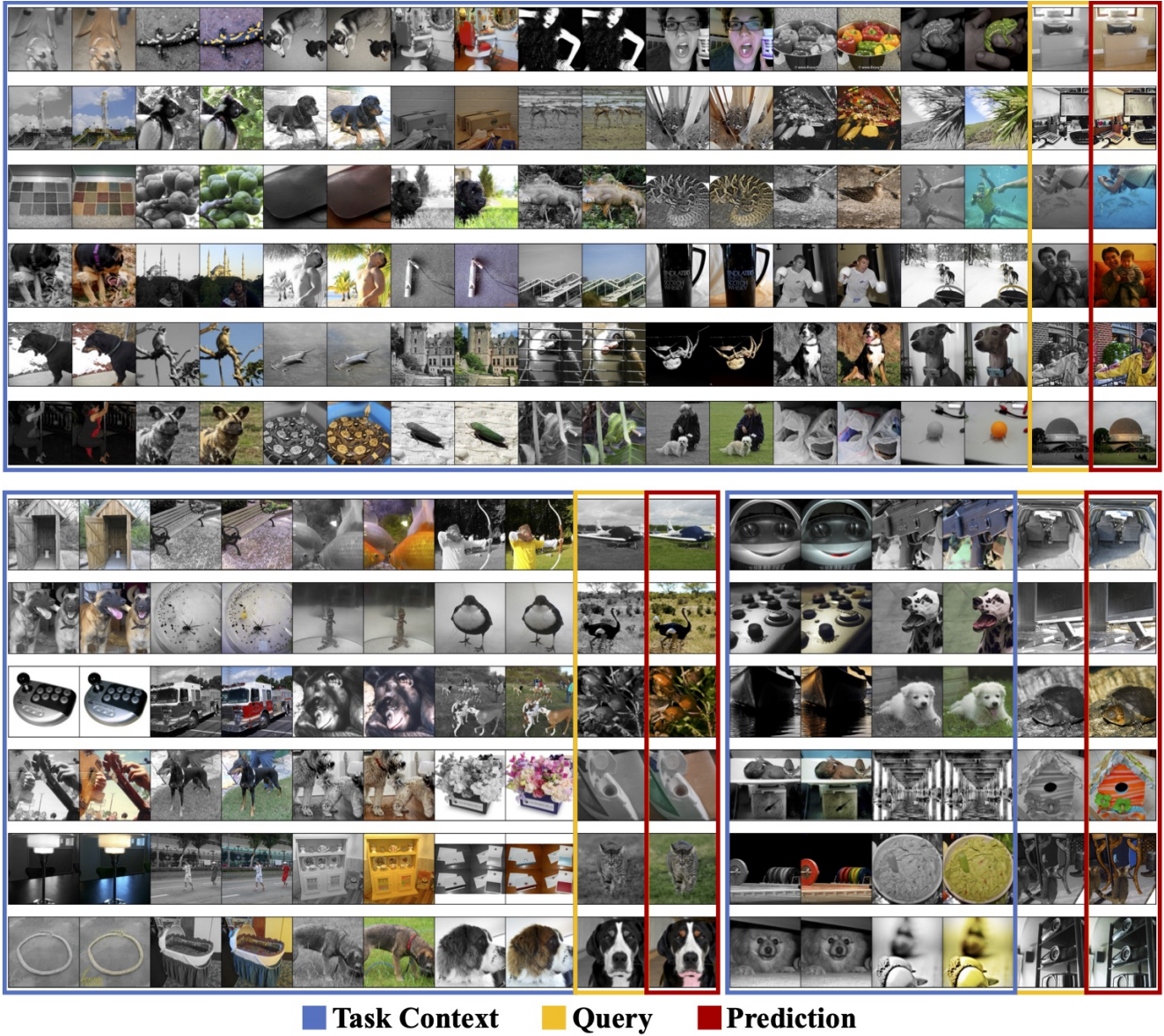
■ **Task Context** ■ **Query** ■ **Prediction**

Figure 14. **Qualitative results on depth-to-image generation.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*

**Figure 15. Qualitative results on surface normal estimation.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*
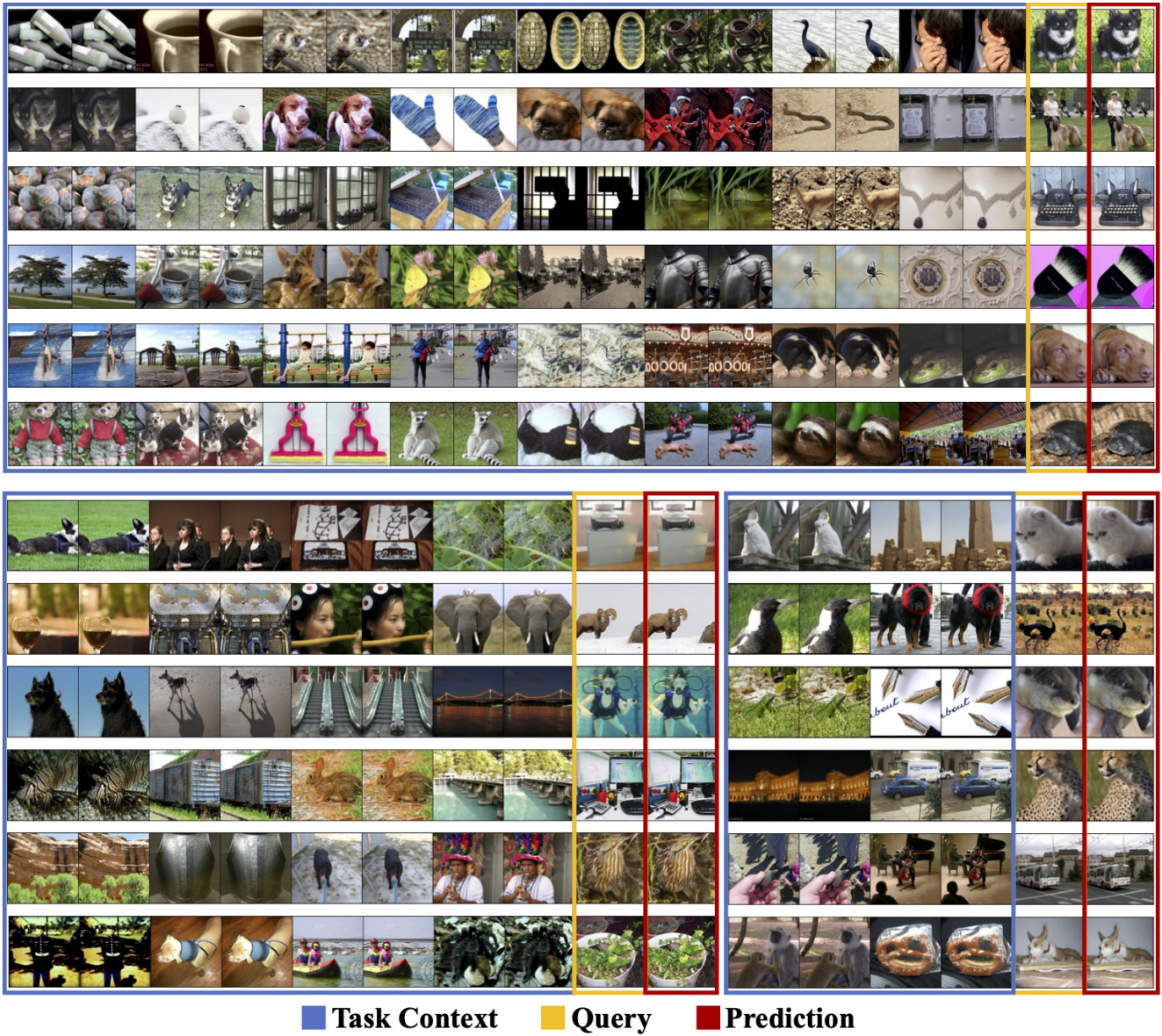
**Task Context**    **Query**    **Prediction**

Figure 16. **Qualitative results on normal-to-image generation.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*
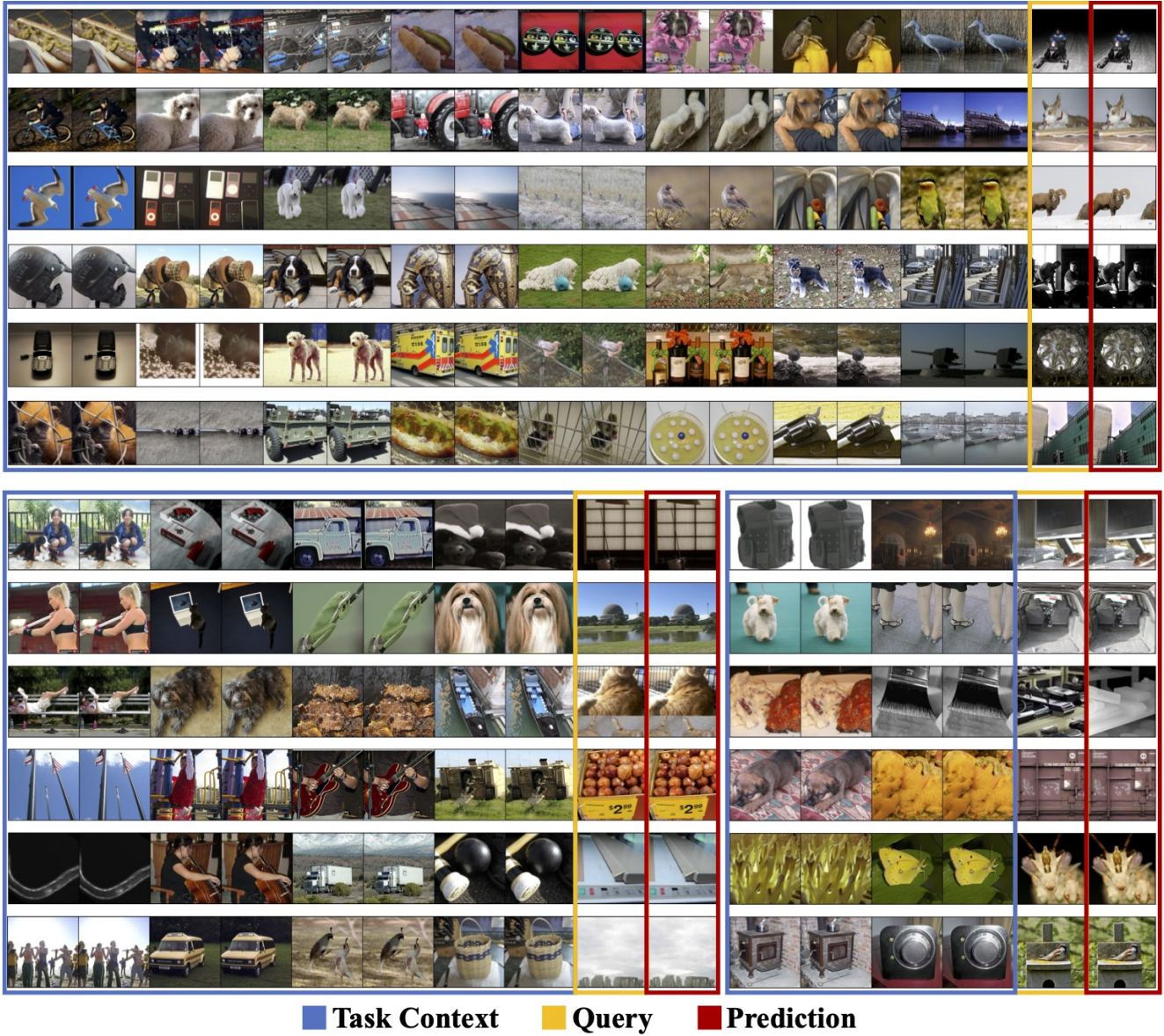
Figure 17. **Qualitative results on edge detection.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*

Figure 18. **Qualitative results on inpainting.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*
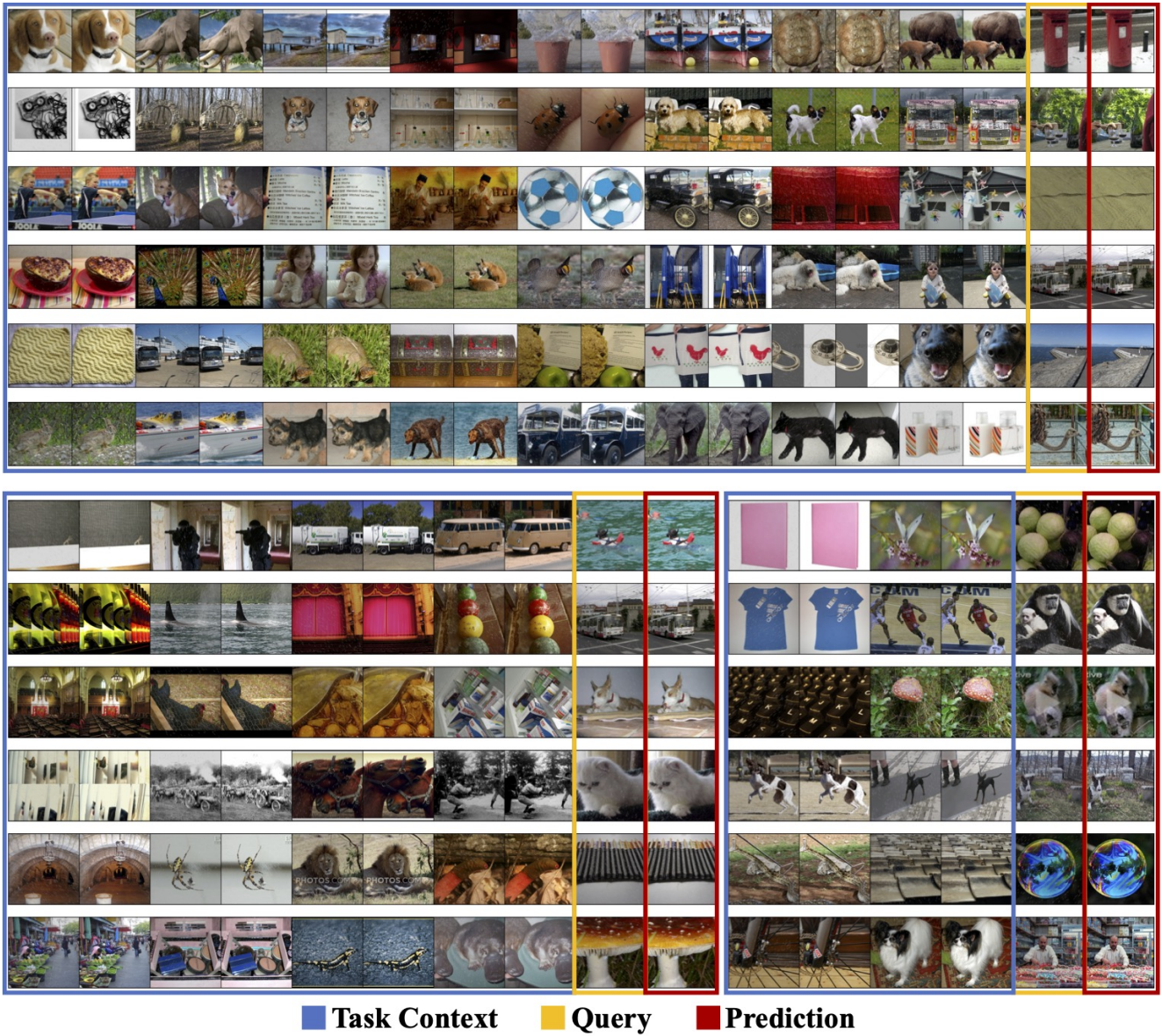
Figure 19. **Qualitative results on image colorization.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*

**Task Context** ■ **Query** ■ **Prediction** ■

Figure 20. **Qualitative results on de-glass blur.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*
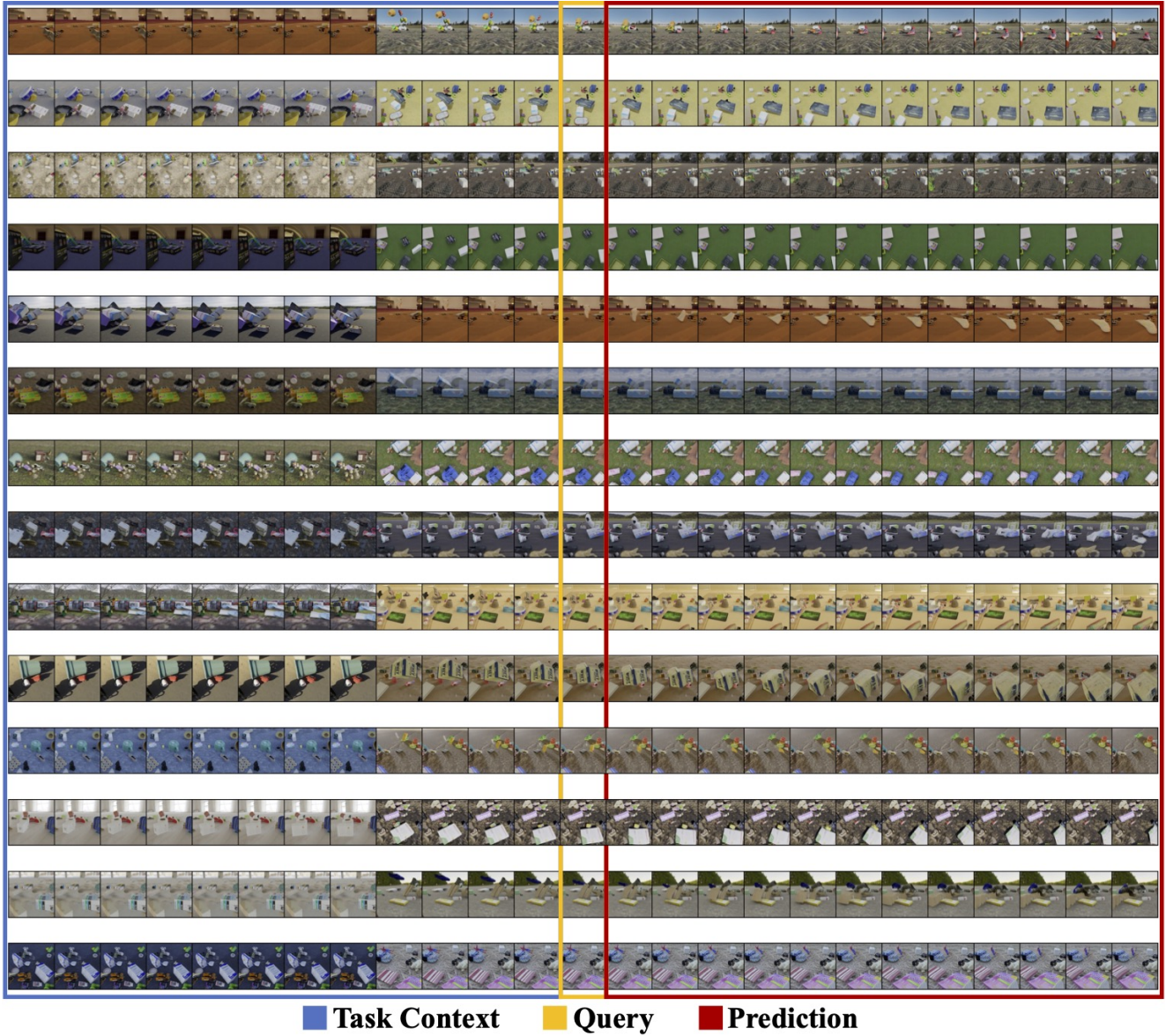
27

Figure 21. **Qualitative results on de-motion blur.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*

Figure 22. **Qualitative results on de-raining.** Each row contains a sequence of images interleaved with annotations, followed by a query. The last image is predicted by the model (marked in red). *Best viewed in color.*

Figure 23. **Qualitative results on frame prediction.** Each row includes a video sequence with a series of target frames as task context (marked in blue), followed by a query frame (marked in yellow). A set of frames in the red box indicates the model's predictions. Due to the length of the sequence, a portion of the task context is hidden. *Best viewed in color.*
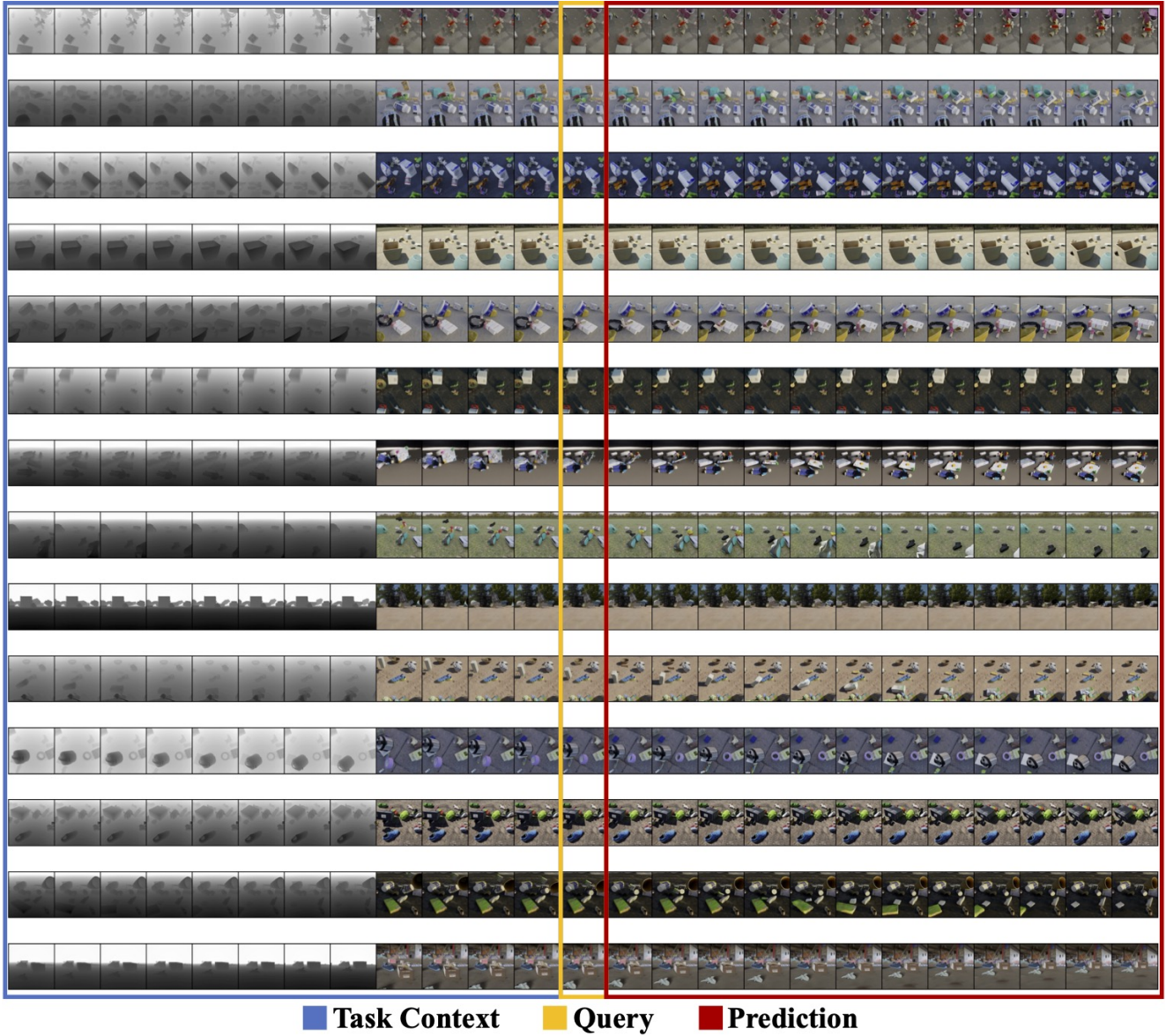
**Task Context**    **Query**    **Prediction**

Figure 24. **Qualitative results on video depth estimation.** Each row includes a video sequence with a series of target frames as task context (marked in blue), followed by a query frame (marked in yellow). A set of frames in the red box indicates the model's predictions. Due to the length of the sequence, a portion of the task context is hidden. *Best viewed in color.*
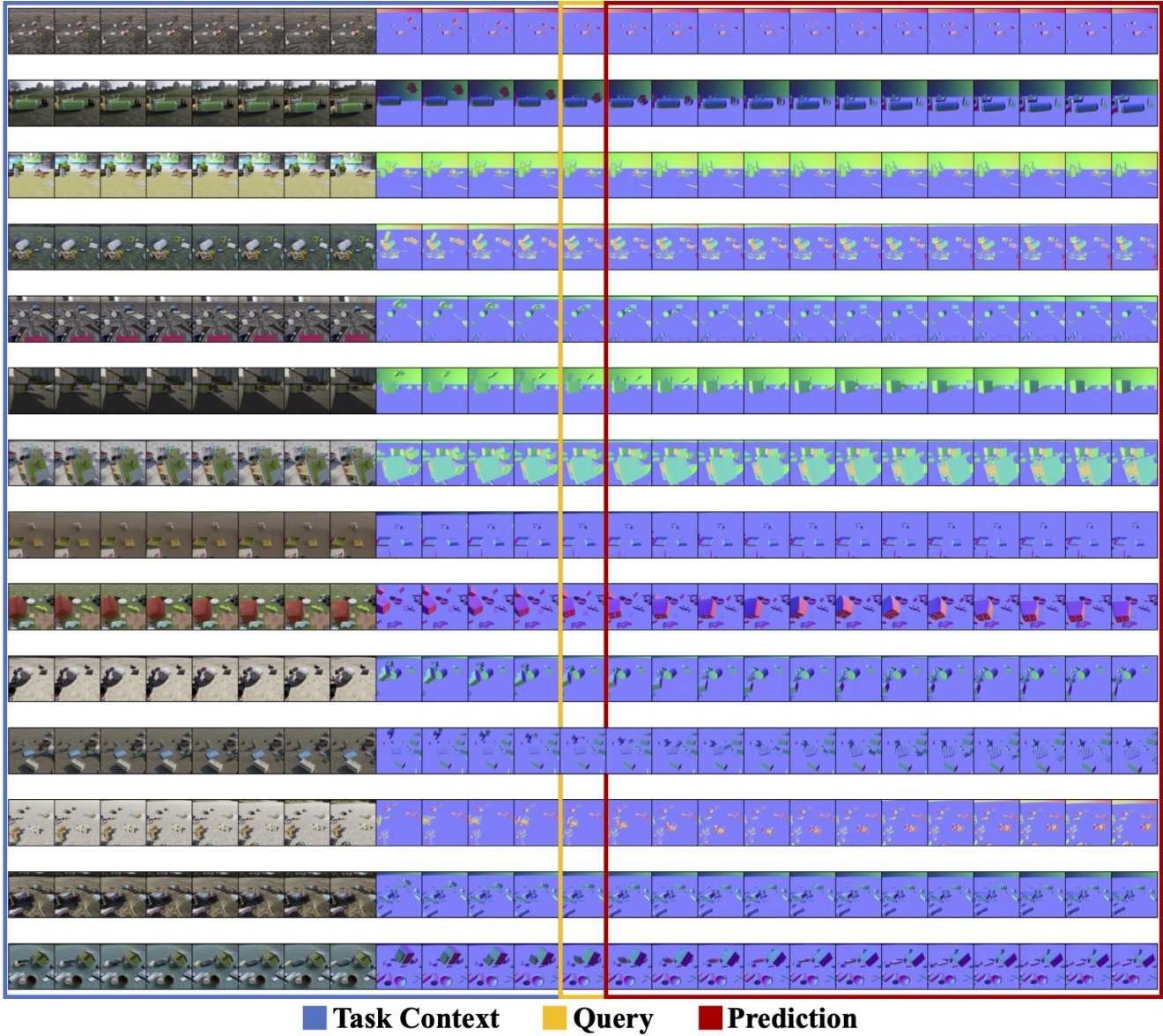
**Task Context**　　**Query**　　**Prediction**

Figure 25. **Qualitative results on depth-to-video generation.** Each row includes a video sequence with a series of target frames as task context (marked in blue), followed by a query frame (marked in yellow). A set of frames in the red box indicates the model's predictions. Due to the length of the sequence, a portion of the task context is hidden. *Best viewed in color.*
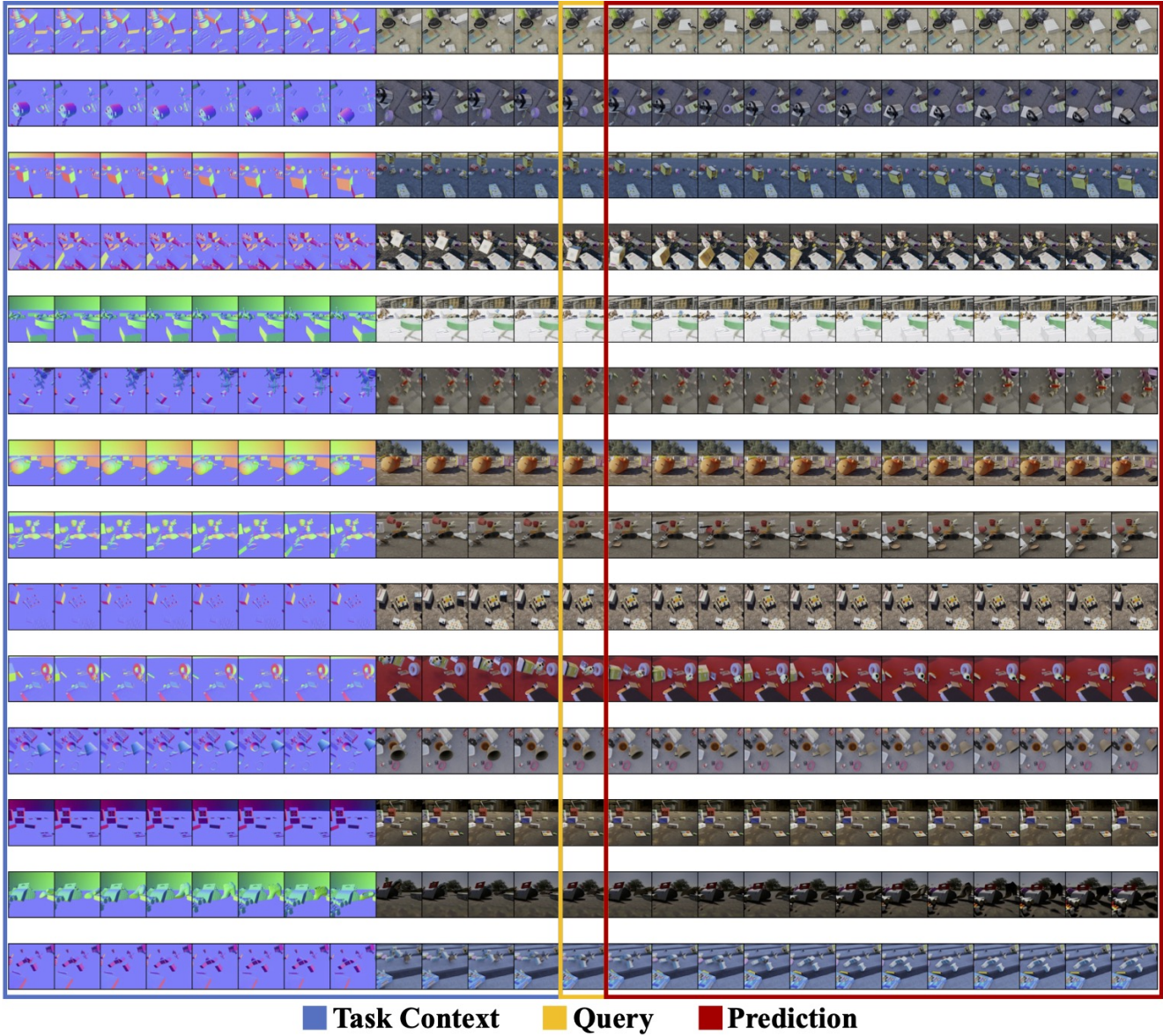
Figure 26. **Qualitative results on video surface normal estimation.** Each row includes a video sequence with a series of target frames as task context (marked in blue), followed by a query frame (marked in yellow). A set of frames in the red box indicates the model's predictions. Due to the length of the sequence, a portion of the task context is hidden. *Best viewed in color.*

Figure 27. **Qualitative results on normal-to-video generation.** Each row includes a video sequence with a series of target frames as task context (marked in blue), followed by a query frame (marked in yellow). A set of frames in the red box indicates the model's predictions. Due to the length of the sequence, a portion of the task context is hidden. *Best viewed in color.*
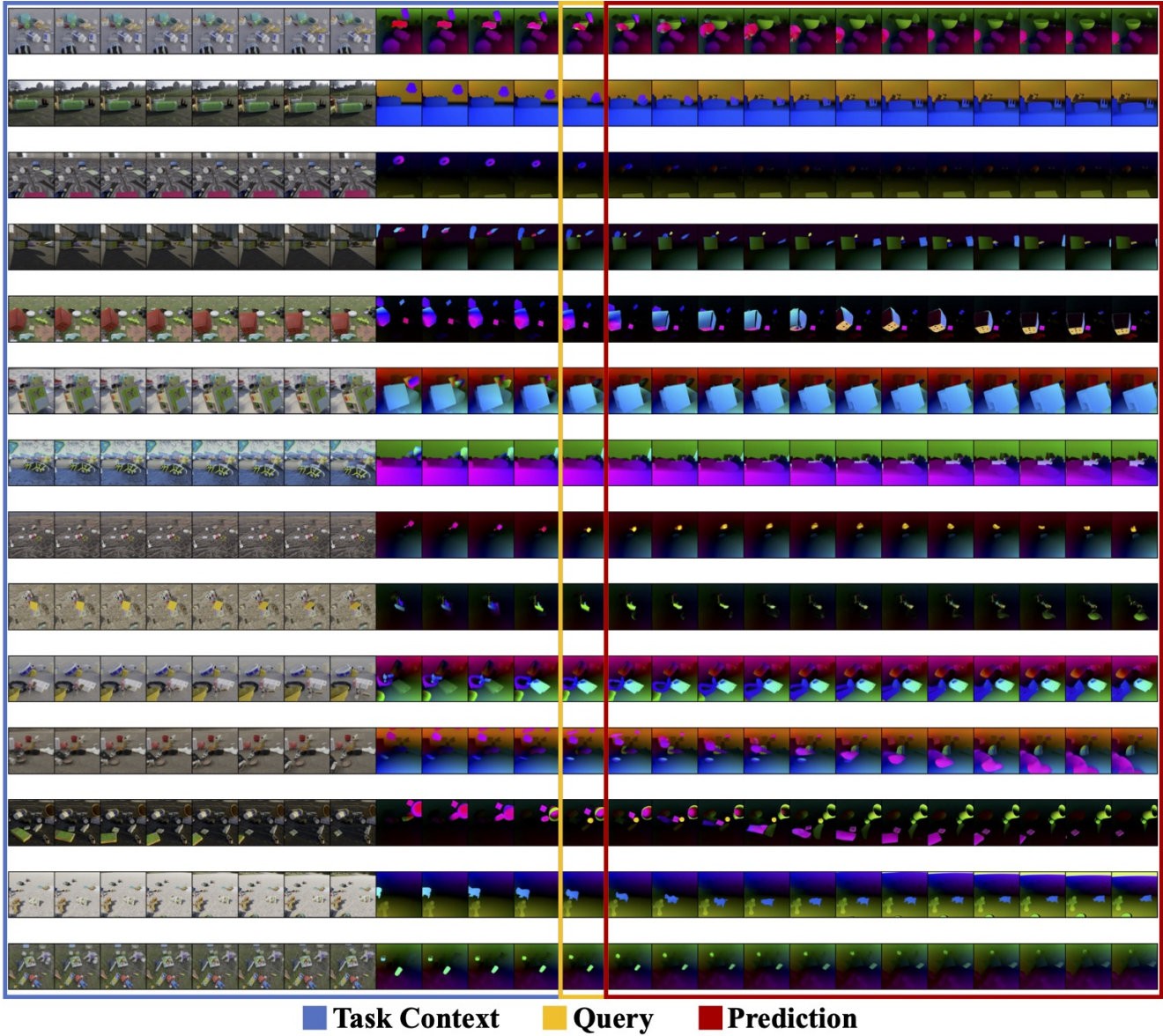
Figure 28. **Qualitative results on optical flow estimation.** Each row includes a video sequence with a series of target frames as task context (marked in blue), followed by a query frame (marked in yellow). A set of frames in the red box indicates the model's predictions. Due to the length of the sequence, a portion of the task context is hidden. *Best viewed in color.*

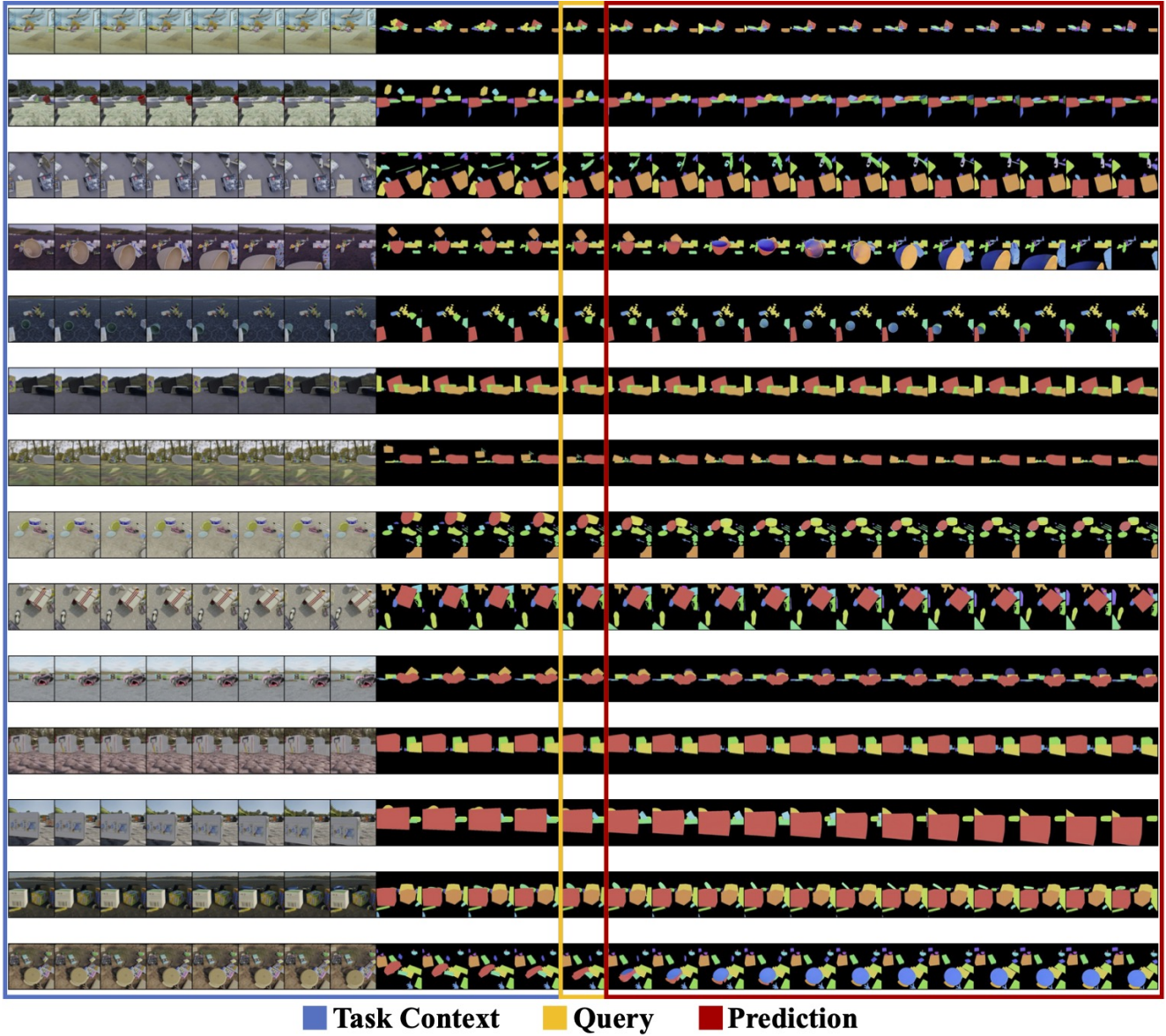**Task Context**  **Query**  **Prediction**

Figure 29. **Qualitative results on video instance segmentation.** Each row includes a video sequence with a series of target frames as task context (marked in blue), followed by a query frame (marked in yellow). A set of frames in the red box indicates the model's predictions. Due to the length of the sequence, a portion of the task context is hidden. *Best viewed in color.*

Figure 30. **Potential application of single-view scene reconstruction.** Given an RGB image and predicted depth map, we lift this image into a 3D space. We illustrate three views of this scene. *Best viewed in color.*