Few-Shot Learning in Video Diffusion Models

Pablo Acuaviva [1] **Aram Davtyan** [1] **Mariam Hassan** [2] **Sebastian Stapf** [1] **Ahmad Rahimi** [2] **Alexandre Alahi** [2]
Paolo Favaro [1]

Abstract

Video Diffusion Models (VDMs) are trained for video generation, yet this objective implicitly induces structured visual representations that extend beyond this task. In this work, we investigate whether such pretrained models can be adapted to perform classical computer vision tasks from only a few examples. We introduce a simple few-shot adaptation framework in which each task is specified by a small set of paired input and output images, encoded as short transition videos. Lightweight LoRA adapters are trained while keeping the VDM backbone frozen, and predictions are obtained from the final frame of generated sequences.

Across a diverse set of image-to-image tasks, including geometric transformations, style transfer, dense prediction, and classification, we find that the model can generalize from as few as one to thirty examples. Performance varies across tasks, with simpler transformations requiring minimal supervision and more structured problems benefiting from additional examples. These results suggest that pretrained VDMs encode reusable visual priors that can be exposed and steered through limited supervision. Overall, our findings position video diffusion models as flexible visual learners with the potential to become vision generalists.

1. Introduction

> *"What I cannot create, I do not understand."*
> — Richard P. Feynman (Feynman, 1989)

Generative models have rapidly advanced in text, image, audio, and video synthesis (OpenAI, 2023; Rombach et al., 2022; Polyak et al., 2024). In language, generative pretraining has also produced models that can adapt to new tasks from only a few examples, either through in-context demonstrations or parameter-efficient fine-tuning (Brown et al., 2020; Liu et al., 2022). A natural question is whether a similar phenomenon is emerging in visual generative models: does learning to create visual content induce priors that support visual understanding?

Video Diffusion Models (VDMs) are a particularly compelling setting for this question. To generate plausible video, a model must maintain object identity, spatial layout, temporal consistency, and fine-grained appearance over a sequence. We hypothesize that this pressure encourages VDMs to internalize structured visual representations that can be repurposed for classical computer vision tasks.

We test this hypothesis with a few-shot adaptation protocol. Given a task specified by a small number of image pairs, such as an image and its segmentation mask, we render each pair as a short video transition from input to target. We then fine-tune LoRA adapters on top of a frozen image-to-video diffusion model. At inference time, the adapted model receives a new input image, generates a transition video, and the final frame is used as the prediction.

The central observation is that many tasks require very little data to steer the model. One-shot adaptation is often sufficient for geometric transformations and style transfer, while more structured tasks such as colorization, inpainting, jigsaw reconstruction, pose estimation, binary segmentation, and grid-based classification improve sharply with the number of demonstrations. Since the backbone is frozen and supervision is scarce, successful adaptation is evidence that the pretrained VDM already contains strong useful visual priors for these tasks.

Our paper makes three main contributions:

- We present a transition-video few-shot adaptation protocol for probing pretrained VDMs on RGB image-to-image visual tasks.
- We show that lightweight LoRA adaptation unlocks generalization across a broad set of classical computer vision tasks using between 1 and 30 examples.
- We explore how adaptation quality depends on design choices including transition trajectory, LoRA placement, LoRA rank, and VDM backbone scale, and show it to be robust to these design choices and scale with model size.

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

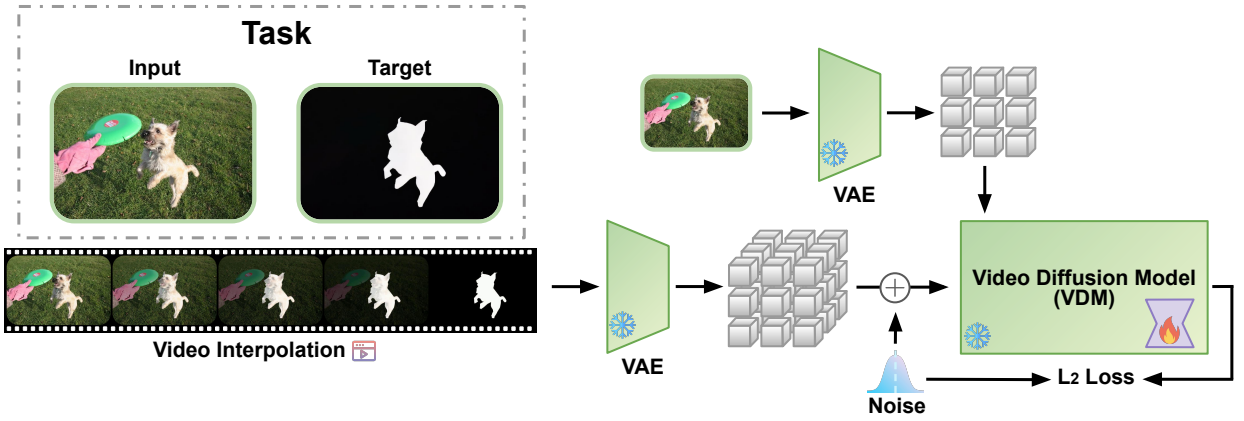


Figure 1. Transition-video adaptation. A task is given by a few input-target image pairs. Each pair is converted into a short video that transforms the input into the target. We fine-tune LoRA adapters on a frozen VDM, then use the final generated frame as the prediction for a new input image.

2. Related Work

Few-shot adaptation of generative models. Few-shot learning is a defining behavior of modern language models (Brown et al., 2020). In vision, parameter-efficient tuning has often focused on personalization or concept learning for text-to-image diffusion models (Ruiz et al., 2022; Liu et al., 2024), while other work uses diffusion features for downstream recognition or dense prediction (Tang et al., 2023; Wang et al., 2024; Helbling et al., 2025; Delatolas et al., 2025). Our focus is complementary: rather than extracting frozen features, we ask whether the native generative interface of a VDM can be steered toward visual tasks from a few input-output examples. Unlike the aforementioned techniques, we instead use LoRA to flexibly steer the model with only a few visual examples.

Generative models as generalist vision learners. Several recent works recast perception as generation. Image inpainting as visual prompting (Bar et al., 2022), image-based in-context learning (Wang et al., 2023a), diffusion-based in-context learning (Wang et al., 2023b), instruction-following diffusion models (Geng et al., 2024), and sequential visual modeling (Bai et al., 2024) all suggest that generative interfaces can support broad visual tasks. RealGeneral (Lin et al., 2025) fine-tunes CogVideoX1.5 with temporal in-context prompts and abundant data. More recently, Gabeur et al. (2026) show that image generators can be instruction-tuned into strong generalists, given sufficient data. Wiedemer et al. (2025) demonstrate that video models can act as zero-shot learners through prompting. We bypass the need for large-scale data or textual prompts, instead using LoRA to flexibly steer the model with only a few visual examples.

Video diffusion models. VDMs extend diffusion modeling to spatiotemporal data (Blattmann et al., 2023). Recent systems such as CogVideoX (Yang et al., 2024), LTX-Video (HaCohen et al., 2025), Wan2.1 (Wang et al., 2025), MovieGen (Polyak et al., 2024), Sora (Qin et al., 2024), and Veo 2 (Google DeepMind, 2024) have substantially improved video quality and controllability. We use these models not for open-ended synthesis, but as pretrained visual learners whose priors can be exposed through few-shot task adaptation.

3. Method

3.1. Image Tasks as Transition Videos

Let a visual task $\mathcal{T}$ be represented by a dataset $\mathcal{D}_{\mathcal{T}} = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i, y_i \in \mathbb{R}^{3\times H\times W}$ are RGB input-target pairs. The goal is to learn a mapping from x to y using only a small number of examples, typically $n \in [1, 30]$.

We convert each pair into a video sequence $v_i = [v_{i,1}, \ldots, v_{i,F}]$ with $v_{i,1} = x_i$ and $v_{i,F} = y_i$. This is done with an interpolation function

$$v_{i,f} = \phi(x_i, y_i, f), \quad f \in \{1, \ldots, F\}. \tag{1}$$

A simple linear transition is

$$\phi(x, y, f) = \left(1 - \frac{f-1}{F-1}\right) x + \frac{f-1}{F-1} y. \tag{2}$$

We also consider quadratic, discrete, and tiled trajectories, since the path between input and target can affect how well the task aligns with the model's video priors.

3.2. LoRA Adaptation

We use a pretrained image-to-video diffusion model conditioned on the first frame and a fixed task-agnostic text

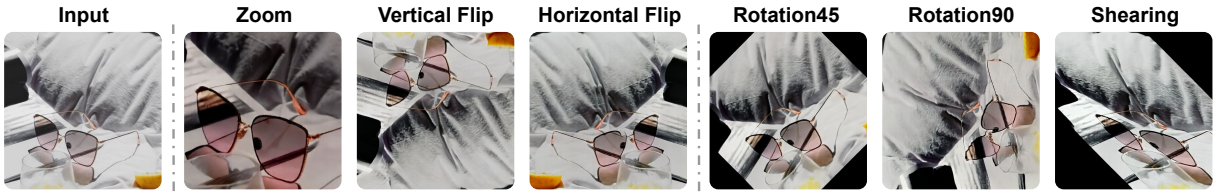


Figure 2. Geometric transformations learned from one (Zoom, Vertical Flip and Horizontal Flip) or three examples (Rotation45, Rotation90 and Shearing), applied to new Input example.

Figure 3. One-shot style transfer. With a single example (first row), the model can learn to transfer style (second row).

prompt. The model is trained with the standard denoising objective

$$\mathcal{L}(\theta) = \mathbb{E}_{v^0,\epsilon,t}\left[\left\|\epsilon - \epsilon_\theta(v^t, t, c)\right\|_2^2\right], \tag{3}$$

where v^0 is a transition video, v^t is its noisy version at diffusion step t, and c contains the conditioning image and text embedding.

To preserve the pretrained model and probe its existing priors, we freeze the backbone and train only LoRA parameters (Hu et al., 2022). For selected weight matrices θ_ℓ, the adapted weights are

$$\hat{\theta}_\ell = \theta_\ell + B_\ell A_\ell, \tag{4}$$

where $B_\ell A_\ell$ is a low-rank update. At inference, a new image x_{test} is used as the conditioning frame, the VDM generates a video sequence, and the last frame is decoded as the prediction $\hat{y}$.

4. Experiments

Unless otherwise specified, experiments use the image-to-video version of CogVideoX1.5 (Yang et al., 2024). We explore tasks encoded as RGB input-output pairs: geometric transformations, jigsaw reconstruction, colorization, inpainting and style transfer from DreamBooth images (Ruiz et al., 2022); binary segmentation and pose estimation from COCO 2017 (Lin et al., 2014); and grid-based object classification from TinyImageNet (mnmoustafa & Ali, 2017). Pose estimation is measured with Match Rate, a lightweight RGB-decoding metric described in Section B.1.

4.1. Few-Shot Visual Adaptation

We first test whether a pretrained VDM can be steered with extremely little supervision. For geometric transformations, a single example often suffices to induce zooming, flipping, and related image transformations on held-out images; three examples are enough to specify other transformations such as rotations and shearing (Figure 2). Similarly, one-shot style transfer can apply the style of a single training image to new content. These results suggest that the LoRA is not learning the transformation from scratch, but selecting and steering priors already available in the pretrained VDM.

More structured tasks benefit from additional examples. In jigsaw reconstruction, colorization, and inpainting, predictions improve markedly as n increases from 1 to 10, as shown in Figure 4. This trend is consistent with the interpretation that few-shot examples help disambiguate the task and align it with existing visual priors.

The progression is also informative, as it reveals what the model appears to learn first. Coarse layout and semantic object identity typically emerge before finer details and local consistency. Even though the model still makes mistakes, clear stepwise improvements are visible even with a very small number of examples.

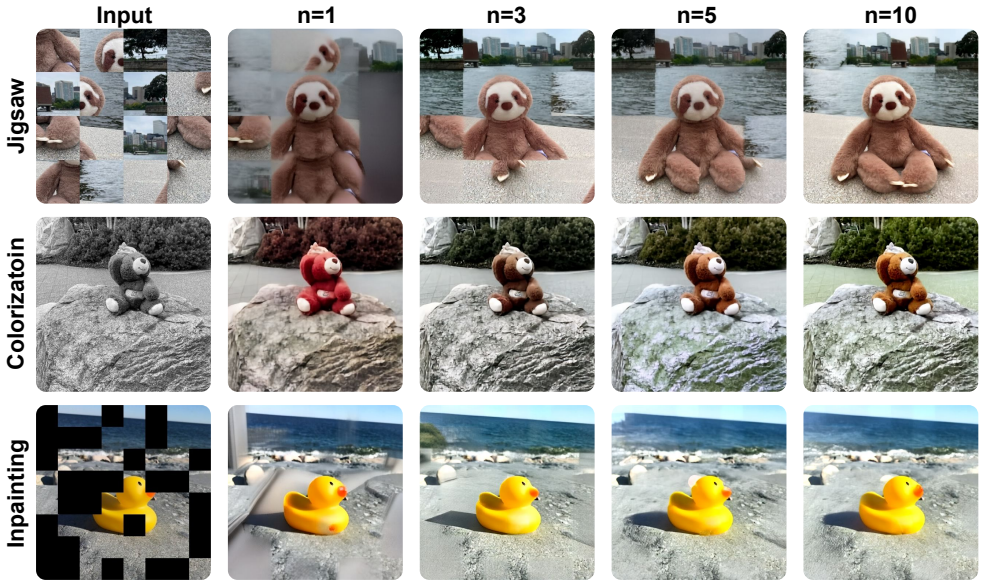
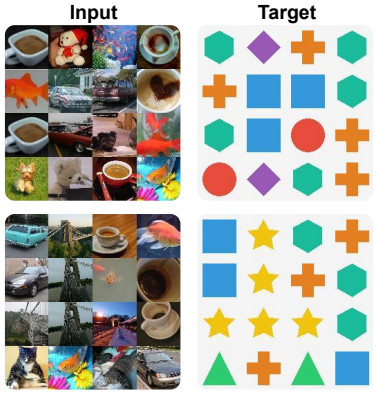
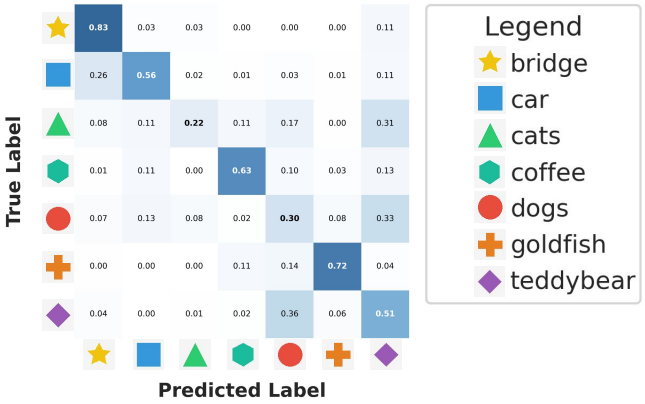


Figure 4. Qualitative Results. Jigsaw reconstruction, colorization, and inpainting improve as the number of examples increases from $n = 1$ to $n = 10$.



(a) Input image grids and symbolic class targets.



(b) Confusion matrix after training with $n = 30$ grids.

Figure 5. TinyImageNet grid classification. Each 4×4 grid contains natural images from seven object categories; targets are grids of category symbols. After $n = 30$ training examples, the model learns to distinguish many visually distinct classes, with confusions remaining mainly for semantically similar categories.

4.2. Classification

To further probe the model's recognition capabilities, we design a grid-based classification task utilizing the Tiny-ImageNet dataset (mnmoustafa & Ali, 2017). To keep the task manageable within our few-shot setting, we restrict the dataset to seven visually distinct classes: bridges, cars, cats, coffee, dogs, goldfish, and teddy bears. This restriction ensures the model observes each category a sufficient number of times during adaptation. In this setup, the model is presented with a 4×4 spatial grid populated by natural images sampled from these selected categories. The objective is to translate this complex visual input into a corresponding output grid composed entirely of discrete category symbols. To facilitate this mapping, we assign each class a unique symbol characterized by a distinct geometry, such as a triangle or rectangle, as well as a specific color. The full setup for this task is illustrated in Figure 5(a).

After fine-tuning the model with these example grids, we observe that the model can learn to distinguish the visually disparate classes, with the resulting confusion matrix shown in Figure 5(b). However, it is important to note that the overall discriminative performance is not exceptionally strong. Misclassifications persist, primarily manifesting as confusions between categories with high semantic overlap or similar low level visual features. This

error pattern indicates that the model struggles to perform robust, fine grained object recognition under this setting.

We hypothesize that this performance bottleneck stems from our tuning strategy rather than a fundamental limitation of the underlying architecture. Recent literature demonstrates that foundational image models exhibit strong zero shot recognition and compositional capabilities (Helbling et al., 2025), traits that video generation models should inherently retain. Therefore, while the underlying capability almost certainly exists within the pretrained weights, our simple adaptation is not fully sufficient to extract and reliably surface it for this task. Even so, the model's ability to map abstract object identities to novel symbolic codes, still demonstrates a meaningful degree of generalization.

4.3. Monocular Depth Estimation

Figure 6. Depth estimation rendered as an gray-scale RGB output.

Monocular depth estimation is a foundational computer vision task that is critical for understanding 3D scene geometry from a single viewpoint. Video diffusion models are naturally positioned to tackle this. Because they must generate temporally coherent movement and maintain consistent geometry across varying frames, these models require internal representations of 3D structure.

As illustrated in Figure 6, our adapted VDM successfully leverages this inherent spatial understanding. The model maps the natural images into a gray-scale RGB output, after fine-tuning only with a few images, effectively distinguishing near surfaces from distant regions.

4.4. Image-to-Segmentation and Segmentation-to-Image

In this section, we explore the model's ability to translate bidirectionally between an image and its semantic map. As shown in Figures 7 and 8, the model performs reasonably well after being fine-tuned in a one-shot setting.

However, in the image-to-segmentation direction, class boundaries lack sharpness, and there are color palette inconsistencies across semantic classes. We hypothesize that successfully solving this task requires the model to satisfy three specific criteria:

- **Boundary identification:** The ability to identify objects and their precise boundaries, a task at which the model excels (see binary segmentation).
- **Few-shot object recognition and classification:** The ability to classify objects within a scene using minimal examples, an area where the model currently struggles (see Section 4.2).
- **Global modulation:** The ability to alter the overarching style of the image to reflect the semantic map, which the model handles well (see Figure 3).

Moreover, the current task suffers from underspecification, limiting the model's ability to grasp the full semantic intent. Resolving this would necessitate a significantly larger and more diverse dataset of image categories, an exploration which is orthogonal to this work.

4.5. Design Exploration

The results, presented in Table 1, reveal that while design choices certainly influence performance, the overarching trends demonstrate a remarkable robustness in our approach. When examining the transition trajectories, we observe that quadratic and discrete methods generally outperform linear interpolation, particularly in the few-shot dense prediction, probably as the model needs to spend less resources modeling a transition between input and target state.

Finally, Table 2 presents an evaluation comparing the performance of various VDMs backbones within our

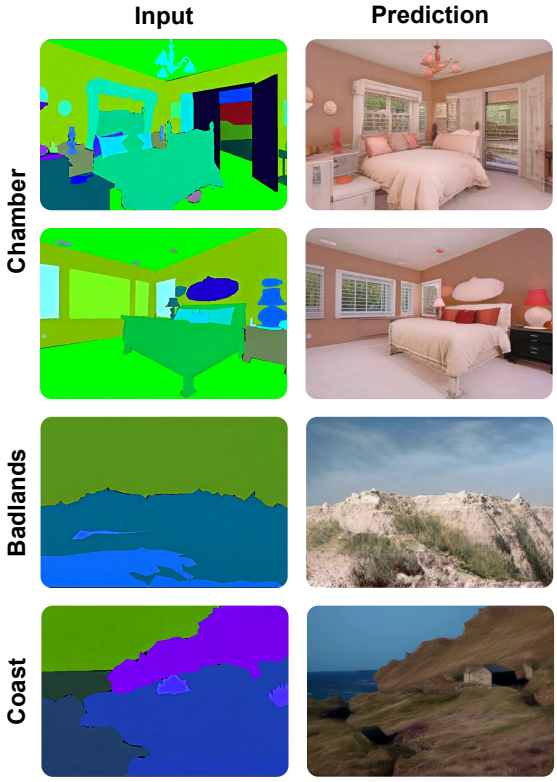


Figure 7. Examples from the Segmentation → Image task in the 1-shot setting. Each environment corresponds to a separate 1-shot training: for Chamber we train on one chamber and test on others, for Coast and Badlands the same protocol applies within their category.

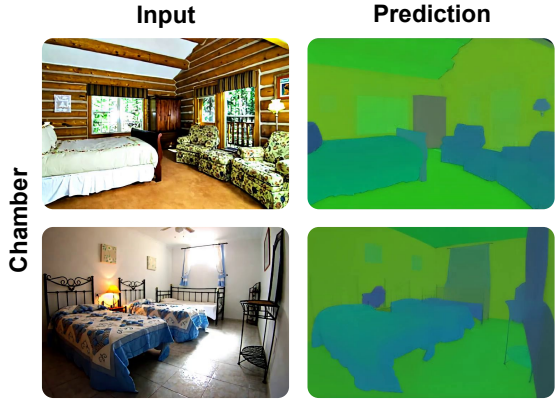


Figure 8. Examples from the Image → Segmentation in 1-shot setting for Chamber.

Table 1. Ablations on different interpolation method, training modules and rank on overall performance for $n = 3$ and $n = 30$. Full results including $n = 5$ and $n = 10$ results are in Table 8.

		Seg. mIoU ↑		Pose MR ↑	
Category	**Method**	$n = 3$	$n = 30$	$n = 3$	$n = 30$
Interp.	Linear	16.56	**55.96**	34.13	64.63
	Quadratic	**21.50**	**58.78**	**39.15**	**65.58**
	Discrete	**19.65**	55.36	**36.29**	65.48
	Tiles	14.85	54.68	30.73	**67.12**
Modules	QK	18.81	51.27	27.71	**61.83**
	VO	**23.87**	55.17	**39.94**	59.10
	QKVO	**21.50**	**58.78**	**39.15**	**65.58**
	All Linear	21.42	**56.34**	35.37	58.02
Rank	16	17.62	49.32	35.88	**59.46**
	32	18.09	48.77	**36.88**	58.95
	64	21.50	**58.78**	**39.15**	**65.58**
	128	**24.48**	**55.00**	36.06	57.66
	256	**21.95**	54.23	33.38	59.35

framework. The results reveal a clear trajectory: transitioning from lighter baselines like LTX Video to larger scale models with CogVideoX1.5 and Wan2.1 yields substantial performance improvements. This noticeable leap strongly suggests the existence of similar scaling behavior as already explored in language where model capabilities emerge and improve with an increase of training data and model size on a general pretraining task.

Table 2. Backbone comparison on binary segmentation and pose estimation. Larger VDMs show stronger few-shot adaptation.

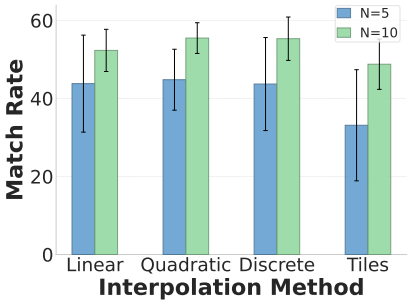
	Seg. mIoU ↑		Pose MR ↑	
VDM	$n = 3$	$n = 30$	$n = 3$	$n = 30$
LTX-Video (2B)	16.41	46.15	14.64	38.47
CogVideoX1.5 (5B)	**34.08**	**58.16**	**46.38**	**65.09**
Wan2.1	**52.88**	**68.38**	**51.74**	**72.51**

5. Takeaways from Few-Shot Adaptation

The experiments should be read as a probe of pretrained visual priors rather than as a fully supervised recipe for each individual task. In the one-shot and low-shot regime, the examples cannot contain enough statistical evidence to learn a new segmentation system, pose estimator, or restoration model from scratch. Instead, the demonstrations specify a target interface: they tell the VDM which latent behavior to expose and how to express the answer in RGB space. When this succeeds, most of the useful structure must come from the pretrained model.

This interpretation is clearest for transformations and style

(a) The four transition trajectories: Linear, Quadratic, Discrete, and Tiles.



(b) Pose estimation Match Rate (n=5 and n=10) for each trajectory.

Figure 9. Transition trajectory comparison. The same input-target pair can be encoded as a smooth linear or quadratic blend, a discrete switch, or a tiled mosaic sequence (a). Non-linear and discrete trajectories generally outperform linear interpolation on dense prediction tasks, especially in the low-shot regime (b).

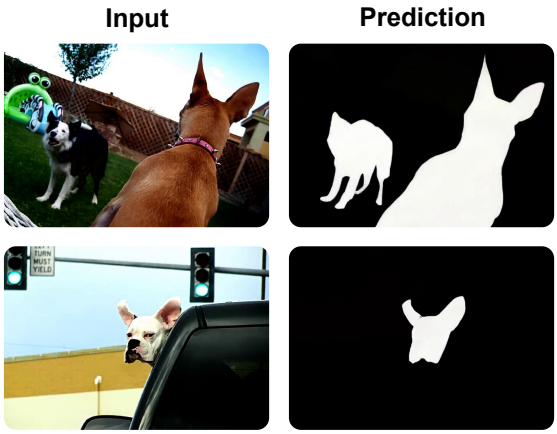


Figure 10. Examples of model's binary segmentations predictions (dogs category) after fine-tuning with $n = 30$ examples.

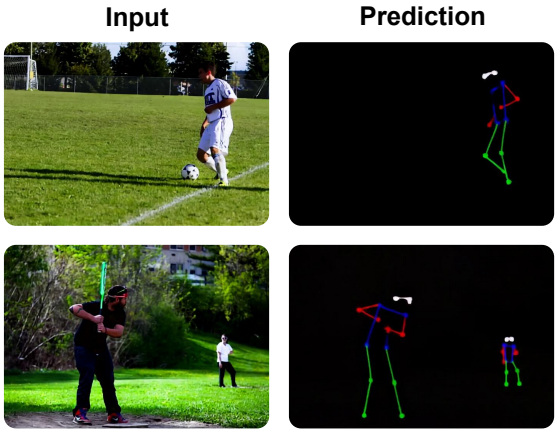


Figure 11. Examples of model pose predictions after fine-tuning with $n = 30$ examples.

transfer. The input and output remain natural images, and a small number of demonstrations can define a global visual operation. The model then applies this operation to held-out objects and scenes without being trained on many instances of the transformed distribution. Restoration tasks are more ambiguous because the model must infer both the task rule and the missing or reordered content. Dense prediction tasks add another layer of difficulty: the output is no longer a natural image, but a structured code for object masks or body parts. The gradual improvement from $n = 3$ to $n = 30$ is therefore important. It suggests that additional examples are not simply adding data, but sharpening the output convention and reducing ambiguity in how the model should use its priors.

The design exploration reinforces this message from a complementary angle. While factors such as transition trajectories, LoRA placement, adapter rank, and backbone scale have a minor impact on the results, none of them fundamentally alter the core learning dynamics. Although these observations do not exhaust the design space, they demonstrate that few-shot visual generalization in this setting is remarkably robust to such architectural variations.

Overall, the strongest claim supported by the evidence is not that Video Diffusion Models (VDMs) are already universal vision systems. Rather, it suggests that pretrained video generators harbor reusable visual knowledge that can be redirected with a surprisingly small number of examples. The classical computer vision tasks examined here offer a controlled method to reveal this knowledge while maintaining a frozen backbone and minimal task-specific supervision. Furthermore, the observed scaling behavior indicates that, much like Large Language Models (LLMs), these models have the potential to become increasingly powerful few-shot generalists as they continue to grow.

6. Limitations and Future Work

Few-shot tasks can be intrinsically underspecified. With one example, a model may infer a plausible but unintended rule, especially when several transformations are consistent with the demonstration. This ambiguity is not unique to our method, but it is important when interpreting few-shot generalization. In practice, we find that the model often behaves sensibly: it relies on object, texture, layout, and motion priors that are useful across many tasks. However, the same priors can encourage shortcuts when the examples do not clearly specify the intended mapping.

The RGB encoding interface is flexible but imposes precision limits for tasks such as metric depth estimation. The direction-agnostic nature of the interface is a strength for generality, but it might pose problems for decoding and evaluating various tasks, as well as for overall performance. Different visual encodings or task-specific output parameterizations in RGB could help address this, as suggested by recent other works, for example for depth (Gabeur et al., 2026).

Our evaluation is intentionally centered on tasks that can be represented and inspected through image outputs. This makes the protocol broad and visually transparent, but it does not replace task-specific evaluation pipelines for every domain. A broader benchmark would need to standardize output encodings, decoding rules, and metrics across many task families.

An additional consideration concerns the role of data diversity and prompt specification in shaping generalization behavior. When the support examples cover only a limited subset of possible variations, the model may overfit to superficial regularities rather than capturing the intended transformation. Expanding the diversity of demonstrations, together with clearer task conditioning, may mitigate such effects and improve robustness. This suggests that progress in few-shot settings will depend not only on model capacity and training procedures, but also on principled approaches to example selection and task representation.

Finally, LoRA fine-tuning and diffusion inference remain computationally expensive. Future work could study composable adapters, faster samplers, or shared multi task LoRAs to reduce adaptation cost while preserving the benefits of pretrained visual priors. Looking ahead, a promising direction involves moving beyond task-specific adaptation entirely. Just as LLMs evolved from requiring specialized fine-tuning to achieving alignment through broad instruction tuning, we envision a similar trajectory for VDMs. By pretraining on diverse, multi-task datasets and performing large-scale alignment with in-context examples, future models could learn to interpret few-shot demonstrations natively. This would allow the model to infer the intended transformation and align its output directly from the provided examples, eliminating the need for per-task parameter updates like LoRA and moving toward a truly zero-shot or in-context inference paradigm for generalist vision.

7. Conclusion

We studied whether pretrained VDMs can be adapted to classical computer vision tasks from only a few examples. By converting image-to-image tasks into transition videos and fine-tuning lightweight LoRA adapters, we find that VDMs can generalize across transformations, style transfer, restoration, dense prediction, and classification with between 1 and 30 demonstrations. These results support the view that video generative pretraining induces useful visual priors, and that few-shot transition based adaptation provides a simple way to expose and steer them.

Beyond empirical performance, our findings suggest generative video models can act as general purpose visual learners rather than task-specific systems. Interpreting tasks as transitions offers a unified interface for many vision problems, where adaptation is driven by examples instead of explicit supervision or architectural changes. This perspective aligns with flexible task-specification and indicates the boundary between generative modeling and discriminative vision is less rigid than traditionally assumed.

However, our results also reveal important limitations and directions for future work. Few-shot purely visual adaptation remains sensitive to ambiguous examples, with performance depending heavily on how clearly the intended transformation is specified. Improving robustness may require better mechanisms for encoding task intent, diverse demonstration sets, or hybrid approaches combining structured textual supervision. Additionally, reducing the computational costs of fine-tuning and inference is essential for practical deployment.

Overall, our study provides evidence that pretrained VDMs constitute a promising foundation for generalist vision systems. We hope that this work motivates further exploration on the field, with more expressive adaptation mechanisms, and broader evaluation protocols that can fully characterize the capabilities and limits of these models.

References

Bai, Y., Geng, X., Mangalam, K., Bar, A., Yuille, A. L., Darrell, T., Malik, J., and Efros, A. A. 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.

Bar, A. et al. Visual prompting via image inpainting. *CoRR*, abs/2209.00647, 2022. URL https://doi.org/10.48550/arXiv.2209.00647.

Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., Jampani, V., and Rombach, R. Stable video diffusion: Scaling latent video diffusion models to large datasets. *CoRR*, abs/2311.15127, 2023. URL https://doi.org/10.48550/arXiv.2311.15127.

Brown, T. B. et al. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL https://arxiv.org/abs/2005.14165.

Delatolas, T. et al. Studying image diffusion features for zero-shot video object segmentation. *arXiv preprint arXiv:2504.05468*, 2025.

Feynman, R. P. Feynman's office; the last blackboards. *Physics Today*, 42(2):88–88, February 1989. ISSN 0031-9228. doi: 10.1063/1.2810904. URL https://doi.org/10.1063/1.2810904.

Gabeur, V. et al. Image generators are generalist vision learners, 2026. URL https://arxiv.org/abs/2604.20329.

Geng, Z. et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 12709–12720, 2024.

Google DeepMind. Veo 2. https://deepmind.google/technologies/veo/veo-2/, December 2024. URL https://deepmind.google/technologies/veo/veo-2/. Accessed: 2025-05-12.

HaCohen, Y. et al. Ltx-video: Realtime video latent diffusion. *CoRR*, abs/2501.00103, January 2025. URL https://doi.org/10.48550/arXiv.2501.00103.

Helbling, A. et al. Conceptattention: Diffusion transformers learn highly interpretable features. *CoRR*, abs/2502.04320, February 2025. URL https://doi.org/10.48550/arXiv.2502.04320.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Lin, T.-Y., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

Lin, Y. et al. Realgeneral: Unifying visual generation via temporal in-context learning with video models. *arXiv preprint arXiv:2503.10406*, 2025.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *CoRR*, abs/2205.05638, 2022. URL https://doi.org/10.48550/arXiv.2205.05638.

Liu, J. et al. Ada-adapter: Fast few-shot style personalization of diffusion model with pre-trained image encoder. *CoRR*, abs/2407.05552, 2024. URL https://doi.org/10.48550/arXiv.2407.05552.

mnmoustafa and Ali, M. Tiny imagenet. https://kaggle.com/competitions/tiny-imagenet, 2017. Kaggle.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. URL https://doi.org/10.48550/arXiv.2303.08774.

Polyak, A. et al. Movie gen: A cast of media foundation models. *CoRR*, abs/2410.13720, 2024. URL https://doi.org/10.48550/arXiv.2410.13720.

Qin, Y., Shi, Z., Yu, J., Wang, X., Zhou, E., Li, L., Yin, Z., Liu, X., Sheng, L., Shao, J., Bai, L., Ouyang, W., and Zhang, R. Worldsimbench: Towards video generation models as world simulators. *CoRR*, abs/2410.18072, 2024. URL https://doi.org/10.48550/arXiv.2410.18072.

Roberts, M. et al. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *International Conference on Computer Vision (ICCV) 2021*, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. 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.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *CoRR*, abs/2208.12242, 2022. URL https://doi.org/10.48550/arXiv.2208.12242.

Silberman, N. et al. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

Tang, L. et al. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023.

Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Meng, X., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.-F., and Liu, Z. Wan: Open and advanced large-scale video generative models. *CoRR*, abs/2503.20314, March 2025. URL https://doi.org/10.48550/arXiv.2503.20314.

Wang, Q. et al. Zero-shot video semantic segmentation based on pre-trained diffusion models. *CoRR*, abs/2405.16947, 2024. URL https://doi.org/10.48550/arXiv.2405.16947.

Wang, X. et al. 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.

Wang, Z. et al. In-context learning unlocked for diffusion models. *Advances in Neural Information Processing Systems*, 36:8542–8562, 2023b.

Wiedemer, T. et al. Video models are zero-shot learners and reasoners, 2025. URL https://arxiv.org/abs/2509.20328.

Yang, Z. et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *CoRR*, abs/2408.06072, 2024. URL https://doi.org/10.48550/arXiv.2408.06072.

Zhou, B. et al. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Zhou, B. et al. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

A. Experimental Details

A.1. Datasets

COCO 2017 (Lin et al., 2014). We use COCO 2017 for binary segmentation and human pose estimation. Binary masks are converted into black-and-white target images. For pose estimation, body components are rendered with a compact color code: white for head, blue for torso, red for arms, and green for legs.

DreamBooth (Ruiz et al., 2022). We use DreamBooth images for geometric transformations, style transfer, and restoration-style tasks. Task-specific targets are generated by applying transformations such as colorization, masking for inpainting, or patch shuffling for jigsaw reconstruction.

TinyImageNet (mnmoustafa & Ali, 2017). We use TinyImageNet to construct a grid-based classification task. We select seven categories with 100 images each at resolution 64×64. We use 80 images per class for training grids and 20 for validation grids. Inputs are 4×4 image grids, and targets are 4×4 grids of class-specific colored symbols.

Additional Dense Prediction Datasets. We also use NYUv2 (Silberman et al., 2012), ADE20K (Zhou et al., 2017; 2019), and ML-Hypersim (Roberts et al., 2021).

A.2. Hyperparameters

Table 3 lists the main hyperparameters. CogVideoX1.5 uses task-specific training schedules and resolutions, detailed in Tables 4 and 5.

Table 3. Hyperparameters used in CogVideoX1.5, LTX-Video, and Wan2.1 experiments.

Parameter	**CogVideoX1.5**	**LTX-Video**	**Wan2.1**
LoRA Configuration			
Rank	64	64	128
Alpha	32	32	128
Target modules	QKVO	All linear	QKVO + FF
Video Configuration			
Resolution (FxHxW)	task-specific	9x480x704	9x480x720
Interpolation	task-specific	Convex	Convex
Training Configuration			
Seed	42	42	42
Batch size	2	16	1
Training steps	task-specific	3000	4000
Validation steps	task-specific	200	1000
Gradient accumulation steps	1	1	1
Optimizer Configuration			
Optimizer	AdamW	AdamW	AdamW
Learning rate	1e-4	1e-3	2e-5
Scheduler	Constant	Constant	Const. + Warmup (1000)
β_1	0.90	0.90	0.90
β_2	0.95	0.99	0.99
Weight decay	1e-3	0.0	1e-4
ϵ	1e-8	1e-8	1e-8
Max grad norm	1.0	1.0	1.0

For CogVideoX1.5 and Wan2.1, Classifier-Free Guidance was not used during inference. For the CogVideoX1.5 binary segmentation and pose ablations, the number of diffusion sampling steps was reduced from 50 to 5 after verifying comparable behavior following fine-tuning.

Table 4. Training and validation steps for CogVideoX1.5.

Task	Training Steps	Validation Steps
Binary Segmentation		
$n = 3$	1000	1000
$n = 5$	1000	1000
$n = 10$	2000	1000
$n = 30$	4000	1000
Pose Estimation		
$n = 3$	1000	1000
$n = 5$	1000	1000
$n = 10$	2000	1000
$n = 30$	4000	1000
TinyImageNet classification	500	500
Remaining tasks	1200	300

Table 5. Resolution configurations for different tasks.

Task	Resolution (FxHxW)
Binary Segmentation	9x480x720
Pose Estimation	9x480x720
TinyImageNet classification	9x256x256
Remanining tasks	9x480x720 / 9x480x480

A.3. Computational Requirements

Most experiments were run on NVIDIA RTX 4090 GPUs with 24GB of VRAM. Each individual run used a single GPU, although multiple runs were executed in parallel on multi-GPU nodes. Wan2.1 experiments used H100 GPUs. Tables 6 and 7 report finalized training runs and exclude development/debugging time.

Table 6. Training times for CogVideoX1.5, LTX-Video, and Wan2.1 across tasks and number of training samples. Total time reflects cumulative runtime for each configuration.

Task	CogVideoX1.5			LTX-Video			Wan2.1		
	Time (h)	#	Total (h)	Time (h)	#	Total (h)	Time (h)	#	Total (h)
Binary Segmentation									
$n = 3$	0.50	3	1.50	1.50	3	4.50	1.00	3	3.00
$n = 5$	0.50	3	1.50	1.50	3	4.50	1.00	3	3.00
$n = 10$	1.00	3	3.00	1.50	3	4.50	2.00	3	6.00
$n = 30$	2.00	1	2.00	1.50	1	1.50	2.00	1	2.00
Pose Estimation									
$n = 3$	0.75	3	2.25	1.75	3	5.25	1.00	3	3.00
$n = 5$	0.75	3	2.25	1.75	3	5.25	1.00	3	3.00
$n = 10$	1.50	3	4.50	1.75	3	5.25	2.00	3	6.00
$n = 30$	3.00	1	3.00	1.75	1	1.75	2.00	1	2.00
Total			**20.00**			**32.50**			**28.00**

B. Evaluation Details

B.1. Match Rate

We use Match Rate as a lightweight proxy for pose quality. The metric is designed for our RGB pose representation and avoids expensive conversion into full keypoint predictions.

Table 7. Training times for CogVideoX1.5 general visual tasks and ablations.

Task	Training Time (h)	# Runs	All Runs (h)
Binary Segmentation			
$n = 3$	0.75	3	2.25
$n = 5$	0.75	3	2.25
$n = 10$	1.50	3	4.50
$n = 30$	3.00	1	3.00
Pose Estimation			
$n = 3$	1.25	3	3.75
$n = 5$	1.25	3	3.75
$n = 10$	2.50	3	7.50
$n = 30$	5.00	1	5.00
Time Per Ablation			32.00
Total Ablation Time			352.00
General Visual Tasks			
TinyImageNet			25.00
Remaining tasks			150.00

1. Given a predicted pose image, we use the color channels to segment four body components: head, torso, arms, and legs.
2. For each connected component, we compute its centroid.
3. We match predicted centroids to centroids derived from annotated keypoints for each body part.
4. A match is valid if its Euclidean distance is below 1.5 times the average inter-head distance. If no heads are available, we use a default threshold of 20 pixels.

The main advantage of Match Rate is that it evaluates pose quality from a single RGB prediction. Its main limitation is that it depends on stable color decoding and does not penalize every spurious component. In practice, visual inspection indicates that false positives outside the intended figure are rare.

B.2. TinyImageNet Decoding

For each TinyImageNet grid, we decode 16 classification patches. We apply L2 nearest-neighbor matching against the set of known class symbols, aggregate patch predictions, and then evaluate the results as a standard multi-class classification problem.

C. Full Design Exploration Results

The four transition trajectory types (Linear, Quadratic, Discrete, Tiles) are illustrated in Figure 9a of the main paper, together with a quantitative comparison on pose estimation.

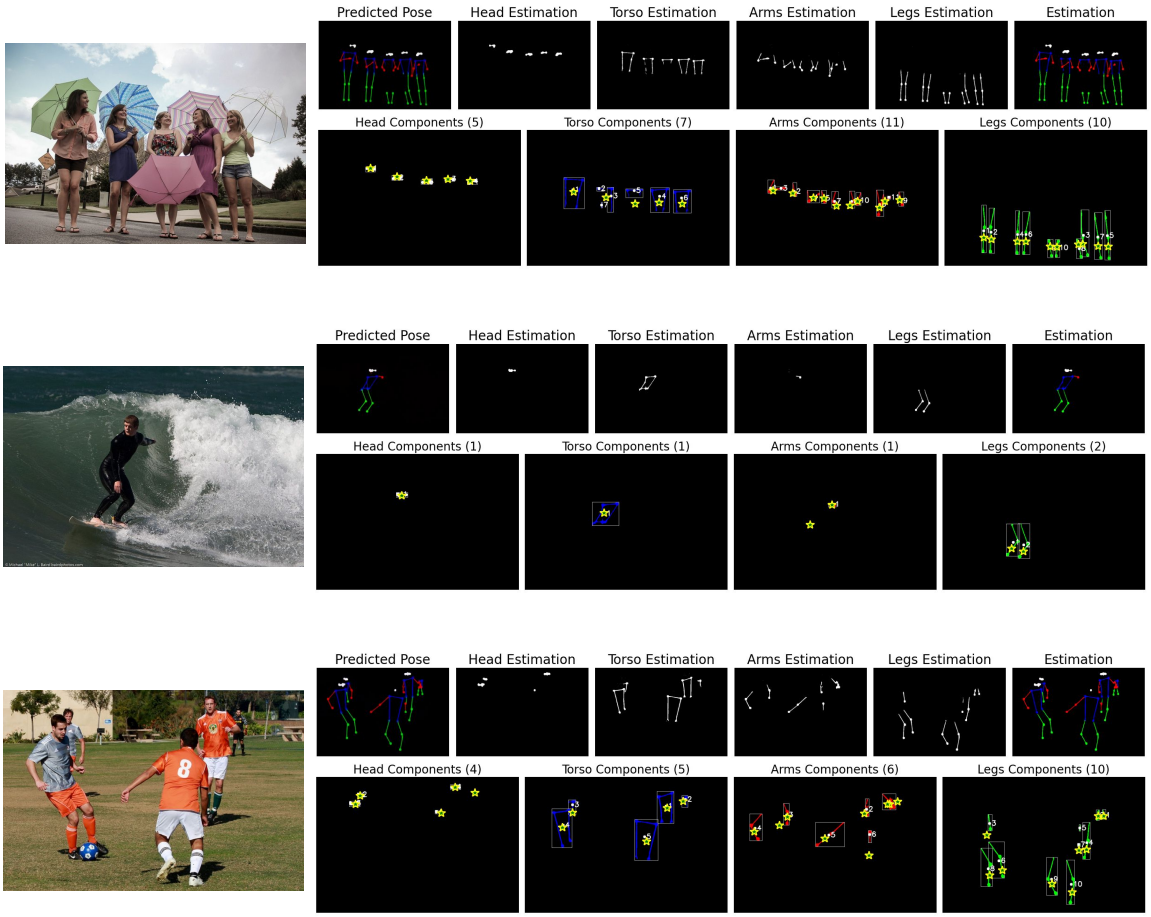


Figure 12. Match Rate calculation. Left: original image. Top row: model prediction and estimated components. Bottom row: component centroids from predictions, with annotated centroids shown as stars.

Table 8. Full exploration of transition trajectories and LoRA strategies across binary segmentation and pose estimation. **Best** and **second-best** results are highlighted within each group.

		Binary Segmentation (mIoU ↑)				Pose Estimation (Match Rate ↑)			
Category	**Method**	$\mathbf{n=3}$	$\mathbf{n=5}$	$\mathbf{n=10}$	$\mathbf{n=30}$	$\mathbf{n=3}$	$\mathbf{n=5}$	$\mathbf{n=10}$	$\mathbf{n=30}$
Interpolation	Linear	16.56	30.36	**46.71**	**55.96**	34.13	43.80	52.77	64.63
	Quadratic	**21.50**	**30.50**	**48.97**	**58.78**	**39.15**	**44.80**	**55.48**	**65.58**
	Discrete	**19.65**	**30.72**	46.56	55.36	**36.29**	**43.69**	**57.23**	65.48
	Tiles	14.85	29.95	39.96	54.68	30.73	33.13	48.77	**67.12**
LoRA Modules	QK	18.81	25.89	41.28	51.27	27.71	29.05	39.83	**61.83**
	VO	**23.87**	**31.90**	**45.46**	55.17	**39.94**	**44.36**	**54.36**	59.10
	QKVO	**21.50**	30.50	**48.97**	**58.78**	**39.15**	**44.80**	**55.48**	**65.58**
	All Linear	21.42	**33.32**	42.61	**56.34**	35.37	42.84	52.42	58.02
LoRA Rank	Rank 16	17.62	27.46	**44.52**	49.32	35.88	35.03	46.25	**59.46**
	Rank 32	18.09	30.51	**44.25**	48.77	**36.88**	38.87	47.63	58.95
	Rank 64	21.50	**30.50**	**48.97**	**58.78**	**39.15**	**44.80**	**55.48**	**65.58**
	Rank 128	**24.48**	30.11	41.98	**55.00**	36.06	**44.72**	49.00	57.66
	Rank 256	**21.95**	**31.14**	43.67	54.23	33.38	40.25	**49.97**	59.35

D. Additional Qualitative Results

Here we include some additional qualitative examples for the experiments on the main paper.

D.1. Task Underspecification

Here we outline further limitations of our approach, contextualized by the additional examples provided. Some challenges are intrinsic to the few shot learning paradigm, while others stem from specific choices and constraints in our current implementation.

Ambiguity in Task Definitions. Few shot tasks, especially in the one shot setting, are inherently underspecified and introduce significant ambiguity. A single example is rarely sufficient to capture the full variability of a class or to clearly convey the intended concept without strong inductive priors. The unexpected behaviors we observe in these edge cases likely stem from this inherent task underspecification, though they could also be a consequence of overfitting to the strictly limited training data.

In our setting, even if the model **has access to the necessary representational features** required to support a generalizable solution, there is no guarantee that the LoRA adaptation will utilize them effectively. We observe that the model typically relies on these features when they offer a low effort solution, indicating they serve as strong priors. However, in cases where only a single example is available, the model may instead exploit superficial correlations or shortcuts that solve the immediate training task without achieving true generalization.

As shown in Figure 18, a model trained for inpainting on a single image can sometimes introduce features into validation examples that reflect artifacts of the original training task. A related pattern emerges in the binary segmentation setting, where the model might adopt overly broad interpretations, such as learning to segment animals instead of the specific target of segmenting dogs. It may even default to segmenting any centered object when trained with minimal data (see Figure 19). These behaviors highlight how underspecification and/or overfitting can cause outputs to diverge from intended outcomes.

A more extreme instance of this behavior is shown in Figure 20. The model clearly learns to texturize the segmentations, which is an entirely sensible outcome based on the training data, rather than deriving any semantic understanding from the shapes to construct the images. Consequently, it completely fails to handle out of distribution images. In this example one cannot see if the problem comes from the model don't understanding the tasks or failing to generalize from its semantic understanding.

Figure 13. Dense prediction results with $n = 30$ examples. Left: pose estimation predictions rendered as coarse body components. Right: binary dog segmentation predictions.

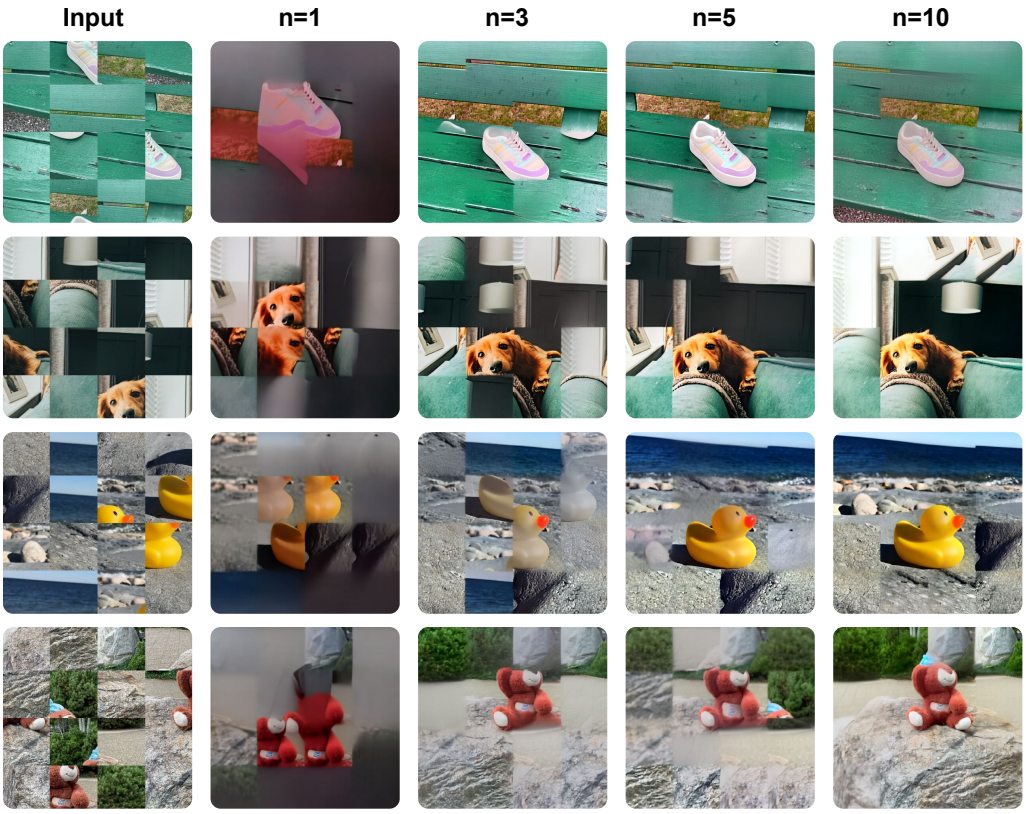


Figure 14. Progression of jigsaw reconstruction results as the number of training examples increases.

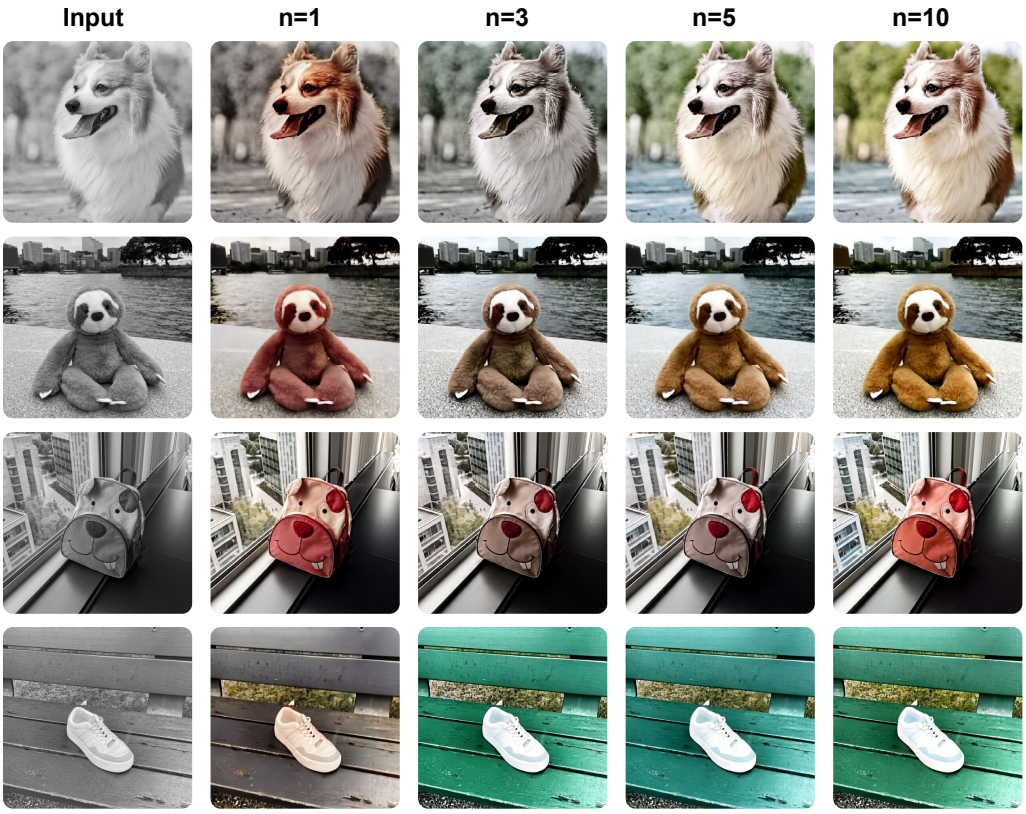


Figure 15. Progression of colorization results as the number of training examples increases.

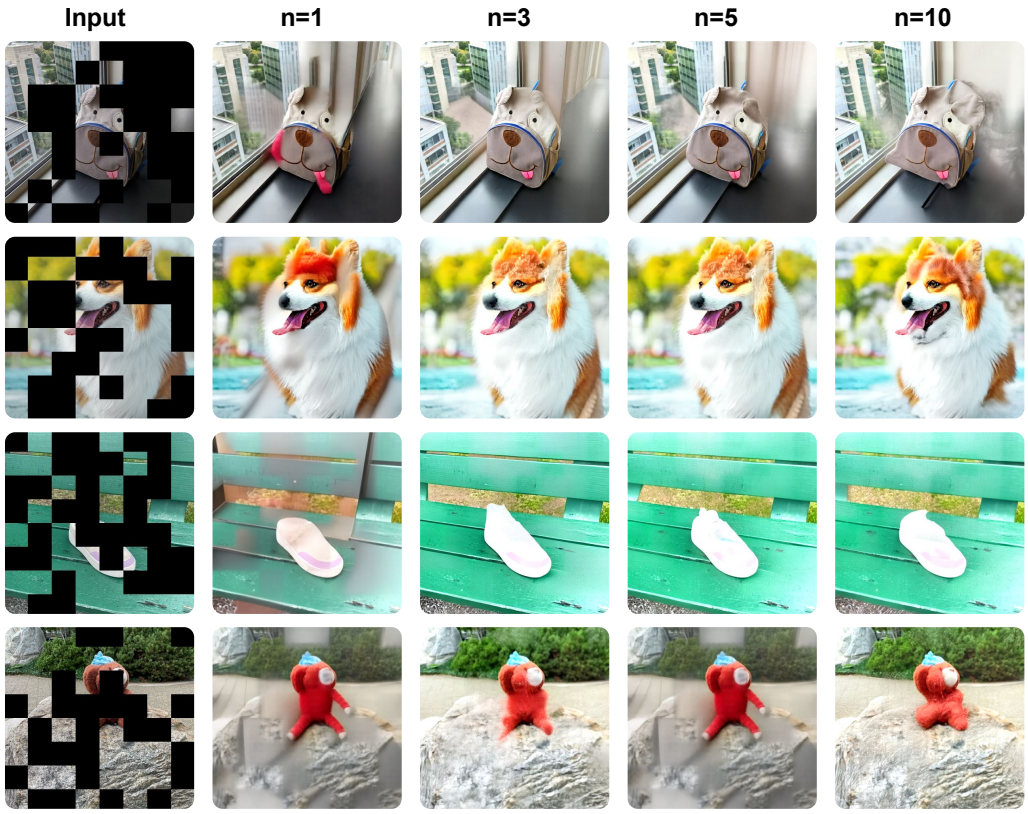


Figure 16. Progression of inpainting results as the number of training examples increases.

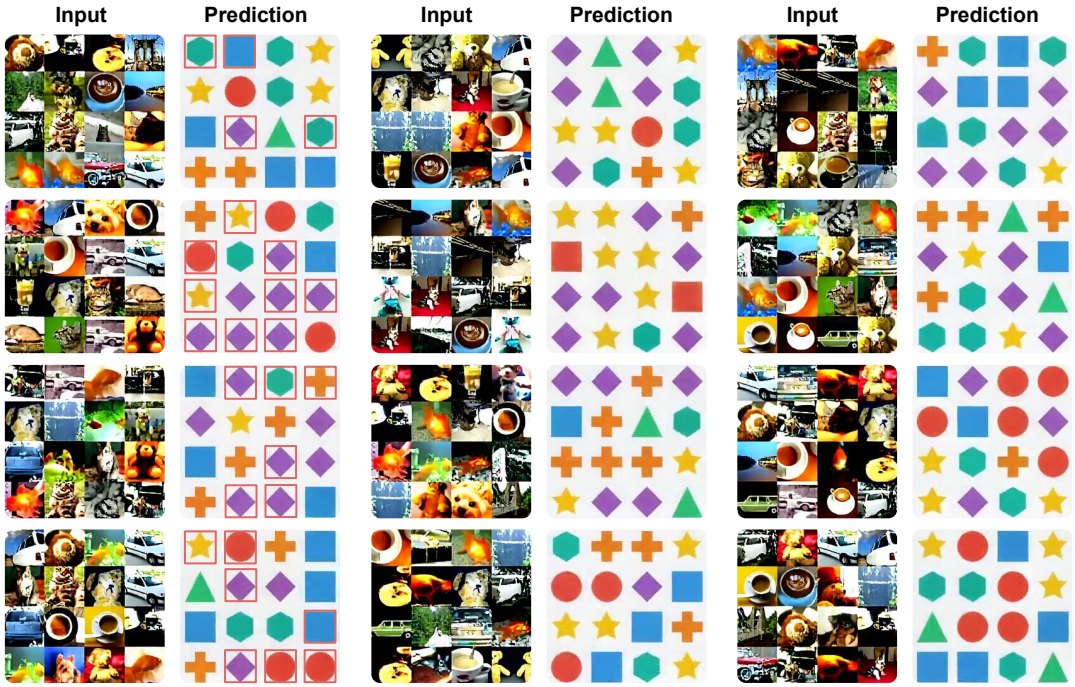


Figure 17. Validation examples from the TinyImageNet grid classification task with $n = 30$ training examples. Prediction errors in the first column are highlighted with red boxes.

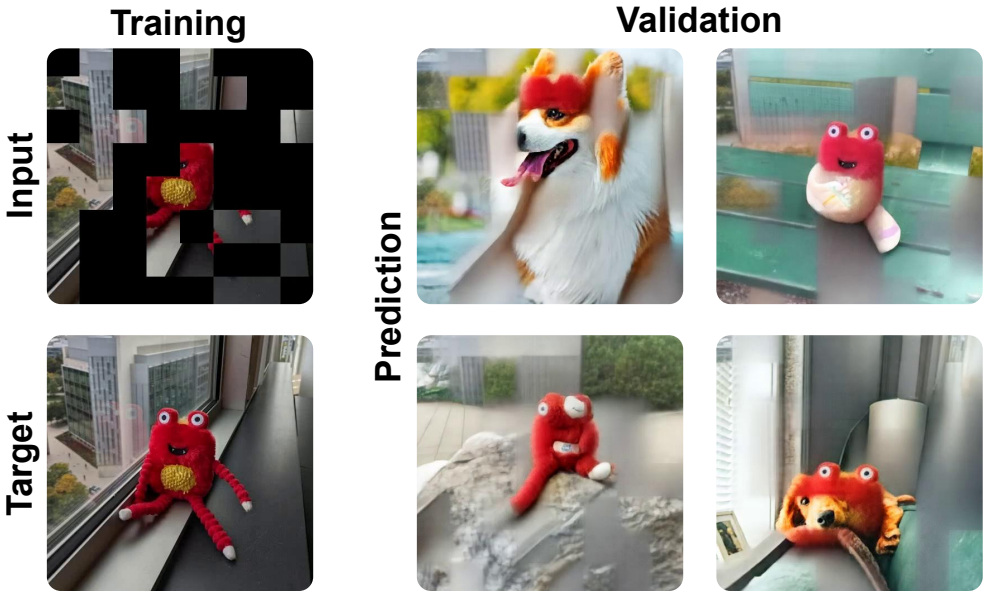


Figure 18. Left: a training sample used in the inpainting task (one of several masked-region variations). Right: representative validation outputs, where the model erroneously inserts features reminiscent of the original image.

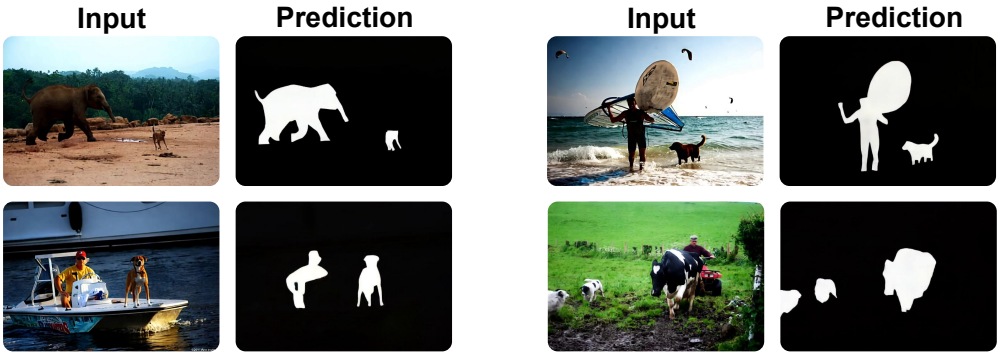


Figure 19. Examples from the binary segmentation task (validation), where the model, trained to segment dogs, instead segments broader categories like all animals or defaults to spatial biases such as centered objects.

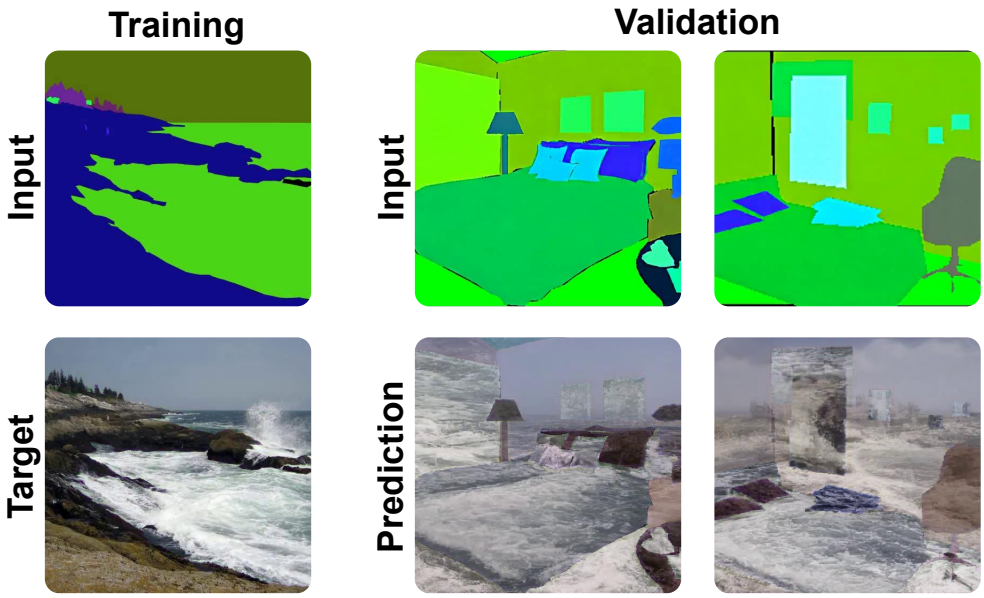


Figure 20. Examples for Segmentation to Image with out-of-distribution validation samples.