A. Appendix

A.1. Probing Across Layers

As shown in Figure 4 for the ImageNet classification task, different layers of the model contribute differently to the task; a behavior also observed in iGPT [9]. To study this behavior across multiple tasks, we train probing layers for action recognition, object tracking, and robot manipulation. Figure 9 shows the probing performance across layers, model sizes, and tasks. It reveals that action recognition follows a similar trend to ImageNet classification—peaking in the middle of the model stacks. While object tracking shows a comparable trend, object manipulation interestingly exhibits strong performance in the last layers, similar to the middle layers. Compared to the first three tasks, robot manipulation benefits from the generative nature of the task. In encoder models [7] or encoder-decoder models [3, 25], the last layer of the encoder typically has richer semantic features. *This may suggest that in a decoder-only model the first half of the network starts to behave like an encoder—compressing information—while the remaining layers project the compressed semantic features back to the input space.*

A.2. Limitations

Our study suggests several important limitations and opportunities for future work. A significant limitation stems from the use of internet videos, which, unlike carefully curated datasets, introduces challenges related to data quality and diversity. This variance in data quality can impact model performance, especially when compared to models trained on more curated datasets. Another limitation is the use of tokenizer, this makes the learning not end-to-end, and the representation and generation quality is bounded by the quality of the tokenizer, and with quantized vectors, the quality is very much limited, this needs further explorations to build a universal visual tokenizer. Another fundamental limitation is training on videos for next token prediction task. The added redundancy in video frames, can hurt quality of the learned representations. See Appendix A.3 for more discussion on this topic. Additionally, our exploration of various design choices are based on ImageNet classification. While it does transfer to most of the tasks we considered in this paper, it may not be the optimal configuration for many other tasks. Furthermore, we have not yet fully assessed our method's effectiveness in dealing with dense prediction tasks, fine-grained recognition, or comprehending complex temporal dynamics over extended time frames. These areas represent key opportunities for further research, aiming to broaden the fruitfulness of autoregressive pretrained models.

A.3. Video Tokens for Pre-Training

The next patch prediction for visual pre-training is equivalent to the next token prediction in large language models. However, most languages have a clear sequential nature, therefore there is a clear definition for the next word. This also makes the next word prediction task relatively harder, since the model requires learning to extrapolate the data. On the other hand, images and videos, especially over the spatial dimensions lack a sequential nature. We follow the previous works [9, 57] to make the images and videos into a 1D sequence by scanning the patches in raster order. While this ordering allows for example to learn to predict the bottom half of the image from the top part of the image, in many places, the tokens can be predicted by interpolating rather than extrapolating.

On the time axis, yes, there is a clear sequential nature, however, video frames compared to text tokens are more redundant, making the next frame prediction task much easier. Figure 10 shows average validation loss over 4096 token, in kinetics 400 dataset [29], on *Toto*-large model. This shows there is high loss of the first frame, but the subsequent frames have relatively lower loss compared to the first frame. This is because, even with reasonably lower sampling rate, frames following the first frame has some redundancy, and hinders the learning, since these tokens are relatively easy to predict. This also could be attributed by emergence of induction heads [35]. While we focused on learning from unfiltered internet scale video with minimal inductive bias, to learn efficiently from videos, need further research in this direction.

A.4. SD-VAE discrete tokenizer:

We also explored quantizing continuous tokens from SD-VAE tokenizer [16]. The simplest option is to create very large number of randomly initialized discrete code books and train the model to predict these discrete code books. This only works when the code book dimension is very small, and in our case SD-VAE returns 4 dimension vector. From the SD-VAE we will get a 4 dimensional vector for each patch and we simply find a nearest neighbor from the randomly initialized code book as the discrete token. This allow us to change the vocabulary size without any training.

Method	**Tokens**	**Vocabulary**	**Top1**
Toto-large-dVAE	32x32	8k	64.4
Toto-large-SDVAE	32x32	32k	73.8
Toto-1b-SDVAE	32x32	32k	78.8

Table 13. **Random Vocabulary:** We trained our models on randomly initizalied code books on top of SD-VAE latents. This preserves the image fidelity much higher than dVAE or VQGAN.

Figure 7. **Semi-Supervised Tracking:** We follow the protocol in STC [27] by starting with the GT segmentation mask and propagating the labels using features computed by *Toto*-large. The mask was propagated up to 60 frames without significant loss of information.

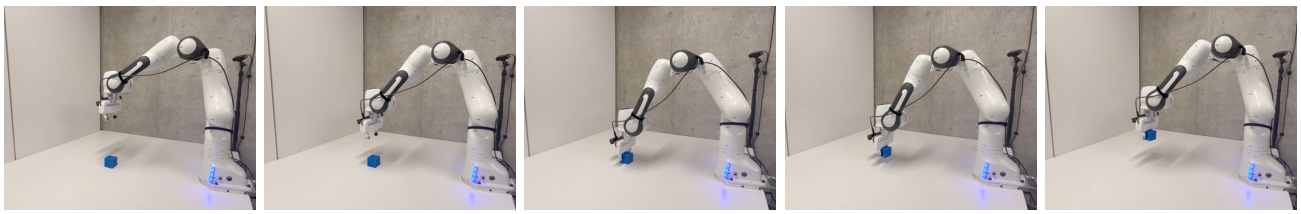


Figure 8. **Real-world Deployment:** An example episode of our policy performing a cube-picking task on a Franka robot in the real world. Using *Toto*-base enables real-time control, achieving about 63% success rate.

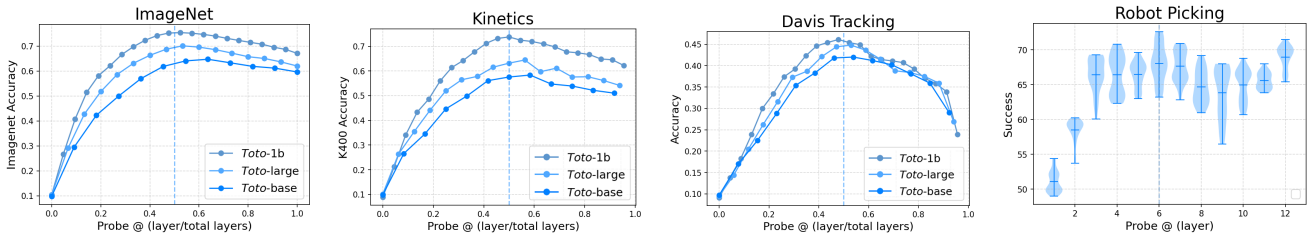


Figure 9. **Probing Across Layers, Models, and Tasks:** We study the behavior of our models across multiple layers and tasks. For image classification, action recognition, and object tracking, all the models behave similarly and peak around 50% of the model depth. This behavior is observed across all model sizes. Robot tasks show a similar behaviour, where the middle layers perform good at picking the objects, but last layers also perform good as middle layers. These plots suggests, in decoder-only model, first half of the model starts to behave like an encoder, and compress the information, and then rest of the model, projects the compressed semantic features back to input space.

Method (Res/Patch)	J&F	J	F
DINO-base (224/16)	33.1	36.2	30.1
Toto-base-dVAE (256/16)	20.4	19.1	21.6
Toto-base-SDVAE-400ep (256/16)	29.9 (+9.5)	33.4 (+14.3)	26.4 (+4.8)

Table 14. **DAVIS with SD-VAE tokenizer**: We also test SD-VAE tokenizer trained models on tracking, and it performs much better than dVAE trained *Toto* models.

A.5. Prefix attention

During fine-tuning, we experimented with causal and full attention. On ImageNet, our base model achieved full attn: 82.6% vs causal attn: 82.2%. Even though our models are *not pre-trained with prefix attention*, still able to utilize full attn at fine-tuning. This is an unrealized benefit of training with videos, (a middle token in say, 8th frame won't see the rest half of the 8th frame, but have seen all the tokens from 7th frame, which are similar because of video, hence approximating full attention at pre-training)

A.6. Full fine-tuning

We fine-tuned our models on ImageNet, and performance is close to SOTA, compared to linear probing (where we only use causal attention). But during the fine-tuning, we use full attention.

Model	Params	Dimension	Heads	Layers
a1	14.8M	256	16	12
a2	77.2M	512	16	16
a3	215M	768	16	20
a4	458M	1024	16	24
a5	1.2B	1536	16	28
a6	1.9B	1792	16	32

Table 17. ***Toto* Varients:** We scale *Toto* models by increasing hidden dimension and number of layers linearly while keeping number of heads constant following [55, 65].

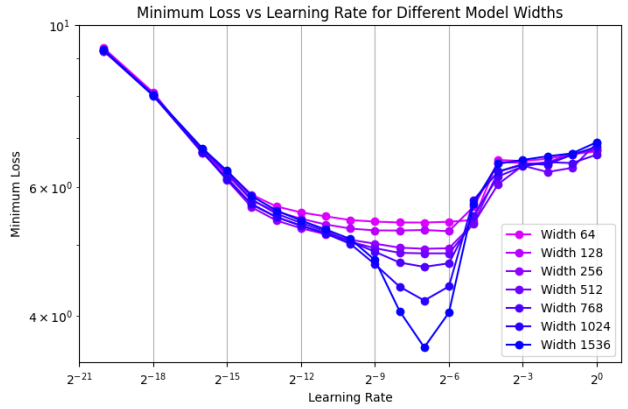


Figure 11. **μ-Parameterization Learning Rate:** We show that μ-Parameterization [65], we can train all width *Toto* models, with an single optimal learning rate of 2^{-7}.

DINO	MoCo v3	BEiT	MAE	*Toto*
82.8	83.2	83.2	83.6	82.6

Table 15. **Full Fine Tuning Performance:** Comparison of different methods performance on ImageNet-1K.

A.7. iGPT vs *Toto* on ImageNet

Table 7 shows ImageNet evaluation performance. However, iGPT [9] models are evaluated only using linear probing. To have a fair comparison, between iGPT and *Toto*, we reevaluated our models using linear probing. Both models have causal attention and are trained on auto-regressive objectives. On the same model sizes, about 1 billion parameters, our achieve 66.2% while the similar iGPT model's ImageNet performance is 65.2%. This fair evaluation suggests the modifications made on *Toto* have clear benefits over iGPT.

Method	Arch	#θ	Top1
iGPT-L [9]	GPT-2	1386	65.2
Toto-1b	LLaMA	1100	66.2

Table 16. **ImageNet Linear Probing Results:** *Toto* performs better than similar size iGPT models.

A.8. μ-Parameterization

To study the scaling behaviours of *Toto* using μ-Parameterization [65]. First we train various models a1-a6 (in Table 17), with hidden sizes (64-1536) and number of layers (12-48), increasing linearly and we used VQGAN tokenizer [15]. Then we tune the learning rate for these models, with fixed depth using μ-Parameterization [65]. Figure 11 shows optimal learning rate of 2^{-7} for all the model widths. Once we find the optimal learning rate, we train a1-a6 models on the mixture of image and video data, as mentioned in Table 2.

A.9. n-gram distribution

In this section, we compare the 2-gram and 3-gram distribution of dVAE [45], VQGAN [15] image tokeizers. We compute 2-gram and 3-gram distributions on the discrete tokens of 10000 ImageNet validation images. Figure 12 and Figure 13 show the distributions of these tokenizers respectively. On 2-gram distribution, dVAE [45] has more discrete combination of tokens compared to both VQGAN-1K and VQGAN-16k tokenizers.

A.10. Attention probing variants on K400

We also evaluate our models and baselines on the Kinetics 400 dataset using a variant of attention probing. In the main paper, we use attention probing, with only learning W_k, W_v matrices, and a single learnable query vector. We also test with cross attention with MLP layers as the attention classifier, to give more capacity to the learnable head. Table 18 show the performance on the attention classifier with an additional MLP head. This helps to improve performance across over all models.

Method	Arch	Top1
Hiera [49]	Hiera-L/14	74.2
Hiera [49]	Hiera-H/14	75.2
VideoMAE [59]	ViT-B/14	65.4
VideoMAE [59]	ViT-L/14	74.8
Toto-base	LLaMA	61.2
Toto-large	LLaMA	65.8
Toto-1b	LLaMA	74.8

Table 18. **K400 Results:** We evaluate our models using cross attention and MLP layer as the classification head. Overall using a high-capacity head improves the performance across all models.

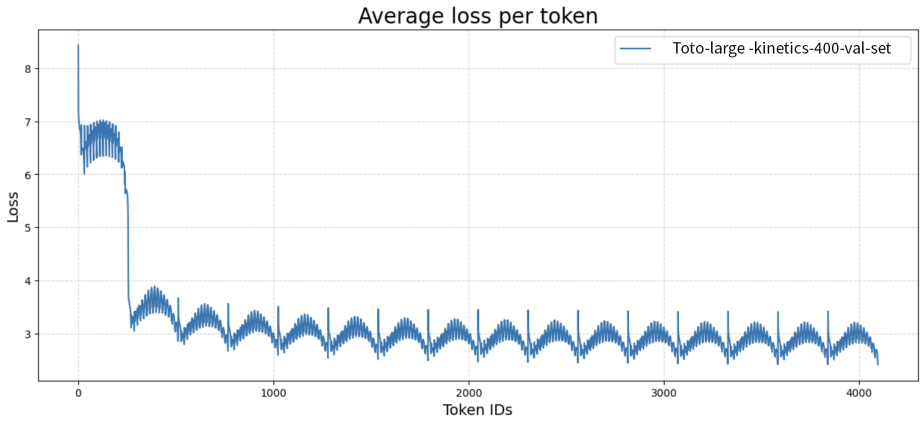


Figure 10. **Average Validation Loss Over Tokens:** We show the average loss per token for kinetics validation set. It clearly shows the redundancy in videos, as the first frame has higher prediction loss, and rest of the frames on average has lower loss than the first frame.

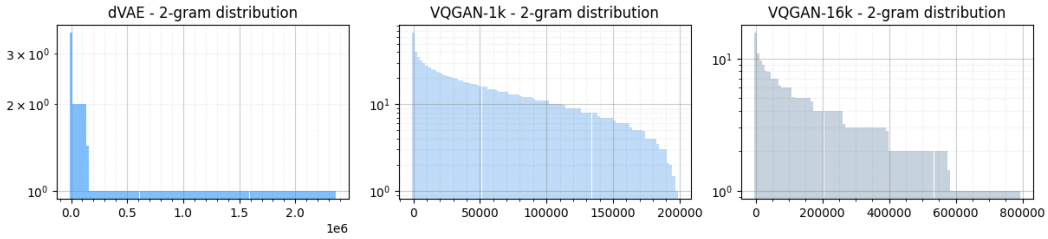


Figure 12. **2-gram Distribution of Various Tokens:** We compute the 2-gram distribution on 10000 images from the ImageNet validation set. Compared to VQGAN 1k and 16k vocabulary tokenizers, the dVAE tokenizer has a larger set of token combinations.

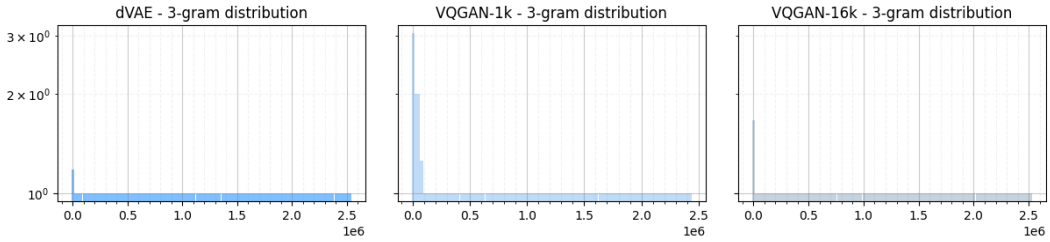


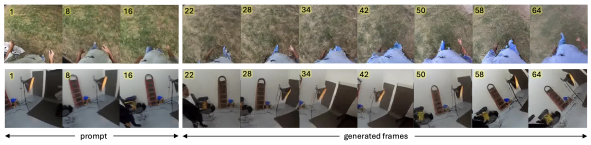
Figure 13. **3-gram Distribution of Various Tokens:** We compute the 3-gram distribution on 10000 images from the ImageNet validation set. All the tokenizers has similar almost flat distribution when it comes to 3-gram tokens.

A.11. Additional Layer-wise Probing Results

We probe the multiple variants of our models at each layer for the best ImageNet performance. First, we test the models on linear probing, on both sizes of 128 and 256 resolution. Figure 14 presents the probing curves of the models trained with attention probing at 128 resolution. Across all models, the performance has a similar behavior to the pre-trained models, with peak performance around the middle of the depth of the model.

A.12. Generation samples

long video generation: we can generate up to 64 frames, first raw: periodic motion, second raw: object permanence (light stand).



Comparison of Generation Quality on UCF101 Additionally, we compare our model's generation quality (FVD) on UCF101. Even though *Toto* is not trained on curated videos, it is competitive with state-of-the-art methods.

Method	FVD
Toto	290
Video-LDM	550
ModelScope	410
PYoCo	355
PixelDance	243
ViD-GPT	278

Table 19. FVD scores on UCF101. Lower is better.

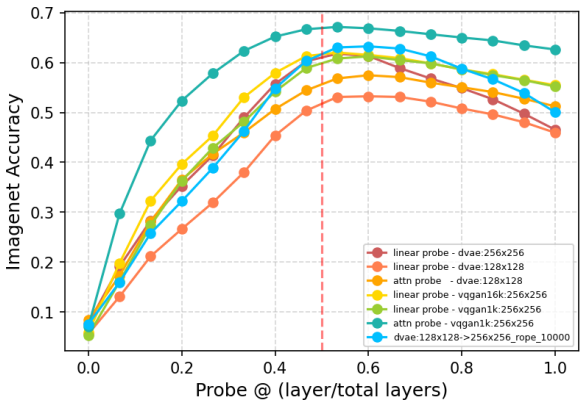
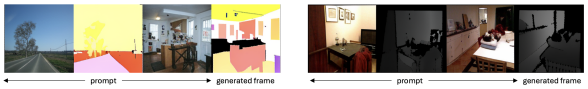


Figure 14. **Training Loss Curves:** We show the training loss curves for multiple variants of our models.

prompting (pre-trained model): shows 3D rotation

prompting (finetuned model): A small 1000-step finetuning leads to a promptable model for various vision tasks.



A.13. Joint training of Images and Videos

During Toto training, we use special tokens to separate video and image data. Fig 15 shows generation for video/image start tokens.

Figure 15. With [1] start token, we see the model generates video, with [3] it generates sequence of images. Please zoom for quality.