

Any-to-Any Video Modeling in Multimodal Abstract Spaces

Muhammad Uzair Khattak¹ Won Jun Kim¹ Reza Abbassi¹ Michael Murphy¹
 Albias Havolli¹ Amir Zadeh³ Chuan Li³ Oğuzhan Fatih Kar¹ Muhammad Ferjad Naeem²
 Andrei Atanov¹ Roman Bachmann¹ Federico Tombari² Amir Zamir¹
¹EPFL ²Google ³Lambda

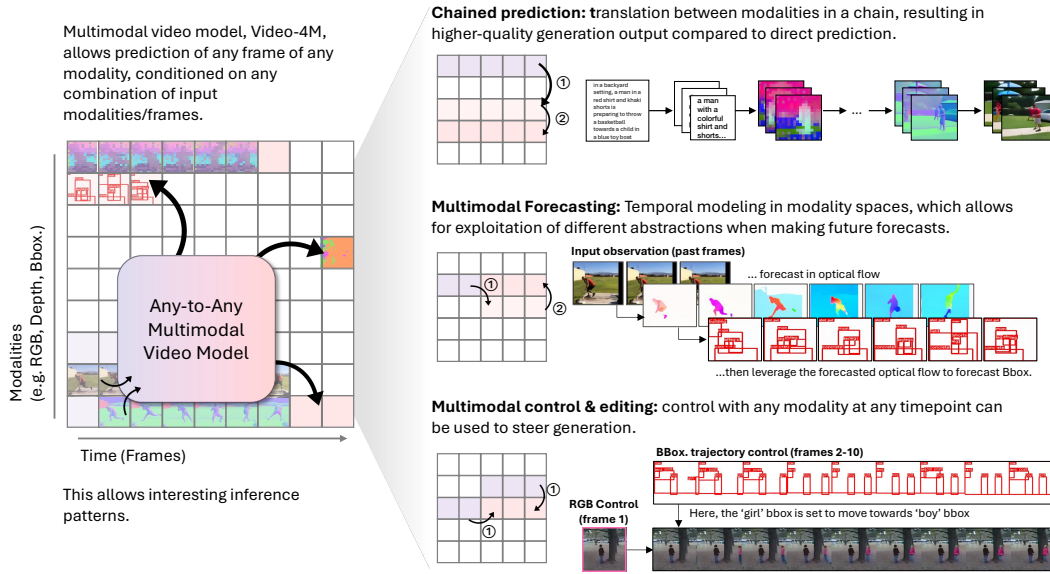


Figure 1: We introduce Video-4M, an any-to-any multimodal video model that is able to predict any modality at any point in time, given any combination of observations from any other modalities. It is thus able to traverse along both the modality and temporal axes, enabling flexible, diverse inference schemes: **chained generations**, where we decompose a task, e.g., caption-to-video, into a chain of simpler tasks of other intermediate modalities; **temporal modeling in abstract modalities**, where given a sequence of past observations, the future can be forecasted in any of the modality spaces; and **multimodal control & editing** where a generation rollout is explicitly constrained by conditioning given as any of the supported modalities specified at any point in time.

Abstract

Video modeling aims to capture spatiotemporal relationships among features of an observation. Most video models focus on modeling this structure using RGB. While this can happen in RGB space, doing so is challenging, as it requires simultaneously learning high-level semantics (e.g., what is happening and where) and low-level appearance (e.g., pixel-level details). It is also often unnecessary: for certain predictive tasks, estimating where an object will be in future is often sufficient for decision-making, and more actionable than generating its image in that future. While certain models have gone beyond RGB by incorporating other modalities (e.g., depth and segmentation maps), jointly modeling a large number of modalities and studying the consequences for the relationships between them is of fundamental interest. In this work, we study the role of multimodal

modeling for video. We first develop a generic multimodal any-to-any video model, which supports a diverse set of modalities: depth, surface normals, optical flow, bounding boxes, text, and neural network feature maps etc. This encapsulates multiple specialist models within one model and performs diverse tasks (e.g., depth estimation or caption-to-video). It also enables novel capabilities, such as temporal modeling through any modality, and mappings for which no specialist exists (e.g., depth to caption). We explore three instantiations of this framework. In chained generations, we decompose a task, e.g., caption-to-video, into a chain of simpler tasks of other intermediate modalities. We find that chained inference yields more consistent and higher-quality generations than direct generation. Secondly, we explore *temporal modeling in abstract modalities* and find that it can lead to more accurate forecasting of information such as distances (via depth) and object motion (via bounding boxes) than modeling in RGB modality. Finally, we demonstrate the fine-grained and controllable multimodal generation capabilities of our framework.

1 Introduction

Video modeling aims to capture spatiotemporal relationships among features of an observation. Most video models operate in RGB space, with some variance in input conditioning formats such as caption, class labels, depth, and canny edges [45, 50, 48, 21, 10]. While this can happen directly in RGB space, we posit that RGB is not always be the most suitable space to operate in - it entangles both low-level appearance (pixel details) with high-level semantic changes across time. This is an inherently a difficult task for a video model, as it aims to learn these two aspects simultaneously. In addition, it is also often unnecessary: for certain predictive tasks, estimating where an object will be in future is often sufficient for decision-making, and more actionable than generating its image in that future [3, 52]. Jointly formulating multiple operating spaces for video modeling, and studying the consequences for the relationships between them, is therefore of fundamental interest.

In this work, we explore video modeling using abstract spaces beyond RGB, defined through multimodality. We first develop a generic any-to-any video model, Video-4M (Fig. 1) that supports a diverse set of modalities: depth, normals, optical flow, bounding boxes, text, and neural network feature maps etc. These modalities capture different aspects of an underlying scene, and naturally forms a hierarchy depending on type and complexity of information each captures: depth and normals, for instance, encode more geometric detail about a video than modalities such as captions or feature maps, which are more semantic. Video-4M multiple specialist models within one model and performs diverse tasks (e.g., depth estimation or caption-to-video). It also enables novel capabilities by allowing traversing along both the time and modality axis, such as temporal modeling through any modality, and mappings for which no specialist model exists (e.g., depth-video-to-caption).

We study the usefulness of these multimodal abstractions in three settings. In *chained generations*, we decompose a downstream task, e.g., caption-to-video, into a chain of potentially simpler prediction tasks of other intermediate modalities. We find that chained inference yields more consistent and higher-quality generations than direct generations. Secondly, we explore *temporal modeling in abstract modalities* and find that it can lead to more accurate forecasting of information such as distances (via depth) and object motion (via bounding boxes) than modeling in the RGB modality. Finally, we demonstrate the fine-grained and controllable *multimodal video generation capabilities* of Video-4M (e.g., conditioning output generations using bounding boxes, surface normals, or a combination of both).

In summary, we make the following contributions: (1) We develop Video-4M, an any-to-any multimodal video model. Video-4M allows traversing both both modality and the temporal axis within a single model that allows diverse inference schemes. (2) We demonstrate the benefit of chained generations through intermediate modalities for video generation tasks, and (3) We investigate forecasting in different modality spaces, and show its advantages compared to modeling directly in the RGB modality space.

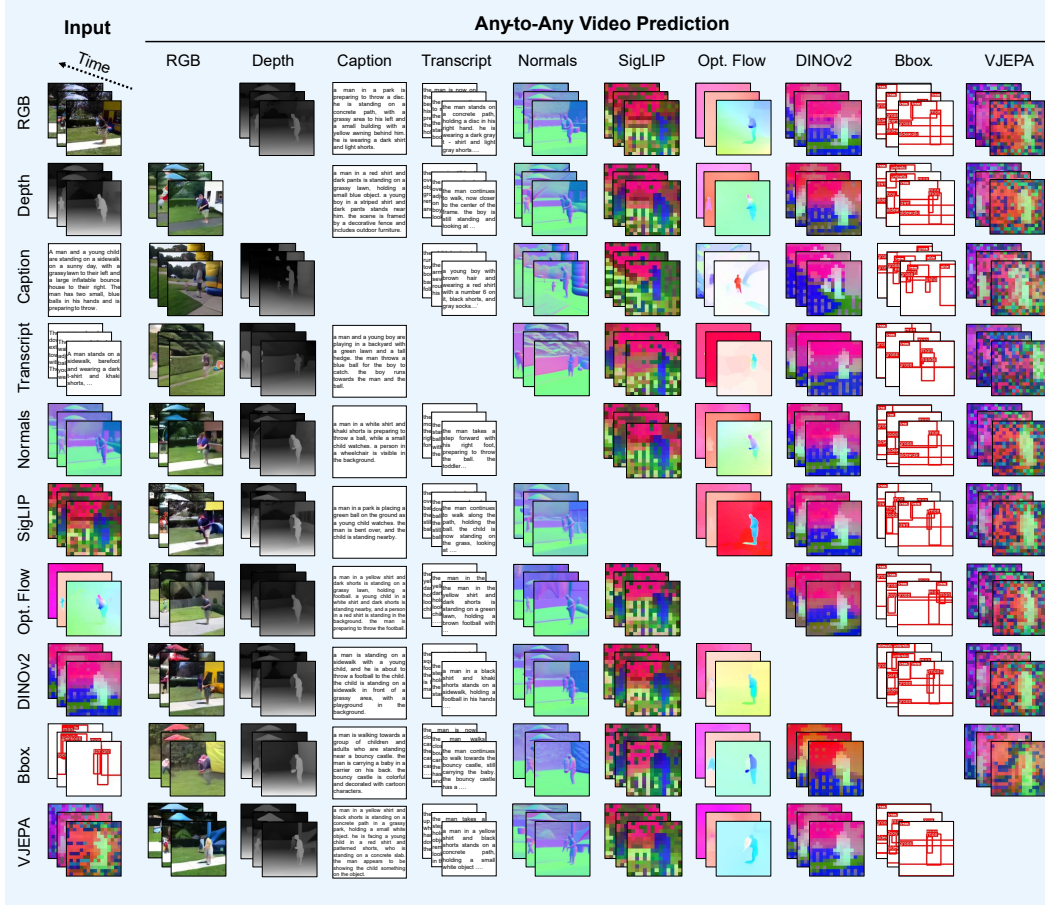


Figure 2: **Any-to-any capabilities.** Video-4M allows predictions from any input modality to any target modality using a single model. Each row shows predictions in a chained manner from different input modalities, which all correspond to the same video sample. High consistency is observed across predictions, even from sparse modalities such as text $\rightarrow$ other and transcription $\rightarrow$ other modalities.

2 Related Work

Multimodality in video modeling. Many video models incorporate different modalities as conditioning inputs to guide video generation: Text-to-video [21, 40, 20, 50], depth/edges/human pose/etc. with ControlNet [51] and ControlVideo [10], sketch sequences/reference videos/motion vectors with VideoComposer [46], text/audio/optical flow/depth with the LLM-based VideoPoet [27]. In these methods, most modalities serve only as conditioning inputs, while videos themselves are modeled in RGB space. While there have been multimodal video models, especially in the service of robotics or driving, which model videos in other modalities such as segmentation maps under specific settings [49, 52, 26, 37, 25], there is comparably little investigation into unified and general multimodal settings that enable exploration into the modality axis of video modeling.

Any-to-any models. These are class of models that can accept any modality from a supported set as input and generate any modality as output. These models aim to address the long-standing goals of model unification and multi-task learning. UnifiedIO [31] and 4M [33, 15] are powerful examples in the image domain. To our knowledge, any-to-any modeling has not been explored in the video domain. Video-4M adopts the design choices of 4M [33] and tailors it for the video domain.

Forecasting in abstract spaces. DINO-WM [52] and DINO-Foresight [24] explore modeling in feature spaces of pre-trained vision models. Chiu et al. [11] and Luc et al. [32] explore predicting the future in semantic segmentation space conditioned on input RGB frames. However, these methods are limited to operating in a single or a few modality spaces. Video-4M supports multiple abstractions, and enables joint analysis of the modality and the temporal axis and adjusting the choice of modalities based on the downstream setting or a particular sample.

3 Method

Development of Video-4M entails several components, namely, multimodal video dataset curation, tokenization, and model architecture & pretraining. Below, we present each component in detail.

3.1 Multimodal pretraining video dataset

Multimodal video data with alignment across more than just video-caption pairs is scarce. We overcome this by pseudolabeling a base video dataset using specialist models. Starting from video-class pairs from the Kinetics-600 dataset [7] (600 action categories across $\sim 400k$ videos of ~ 10 seconds each), we generate 10 different modalities that can be categorized into 4 types: **(i) *geometric modalities*** including depth maps, surface normals and 3D human poses, **(ii) *motion modality*** including optical flow, **(iii) *semantic modalities*** including neural network feature maps from pretrained vision foundation models and bounding boxes, and **(iv) *1D/high-level modalities*** including captions, and transcriptions. Additional details on pseudolabelers and modalities are provided in the appendix B.

3.2 Tokenization

We use discrete tokenizers that transform each modality input into a discrete tokens. This allows us to train a unified model without introducing modality-specific architectural changes, since all modalities are in discrete token space. Below we discuss tokenization details for different modality types.

Modalities with a dense 2D structure. To avoid the excessive token count of frame-wise image tokenizers, we adopt video tokenizers that perform compression both spatially and temporally. For modalities with a 2D spatial structure such as RGB, depth, normals, and optical flow, we use the VidTok tokenizer [41]. This results in 1280 (flattened from a $5 \times 16 \times 16$ token tensor) tokens per 4-second 128 resolution video clip sampled at 4 FPS. For feature maps, we modify the input and output projection layers of VidTok’s architecture to be compatible with the feature map input dimensions, and replace the L2 based reconstruction objective with a cosine similarity loss. For feature-map modalities, we train tokenizers from scratch; for others, we fine-tune the pretrained RGB VidTok tokenizer.

Modalities that are represented sequentially. For modalities such as captions, transcriptions, human-poses and bounding boxes, we treat them as text tokens and encode using WordPiece tokenization [12]. For detections and human poses, we introduce special frame markers [FRAME_1], [FRAME_2], etc., to delineate frame boundaries which allows the model to be aware of the temporal structure. For transcriptions, we use four special temporal tokens [SEC_1], [SEC_2], [SEC_3], and [SEC_4] to mark the start of each one-second transcription segment aligned with the 4-second video clips.

We qualitatively show the pseudo-labeling and tokenizer reconstruction results in the Appendix C. In total, the tokenized pretraining data amounts to approximately 34B tokens, which are used for pretraining Video-4M. We denote this dataset as K600-MM.

3.3 Developing Video-4M

As shown in Fig. 3, Video-4M is an encoder-decoder transformer model and uses a multimodal masked modeling objective [33] for pretraining. During pretraining, one subset is randomly selected from all modality tokens (across both space and time) as input to the encoder and another random subset as masked (or target) tokens which is predicted by the decoder. For dense modalities, tokens are sampled randomly for input and target, while for sequential modalities, we perform span masking [38]. The model is trained with a cross-entropy loss to predict the output tokens.

Adaptations for temporal modalities. We apply the following adaptations to the image-based 4M’s architecture [33, 15] for video modeling.

(i) *Spatio-temporal positional embeddings.* For modalities with explicit spatial and temporal structure (RGB, depth, optical flow, etc.), we use factorized additive learnable spatio-temporal positional embeddings. For sequential modalities such as detections and human poses, which do not have a spatial grid structure, we use learnable 1D positional embeddings and frame marker tokens (e.g., [FRAME_1]) to define the frame boundaries.

(ii) *Time-aware masking during pretraining.* When selecting the input and output token subsets for the encoder and decoder respectively, we ensure that frame marker tokens (e.g., [FRAME_1], [SEC_1]) are never masked for sequential modalities and always remain part of the encoder input. This keeps the encoder context temporally grounded and allows the decoder to infer each token’s temporal location when predicting masked tokens.

(iii) *Masking modes for future prediction.* 4M employs random and span masking [38] for non-sequential and sequential modalities, respectively. Random masking selects a subset of tokens to mask or keep visible from a given modality that simulates a diverse set of video tasks such as inpainting, outpainting, and past or future prediction etc. We introduce two additional masking strategies for non-sequential and one for sequential modalities to reinforce Video-4M future prediction capabilities:

- **Frame-Conditioned Future Frame (FCFF):** The input tokens are sampled causally from the initial frames, and target tokens are taken exclusively from subsequent future frames, also in a causal manner. This enforces a temporal separation between input and output tokens.
- **Frame-Conditioned Any-Future (FCAF):** Input tokens are sampled causally from the initial frames similarly to FCFF, but the output tokens are sampled randomly in a non-causal manner from any of the subsequent future frames.
- **Causal Frame-level Prediction (CFP):** Instead of random span masking, input tokens are selected causally starting from the first token, and masked tokens begin immediately after the last input token. This mirrors FCFF but is designed for sequential modalities. In our ablations, we found that incorporating CFP is essential for our inference strategy of feeding back predicted tokens to encoder, as it simulates such cases during training. Without this masking mode, model generations significantly deteriorates when predicted tokens are fed to the encoder.

During training, one of the three masking modes (random, FCFF, FCAF) is selected with equal probability for non-sequential modalities. For sequential modalities, CFP is sampled alongside the span masking at a probability of 35%. We find that incorporating these frame-level causal masking modes improves performance on RGB generation tasks, as shown later in our ablation studies.

3.4 Chained generation

After pretraining on K600-MM, Video-4M can perform any-to-any generation in multimodal abstract spaces. Given one or multiple input modalities, it can generate any desired target modality. This property gives rise to chained predictions,¹ where the model is iteratively tasked to generate one modality at a time, feeding each prediction back to the model as additional context before predicting the next target modality. We illustrate this technique in Fig. 4.

Chaining enables consistent generations across modalities, e.g., depth that is in-agreement with predicted semantics, and allows multimodal conditioning at test time without requiring annotated intermediate modalities that often require external models. In addition, we can use chaining to decompose a task into a sequence of potentially simpler sub-tasks by routing through different abstract spaces. Below, we discuss one such instantiation for the RGB generation tasks.

RGB video generation using chaining in abstract spaces. We consider the class-, caption-, and frame-conditional (1 and 5 frames) video generation tasks. Video generation can be decoupled into sequential sub-tasks of first modeling high-level semantic content (such as object positions, motion patterns, and scene composition), followed by modeling the low-level pixel details. Instead of jointly performing these steps (direct RGB generation), decomposing this into intermediate abstract spaces can potentially make it more tractable. Using an initial condition such as a caption, Video-4M generates intermediate abstract spaces sequentially before generating RGB in the final step.

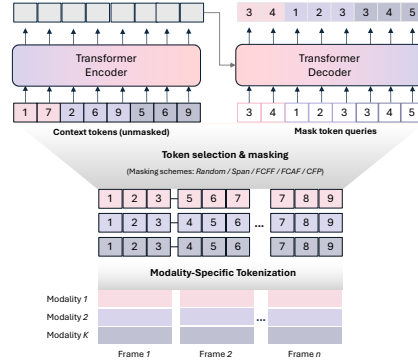


Figure 3: Video-4M framework. Context tokens (*i.e.*, input to the transformer encoder) and masked tokens (*i.e.*, target tokens to be predicted by the transformer decoder) are sampled from different modality tokens. The masking strategy is a combination of random masking and future prediction masking modes tailored for both sequential and non-sequential modalities.

¹Chaining is broadly supported in any-to-any models, as demonstrated in 4M and 4M-21 [33, 15] for the image domain.

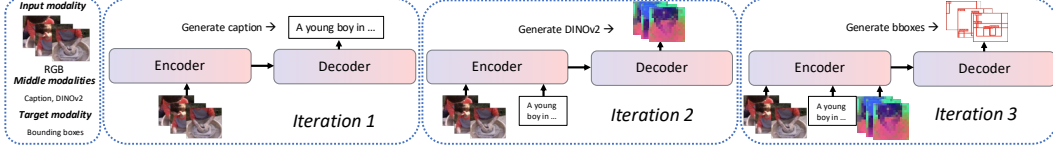


Figure 4: **Chained generations with Video-4M.** We illustrate the chaining capability of Video-4M for the task of predicting bounding boxes given RGB video. Using caption and DINOv2 as the middle modality, Video-4M predicts one modality at a time in a sequential manner. Though each modality is visualized here, the model operates in shared token space of all modalities.

Chain design. We use a coarse-to-fine strategy: the chain starts from high-level semantic abstractions towards modalities that represent low-level geometric structure. We empirically find that such chains result in improved RGB generation compared to randomly assigning the chain order. To reduce the search space, we select a subset of Video-4M’s modalities and categorize them into three groups: sparse (captions, transcriptions), semantic (SigLIP 2, bounding boxes), and geometric (depth), and use Sparse $\rightarrow$ Semantic $\rightarrow$ Geometric chains across all tasks.

Optimal chain design. The ideal chain configuration is likely task-dependent. With n abstract spaces, there exist $n!$ possible orderings, making an exhaustive search computationally prohibitive as n grows. In this work, we focus on RGB video generation and demonstrate that our coarse-to-fine chaining strategy consistently improves generation quality. We leave a systematic search for optimal chain configurations across all tasks to future work.

3.5 Forecasting through modality spaces

Video models can forecast rollouts into the future given past frames. This *future prediction* ability has sparked interest in their capacity as world models [5, 1]. While useful future prediction models with applications in robotics and driving are already emerging, existing models roll out the future in one or two modalities [49, 52, 26, 37, 25]. Video-4M allows exploration of the effect of extending the prediction axis to many modalities, specifically RGB, depth, normal, bounding boxes, transcription, DINOv2, and SigLIP 2 for this investigation.

Setup: we experiment with forecasting through different modalities by giving as input some observed RGB frames and using Video-4M to generate the future frames of a chosen target modality. We test both direct prediction of the target modality (e.g. RGB $\rightarrow$ future Depth) or a single-hop chained prediction (e.g., RGB $\rightarrow$ future DINOv2 $\rightarrow$ future Depth). We evaluate the generated future frames.

Evaluation: There are many valid rollouts into the future given a single past sequence. This poses a problem for evaluating different generated rollouts because each video in the test set provides only one realized future. We utilize Gneiting & Raftery’s energy score [17], which provide a quantification of the quality of a predictive distribution against a single realized outcome. Given sampled rollouts $\hat{x}^{(1)}, \dots, \hat{x}^{(K)}$ and the observed rollout x , the energy score is calculated as follows: $ES = \frac{1}{K} \sum_{i=1}^K d(\hat{x}^{(i)}, x) - \frac{1}{2K^2} \sum_{i=1}^K \sum_{j=1}^K d(\hat{x}^{(i)}, \hat{x}^{(j)})$, where d is a task-appropriate distance metric. Explanation and intuition behind this score is explained in Appendix F. For the distance function, we use square root of the scale-invariant logarithmic loss (Sqrt-SiLog) [14], L2 in occupancy maps, L2 in DINOv2 feature space, and L2 distance in Inception V3 feature space for depth, bounding boxes, DINOv2, and RGB respectively.

4 Experiments

4.1 Video-4M as a generalist video foundation model

After pretraining, Video-4M encapsulates multiple video tasks within a *single* neural network. As shown in Fig. 5, Video-4M predictions for various modalities closely align with those of the specialized pseudolabelers used to generate its pretraining data. Given RGB video as input, Video-4M can generate different modalities in a highly consistent manner. We provide the results of the quantitative evaluation in the Appendix D.

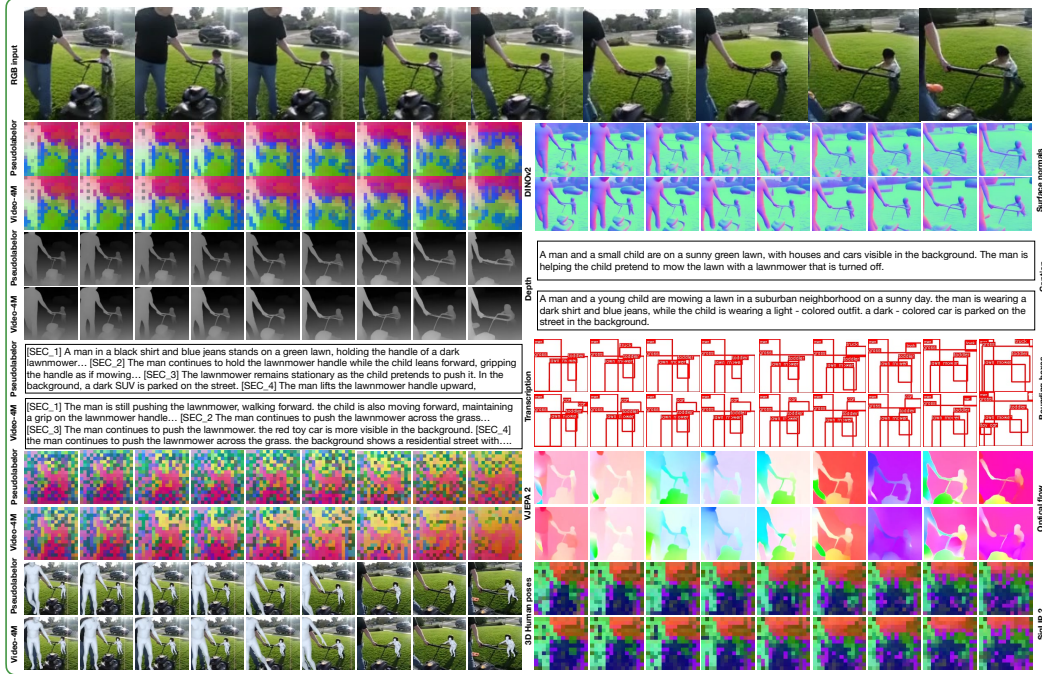


Figure 5: Video-4M as a generalist video model for RGB $\rightarrow$ other tasks. While being a *single* model, Video-4M can perform different video tasks given RGB video as input. We compare its predictions with those of its pseudolabelers. Note that Video-4M uses tokenized 128-resolution input videos, while pseudolabelers use their native, often higher-resolution, pre-tokenized input videos.

In addition to RGB-conditioned tasks, the any-to-any nature of Video-4M allows traversing along the modality axis with arbitrary selection of input and output modalities. In Fig. 2, we demonstrate this by showing one-to-all generation, where Video-4M sequentially generates each modality from a single input. With consistent predictions across different modalities, Video-4M successfully learns the supported abstract spaces. In the following section, we show the usefulness of these abstract spaces for RGB video generation task.

4.2 Video generations using abstract spaces

RGB video generation requires jointly modeling high-level semantics in space and time, and fine-grained pixel structure and appearance. Using Video-4M, we show that RGB generations can benefit from chaining through abstract spaces. For chained generations, we adopt the coarse-to-fine chain design as discussed in Sec. 3.4. As our main baseline, we use Video-4M to generate RGB tokens conditioned only on the input denoted as Video-4M-direct. Both direct and chained generations use the same trained model. We also train a variant of Video-4M using only RGB, caption, and class modalities, using the same pretraining schedule as Video-4M. This serves as our RGB-only baseline, and allows us to disentangle the effect of training exposure from the benefit of chaining.

We show the results in Tab. 1. Across different model sizes, Video-4M with chaining in abstract spaces outperforms the direct prediction baseline. Overall, the trend is consistent across class-to-RGB, caption-to-RGB, and frame-conditional RGB generation tasks. For direct baselines, we optimize the generation hyperparameters using grid search, including CFG scale to optimize their performance.

Table 1: **Chained vs. direct modeling for RGB video generation.** Overall, chained predictions improve over directly modeling in RGB space.

Model	Video Generation				Video Prediction	
	Class $\rightarrow$ RGB		Caption $\rightarrow$ RGB		Frame conditional	
	FVD	Top 1 Acc.	FVD	Text Align.	1-frame	5-frame
RGB Baseline	490.6	40.9	304.3	19.2	318.6	172.6
Video-4M-B Direct	325.0	53.3	418.6	17.9	230.0	138.0
Video-4M-B Chained	263.7	52.9	250.0	19.2	155.0	119.0
Video-4M-L Direct	360.3	49.5	231.0	20.6	229.3	132.3
Video-4M-L Chained	249.0	57.1	191.3	20.2	143.0	105.0

Qualitative results. We show generation results in Fig. 6 for different tasks. Videos generated using the direct generation baseline often exhibit noticeable structural artifacts. While the overall semantics are largely preserved (e.g., a person attempting to perform a

side kick when conditioned on this action), direct prediction struggles with low-level details such as object boundaries, particularly in frames where sudden motion occurs.

In chained generations, intermediate modalities, such as semantic features and bounding boxes, outline the semantics and motion trajectories, while geometric modalities, like depth, contribute in distinguishing objects from each other. We hypothesize that this leads to RGB generations with coherent motion and fine-grained details. Additional qualitative results are shown in Appendix E.

Table 2: Which modality group matters most in chaining?

Configuration	Video Generation		Video Prediction	
	Class→RGB	Caption→RGB	Frame conditional	
	FVD ↓	FVD ↓	1-frame ↓	5-frame ↓
Sparse (Caption → Transc. → RGB)	431.0	367.3	198.3	122.3
Semantic (SigLIP 2 → BBox → RGB)	266.3	191.7	150.7	107.7
Geometric (Depth → RGB)	439.7	309.3	209.0	122.0
Coarse-to-fine chain (default)	249.0	191.3	144.0	102.0

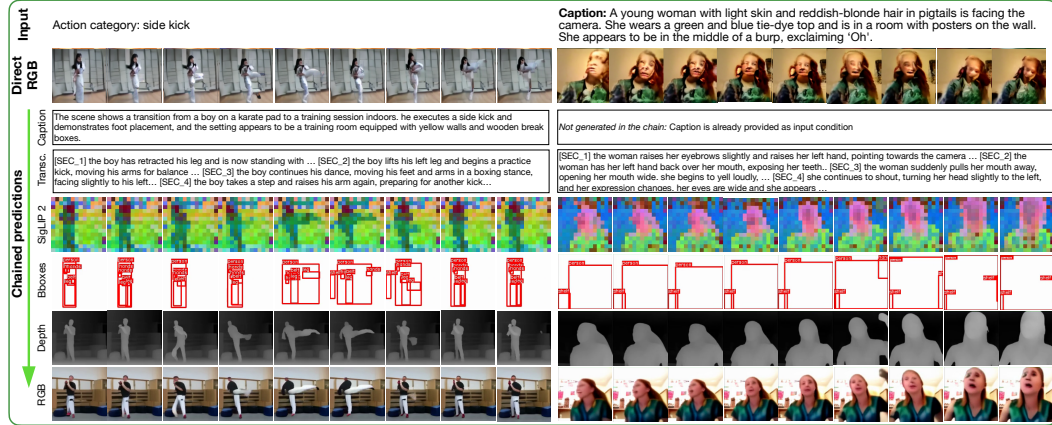


Figure 6: **RGB video generations with chained predictions in abstract spaces.** For different RGB tasks (*left*: class-to-video, and *right*: caption-to-video). Generating intermediate abstract modalities such as transcriptions, SigLIP 2 feature maps, detections, and depth as middle modalities leads to improved RGB generations. In contrast, frames generated using direct RGB often show artifacts, especially for sudden motion changes, and lack geometric consistency.

Analysis on chain design. We study two questions regarding chain design: which abstract space group contributes most, and how the ordering of groups affects final generation quality. In Tab. 2, we perform generation using individual modality groups and compare it with full chain. Semantic modalities contribute most to generation quality, compared to the geometric and sparse groups. Semantic modalities like SigLIP 2 represent global semantics of the video, which are more readily translated to the RGB space, compared to geometric modalities like depth which emphasizes more on low-level structural details. In Tab. 3, we investigate the effect of ordering of modalities in the chain on the quality of the generation.

We reverse the chain order such that geometric modalities are predicted first, followed by sparse and semantic modalities. This reverse ordering performs worse than the semantic-first chain order. From qualitative comparisons, we note that when the geometric modalities are generated first, subsequent modality predictions are constrained by the the geometric modality. This strictly limits the ability of semantic modalities to flexibly guide the generation trajectory (except for those details where geometric modalities does not contain, such as color info), even when it is part of the chain. Therefore, not only the presence of semantics but also their order in the chain design significantly affects generation quality. Refer to Appendix E for additional analysis.

Table 3: Order of modality chain matters.

Chain ordering	FVD ↓	Text Align. ↑
Caption → Geometric → Semantic → Sparse	239.0	19.9
Caption → Sparse → Semantic → Geometric (default)	191.6	20.2

5 Forecasting through different modalities

Fig. 7 shows the results of forecasting evaluated with energy scores [17]. We first note the expected findings: (1) for most forecasts, the energy scores increase with longer prediction horizon, since predictions of the farther future is less likely to be close to the observed rollout; (2) the static baseline,

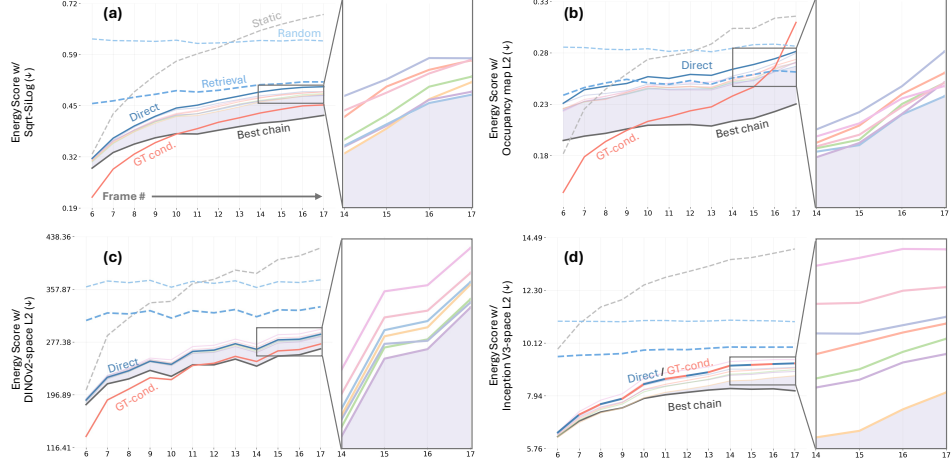


Figure 7: **Forecasting future (a) Depth, (b) Bounding Boxes, (c) DINOv2, and (d) RGB frames given 5 starting frames of RGB.** **Direct:** prediction of the future frames of the target modality directly from the 5 RGB frames, e.g. RGB[:5] $\rightarrow$ Depth[5:17]. Chained predictions (e.g. RGB[:5] $\rightarrow$ DINOv2[:17] $\rightarrow$ Depth[5:17]) with RGB, optical flow, SigLIP 2, Bbox., DINOv2, transcription, depth, normal as middle modalities. **GT-cond:** prediction conditioned on the 5 ground-truth frames of the target modality, e.g. Depth[:5] $\rightarrow$ Depth[5:17]. **Static** baseline: last input frame is used as forecast. **Random** baseline: 16 random ground-truth test samples are used as forecasts. **Retrieval** baseline: 16 closest neighbours selected from the ground-truth test set based on the conditioning frames, based on the modality-specific distance function, are used as forecasts. **Best chain:** result if the best modality chain for each sample was used. : Gap between per-sample best chain result and the best single chain result.

which is just simple repetition of the last given frame used as the ‘forecast’, performs well in the initial few frames but rapidly degrades over time while the random and retrieval baselines, which are random and nearest-neighbour samples from the ground-truth test set respectively, show static performance over the entire horizon – with model prediction-based forecasts getting closer to the retrieval baseline with time; and (3) the ground truth-conditioned prediction performs well especially in the earlier frames because past ground-truth frames allow for closer predictions to the ground-truth future frames.

One finding is that the chained predictions generally perform better than the direct predictions for forecasting, reinforcing the findings in Sec. 4.2. But more importantly, while the performance gaps between different chains are not very significant – especially in the context of magnitude of the energy score difference between the two lower bound baselines and the GT-conditioned prediction – the performance of the best per-sample chain is significantly better than any of them, overtaking the ground truth-conditioned predictions in the later frames and showing much slower degradation over time. This suggests that while different modality chains show similar performance on average, the optimal modality chain is highly sample-dependent. It strengthens the hypothesis that finding the right abstraction (represented by each modality) through which to make predictions can lead to significant improvements in forecast quality. However, the problem of finding the best chain per sample was non-trivial, so in this investigation we limit our contribution to the demonstration of the potential of optimizing over modality chains and leave the chain optimization investigation to future work.

6 Multimodal video generation capabilities

Different modalities capture complementary spatio-temporal information in videos. As Video-4M can accept multiple input modalities as conditions, this enables many exciting video editing and generation possibilities through multimodal input probing. We demonstrate a subset of these capabilities in Fig. 8. For instance, given the first RGB frame, we can generate its future RGB frames by conditioning the video using caption as the probing modality. By changing the caption (e.g.,

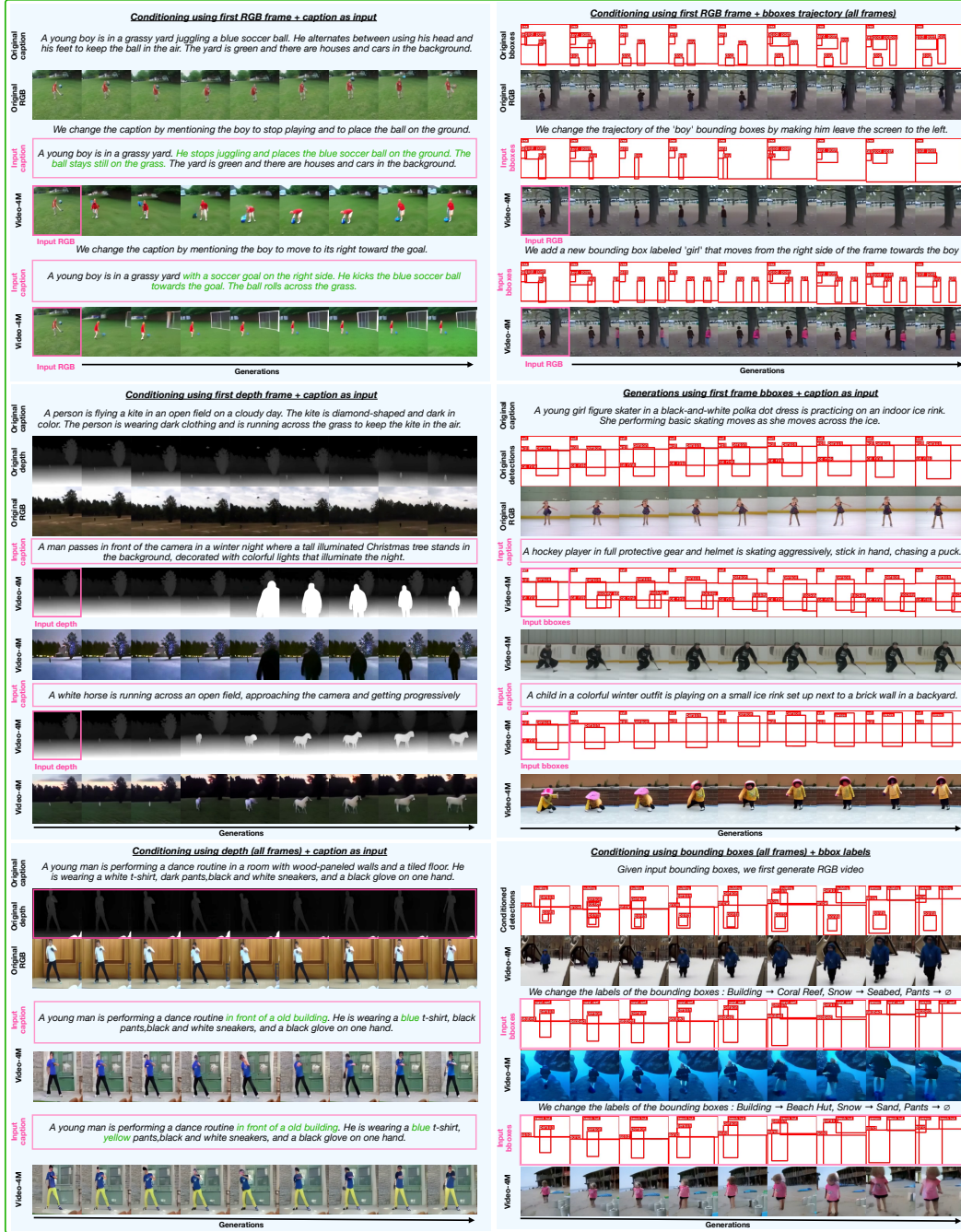


Figure 8: **Fine-grained and controllable video generation using multimodal input conditioning.** We showcase several instantiations of video generation through multimodal input probing. Video-4M allows any combination of modalities as input to generate output videos. In most scenarios, we provide one conditioning modality (e.g., first frame of depth/RGB/bounding boxes), and vary the output generation using a probing modality (e.g., caption/bounding box labels or coordinates).

stopping the football juggling or kicking the soccer ball), we can generate plausible future frames of video that all start from the same input image. Moreover, instead of an RGB frame as input, we can generate videos by providing the first frame in other modalities such as depth or bounding boxes. This flexibility to probe and condition across modalities allows Video-4M to support a broad range of controllable video generation tasks beyond what any single modality could achieve alone.

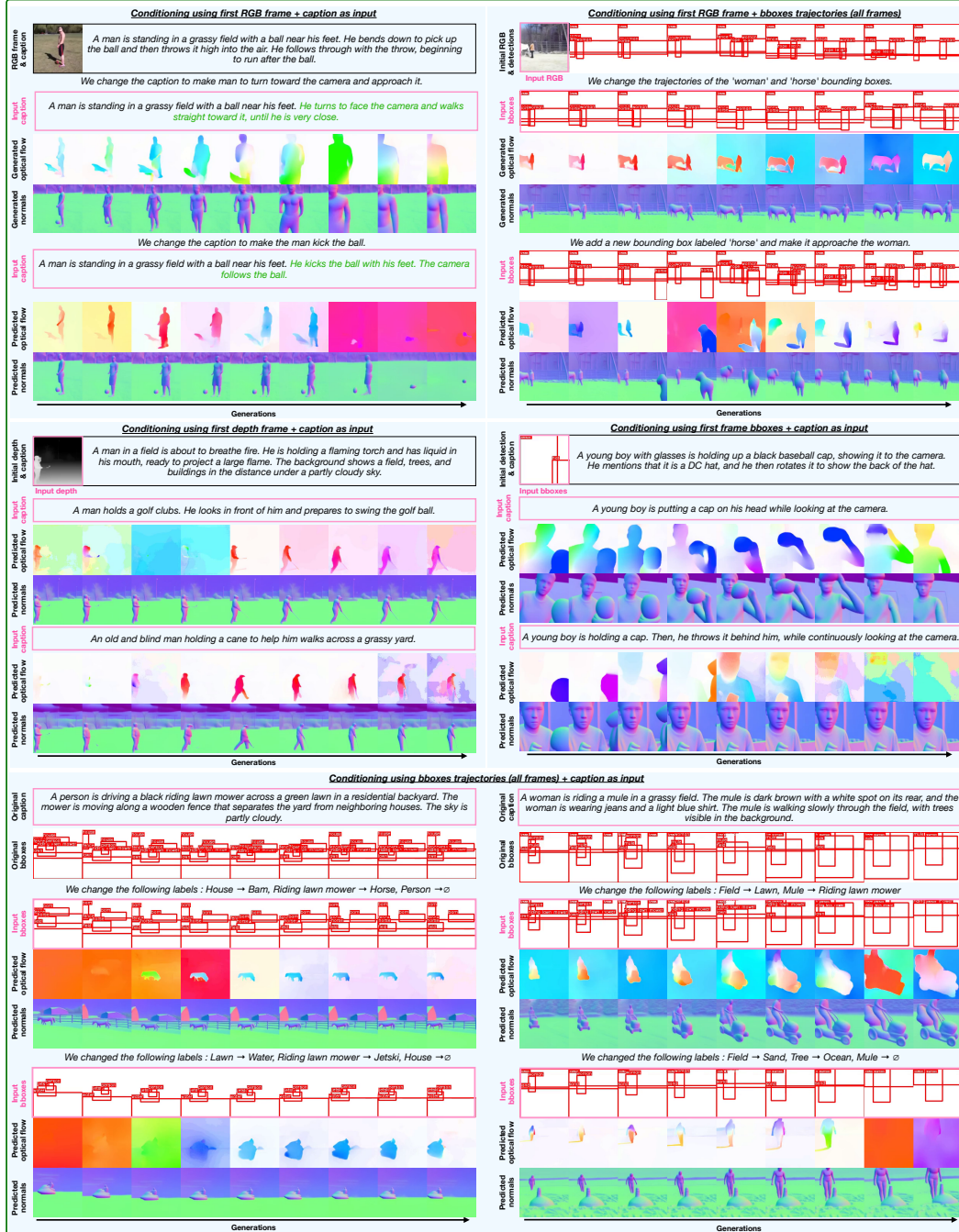


Figure 9: **Multimodal input probing for controllable generations in non-RGB modality spaces.** We extend our probing analysis to non-RGB modality spaces as outputs. For instance, we can control how surface normals in the future will look by varying the probing modality, such as captions, the first RGB frame, or bounding boxes.

Unlike most video models that generate in RGB space, Video-4M can generate and predict observations directly in modality spaces of interest, while still supporting multimodal input conditioning. As shown in Fig. 9, we use different multimodal inputs but generate in the output space of other modalities such as optical flow and surface normals. For instance, using the first RGB frame, we can generate future surface normals and optical flow by probing with the caption modality. These scenarios are similar to the previous paragraph, except that the output space is non-RGB. This

decoupling of input and output modalities highlights Video-4M’s generality beyond RGB-centric video generation.

7 Conclusion

We develop Video-4M, an any-to-any multimodal video model that supports a diverse set of abstract spaces, moving beyond the conventional RGB-centric video modeling paradigm. These abstract spaces are defined through multimodality, which provides complementary information about the underlying temporal observation. Using Video-4M, we investigate the role of abstract spaces in video modeling for generation and forecasting. For generation, we show the output quality and consistency could be improved by routing through different abstract spaces using chained generations, as the model is relieved of having to generate both semantics and low-level details in a single step. For forecasting, we show that different modality spaces allow for unique rollouts such that the sample-dependent optimal chain enables significantly higher-quality forecasts, compared to forecasting directly in a single modality space. We also demonstrate the flexible input-output capabilities of Video-4M for fine-grained and controlled video editing tasks.

References

- [1] Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. [arXiv preprint arXiv:2511.00062](#), 2025.
- [2] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. [arXiv preprint arXiv:2506.09985](#), 2025.
- [3] Federico Baldassarre, Marc Szafraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the features: Dino as a foundation for video world models. [arXiv preprint arXiv:2507.19468](#), 2025.
- [4] Yanrui Bin, Wenbo Hu, Haoyuan Wang, Xinya Chen, and Bing Wang. Normalcrafter: Learning temporally consistent normals from video diffusion priors. [arXiv preprint arXiv:2504.11427](#), 2025.
- [5] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. [OpenAI Blog](#), 1(8):1, 2024.
- [6] Neil Burgess, Jelena Milanovic, Nigel Stephens, Konstantinos Monachopoulos, and David Mansell. Bfloat16 processing for neural networks. In [2019 IEEE 26th Symposium on Computer Arithmetic \(ARITH\)](#), pages 88–91, 2019.
- [7] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. [arXiv preprint arXiv:1808.01340](#), 2018.
- [8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In [proceedings of the IEEE Conference on Computer Vision and Pattern Recognition](#), pages 6299–6308, 2017.
- [9] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. [arXiv preprint arXiv:2501.12375](#), 2025.
- [10] Weifeng Chen, Yatai Ji, Jie Wu, Hefeng Wu, Pan Xie, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. [arXiv preprint arXiv:2305.13840](#), 2023.
- [11] Hsu-kuang Chiu, Ehsan Adeli, and Juan Carlos Niebles. Segmenting the future. [IEEE Robotics and Automation Letters](#), 5(3):4202–4209, 2020.

- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In North American Chapter of the Association for Computational Linguistics, 2019.
- [13] Sai Kumar Dwivedi, Yu Sun, Priyanka Patel, Yao Feng, and Michael J Black. Tokenhmr: Advancing human mesh recovery with a tokenized pose representation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1323–1333, 2024.
- [14] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Advances in Neural Information Processing Systems, 2014.
- [15] Bachmann et al. 4M-21: An any-to-any vision model for tens of tasks and modalities. Neurips, 2024.
- [16] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. Advances in neural information processing systems, 35:35946–35958, 2022.
- [17] Tilman Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association, 102(477):359–378, 2007.
- [18] Google. Gemini, 2023.
- [19] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. ArXiv, abs/1706.02677, 2017.
- [20] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303, 2022.
- [21] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. Advances in neural information processing systems, 35:8633–8646, 2022.
- [22] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 21807–21818, 2024.
- [23] Glenn Jocher, Jing Qiu, Mengyu Liu, Shuai Lyu, Fatih Cagatay Akyon, and Muhammet Esat Kalfaoglu. Ultralytics yolo26: Unified real-time end-to-end vision models. arXiv preprint arXiv:2606.03748, 2026.
- [24] Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Dino-foresight: Looking into the future with dino. arXiv preprint arXiv:2412.11673, 2024.
- [25] Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Advancing semantic future prediction through multimodal visual sequence transformers. In Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR), pages 3793–3803, June 2025.
- [26] Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. DINO-foresight: Looking into the future with DINO. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025.
- [27] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. arXiv preprint arXiv:2312.14125, 2023.
- [28] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In Proceedings of the IEEE/CVF international conference on computer vision, pages 19948–19960, 2023.

- [29] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In European Conference on Computer Vision, pages 38–55. Springer, 2024.
- [30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In International Conference on Learning Representations, 2019.
- [31] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 26439–26455, 2024.
- [32] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In Proceedings of the IEEE international conference on computer vision, pages 648–657, 2017.
- [33] David Mizrahi, Roman Bachmann, Oğuzhan Fatih Kar, Teresa Yeo, Mingfei Gao, Afshin Dehghan, and Amir Zamir. 4M: Massively multimodal masked modeling. In Advances in Neural Information Processing Systems, 2023.
- [34] Henrique Morimitsu, Xiaobin Zhu, Roberto M Cesar, Xiangyang Ji, and Xu-Cheng Yin. Dpflow: Adaptive optical flow estimation with a dual-pyramid framework. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 17810–17820, 2025.
- [35] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In ECCV, 2012.
- [36] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193, 2023.
- [37] Enrico Pallotta, Sina Mokhtarzadeh Azar, Shuai Li, Olga Zatsarynna, and Juergen Gall. Syncvp: Joint diffusion for synchronous multi-modal video prediction. In Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR), pages 13787–13797, June 2025.
- [38] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 21(1):5485–5551, 2020.
- [39] Noam Shazeer. GLU variants improve transformer. ArXiv, abs/2002.05202, 2020.
- [40] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792, 2022.
- [41] Anni Tang, Tianyu He, Junliang Guo, Xinle Cheng, Li Song, and Jiang Bian. Vidtok: A versatile and open-source video tokenizer. arXiv preprint arXiv:2412.13061, 2024.
- [42] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.
- [43] Michael Tschanen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. arXiv preprint arXiv:2502.14786, 2025.
- [44] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. arXiv preprint arXiv:1908.00463, 2019.
- [45] Hanyu Wang, Saksham Suri, Yixuan Ren, Hao Chen, and Abhinav Shrivastava. Larp: Tokenizing videos with a learned autoregressive generative prior. arXiv preprint arXiv:2410.21264, 2024.

- [46] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. Advances in Neural Information Processing Systems, 36:7594–7611, 2023.
- [47] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multi-modal understanding and generation. In International Conference on Learning Representations, volume 2024, pages 42055–42079, 2024.
- [48] Yuqing Wang, Tianwei Xiong, Daquan Zhou, Zhijie Lin, Yang Zhao, Bingyi Kang, Jiashi Feng, and Xihui Liu. Loong: Generating minute-level long videos with autoregressive language models, 2024.
- [49] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. arXiv preprint arXiv:2310.06114, 2023.
- [50] Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David C. Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation. 2023.
- [51] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In Proceedings of the IEEE/CVF international conference on computer vision, pages 3836–3847, 2023.
- [52] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning, 2024.

Appendix

A Additional results for multimodal video modeling capabilities

In Sec. 6, we demonstrated the fine-grained video generation capabilities of Video-4M using chained generations. Here we provide additional qualitative results and specifically show comparisons with the direct-generation inference strategy. Fig. 10 shows the results for video generations by conditioning on the first RGB frame and caption. For direct generations, the resulting videos are of lower quality when compared to chained-generation variants. These observations are consistent with the chained-generation results and analysis for class-, caption-, and frame-conditional RGB generation tasks that we discussed in Sec. 4.2.

B Pseudolabeling

To construct our multimodal video dataset, we start from the Kinetics-600 dataset [7] and obtain additional modalities through pseudolabeling using task-specific models. Tab. 4 provides an overview of all the modalities supported by Video-4M.

Modalities with a dense 2D structure. For modalities like feature maps, depth, normals, and optical flow, we directly use off-the-shelf models to generate them. During pseudolabeling, we generate modalities using the native resolution of the videos and later perform spatial resizing in the modality space to obtain 128×128 resolution samples. For feature maps, we use the default resolution of the neural network (224 for SigLIP 2 [43] and DINOv2 [36], and 256 for VJEPa 2 [2]), as inference at 128×128 resolution would result in out-of-distribution inputs for these models.

Modalities that are represented sequentially. For generating captions and transcriptions, we use Gemini-2.0 Flash [18] by providing the input video alongside a prompt instruction. Additionally, we obtain metadata and object tags for each video in our dataset.

Object tags consist of category names of active (e.g., objects involved in the main actions) and passive (e.g., static or background objects) entities. For the bounding box modality, we use object tags from Gemini as class labels for Grounding DINO, which generates the detections. The diverse nature of object tags also enables open-vocabulary detection capabilities in Video-4M.

For human poses, we first generate 3D human poses using TokenHMR [13]. We noticed that the pseudolabelers also generate complete human poses for partially visible human instances in the videos (e.g., where only hands or fingers are visible). Additionally, there were false positive predictions for out-of-distribution videos. We filter such detections by first estimating 2D poses [23] on the predicted humans and retaining only those instances for which the 2D keypoint prediction confidence for the nose, shoulders, and hips is above 0.85. This significantly reduced the number of human pose predictions for partially visible human instances, for which complete pose prediction was deemed invalid.

Table 4: **Overview of Modalities.** We group different modalities into four categories based on the type of information they capture. In total, Video-4M supports 12 multimodal abstractions. For class category, we learn a token embedding layer.

Modality	Pseudo Labeler	Tokenizer
RGB	N/A	ST FSQ
<i>Geometric modalities</i>		
Depth	VDA-v2 [9]	FSQ
Surface Normal	NormalCrafter [4]	FSQ
3D Human Pose	TokenHMR [13]	VQ-VAE
<i>Motion modality</i>		
Optical Flow	DPFlow [34]	FSQ
<i>Semantic modalities</i>		
SigLIP 2	SigLIP 2 [43]	FSQ
DINOv2	DINOv2 [36]	FSQ
V-JEPa 2	V-JEPa 2 [2]	FSQ
Detection	Grounding DINO [29]	WordPiece
<i>1D/High-Level Modalities</i>		
Caption	Gemini-2.0 Flash [42]	WordPiece
Class category	N/A	Embed.
Transcriptions	Gemini-2.0 Flash [42]	WordPiece

C Tokenization

Video-4M is pretrained on a multimodal video dataset that is tokenized into a discrete token space. As discussed in Sec. 3.2, we train modality-specific discrete tokenizers. In Fig. 11, we show the reconstruction quality of our trained tokenizers for various modalities. As we sample videos at 4 FPS, most of our training samples contain abrupt motion changes (e.g., a person in a standby position in one frame and in a kicking position in the very next frame), which is more challenging for the tokenizer to reconstruct, compared to video samples where motion change is more gradual (e.g., a

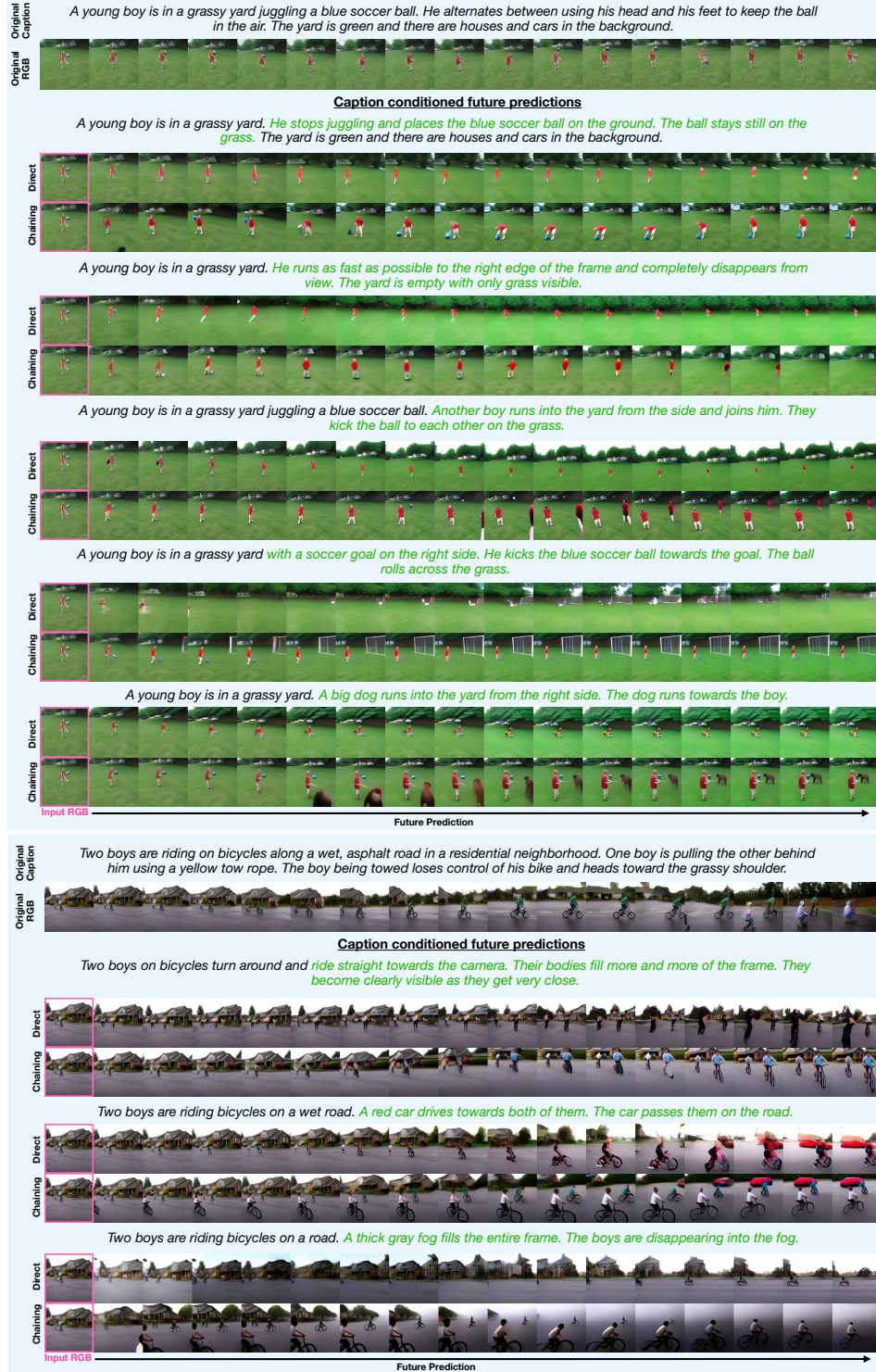


Figure 10: **Caption-conditioned with direct and chaining future predictions.** Given a single RGB frame and a caption text as inputs, we use Video-4M to predict different future RGB trajectories by changing the caption. Notice how the chaining improves the future prediction.

person using a sewing machine). We qualitatively show reconstructions using these two representative cases. For more abstract modalities such as feature maps, depth, or optical flow, reconstructions

are highly consistent with the raw modalities. For the RGB modality, reconstructed videos contain small artifacts when fast or abrupt motion changes occur. Note that each tokenizer is finetuned on the K600-MM dataset using its respective modality-specific data.

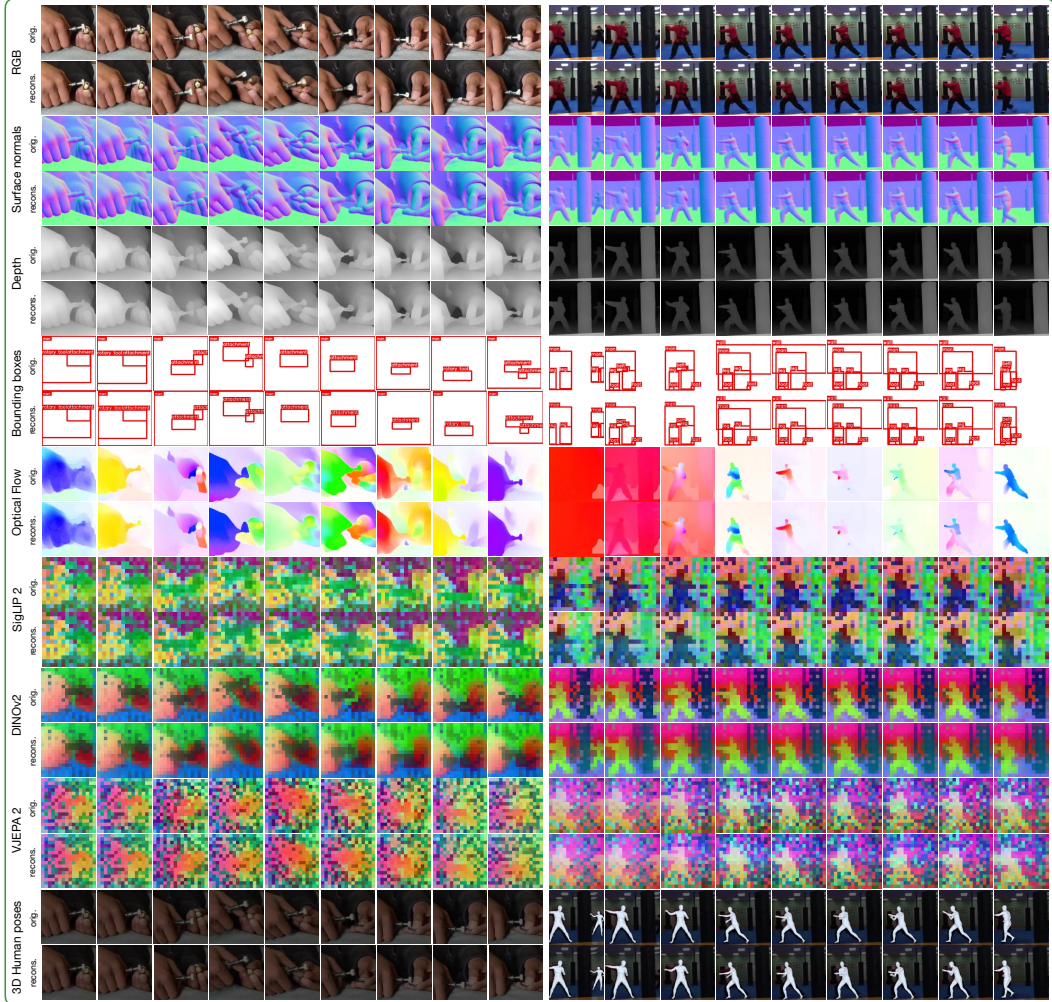


Figure 11: **Tokenizer reconstruction quality across modalities.** For each of the tokenized modalities, we show the original pseudolabeled input (orig.) alongside its reconstruction after tokenization (recons.), for two representative K600-MM clips: one with gradual motion (a person using a sewing machine, left) and one with abrupt motion changes (a person performing a kick, right). Reconstructions of abstract modalities such as feature maps, depth, and optical flow are highly consistent with the raw modalities, while RGB reconstructions contain small artifacts when fast or abrupt motion occurs.

D Additional results and ablation studies

Ablations studies. For pretraining Video-4M, we use a combination of multiple masking strategies. As discussed in Sec. 3.3, in addition to random masking, we introduce causal masking modes that incentivize future prediction across both sequential modalities (e.g., detections, transcriptions etc.) and non-sequential modalities (e.g., RGB, Depth etc.). Tab. 5 shows the ablations of these design choices using chained generations. Removing causal masking modes for non-sequential modalities improves Caption→RGB generation in isolation, but significantly degrades video prediction performance. Incorporating masking modes for both sequential and non-sequential modalities yields the best overall results across tasks.

Table 5: **Ablation on future prediction masking modes.** Incorporating masking modes that target future prediction during pretraining improves performance across generation and prediction tasks.

Configuration	Video Generation	Video Prediction
	Caption→RGB FVD ↓	1-frame FVD ↓
Video-4M (default)	310.0	238.0
Video-4M - future pred. modes (non-seq. modalities)	289.0	<u>331.0</u>
Video-4M - future pred. modes (seq. modalities)	316.0	240.0

Zero-shot evaluation results. In Tab. 6, we report the results of the zero-shot evaluation on the depth and surface normal estimation on the DIODE [44] and NYUv2 [35], and compare Video-4M with its pseudolabelers. Since Video-4M operates on tokenized inputs of 128^2 resolution, naively evaluating pseudolabelers at the same resolution yields significantly lower results, as they were trained on high-resolution images and videos [9, 4]. We therefore compare with two pseudolabeler variants: one evaluated at their native resolution and one that processes high-resolution inputs through downsampling followed by upsampling with a tokenizer reconstruction offset, to better reflect the constraints under which Video-4M operates. Overall, Video-4M performs competitively with its pseudolabelers in the latter comparable evaluation settings. Qualitative results are shown in Fig. 12.

Table 6: **Zero-shot depth and surface normal estimation.** We report results under two settings: pseudolabelers evaluated at original resolutions, and evaluated using the same settings as Video-4M (downsampled to 128^2 and resampled + tokenization offset). Under the comparable evaluation settings, Video-4M performs closely to their pseudolabelers *while being a single model for all tasks*.

Model	Depth				Normal			
	DIODE		NYUv2		DIODE		NYUv2	
	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	Mean ↓	< 11.25° ↑	Mean ↓	< 11.25° ↑
<i>Upper bounds (original resolution, no tokenization)</i>								
Video Depth Anything	0.564	0.681	0.192	0.859	-	-	-	-
NormalCrafter	-	-	-	-	16.831	42.501	15.980	60.521
<i>Downsampled to 128^2, resampled to original resolution + tokenization offset</i>								
Video Depth Anything	0.571	0.673	0.194	0.855	-	-	-	-
NormalCrafter	-	-	-	-	25.160	21.010	18.760	47.400
<i>Inference at 128^2 resolution with tokenized inputs</i>								
Video-4M-L	0.587	0.646	0.209	0.831	20.601	38.700	18.801	50.428

E Additional results for chained generations

E.1 Qualitative results

In Sec. 4.2, we presented qualitative results for class and caption condition RGB generation setting. In Fig. 13, we show results for 5-frame conditional generations for both chained and direct generations using Video-4M. In the chained generation variants, high-level textual modalities such as caption, and transcription generate the global context of the future frames, followed by semantic modalities that predicts the semantics, which are subsequently mapped to geometric depth and RGB spaces respectively.

Table 7: **Coarse-to-fine chain design ablation.**

Chain	FVD ↓	Top-1 Acc. ↑
Sparse only	439.0	30.09
Sparse → Semantic	246.3	55.4
Sparse → Semantic → Geometric	249.0	57.1

E.2 Additional analysis and ablation studies

Reversing the chain order: As discussed in Sec. 3.4, changing the order of modalities in the chain affects the generation results. In Fig. 15, we show generation examples using the coarse-to-fine chain and reverse-order chain variants. We observed that, on average, generated videos using geometric

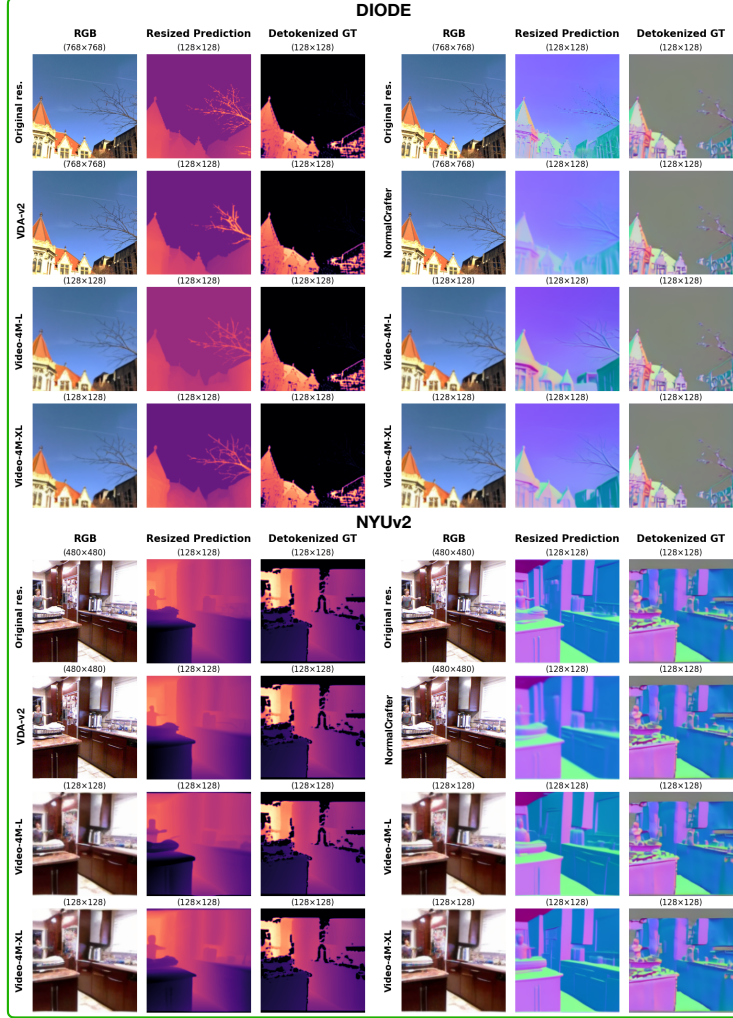


Figure 12: **Zero-shot qualitative examples for depth (left) and surface normal (right) estimation tasks.** For pseudo-labelers, we use the following configuration for comparable evaluation settings: *downsampling of input video to 128×128 , resampling to the original resolution before prediction, and passing the prediction through the tokenizer with a reconstruction offset.* The tokenized GT represents the reconstruction upper bound after tokenization.

modalities at the start of the chain struggle to follow the video semantics given in the input condition. Secondly, the visual content in the generated videos has relatively less motion compared to the default coarse-to-fine chain design. We hypothesize that understanding semantic conditions from an input condition such as a caption using low-level modalities like depth is relatively harder compared to other semantic modalities such as SigLIP. Geometric modalities focus more on low-level details such as edges and boundaries than on integrating high-level semantics from input conditions.

Despite the presence of semantic modalities in the later part of the chain, they are restricted to guide the generation, as the depth map has already generated the overall video trajectory. During training, modalities from the same training sample are always consistent with each other; therefore, the subsequent modalities in the chain are expected to be consistent with earlier predicted modalities, and thus help little in steering the video content once the layout is generated by modalities earlier in the chain.

Progressively adding modality groups in chain design: In Tab. 7, we study the effect of progressively adding modality groups in our coarse-to-fine chain design. Using only the sparse modality group, which includes caption and transcription, leads to the lowest overall performance. Including

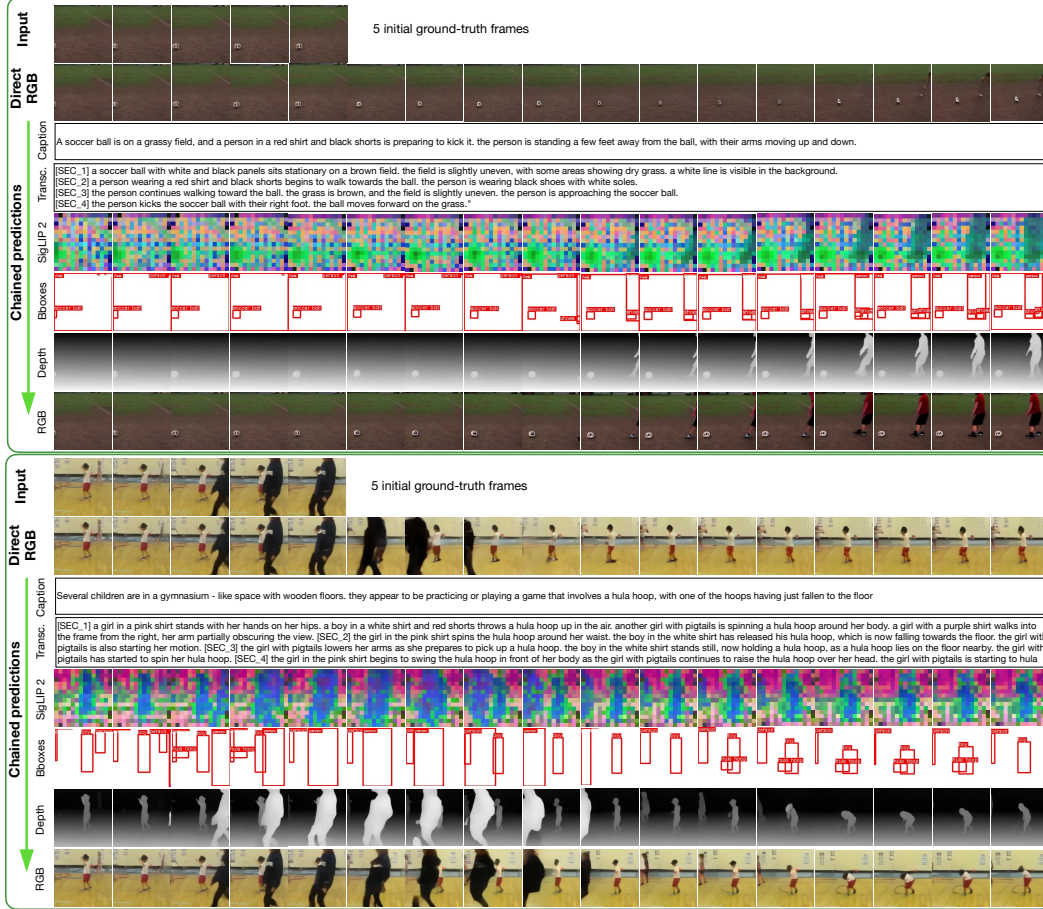


Figure 13: **Chained vs. direct predictions for 5-frame conditional RGB generation.** Given 5 initial ground-truth RGB frames as input, we compare direct RGB generation with chained generation, where Video-4M sequentially predicts caption, transcription, SigLIP 2 features, bounding boxes, and depth as intermediate modalities before generating RGB. The high-level textual modalities establish the global context of the future frames, semantic modalities outline object layout and motion, and the geometric modality refines low-level structure, leading to more coherent generations than the direct baseline.

the semantic group alongside the sparse group significantly improves performance and leads to the best FVD. Further adding the geometric modality improves the top-1 classification score while slightly reducing the FVD.

Table 8: **Geometric modalities in chaining improves edge-based metrics.**

Chain	FVD ↓	Top-1 Acc. ↑	RGB Edge Sharp. ↑	Depth Edge Sharp ↑
Sparse → Semantic → Geometric (default)	249.0	57.1	70.81	18.47
Sparse → Semantic	246.3	55.4	68.39	17.81

From our qualitative analysis, we note that adding geometric modality to the chain improves low-level structural details in the generations, but this improvement is not reflected in the FVD metric. We show generation comparisons in Fig. 14 (top). Using identical hyperparameter and seed settings, we generate two RGB videos, one using depth in the chain and the second without. Adding depth improves low-level frame details such as background clarity, inter-object boundaries, and object sharpness. In terms of FVD, this improvement is not reflected. We attribute this to the fact that FVD, which relies on I3D features [8] primarily tuned for action recognition, is potentially less sensitive to fine-grained spatial improvements such as object sharpness and boundary clarity. In Fig. 14 (middle), we generate depth maps using Video-4M on videos generated without depth in the chain,

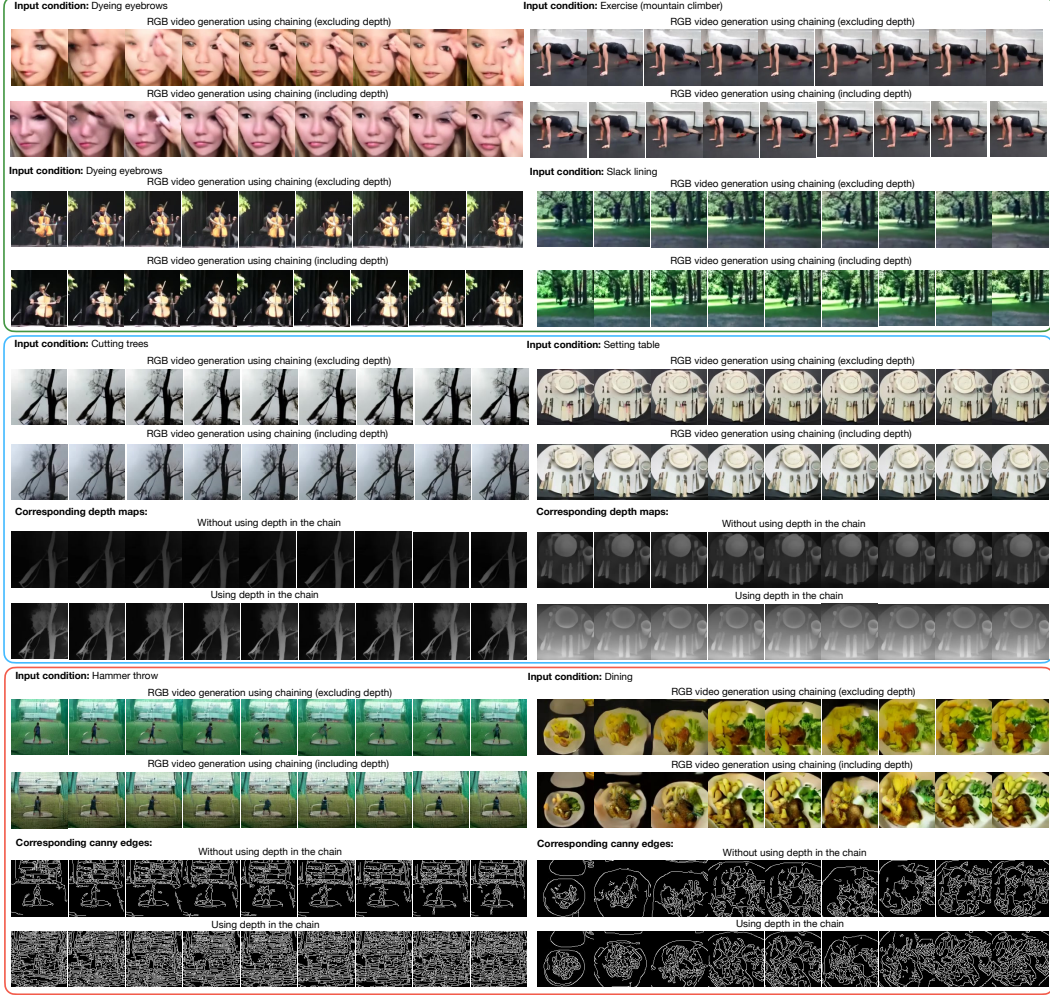


Figure 14: **Effect of including depth in the chain.** Top: RGB generations for different input conditions using the coarse-to-fine chain with and without depth in the chain, under identical hyperparameter and seed settings. Middle: depth maps corresponding to the two chain variants. Bottom: Canny edge maps of the generated videos. Including depth improves low-level structural details such as object boundaries, background clarity, and sharpness.

and compare them with depth maps from the chained generation. In the depth space, fine-grained differences are more noticeable. Depth maps used in the chain have more fine-grained details and more accurately capture the separation between objects. To quantify this analysis, we use an edge sharpness metric that computes edges on the generated videos using a Sobel filter and averages the edge magnitude score. We compute this metric on both depth maps and RGB videos, and report the results in Tab. 8. As expected, the edge sharpness metric is higher for the chain that uses depth compared to the no-depth chain design. We visualize the edge detections in Fig. 14 (bottom). For example, in the hammer throw example, the chain design using depth generates fine-grained net boundaries that commonly appear in this sport activity, which were absent in generations that did not use depth in the chain.

Ablating on the chain design: Tab. 9 shows results for ablating modality groups in our coarse-to-fine chain design. Consistent with our earlier analysis, removing semantic modalities from the chain leads to lowest generation results. Overall, using all modality groups results in the best trade-off between FVD and top-1 accuracy.

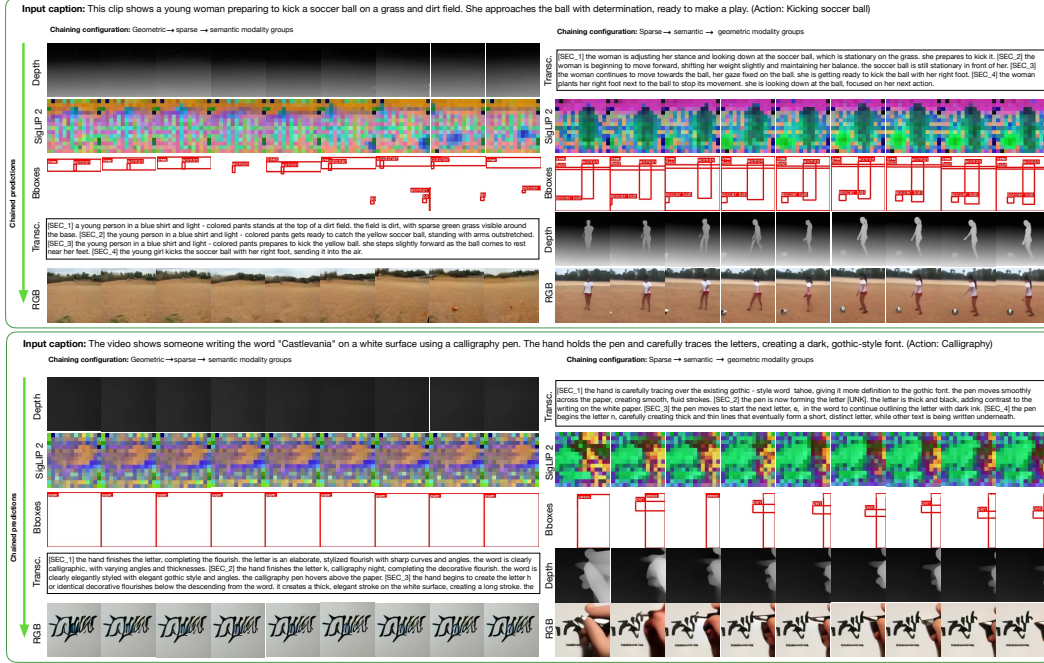


Figure 15: **Effect of reversing the chain order.** For the same input caption, we compare generations using the reversed chain (left: Geometric → Semantic → Sparse) and the default coarse-to-fine chain (right: Sparse → Semantic → Geometric). When the geometric modality is predicted first, the early depth prediction constrains the subsequent modalities, and the resulting videos struggle to follow the semantics of the input condition and exhibit less motion compared to the coarse-to-fine ordering.

Table 9: **Ablating modality groups in the chain design.**

Chain	FVD ↓	Top-1 Acc. ↑
Sparse → Semantic → Geometric (default)	249.0	57.1
<i>Leave-one-out</i>		
Sparse → Geometric	433.0	35.4
Sparse → Semantic	246.3	55.4
Semantic → Geometric	273.9	62.9

F Future Prediction Evaluation: Energy Scores

Gneiting & Raftery [17] propose proper scoring rules for evaluating probabilistic forecasts, i.e., functionals that assign a score to a predictive distribution against the (single) real outcome that is actually observed. A scoring rule is called strictly proper if in expectation it is uniquely optimized when the predicted distribution matches the true distribution. They introduce the energy score, a strictly proper scoring rule for multivariate outcomes that generalizes the continuous ranked probability score (CRPS). Crucially, the energy score allows evaluation of a predictive distribution from a single observed realization. Thus, rather than requiring access to multiple ground-truth futures for the same past, which would be impossible to obtain, it compares the single observed rollout to samples drawn from the model’s predictive distribution, while also accounting for the internal dispersion of those samples. This makes it well-suited for future prediction settings, where each past sequence is associated with only one realized future, but the model is expected to represent a distribution over many plausible futures.

Given sampled prediction rollouts $\hat{x}^{(1)}, \dots, \hat{x}^{(K)}$ and the single real (“observed”) rollout x , we estimate the energy score as

$$\text{ES} = \frac{1}{K} \sum_{i=1}^K d(\hat{x}^{(i)}, x) - \frac{1}{2K^2} \sum_{i=1}^K \sum_{j=1}^K d(\hat{x}^{(i)}, \hat{x}^{(j)}),$$

where d is a task-appropriate distance between rollouts. For the energy score to define a proper scoring rule, the distance metric d must be of negative type [17]. The first term measures how close the predicted futures are to the realized future, while the second term corrects for the spread of the predictive distribution. Intuitively, the score does not require observing all plausible futures for the same past; instead, it evaluates the distance from the realized future to the model’s sampled predictive distribution, with a correction for the distribution’s own sample-to-sample dispersion so that, in expectation, the score favors calibrated predictive distributions over collapsed or artificially overdispersed ones.

G Future Prediction: Effect of Forecasting Modality

Figure 16 shows forecasts made in different modality spaces: **SigLIP 2**, **Bbox.**, and **Normal** using four seeds each.

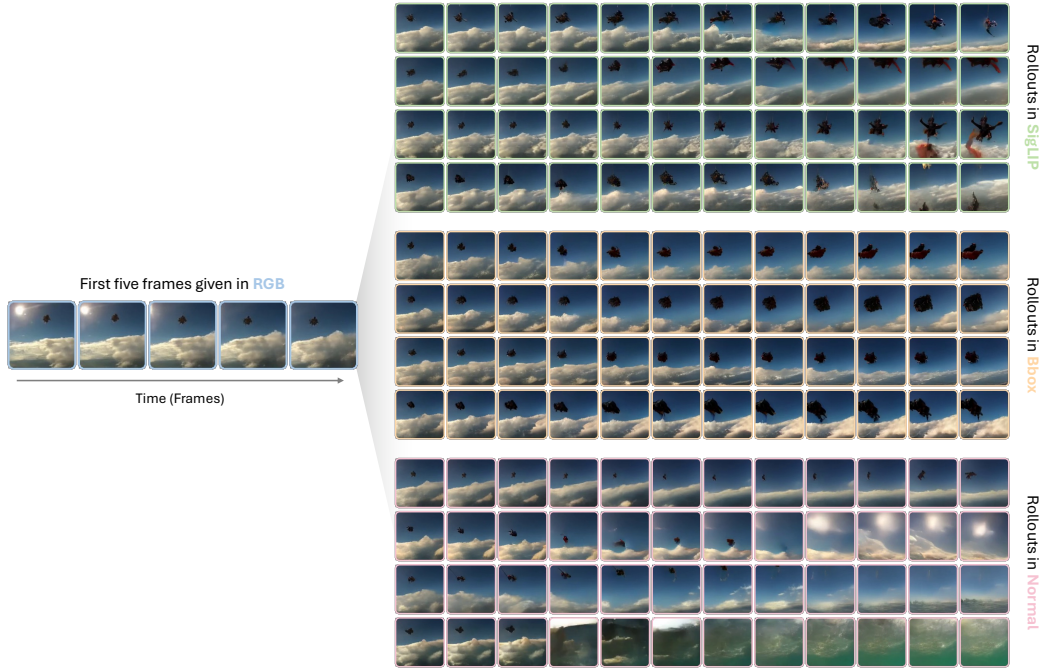


Figure 16: **Rollouts of Future Frames (Forecasting) in Different Modalities.** Future frames are rolled out in **SigLIP 2**, **Bbox.**, and **Normal** modality spaces, from the same five initial RGB frames, using 4 different seeds each. Forecasts in **SigLIP 2** tend to preserve the semantics of the scene (skydiving and parachute), **Bbox.** the object locations, **Normal** the surface characteristics (sometimes at the cost of semantic change from clouds to water surface). Note, this is an example selected to intuitively demonstrate the effect that the forecasting modality space can have; for other cases, the effect of forecasting modality is not always obvious or interpretable. However, this visualization is intended to strengthen the intuition behind how multimodal forecasts allow for exploration of a wider space of possible futures and add support to the finding in Fig. 7 that the best forecasting modality depends on the sample and the task at hand.

H Experimental Setup details

H.1 Evaluation details.

For RGB generation tasks, we use Fréchet Video Distance (FVD) as the primary metric, computed over 2,000 videos from the test set. Results are averaged over three seeds, with each seed randomly sampling 2,000 videos from the test set; the sampled videos for each seed remain consistent across experiments. For abstract-space modality prediction tasks, we similarly evaluate on 2,000 videos,

but average results over 16 random seeds. We additionally report classification accuracy and a text similarity score as alignment metrics with respect to the input condition. Following VBench [22], we use a UMT-L [28] model finetuned on Kinetics-600 classification to measure class-video alignment (using the first 16 of 17 frames), and ViCLIP-InternVid-10M-FLT [47] to measure text-video alignment (subsampling 8 of 17 frames with a temporal stride of 2). In both cases, videos are interpolated to 224×224 resolution.

H.2 Pre-training details

Implementation details. Video-4M is trained at three model scales: Video-4M-B with 198M parameters, Video-4M-L with 705M parameters, and Video-4M-XL with 2.5B parameters. For Video-4M-B, we use a batch size of 16,384 (64 per GPU x 256 GPUs) and train for 1,000B tokens, taking approximately 2 days on H200 GPUs. For Video-4M-L, we use a batch size of 14,336 (28 per GPU x 512 GPUs) and train for 1,500B tokens over approximately 3.5 days. For Video-4M-XL, we use a batch size of 36 per GPU across 40 nodes with 4 gradient accumulation steps, resulting in an effective batch size of approximately 23,040, and train for approximately 10 days. Our implementation is based on PyTorch, using Distributed Data Parallel (DDP) for Video-4M-B and Fully Sharded Data Parallel (FSDP) for Video-4M-L and Video-4M-XL.

Initialization. For training Video-4M, we start from scratch and pretrain for 1.5T tokens. Exact hyperparameters are shown in Tab. 10.

Configuration	Base	Large
Training length (n tokens)	1000B	1500B
Warmup length (n tokens)	10B	
Optimizer	AdamW [30]	
Opt. momentum	$\beta_1, \beta_2 = 0.9, 0.95$	
Base learning rate [19]	$2e-5$	
Batch size	16.3K	14.3K
Weight decay	0.05	
Learning rate schedule	Cosine decay	
Feedforward activation	SwiGLU [39]	
Input token budget	512	512
Target token budget	512	512
Input and target α	Mixture [15]	
Masking strategy	Mixture [15]	
Image resolution	128^2	
Augmentation	Center Crop	
Repeated sampling [16]	4	
Data type	bfloat16 [6]	

Table 10: **Pre-training settings for scratch initialization.**

I Limitations

While Video-4M demonstrates the benefits of abstract space modeling for video generation and prediction, several limitations remain. Chained predictions are susceptible to error propagation, where mistakes in early chain steps can compound and negatively affect subsequent modalities and the final RGB output. The quality of abstract spaces is also bounded by the pseudo labelers and tokenizers used to generate them, and inaccuracies in these models may limit the quality of learned representations. Finally, evaluating future prediction remains an open challenge as current metrics assume a single ground truth future, whereas multiple plausible futures may exist for a given observation. We believe developing better evaluation protocols for future prediction is an important direction for future work.

J Broader Impact

This work studies multimodal video modeling and future forecasting. More broadly, advances in generative world modeling may contribute to progress in areas such as robotics, autonomous systems, simulation, video generation, and multimodal scene understanding. Models capable of predicting future states across multiple modalities may also help improve data efficiency and representation learning in settings where different sensory measurements provide complementary information about the same underlying scene.

At the same time, improvements in video generation and forecasting technologies may contribute to risks associated with synthetic media, including the generation of misleading or deceptive visual content. More capable predictive and generative systems could also be used for surveillance or behavioral inference in sensitive settings. While our work is primarily methodological and does not introduce a deployment-ready system for such applications, it may indirectly contribute to broader progress in generative video modeling capabilities.

Our experiments are conducted on standard research datasets and focus on scientific understanding of multimodal future prediction rather than human-centered or identity-sensitive applications. We believe continued research into evaluation, controllability, uncertainty estimation, and transparency of generative world models will be important as these systems become more capable.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [\[Yes\]](#)

Justification: We discuss the contributions and claims of our paper in both the abstract and the introduction (Sec. 1).

Guidelines:

- The answer [\[N/A\]](#) means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [\[No\]](#) or [\[N/A\]](#) answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Yes, we discuss the limitations of our work in Sec. 7.

Guidelines:

- The answer [\[N/A\]](#) means that the paper has no limitation while the answer [\[No\]](#) means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate “Limitations” section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren’t acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[N/A\]](#)

Justification: The paper does not include any theoretical assumptions, proofs, or results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in the appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sec. H of the appendix provides the implementation details of our work.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: At the time of submission, the code is not yet publicly available. We plan to release it upon acceptance, after appropriate cleaning and documentation.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Sec. H of the appendix provides the implementation details of our work.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We follow the best practices from previous work, like 4M-21 and UnifiedIO. The error bars are not reported due to the high computational cost of running each experiment.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The information regarding the compute resources can be found at Section H.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Yes, we follow the Neurips code of ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Yes, we discuss the broader impact in Sec. J.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [\[Yes\]](#)

Yes, we discuss the broader impact in Sec. J. We will publicly release our models, code, and data with standard safeguards.

Guidelines:

- The answer [\[N/A\]](#) means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: Yes, we cite the assets used in the paper accordingly.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce any new assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, a declaration is not required.

Answer: [N/A]

Justification: The core methodology of the paper is to do video modelling in abstract spaces via any-to-any multimodal modelling. It does involve LLM use in any of its core methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.