LADDER TRANSFORMER: SELF-SUPERVISED LEARNING FOR CONTINUOUS TIME MULTIMODAL STREAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present the Ladder Transformer, an encoder-decoder architecture capable of representation learning and autoregressive generation across continuous and discrete input spaces. We treat modalities like text, video, audio, and proprioception as a single token stream, and our model supports heterogeneous sampling rates natively. The model is trained using only input-space losses, with no EMA teacher, auxiliary losses, or anti-collapse regulariser. On frozen-feature video classification, our learned representations show stronger temporal understanding than existing masked modelling methods. We demonstrate the versatility of our model in a variety of autoregressive generation tasks.

1 INTRODUCTION

Accurate prediction alone does not guarantee useful representations. Predicting masked or future data in the input space is generally held to produce weaker representations compared to discriminative methods, because representation capacity is wasted on noisy low level signals (LeCun, 2022; Assran et al., 2023). We argue that this weakness may be due more to network architecture than the training objective. We investigate hierarchical information flow as a basis for learning representations and generating observations within a shared predictive framework. This perspective revisits a principle underlying Ladder Networks (Valpola, 2015; Rasmus et al., 2015): lateral encoder–decoder connections let lower-level details reach the reconstruction without passing through the deepest layer.

In this paper, we introduce the *Ladder Transformer*, which implements this principle through mirrored cross-attention between a bottom-up encoder and a top-down decoder. The decoder combines deep contextual features with progressively shallower information when predicting target tokens. This provides an architectural route for retaining predictive detail while allowing deeper features to learn abstract, invariant features. We evaluate this design through controlled changes to the encoder–decoder connectivity. We develop the architecture specifically for multimodal sequential data, where observations differ in their tokenization schemes and sampling rates. We represent these inputs as a shared stream of timestamped tokens. Learned positional encodings describe within-modality structure, and rotary time embeddings and time-based attention windows capture temporal

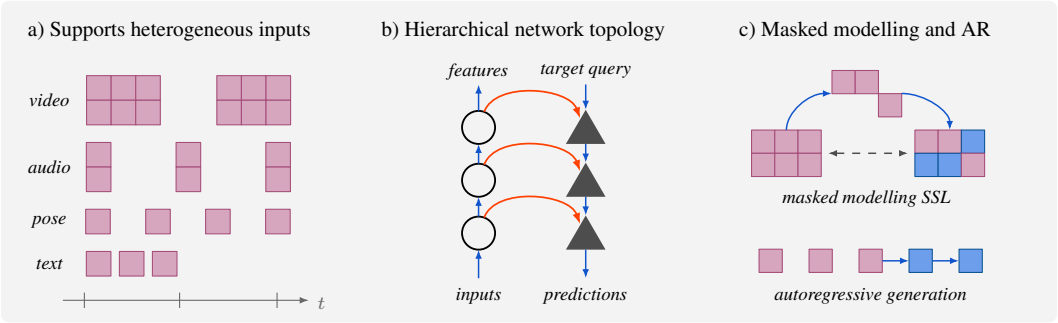


Figure 1: **Training and inference with a Ladder Transformer:** Timestamped tokens with heterogeneous sampling rates and tokenizations. Mirrored skip-connections allow noisy signals to bypass the deep layers. The model supports both masked modelling and autoregressive generation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

relationships. This formulation accommodates heterogeneous sampling rates and preserves absolute positional information, such as screen-space coordinates of video patches.

Contributions We summarize our main contributions as follows.

- We introduce the Ladder Transformer architecture with mirrored hierarchical cross-attention and hybrid spatiotemporal positional encoding, and show that this network topology improves frozen video classification over same-depth and deepest-layer-only alternatives in controlled ablations.
- We formulate masked prediction and autoregressive generation within a common timestamp-based framework for multimodal streams with heterogeneous sampling rates.
- We demonstrate that our model is capable of both representation learning and autoregressive generation. Our model demonstrates superior fine-grained motion understanding in frozen evaluations compared to other SSL methods, and achieves state-of-the-art performance in streaming text-to-motion.

2 Related work

Our work combines three lines of research: self-supervised learning by input-space prediction, autoregressive generation of continuous signals, and unified multimodal modelling.

Self-supervised learning The goal of self-supervised learning (SSL) is to learn useful representations of data by deriving supervisory signals from patterns and correlations in the data themselves (Becker & Hinton, 1992; de Sa, 1993; Balestriero et al., 2023). An effective way to accomplish this is to train the model to perform contextual prediction of unseen samples, either via masked modelling (Devlin et al., 2019; Wang et al., 2023b) or next-element prediction (Bengio et al., 2003; Srivastava et al., 2015; Radford et al., 2018). A problem from the perspective of feature learning is that the model needs to use representation capacity on noisy, low-level signals (Rasmus et al., 2015; LeCun, 2022).

Discriminative methods (Chen et al., 2020; Grill et al., 2020; Radford et al., 2021; Balestriero & LeCun, 2025) and joint-embedding predictive architectures (LeCun, 2022; Assran et al., 2023) avoid this by formulating learning objectives in latent space only. However, these methods require the use of contrastive losses, additional regularizers, or EMA teachers to prevent the collapse of the latent space (Grill et al., 2020; Zbontar et al., 2021; Caron et al., 2021; Balestriero & LeCun, 2025).

Another solution to the same problem is to use hierarchical network architectures that allow deep layers to focus on abstract, invariant features, while noisy low level signals are transmitted by skip-connections in the early layers (Valpola, 2015; Zhao et al., 2015). Similar network architectures have been applied in various image and video SSL methods (Wei et al., 2023; Liu et al., 2023c; Tang et al., 2025). Our encoder-decoder transformer design is based around this principle, and it allows us to learn effective video representations with only a single input space loss term.

RVM (Zoran et al., 2026) is another recent approach that narrows the performance gap between latent and pixel-space prediction. They pair a per-frame ViT with a recurrent core and train it to reconstruct heavily masked frames up to several seconds ahead, using a pixel loss alone with no teacher or regulariser. At ViT-L scale it comes within one to two points of V-JEPA 2 on SSv2 and surpasses pixel-reconstruction models up to an order of magnitude larger. RVM attributes this to predicting across a temporal gap through a compact recurrent state, an information bottleneck.

Autoregressive generation Autoregressive generation requires sampling the model's learned predictive distribution. A point estimate trained via a regression loss converges to the conditional mean, producing blurry or implausible outputs (Bishop, 1994; Mathieu et al., 2015). Vector quantization (VQ) discretizes the output space using a fixed-size codebook, so the model can predict a categorical distribution, at the cost of quantization error and added complexity of training (Mentzer et al., 2024).

Diffusion and flow matching based methods outperform vector quantization in terms of output quality, but require iterative sampling during inference, making them poorly suited for tasks requiring low latency. To get around this, many methods combine a large causal model with a small diffusion or flow matching head (Li et al., 2024; Octo Model Team et al., 2024; Xiao et al., 2025; Peng et al., 2025). The causal model then generates efficiently, and only the small head is iterated to draw samples. We use this design for text-to-speech and text-to-motion.

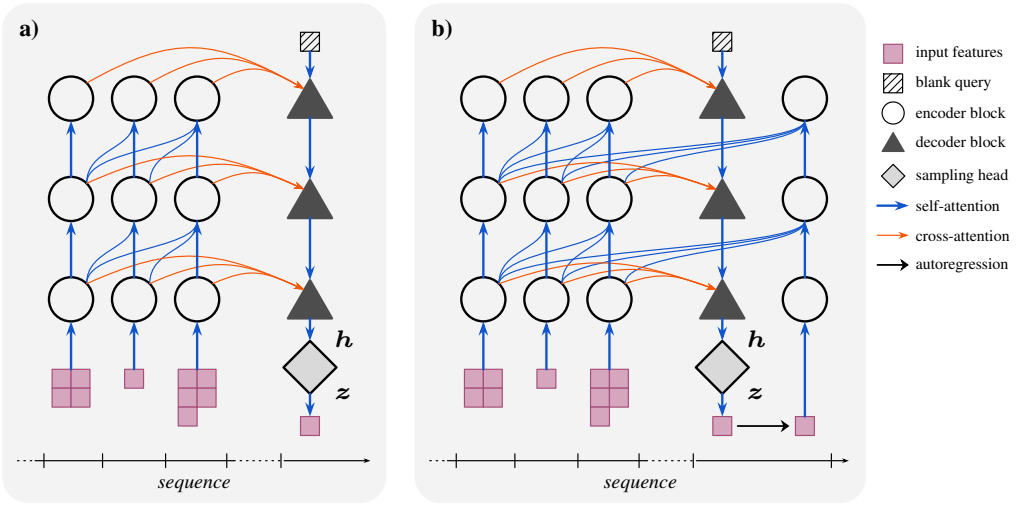


Figure 2: **Ladder Transformer architecture**. Only three layers are pictured for clarity, the actual models we trained have between 8 and 24. a) In masked modelling, the target tokens are removed from the encoder input, and the outputs of the decoder are compared with the ground truth b) In autoregressive generation, the output of the decoder is fed back to the encoder in subsequent steps.

Multimodal models Most large multimodal methods rely on modality specifica machinery: as per-modality encoders (Alayrac et al., 2022; Liu et al., 2023a; Goel et al., 2024; Xu et al., 2025), discretization step that maps every input into a shared token vocabulary (Wang et al., 2024; Chameleon Team, 2024; Bachmann et al., 2024; Lu et al., 2024) or modality-dependent forward passes (Black et al., 2024; Zhou et al., 2025). In each case, the tokenizer expects at fixed frame rate, and time enters the model as sequence index.

Perceiver (Jaegle et al., 2022) and data2vec (Baevski et al., 2022) do away with the per-modality trunk, and OMCAT (Goel et al., 2024) and Qwen2.5-Omni (Xu et al., 2025) encode physical time with rotary embeddings. In our method, each token carries a physical timestamp that drives the rotary positional embedding, capturing continuous time. We also limit the modality-specific components to the feature tokenizer, and optionally a sampling head on the output side, so adding a modality, changing tokenization, or adding a pre-trained encoder requires no changes to the transformer model.

3 Methods

The Ladder Transformer is a self-supervised learning method that treats various time varying input signals as a unified causal token stream and handles timestamps of inputs explicitly. The method comprises a encoder-decoder transformer model with mirrored skip-connections, small per-modality sampling heads, and a self-supervised training procedure with only input space losses. An overview of the model architecture is presented in Fig. 2. The motivation of the hierarchical network design is to allow noisy low-level signals be transferred via early layer skip connections, so deep layers can learn invariant abstract representations, following Ladder networks (Valpola, 2015). We adapt this topology to the transformer architecture similarly to DiffMAE (Wei et al., 2023), where lateral connections become cross-attention, and input space losses instead of the latent denoising objective of Ladder networks (Rasmus et al., 2015). Our method does not require any regularization of the latent space, nor additional contrastive or alignining losses. We aim for our model and training procedure to serve as a widely applicable method for multimodal, continuous time applications.

3.1 Model architecture

The transformer (Vaswani et al., 2017) part of the model consists of a bottom-up stack that encodes observed tokens into progressively more high-level representations, and a top-down stack that predicts unseen observations in the input space by cross-attending to the embeddings of the bottom-up stack in a mirrored hierarchical fashion. In Sec. 4.2, we compare our topology with two alternatives and

show that it significantly outperforms matched-depth and apex-only. We refer to the bottom-up stack as the *encoder* and the top-down stack as the *decoder*[1]. The encoder receives feature tokens as input with learned posenc applied (Devlin et al., 2019). Rotary time embedding (RoTE; Goel et al., 2024), a variant of rotary position embedding (Su et al., 2024), in which continuous time drives the rotation, is used in in the attention blocks to capture temporal dependencies. The inputs of the decoder are learned blank tokens, similar to the mask tokens of MAE (He et al., 2022), with posenc and timestamp of the corresponding target token applied to them. The positional encoding and timestamp appied to the blanks specify the identity of the prediction target to the decoder.

The layers of the encoder stack are typical pre-LayerNorm transformer blocks (Ba et al., 2016; Xiong et al., 2020), with the addition of the rotary time embedding (Goel et al., 2024) in the self-attention. The decoder layers, illustrated in figure Fig. 3, perform both self-attention and cross-attention, similar to the *cross-self* decoder blocks of DiffMAE (Wei et al., 2023). Our implementation differs in two ways: 1) a single attention block performs both self-attention and cross-attention and 2) the decoder does not have it's own key/value projection for the encoder side embeddings, instead using the same projection weights as the encoder self-attention. In the attention block, we concatenate the decoder and encoder embeddings, and use attention masks to control which positions can attend to each other, similar to the *mixed attention* by He et al. (2018). Wei et al. (2023) found that applying both cross-attention and self-attention in the decoder blocks greatly improves the intra-frame coherence of the outputs.

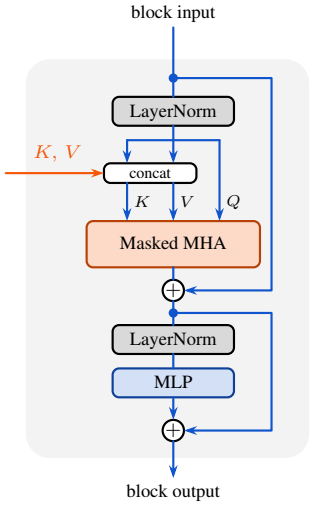


Figure 3: **Our decoder block.** Pre-LayerNorm, with a single attention operation over the decoder's own keys and values concatenated with those of the mirrored encoder block (orange), which are the ones its self-attention already computed.

3.2 Tokenization and positional encoding

Inputs enter the encoder as tokens. Various schemes can be used to tokenize the input, such as the BPE for text (Radford et al., 2019), or projected pixel patches for images and video frames (Dosovitskiy et al., 2021). Each token has two coordinates: a within timestep positional identity and a presentation timestamp t. The reason we opt for learned posenc instead of 3D RoPE (Heo et al., 2024; Assran et al., 2025) is that we want the model to retain absolute within-timestep positional information of the tokens. This can correspond to various notions of position, such as screen-space position of image patches or frequency bands of spectrogram tokens, and is fully learnable by the model. Rotary embeddings encode only relative offsets, which makes attention invariant to translation along each rotated axis.

The learned posenc is applied to input tokens additively at the beginning of the encoder stack, as well as to the blank tokens that serve as the input to the decoder. Time is captured by rotating the query and key vectors in the self-attention and cross-attention blocks (Su et al., 2024). The rotation is driven by the clip-relative timestamps of each token, similar to OMCAT (Goel et al., 2024). We apply the rotation to only half of each head's dimensions, which leaves the other half unrotated so that positional content can be matched between tokens independently of their relative timestamps. RoTE offers a wide dynamic range of temporal resolution from fractions of a second to minutes. We demonstrate strong performance on fine-grained temporal classification in Sec. 4.1.

3.3 Windowed attention

To counteract the quadratic computational cost of self-attention, we make extensive use of sliding window attention masking (Beltagy et al., 2020; Zaheer et al., 2020; Liu et al., 2022). We specify the extent of the window in terms of time instead of token index. A query at time t reads only keys with timestamps in $[t - w, t]$. During training and inference, the tokens of all modalities are merged into

[1]Our terminology does not follow the delineation of encoder and decoder introduced in *Attention is All You Need* (Vaswani et al., 2017).

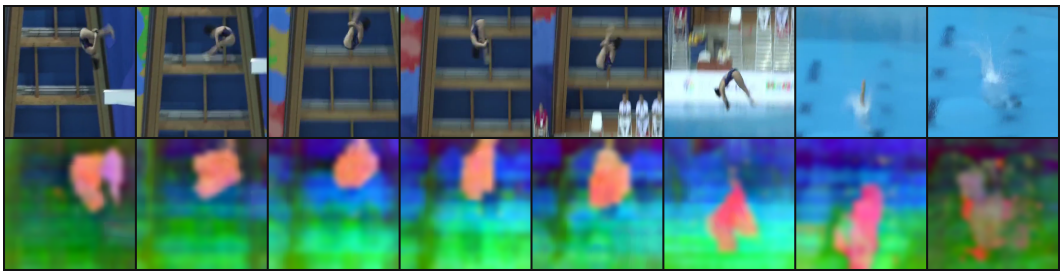


Figure 4: **Encoder output features on a Diving-48 clip.** *Top*: input frames. *Bottom*: the first three principal components of the final-layer encoder outputs, mapped to RGB. The outline of the diver is clearly visible in the features throughout, despite fast movement and motion blur.

a single sequence, and sorted by timestamp, so the attention mask is a band around the diagonal. This allows us to compute attention efficiently with block sparse FlexAttention (Dong et al., 2024).

Because the streams are sampled at different and possibly varying rates, the number of keys in the window depends on t. If $r_{\max}$ is the peak combined token rate of all modalities over the clip, then the cost of an attention layer over N tokens is $\mathcal{O}(Nr_{\max}w)$ instead of $\mathcal{O}(N^2)$ (Beltagy et al., 2020; Zaheer et al., 2020). A short window does not limit the model to only contextualize observations within w seconds. Each layer extends the span a token reaches a further w seconds to the past, so a token at the top of an L-layer encoder can be influenced by inputs up to Lw earlier (Beltagy et al., 2020; Jiang et al., 2023a), in the same way that stacked causal convolutions grow their receptive field with depth (van den Oord et al., 2016). The nominal span is an upper bound: information from its far end must be relayed through L attention steps, so the effective receptive field is concentrated nearer the query (Luo et al., 2016).

To align training with streaming operation, we prevent attention to future positions in all self-attention and cross-attention, but allow concurrent time tokens to attend to each other. Three different windows can be specified independently: the self-attention in the encoder stack, the cross-attention between decoder embeddings and encoder embeddings, and the self-attention between decoder embeddings. The exact window settings we used in training our models are listed in Table A8.

For representation learning we mask spatiotemporal tubes of patch tokens and remove them from the encoder input. Each masked token becomes a decoder query that reads the visible tokens of its own frame and the preceding second and reconstructs its pixels, so the objective is repair from same-frame and past context, not future prediction. Details are included in (App. B.3.1).

4 Experiments

We evaluate our method through frozen evaluations, controlled ablation, and by training it for cross-modal generation. We validate the quality of our learned feature representations in frozen video classification probes that require understanding of temporal dynamics. The central claim of this work is that a single hierarchical encoder–decoder transformer, coupled through cross-attention, serves both masked modelling and autoregressive generation. We quantify the effect of the cross-attention topology under probing, holding the pretraining data, objective, and compute fixed. We also train it for autoregressive text-to-speech (T2S) and text-to-motion (T2M).

4.1 Frozen evaluations

We evaluate representation quality with frozen attentive probes, following the V-JEPA 2 protocol (Assran et al., 2025). The encoder stays frozen and only a light attentive head is trained on its outputs, so the score reflects what pretraining learned rather than what task supervision could add (He et al., 2022; Bardes et al., 2024). Fig. 4 illustrates the features that the frozen backbone outputs. For all of our results we read the final-layer outputs of the encoder and train the probe for 20 epochs. As a test of our probe training procedure, we probe the published V-JEPA 2 ViT-L on SSv2. Our procedure gives a reading of 73.75% against their published 73.7%. Probe hyperparameters are selected as in V-JEPA 2, on the Diving-48 test split and the SSv2 and Jester validation splits (App. B.4).

Table 1: **Frozen evaluations using attentive probing.** We compare against other ViT-L scale video SSL models using the V-JEPA 2 attentive probe protocol (Assran et al., 2025). We probe the final layer outputs of the encoder. Params counts only the frozen encoder that the probe reads. Bold marks the best ViT-L scale attentive probe read.

Model	Params	Diving-48	SSv2	Jester
Ladder Transformer (ours)	278M	**90.3**	67.9	**97.7**
SALT (Li et al., 2025a)	300M	–	**74.9**†	–
V-JEPA 2 ViT-L (Assran et al., 2025)	300M	89.0‡	73.7†	97.6‡
V-JEPA 2.1 ViT-L§ (Mur-Labadia et al., 2026)	300M	88.1‡	74.2†	–
RVM-L (Zoran et al., 2026)	375M	86.0	71.7	**97.7**
VideoMAE v2 Large (Wang et al., 2023b)	305M	76.7	61.9	96.9
OmniStream (Yan et al., 2026)	300M	–	68.5†	–
Larger models for context				
V-JEPA 2 ViT-H (Assran et al., 2025)	600M	89.8‡	74.0†	97.7‡
V-JEPA 2 ViT-g_{384} (Assran et al., 2025)	1B	90.2‡	77.3†	97.8‡
V-JEPA 2.1 ViT-g (Mur-Labadia et al., 2026)	1B	89.2‡	77.7†	–

† Published number under a matched attentive-probe protocol.
‡ V-JEPA 2 and 2.1 use a four layer, 100 epoch readout for Diving-48 and Jester.
§ Numbers for ViT-L scale trained from scratch by Mur-Labadia et al. (2026).
Our pre-training mix includes the SSv2 training split, as in V-JEPA 2 and VideoMAE v2; it includes no Diving-48 or Jester data (Table A7).

We benchmark against leading video SSL methods at ViT-L scale. V-JEPA 2 (Assran et al., 2025) and V-JEPA 2.1 (Mur-Labadia et al., 2026) predict EMA-teacher embeddings of masked spatio-temporal blocks. SALT (Li et al., 2025a) retains the masked latent-prediction objective of V-JEPA, but replaces the EMA teacher with a frozen pixel-reconstruction teacher. VideoMAE v2 (Wang et al., 2023b) and RVM (Zoran et al., 2026) reconstruct pixels, as we do; RVM is additionally causal, carrying a recurrent state across frames, and is the closest published streaming model to ours. OmniStream (Yan et al., 2026) is a causal streaming ViT-L initialised from DINOv3 rather than trained from scratch.

A benchmark only measures spatiotemporal understanding if the task cannot be solved from appearance alone. We chose tasks where labels are decoupled from appearance by construction. Diving-48 was built explicitly to minimise representation bias; every clip shows the same setting and near-identical athletes, classes are entirely determined by the style and trajectory of the jump (Li et al., 2018). SSv2 the classes are motion templates whose object slots are filled arbitrarily. Jester consists of hand gestures recorded against uncontrolled backgrounds whose classes come in direction-reversed pairs, so the benchmark isolates sensitivity to motion direction (Materzynska et al., 2019). The three cover complementary time scales and granularities: short single-gesture clips (Jester), hand–object interactions of a few seconds (SSv2), and whole-body multi-phase sequences whose classification requires ordering events across the full clip (Diving-48).

V-JEPA 2 (Assran et al., 2025) and V-JEPA 2.1 (Mur-Labadia et al., 2026) use a different probe setup for Diving-48 and Jester, where they concatenate embedding from four different depths of the encoder instead of reading the final layer alone they also train the probes for 100 rather than 20 epochs. They report that this adds 3.8 points on Diving-48 and 1.5 on Jester for ViT-g (Assran et al., 2025, Table 17); V-JEPA 2.1 reports a smaller gain of 0.9 points on Diving-48 for ViT-L (Mur-Labadia et al., 2026, Table 12).

Our results are presented in Table 1. At ViT-L scale, our causal encoder frozen features give the best Diving-48 result of any ViT-L model and match V-JEPA 2 ViT-g. We tie RVM for the best Jester result of any ViT-L model, matching V-JEPA 2 ViT-H and within 0.1 points of ViT-g, despite our single-layer 20-epoch readout against their four-level 100-epoch one. On SSv2 we surpass VideoMAE v2-Large by six points and trail RVM by less than four.

4.2 Topology ablation

In order to quantify the effects of this design, we re-train our model with two alternative cross-attention designs: one where the encoder and decoder are connected at equal depths, and another where all the layers of the decoder cross-attend to the deepest layer embeddings of the encoder. The three arrangements are illustrated in Fig. 5. The *reverse* arrangement is more similar to the attention flow in a typical decoder-only autoregressive transformer model, and *apex* is closer to how something

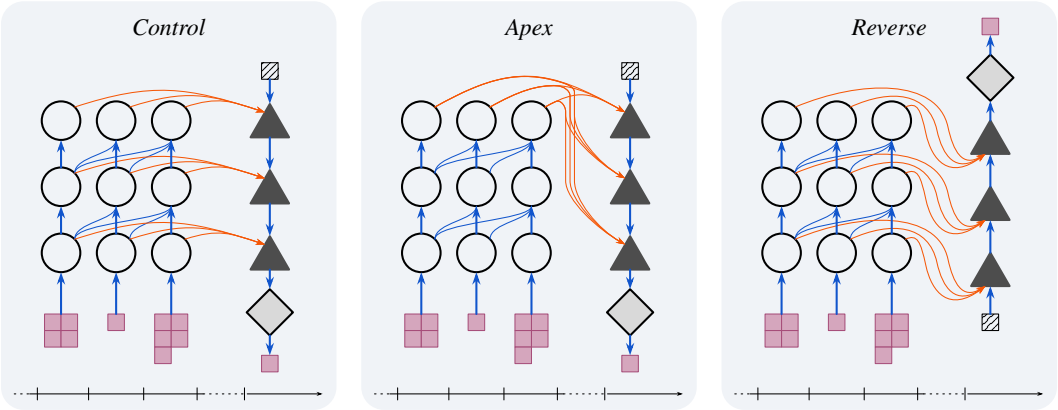
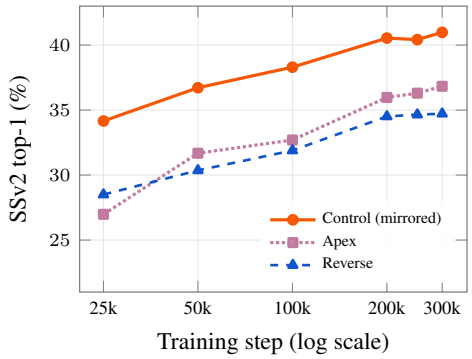


Figure 5: **Alternative cross-attention topologies** *Control*: our mirrored arrangement matching Fig. 2 *Apex*: every decoder block cross-attends to the final layer encoder outputs. *Reverse*: encoder and decoder are connected at equal depth.



Frozen probes at 300k steps

	SSv2	Diving-48
Control	**40.62**	**50.15**
Apex	36.52 (−4.1%)	31.07 (−19.1%)
Reverse	34.87 (−5.8%)	27.51 (−22.6%)

Figure 6: **Frozen evaluations of the topology ablations.** Identical data, optimiser, schedule and seed; only the encoder-to-decoder wiring differs between runs. **Left:** SSv2 probe progression over training. Mirrored leads at every checkpoint and the gap is flat from 25k to 300k steps. **Right:** Attentive probes of the 300k checkpoints. The probes here are a lighter protocol than in Table 1.

like masked autoencoders (He et al., 2022; Wang et al., 2023b) or the original encoder-decoder transformer (Vaswani et al., 2017) pass information between the encoder and the decoder. Only the cross-attention wiring differs; data, masking, objective, hyperparameters and seed are shared.

We trained the model at ViT-B scale for 300k steps. The results are presented in Fig. 6. Tracking the SSv2 probes over the duration of the training shows the mirrored arrangement leading from the first checkpoint onward, and the gap does not close over a 12× span of training. The control gains under a point over the final 100k steps, so the alternatives are climbing against a stationary target during that window and still do not close the gap. The advantage is therefore a property of the connectivity rather than an early-training effect that a longer schedule would retire. On the final 300k checkpoint probes, mirrored also leads even more on Diving-48; the benchmark that depends most on fine-grained motion is the one on which routing shallow encoder features into the decoder matters most. The exact training procedure and probing setup used is described in App. B.4.

4.3 Text-to-motion

Text-to-motion requires the model to map a freeform text caption into high degrees-of-freedom humanoid motion, with a training set orders of magnitude smaller than what is typically used in vision-language or audio-language training. Fig. 7 shows an example generated sequence.

Our pre-training data comprises roughly 276 hours of human motion capture data. We combine the AMASS collection (Mahmood et al., 2019), the BABEL (Punnakkal et al., 2021) and HumanML3D (Guo et al., 2022) annotations, as well as the BONES-SEED (Bones Studio, 2026) datasets. The test

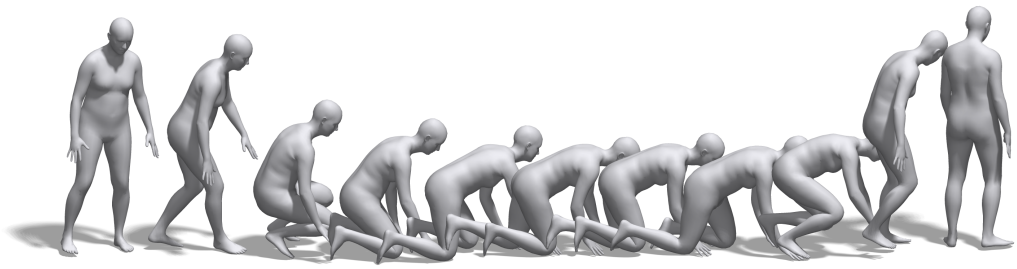


Figure 7: **Example model generated sequence.** Conditioned on caption *"a person kneels down onto all fours, crawls towards the right, and then stands back up"*

Table 2: **Text-to-motion on HumanML3D** with the evaluation protocol of Guo et al. (2022). FID is the Fréchet distance between the feature distributions of generated and real motions. R@k measures text binding by how often the matching caption is retrieved in the top k of 32 candidates from the generated motion. MM-Dist is the mean feature distance between a generated motion and its caption. Div is the mean feature distance between random pairs of generated motions and should match the real-motion value. Our Ladder Transformer gives best results within causal models (bolded).

Model	FID↓	R@1↑	R@2↑	R@3↑	MM-Dist↓	Div→
Real motion	0.002	0.513	0.702	0.797	2.971	9.531
Causal models						
Ladder Transformer (ours)	**0.063**±.003	**0.528**	**0.721**	**0.818**	2.868±.008	9.500±.090
MotionStreamer (Xiao et al., 2025) †	0.092±.003‡	0.513	0.705	0.802	2.909±.015	9.722±.037
T2M-GPT (Zhang et al., 2023)	0.116	0.491	0.680	0.775	3.118	9.761
MotionGPT (Jiang et al., 2023b)	0.232‡	0.492	0.681	0.778	3.096	9.528
Bidirectional in time (reference)						
MoMask (Guo et al., 2024)	0.052±.003	0.518	0.712	0.807	2.961±.005	9.605±.097
MoScale (Zheng et al., 2026)	0.046‡	0.540	0.727	0.817	2.830	9.525
LaMP (Li et al., 2025b)	0.032‡	0.557	–	0.843	2.759	9.571
Motion Anything (Zhang et al., 2025)	0.028‡	0.546	–	0.829	2.859	9.521

† MotionStreamer re-trained on the 263-D representation by Cai et al. (2026) (their Table 1).
‡ Published number, not re-run in our harness.

set of HumanML3D was excluded from pre-training and fine-tuning. A corpus of this size is too small for learning the semantics of open-vocabulary captions so we compute text embeddings using a frozen flan-T5-XL (Chung et al., 2022) encoder. We tokenize the motion capture trajectories using a causal temporal autoencoder following Xiao et al. (2025). The Ladder Transformer trunk emits continuous state at 3 Hz, and a flow matching head is used to sample the motion code. The decoder of the temporal autoencoder turns each code into four frames at 12 Hz. We prepend the text embedding to the pose tokens, and train the model using next-token prediction. Here we use full-causal attention to allow the model to attend to the full caption and motion history. After pretraining, we freeze the autoencoder, and fine-tune the autoregressive model for 6k steps on HumanML3D train split.

We evaluate our model on the paradigm of Guo et al. (2022), with results presented in Table 2. Ladder Transformer achieves the best FID and the best R-precision at every rank among causal models, and lies within one interval of the real diversity. A detailed description of the model and training is included in App. B.3.3.

4.4 Text-to-speech

To perform text-to-speech generation, the model needs to be able to match the corresponding sections of the discrete source text to the continuous time, variable rate spoken word audio frames, as well as navigating the idiosyncratic pronunciation of the English language.

We train Ladder Transformer for the task using script–audio pairs from the LibriHeavy corpus (Kang et al., 2024). For the text input, we use an off-the-shelf 1024-piece SentencePiece subword tokenizer (Kudo & Richardson, 2018) and learn its embedding table from scratch. We don't use any phonetic

Table 3: **Zero-shot text-to-speech** with the continuation procedure of Wang et al. (2023a). The first 3 s of audio and the full transcript are given as the prompt, and the model generates the rest of the 4–10 s utterances. WER-H is scored with HuBERT-Large CTC (Hsu et al., 2021). SIM is the WavLM-TDNN (Chen et al., 2022) cosine similarity between the prompt and the generation.

Model	Params	Data (h)	WER-H↓	SIM↑	SIM / GT↑
Our harness					
Ground truth	–	–	2.20	0.618	100%
VibeVoice codec round trip	–	–	2.40	0.576	93%
Ladder Transformer	185M	40k	2.51±.03	0.525	85%
Simple-TTS (Lovelace et al., 2024)	243M	45k	2.66±.03	0.448	73%
Published, same protocol					
Ground truth	–	–	2.15	0.668	100%
MELLE (Meng et al., 2025)	160M	50k	**1.98**	0.508	76%
Llasa (Ye et al., 2025)	1B	250k	2.47	**0.627**	–
CosyVoice 2 (Du et al., 2024)	0.5B	167k	2.97	0.587	88%
VALL-E 2 (Chen et al., 2024)	400M	50k	2.32	0.504	75%
VALL-E (Wang et al., 2023a)	400M	60k	3.80	0.508	76%
Voicebox (Le et al., 2023)	330M	60k	2.00	0.593	89%

tokenizer, so the model needs to learn the mapping from subwords to acoustics purely from the data. Audio is tokenized with the pre-trained VibeVoice acoustic VAE (Peng et al., 2025), which encodes 24 kHz audio into 64-D continuous latents at 7.5 Hz; we upsample the 16 kHz corpus audio to its input rate. Both the audio tokenizer and the Ladder Transformer are fully causal, so generation can be streamed. The text and audio tokens are concatenated so that every audio position attends to the full script while attending causally to its own past. The Ladder Transformer trunk emits continuous states at 7.5 Hz, and a flow matching head of the same design as in T2M samples the next audio latent from them. A small stop head, trained afterwards on the frozen model, predicts the end of the utterance.

We benchmark our model against leading autoregressive TTS models on the audio continuation protocol of (Wang et al., 2023a). We also benchmark (Lovelace et al., 2024), another method that aims to streamline the training of TTS models. They use a non-autoregressive latent diffusion model that denoises whole utterances on a fixed-length canvas, conditioned through cross-attention on a frozen 1.2B-parameter ByT5-large text encoder (Xue et al., 2022). Like our approach, they use neither phonemes nor forced alignment.

The word-error-rate (WER) and speaker similarity numbers are presented in Table 3. Ladder Transformer is ahead of Simple-TTS on both axes; it makes fewer recognition errors, and speaker similarity is higher, despite having fewer parameters, no pretrained text encoder, and a causal, streamable decoding procedure. Against the published rows, its WER lies in the range of the codec language models, between VALL-E 2 and CosyVoice 2 and on par with the 1B-parameter Llasa. Speaker similarity, expressed relative to the ground truth of the respective harness, is higher than VALL-E 2 or MELLE. The published numbers were obtained on a different utterance cohort and scoring pipeline, so only the ratio to ground truth is strictly comparable across the two blocks. Ladder Transformer is, together with MELLE, the smallest model in the table, is trained on less data than any other row, and takes under a day to train on a single GPU. It is within 0.11 WER of its own codec round trip, so intelligibility is now bounded by the speech autoencoder rather than by the model.

5 Conclusions

We present a novel, modality agnostic self-supervised learning method for high-dimensional continuous time sequential data. We show that the hierarchical network topology of our backbone transformer model is superior to feedforward and apex-only connectivity. We evaluate the learned feature representations of the model using attentive probing, and find that it matches or outperforms other masked modelling approaches, and joint-embedding methods on Diving-48 and Jester at ViT-L scale. We also demonstrate the applicability of our method as a cross-modal AR trunk in text-to-speech (T2S), and text-to-motion (T2M) tasks. We achieve state-of-the art results in streaming T2M, beating out task-specific architectures, as well as multimodal language models.

AI USE STATEMENT

In this work, we used generative AI tools for literature discovery, polishing some of the appendix sections, and for implementing and optimizing model code. We have not used generative AI tools for generating synthetic datasets, drafting sections of the paper, or proving mathematical claims. We have reviewed all AI-assisted work. All AI generated code was manually reviewed and tested, and all related works suggested by AI were reviewed by one of the authors. We also use AI to compact and anonymize the source code we share as part of the supplementary material. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our text-to-speech model is capable of generating synthetic speech resembling the voice of a real person, and can therefore potentially be used for fraudulent impersonation.

REPRODUCIBILITY STATEMENT

We provide our model and AR generation source code, and dataloader implementation in the supplementary material. We describe the exact composition our training data and the model configurations used in training.

Training and probing procedures for the video and audio experiments are given in App. B.4, and the full text-to-motion data processing, training and inference pipeline in App. B.3.3. All datasets used are publicly available. BONES-SEED (Bones Studio, 2026) is released under the BONES-SEED Dataset License Agreement, which permits academic use but prohibits redistribution of the data in any form, including the SMPL-converted clips we train on. We therefore cannot release our processed motion corpus; the conversion is described in App. B.3.3 and uses publicly released code, so it can be reproduced by anyone who obtains the dataset from the licensor under the same agreement.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4895–4901, 2023.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Roman Bachmann, Oğuzhan F Kar, David Mizrahi, Ali Garjani, Mingfei Gao, David Griffiths, Jiaming Hu, Afshin Dehghan, and Amir Zamir. 4M-21: An any-to-any vision model for tens of tasks and modalities. *Advances in Neural Information Processing Systems*, 37:61872–61911, 2024.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning (ICML)*, pp. 1298–1312, 2022.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Suzanna Becker and Geoffrey E Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355(6356):161–163, 1992.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003.

Christopher M Bishop. Mixture density networks. Technical Report NCRG/94/004, Neural Computing Research Group, Aston University, 1994.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Daniel Li, Piotr Dollár, and Christoph Feichtenhofer. Perception Encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

Bones Studio. BONES-SEED: Skeletal everyday embodiment dataset. https://huggingface.co/datasets/bones-studio/seed, 2026. Motion Data by Bones Studio, https://bones.studio/. 142,220 annotated motion capture animations, 288 hours, 522 performers.

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

Yiyi Cai, Yuhan Wu, Kunhang Li, You Zhou, Bo Zheng, and Haiyang Liu. FloodDiffusion: Tailored diffusion forcing for streaming motion generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2295–2304, 2026.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640, 2021.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6): 1505–1518, 2022.

Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. VALL-E 2: Neural codec language models are human parity zero-shot text to speech synthesizers. *arXiv preprint arXiv:2406.05370*, 2024.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pp. 1597–1607, 2020.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Virginia R. de Sa. Learning classification with unlabeled data. *Advances in Neural Information Processing Systems*, 6, 1993.

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 (NAACL)*, pp. 4171–4186, 2019.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex Attention: A programming model for generating optimized attention kernels. *arXiv preprint arXiv:2412.05496*, 2024.

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

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. CosyVoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*, 2024.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio Set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780, 2017.

Arushi Goel, Karan Sapra, Matthieu Le, Rafael Valle, Andrew Tao, and Bryan Catanzaro. OMCAT: Omni context aware transformer. *arXiv preprint arXiv:2410.12109*, 2024.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "Something Something" video database for learning and evaluating visual common sense. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5843–5851, 2017.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, 2022.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33: 21271–21284, 2020.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3D human motions from text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5142–5151, 2022.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. MoMask: Generative masked modeling of 3D human motions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1900–1910, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988, 2022.

Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Layer-wise coordination between encoder and decoder for neural machine translation. *Advances in Neural Information Processing Systems*, 31, 2018.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision (ECCV)*, pp. 289–305, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations (ICLR)*, 2022.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023a.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. MotionGPT: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023b.

Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey. Libriheavy: A 50,000 hours ASR corpus with punctuation casing and context. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 10991–10995, 2024.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pp. 66–71, 2018.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in Neural Information Processing Systems*, 36:14005–14034, 2023.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf. Version 0.9.2, 2022-06-27.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37: 56424–56445, 2024.

Xianhang Li, Chen Huang, Chun-Liang Li, Eran Malach, Josh Susskind, Vimal Thilak, and Etai Littwin. Rethinking JEPA: Compute-efficient video SSL with frozen teachers. *arXiv preprint arXiv:2509.24317*, 2025a.

Yingwei Li, Yi Li, and Nuno Vasconcelos. RESOUND: Towards action recognition without representation bias. In *European Conference on Computer Vision (ECCV)*, pp. 520–535, 2018.

Zhe Li, Weihao Yuan, Yisheng He, Lingteng Qiu, Shenhao Zhu, Xiaodong Gu, Weichao Shen, Yuan Dong, Zilong Dong, and Laurence Yang. LaMP: Language-motion pretraining for motion generation, retrieval, and captioning. In *International Conference on Learning Representations (ICLR)*, pp. 84238–84250, 2025b.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023b.

Yuan Liu, Songyang Zhang, Jiacheng Chen, Zhaohui Yu, Kai Chen, and Dahua Lin. Improving pixel-based MIM by reducing wasted modeling capability. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5338–5349, 2023c.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video Swin transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3192–3201, 2022.

Justin Lovelace, Soham Ray, Kwangyoun Kim, Kilian Q Weinberger, and Felix Wu. Simple-TTS: End-to-end text-to-speech synthesis with latent diffusion. OpenReview, 2024. URL https://openreview.net/forum?id=m4mwbPjOwb. Submitted to ICLR 2024.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26429–26445, 2024.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 29, 2016.

Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. AMASS: Archive of motion capture as surface shapes. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5442–5451, 2019.

Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The Jester dataset: A large-scale video dataset of human gestures. In *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2874–2882, 2019.

Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

Lingwei Meng, Long Zhou, Shujie Liu, Sanyuan Chen, Bing Han, Shujie Hu, Yanqing Liu, Jinyu Li, Sheng Zhao, Xixin Wu, et al. Autoregressive speech synthesis without vector quantization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1287–1300, 2025.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations (ICLR)*, pp. 51772–51783, 2024.

Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in Time dataset: One million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):502–508, 2019.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2023.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024.

Zhiliang Peng, Jianwei Yu, Wenhui Wang, Yaoyao Chang, Yutao Sun, Li Dong, Yi Zhu, Weijiang Xu, Hangbo Bao, Zehua Wang, Shaohan Huang, Yan Xia, and Furu Wei. VibeVoice technical report. *arXiv preprint arXiv:2508.19205*, 2025.

Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. BABEL: Bodies, action and behavior with English labels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 722–731, 2021.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI tech report*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with Ladder networks. *Advances in Neural Information Processing Systems*, 28:3546–3554, 2015.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using LSTMs. In *International Conference on Machine Learning (ICML)*, pp. 843–852, 2015.

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

Fenghe Tang, Qingsong Yao, Wenxin Ma, Chenxu Wu, Zihang Jiang, and S Kevin Zhou. Hi-End-MAE: Hierarchical encoder-driven masked autoencoders are stronger vision learners for medical image segmentation. *Medical Image Analysis*, pp. 103770, 2025.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 32–42, 2021.

Harri Valpola. From neural PCA to deep unsupervised learning. In *Advances in Independent Component Analysis and Learning Machines*, pp. 143–171. Elsevier, 2015.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. In *ISCA Speech Synthesis Workshop (SSW)*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023a.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14549–14560, 2023b.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoders. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16238–16248, 2023.

Lixing Xiao, Shunlin Lu, Huaijin Pi, Ke Fan, Liang Pan, Yueer Zhou, Ziyong Feng, Xiaowei Zhou, Sida Peng, and Jingbo Wang. MotionStreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10086–10096, 2025.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning (ICML)*, pp. 10524–10533, 2020.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306, 2022.

Yibin Yan, Jilan Xu, Shangzhe Di, Haoning Wu, and Weidi Xie. OmniStream: Mastering perception, reconstruction and action in continuous streams. In *European Conference on Computer Vision (ECCV)*, pp. 53–74, 2026.

Zhen Ye, Xinfa Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi Dai, et al. Llasa: Scaling train-time and inference-time compute for Llama-based speech synthesis. *arXiv preprint arXiv:2502.04128*, 2025.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big Bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning (ICML)*, pp. 12310–12320, 2021.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. T2M-GPT: Generating human motion from textual descriptions with discrete representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14730–14740, 2023.

Zeyu Zhang, Yiran Wang, Wei Mao, Danning Li, Rui Zhao, Biao Wu, Zirui Song, Bohan Zhuang, Ian Reid, and Richard Hartley. Motion Anything: Any to motion generation. *arXiv preprint arXiv:2503.06955*, 2025.

Junbo Zhao, Michael Mathieu, Ross Goroshin, and Yann LeCun. Stacked what-where auto-encoders. *arXiv preprint arXiv:1506.02351*, 2015.

Zhiwei Zheng, Shibo Jin, Lingjie Liu, and Mingmin Zhao. Next-scale autoregressive models for text-to-motion generation. *arXiv preprint arXiv:2604.03799*, 2026.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *International Conference on Learning Representations (ICLR)*, pp. 6446–6469, 2025.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5745–5753, 2019.

Daniel Zoran, Nikhil Parthasarathy, Yi Yang, Drew A Hudson, João Carreira, and Andrew Zisserman. Recurrent video masked autoencoders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17744–17755, 2026.

APPENDIX

TABLE OF CONTENTS

A Additional analysis **18**

A.1 Computational cost 18

A.2 Language modelling 19

B Implementation details **20**

B.1 Transformer model 20

B.2 Flow matching head 21

B.3 Training 22

B.4 Attentive probing 25

A ADDITIONAL ANALYSIS

A.1 COMPUTATIONAL COST

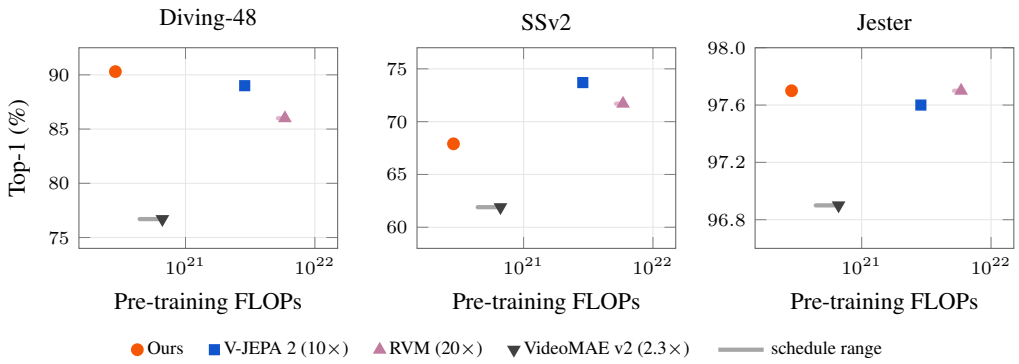


Figure A8: **Frozen-probe accuracy against pre-training compute.** Top-1 accuracy from Table 1. Each marker is a model's pre-training FLOPs, the bars span the schedule details its sources leave open, and the legend gives compute relative to ours.

We compare the computational cost of Ladder Transformer video encoder training to other SSL methods benchmarked in Sec. 4.1. FLOPs count every matrix multiplication, including attention. A trained network costs three forward passes per step and an EMA target encoder one. For Ladder Transformer, we measure the FLOPs of a training step on real training batches using PyTorch's FLOP counter for the linear layers and the tiles computed by the block-sparse attention mask. For the baselines, we compute the FLOPs per training sample from the released architectures. The step counts and batch sizes come from the papers, the released configurations, or the optimizer state stored in released checkpoints. Where these sources leave a detail open, the bars span the alternatives. Inference FLOPs are measured with the same counter on one forward pass of each frozen encoder, or computed from the architecture where the counter confirms the formula exactly. Latencies are measured on one H200 GPU at batch size one, with compiled models and CUDA graphs.

Pre-training FLOPs are presented in Fig. A8. Our ViT-L scale encoder is trained with $10\times$ fewer than V-JEPA 2 and $20\times$ fewer than RVM. Ladder Transformer gives the best Diving-48 result of the four models and ties RVM on Jester, but trails both V-JEPA 2 and RVM on SSv2. Fig. A9 shows full-clip encoding FLOPs and streaming latency. In the case of encoding, our time-windowed attention keeps the cost linear relative to clip length. The full attention of V-JEPA 2 and VideoMAE v2 grows quadratically. At a matched 256 px resolution our encoder costs about as much per frame as RVM. Our encoder emits each new frame in $\sim$2 ms regardless of history, while V-JEPA takes 12 ms per

frame if it re-encodes only a 32-frame window. Our curve sits on RVM's (2.0 vs 1.9 ms) despite 5× fewer FLOPs, because one frame per call is launch-bound for both.

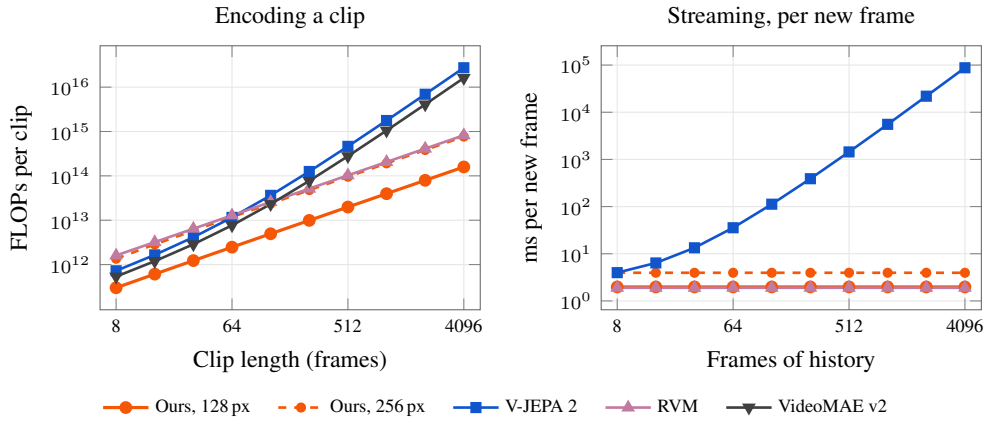


Figure A9: **Encoding cost.** *Left:* forward FLOPs of the frozen encoder to encode a clip. *Right:* latency to produce the features of a new frame given the frames before it, single H200 GPU at batch size one. Our encoder caches the keys and values inside its attention window and RVM carries its recurrent state, while V-JEPA 2 must re-encode the whole history. We read 128 px frames, and 256 px for a matched-resolution comparison. RVM and V-JEPA 2 read 256 px frames, VideoMAE v2 224 px.

A.2 Language modelling

To study how our encoder-decoder architecture compares to the standard decoder-only transformer (Radford et al., 2018) in autoregressive language modelling, we train GPT-2 (Radford et al., 2019) style models, as well as a decoder-only single-stack model built from our own transformer blocks. We picked GPT-2 because its performance and training characteristics are well studied, and because our transformer block design is close to the one it uses. All models are trained on FineWeb-Edu (Penedo et al., 2024) with the GPT-2 (Radford et al., 2019) BPE tokenizer, in batches of 32 rows of 2048 tokens. We use a set 2.09M tokens from held-out documents for validation.

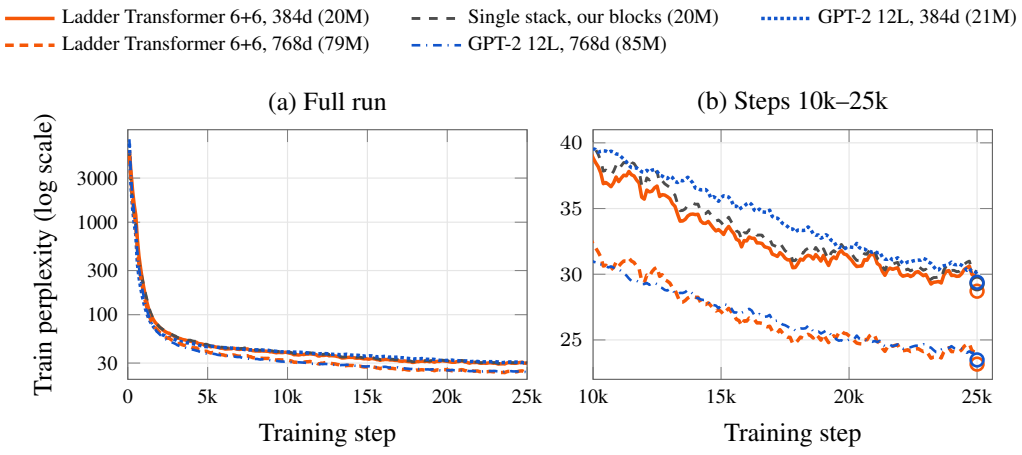


Figure A10: **Language modelling.** 500-step trailing means of the training perplexity over 25k steps. (a) Full run, log scale. (b) Steps 10k–25k, linear scale. GPT-2 trains on random 2048-token windows and our models on document-aligned rows, so their training perplexities are not directly comparable. The open markers show perplexity at 25k on a shared 2.09M-token held-out set.

For compatibility with RoTE, each text token is assigned an equally spaced timestamp. The Ladder Transformer encoder processes token embeddings causally, and the blank query at each position predicts the next token in the sequence. We train models at 6+6 blocks, 384d (19.7M non-embedding

Table A4: **Language modelling, horizon × scale.** Held-out cross-entropy (nats) for Ladder Transformer and the depth-matched 12-layer GPT-2, all at a shared peak learning rate of 6×10^{-4}. Parenthesised values are GPT-2's deficit in nats. Non-embedding parameters as (Ladder Transformer/GPT-2).

	1.6 B tokens (25k steps)		6.5 B tokens (100k steps)	
Width	Ladder Transformer	GPT-2 12L	Ladder Transformer	GPT-2 12L
384d (19.7/21.3 M)	**3.396**	3.417 (+0.021)	**3.226**	3.282 (+0.056)
768d (78.6/85.1 M)	**3.143**	3.156 (+0.014)	**2.950**	2.992 (+0.041)

Table A5: **Language modelling, cross-attention wiring.** Held-out cross-entropy (nats) at 25k steps for 12-block, 384d models trained in the same codebase with the same recipe (peak learning rate 1.2×10^{-3}). Mean ± standard deviation over seeds.

Model	Held-out CE	Seeds
Ladder Transformer, mirrored	**3.368** ± 0.011	3
Ladder Transformer, apex	3.400 ± 0.004	3
Single stack	3.377	1

parameters) and 768d (78.6M), and the single-stack variant at 12 blocks, 384d (19.6M). The GPT-2 models have 12 layers at 384d (21.3M) and 768d (85.1M). Unlike GPT-2, Ladder Transformer uses grouped-query attention (Ainslie et al., 2023) with two query heads per key-value head. All runs use a cosine schedule with 1k warmup steps. At 384d each model runs at the peak learning rate that won its own bracket (1.2×10^{-3} for Ladder Transformer and the single stack, 2.4×10^{-3} for GPT-2 384d). Both 768d models run at 6×10^{-4}, the value Brown et al. (2020) use.

Fig. A10 plots perplexity over a 25k-step run of 1.6B tokens. Ladder Transformer starts slower, trailing the single stack for the first 1k steps. We hypothesize that the blank-query decoder first has to learn to retrieve token identity through cross-attention. On the held out set, Ladder Transformer ends ahead of both the single stack and GPT-2 at 384d, and of GPT-2 at 768d. Over three seeds at each model's own learning rate, it reaches 3.368 ± 0.011 nats against 3.384 ± 0.003 for GPT-2 384d. Table A4 reports a separate set of single runs at two training horizons and two widths. Ladder Transformer is ahead in all four cells. At 1.6B tokens the margins (0.021 and 0.014 nats) are within two seed standard deviations, but with four times the training they grow to 0.056 and 0.041 nats. The gap widens with training at both widths, and shrinks by about a third from 20M to 85M parameters.

These experiments indicate that splitting the stack into a bottom-up encoder and a blank-query top-down decoder costs nothing in autoregressive language modelling at the scales we test, and we claim non-inferiority rather than superiority. We also repeat the topology finding of Sec. 4.2 on text. Table A5 shows that apex wiring is 0.032 nats worse than mirrored, and worse than the single stack.

B Implementation details

B.1 Transformer model

Our transformer blocks, as illustrated in Fig. 3, are pre-LayerNorm with a 4× GELU MLP and biased linear layers. Attention is grouped-query (Ainslie et al., 2023) with a 2:1 ratio of query heads to key/value. Model and training parameters used in Sec. 4.1 and Sec. 4.2 are listed in A10. The video and audio-visual models use LayerScale (Touvron et al., 2021) initialised to 10^{-5} on both residual branches. Linear and embedding weights are initialised from $\mathcal{N}(0, 0.02^2)$, and the attention output and second MLP projections from $\mathcal{N}(0, (0.02/\sqrt{2L})^2)$, where L is the number of encoder blocks. All biases start at zero.

Patch tokens Each continuous modality has its own linear projection to d. For video, a $16 \times 16 \times 3$ patch with values scaled to $[0, 1]$ is projected directly, without per-patch normalisation. Each modality also has a learned position table indexed by the token's position within its timestep. The table is added to the encoder input and to the decoder blanks. The decoder input for a target token is a learned blank for its modality, plus the position-table entry and it uses timestamp of that target for RoTE.

Self-cross-attention Let level ℓ be the output of encoder block ℓ, with level 0 the input. Decoder block j (j=1 nearest the apex) reads level $L-j+1$. For $j \geq 2$, that level is the input to encoder block $L-j+2$, so the decoder reuses the keys and values that block already computed. This avoid additional projections in the hybrid self-cross-attention. In each decoder block, its keys and values are concatenated with the ones from the encoder and one FlexAttention (Dong et al., 2024) call with a fused block mask computes both self-attention and cross-attention.

RoTE The pair k of the first d_r head dimensions of a query or key at time t is rotated by $\theta_k = s\,t\,b^{-2k/d_r}$ and the remaining dimensions are left unrotated. For video, audio-visual and motion we use d_r=32 (half of each head), b=1000 and s=12. With s equal to the nominal 12 fps frame rate, adjacent frames differ by one radian in the fastest pair, and the 16 periods go geometrically from 0.52 s to 340 s. For speech we use d_r=64, b=10, and s=1.

Attention rules Tokens from multiple clips can be packed per row, and they are always blocked from attending to each other. Tokens with the same timestamp see each other in encoder attention. Decoder self-attention has temporal window $0 \leq \Delta t \leq 0.05$ s except in motion generation (Table A8), which at 12–15 fps means the same frame only. Masked decoder positions are keys only for other masked queries. The decoder cross-attention window is $\delta \leq \Delta t \leq \delta + w_{\text{cross}}$ with $\delta > 0$, so a next-token query never sees its own timestep. Masked (repair) queries use δ=0. A decoder query with no readable encoder key is excluded from losses. Block masks are built once per batch, shared by all blocks, and reused across steps when the token metadata is unchanged.

Packing and compaction Clips are packed into rows of up to 8192 tokens and sorted by (clip, timestamp), which makes every mask a band around the diagonal. We remove masked tokens from the encoder input entirely following (He et al., 2022), and the decoder runs only on masked positions. Both sequences are padded to fixed lengths (5760 and 3072 tokens, multiples of FlexAttention's 128-token block), so the compiled kernels see a single shape.

Output head The last decoder block feeds one head per modality. For representation learning, this is a linear map to the patch dimension, trained with MSE between a per-token LayerNorm of the prediction and of the target as in (He et al., 2022). For speech and motion generation, it is a flow-matching head (App. B.2).

B.2 Flow matching head

In text-to-motion and text-to-speech, we use a per-token flow matching head in the style of Li et al. (2024). Let $\mathbf{h}_t \in \mathbb{R}^d$ be the output of the last decoder block at the query position of step t, and let $\mathbf{x}_1 \in \mathbb{R}^{d_x}$ be the continuous token to be generated. The head models $p(\mathbf{x}_1 \mid \mathbf{h}_t)$. The decoder half of the model generates tokens autoregressively in a single pass, and only the head is iterated during sampling. Tokens are therefore sampled independently given $\mathbf{h}_t$, and all dependence between consecutive tokens has to pass through the trunk.

Our FM head is a residual MLP. The noisy input $\mathbf{x}_\tau$ is projected linearly to the head width w. The flow time τ is embedded with 128-D sinusoidal features, concatenated with $\mathbf{h}_t$ and mapped to a width w conditioning vector $\mathbf{c}$ by a two-layer SiLU MLP. The head has n adaptive layer-norm blocks. Each block normalises its input with a non-affine LayerNorm, scales and shifts it by $(1+\boldsymbol{\gamma})$ and $\boldsymbol{\beta}$, applies a $4\times$ GELU MLP, and adds the result to the residual stream multiplied by a gate $\boldsymbol{\alpha}$. $(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\alpha})$ are a linear function of $\mathbf{c}$. A final non-affine LayerNorm and linear layer output the velocity. The modulation layers and the output layer are initialised to zero, as in adaLN-Zero (Peebles & Xie, 2023). At initialisation every block is the identity and the head predicts zero velocity, so it does not disturb the gradients reaching the trunk early in training.

We use the rectified flow parameterisation (Lipman et al., 2023; Liu et al., 2023b). With $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\tau \sim \mathcal{U}(0,1)$ and $\mathbf{x}_\tau = (1-\tau)\,\mathbf{x}_0 + \tau\,\mathbf{x}_1$, the head $\mathbf{v}_\theta$ is trained to regress the constant velocity of the straight path,

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\tau,\mathbf{x}_0,\mathbf{x}_1} \left\| \mathbf{v}_\theta(\mathbf{x}_\tau, \tau, \mathbf{h}_t) - (\mathbf{x}_1 - \mathbf{x}_0) \right\|^2. \tag{1}$$

The loss is averaged over all target positions of a packed row, and its gradient flows through $\mathbf{h}_t$ into the trunk. The regression target is conditioned on the noisy interpolant as well as on $\mathbf{h}_t$. Because the prior has unit variance, the targets are rescaled so that their average per-dimension standard deviation is one.

In training, $\mathbf{h}_t$ is replaced by a learned null vector with probability 0.1. Samples are drawn by integrating $d\mathbf{x}/d\tau = \tilde{\mathbf{v}}$ from τ=0 to 1 with forward Euler steps, starting from $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The guided velocity is $\tilde{\mathbf{v}} = \mathbf{v}_{\text{null}} + w_{\text{cfg}}(\mathbf{v}_{\text{cond}} - \mathbf{v}_{\text{null}})$ (Ho & Salimans, 2022). The two tasks form the null branch differently. In text-to-speech, it is the head evaluated on the learned null vector, so guidance means one extra head evaluation per step. In text-to-motion, the null branch is the head evaluated on a second trunk state, computed with the prompt block replaced by padding. That is the condition dropped with probability 0.1 during training (App. B.3.3). Guidance there is applied only for $\tau \geq 0.125$ (Kynkäänniemi et al., 2024). Per-task settings are listed in Table A6.

Table A6: **Flow matching head settings.** Parameter counts are for the head alone.

	Text-to-motion	Text-to-speech
Target	32-D TAE code, 3 Hz	64-D VibeVoice latent, 7.5 Hz
Condition $\mathbf{h}_t$	768-D	768-D
Width $w \times$ blocks n	1792×9	768×4
Parameters	323M	27M
Condition dropout	0.1	0.1
Null branch	trunk re-run on padded prompt	learned null vector
Euler steps	16	16
Guidance w_{cfg}	4, for $\tau \geq 0.125$	1.5, all steps

B.3 Training

B.3.1 Masked Modelling SSL

Our dataset composition for video SSL using masked modelling is described in Table A7. In Sec. 4.1 we train the model with a 2k-step linear learning-rate warm-up to 2×10^{-4}, hold it constant to 250k steps, and decay it to zero with a cosine schedule over the final 25k steps. Every number of Table 1 is read from the resulting 275k checkpoint. In Sec. 4.2 we use a constant learning rate after the warm-up throughout. The model and training hyperparameters are listed in Table A10. The main 275k training run of Table 1 took 45.7h on 16 NVIDIA GH200 GPUs, for a total of 730 GPU-h of compute.

Masking In every clip exactly one third of the patch tokens are selected as targets (Sec. 3). They are sampled as up to 48 rectangular blocks of 2 to 4 patches per side, each held for 4 to 16 consecutive frames. Blocks may overhang the frame and clip boundaries before being cut to them, so every patch position and every frame is masked with equal probability. With probability 0.5 the mask is inverted: the blocks are then sampled to cover two thirds of the tokens and form the visible context, and their complement is the target set. Half of the targets and half of the visible tokens are then removed at random, as in MAE (He et al., 2022), so the encoder reads one third of the patches of a clip and the decoder predicts one sixth. Decoder queries exist only at the remaining targets. They read visible encoder keys with $0 \leq \Delta t \leq 1.0$ s (Table A8), and each other within the same frame. The loss is the normalised-pixel MSE of App. B.1, averaged over the targets.

Optimisation and data loading We use AdamW with $\beta = (0.9, 0.999)$, weight decay 0.01 and gradient clipping at norm 1.0, in bf16 mixed precision. A training step is 64 rows of 8192 tokens, counted after the token removal above; each row packs several clips (App. B.1). Each clip is decoded from a random start, subsampled by the per-clip integer stride that brings its native frame rate closest to 12 fps, and capped at 128 frames. Clips shorter than 8 frames are dropped. The effective rate is therefore 12 to 15 fps for common source rates, and RoTE reads the true timestamps. Frames are resized to 128×128 px; non-square sources are resized on their shorter side and centre-cropped. No other spatial or photometric augmentation is used.

B.3.2 TTS Training and Inference

The causal TTS models fall into three families. *Codec language models* like VALL-E (Wang et al., 2023a), VALL-E 2 (Chen et al., 2024), and Llasa (Ye et al., 2025) model speech using discrete, neural-codec tokens, and generate using a decoder-only transformer. *Continuous-valued models* like MELLE (Meng et al., 2025) output continuous tokens, regress spectrogram frames with a variational sampling module, and learned stop predictor. *Two-stage systems* such as CosyVoice 2 (Du et al., 2024) use a text LLM to emit semantic speech tokens, and a separate flow-matching model to render

Table A7: **Pre-training data mix for masked modelling.** Hours are the number of clips times the nominal clip duration. *Passes* is the number of clips drawn over the 275k training steps divided by the number of clips in the source. Sampling is with replacement.

Source	Content	Clips	Duration	Hours	Weight	Passes
AudioSet (Gemmeke et al., 2017)[a]	sound events	1.93M	10 s	5,350	0.35	10
PE-Video (Bolya et al., 2025)	stock footage	999k	16.7 s[b]	4,640	0.25	14
Kinetics-400 (Kay et al., 2017)[c]	human actions	240k	10 s	665	0.15	35
SSv2 (Goyal et al., 2017)[d]	hand-object	169k	3.8 s	180	0.10	70
Moments in Time (Monfort et al., 2019)	events	1.06M	3 s	885	0.10	12
Ego4D (Grauman et al., 2022)	egocentric	56k	25.6 s	400	0.05	31
Total		4.45M		12,100	1.00	

[a] Balanced and unbalanced training splits. The AudioSet evaluation split and the 334 clips whose YouTube id is in the VGGSound test split are excluded. The video-only model of Table 1 uses only the frames.
[b] Mean; clip lengths range from 5 to 60 s.
[c] Training split. Clips in VGGSound test or AudioSet evaluation split were excluded.
[d] Training split only. The SSv2 validation and test sets are excluded, and no Diving-48 or Jester data is used in pre-training.

Table A8: **Attention windows in experiments.** All widths are in seconds. The decoder cross-attention band is the range of time offsets $\Delta t = t_{\text{query}} - t_{\text{key}}$ over which a decoder query reads encoder keys.

Experiment	Encoder self-attn	Decoder cross-attn band	Decoder self-attn
Video and audio-visual pretraining	0.5	$[0,\ 1.0]$[a]	0.05
Speech	full clip	$[0.133,\ \infty)$[b]	0.05
Motion generation	2.0	$[0.006,\ 2.006]$[c]	2.0
Text language modelling	full context	$[0.031,\ \infty)$[d]	0.05

[a] The pretraining objective is same-frame repair: decoder queries exist only at masked positions and read visible encoder keys from the same timestamp back to 1.0 s.
[b] Audio latents are 133 ms apart, so the query of each frame reads all earlier frames but not its own. The transcript precedes the audio and is readable by every audio query.
[c] Motion codes are 333 ms apart, so the 2 s windows span six codes and the 6 ms lower bound only excludes a code's own timestamp. Prompt tokens are readable by every motion query regardless of the window.
[d] Text tokens are stamped at 31.25 ms intervals, so the lower bound is one token and the 2048-token context is 64 s. Decoder self-attention covers the query's own and the preceding position.

them into a spectrogram. Our method is a combination of the latter two, but we generate the latent of a speech autoencoder rather than a spectrogram, and sample each frame with a flow-matching head.

Model and objective The trunk is the Ladder Transformer with 12 encoder and 12 decoder blocks of width 768, 12 query and 6 key–value heads, and the flow-matching head of Table A6, 185M parameters in total. A training example is one utterance: its transcript as at most 128 SentencePiece tokens, followed by its 64-D VibeVoice latents at 7.5 Hz. The text tokens are stamped at 7.8 ms intervals ending 50 ms before the first audio frame, so the whole script lies in the past of every audio position. Encoder self-attention spans the entire past of the utterance, and the decoder query at audio frame k reads the encoder states of the script and of the frames before k, so the head predicts each frame from the script and the audio so far. Every audio frame is a target, and the text tokens carry no loss. Utterances are packed into rows of 496 tokens.

Training The model is trained on Libriheavy-large (40k hours) for 300k steps with batch 32 rows, AdamW with weight decay 0.01, 5k warm-up steps to a peak learning rate of 2×10^{-4} and cosine decay, in bf16 on a single GPU. As in text-to-motion, the context is noised while the targets stay clean: the input latents of each utterance receive Gaussian noise whose standard deviation is σ times the per-dimension standard deviation of the corpus, with $\sigma \sim \mathcal{U}(0.1, 0.5)$ drawn per utterance. Without it, the same model scores 3.44 WER-H and 0.530 SIM under the protocol of Table 3, so the noise removes about a quarter of the word errors at a cost of 0.005 SIM.

Stop head The end of the utterance is predicted by a two-layer MLP (width 384, SiLU) on the final decoder state at each audio position, trained with binary cross-entropy with positive weight 20 to fire on the last two frames of each utterance. Trained jointly with the trunk, this loss broke the text conditioning, so the head is trained afterwards on the frozen model, for 20k steps at learning rate 2×10^{-4}, which takes about 35 minutes on one GPU.

Inference and evaluation The prompt is the full transcript and the latents of the first 2.93 s (22 frames) of the reference utterance. Frames are then generated one at a time, each sampled with 16 Euler steps and classifier-free guidance w_{cfg}=1.5, and generation ends at the first frame whose stop probability reaches 0.5, with a cap at 8 s beyond the reference length. The causal VibeVoice decoder converts the latents to 24 kHz audio. The guidance scale, number of Euler steps and stop threshold were selected on 117 utterances of LibriSpeech dev-clean, and the test cohort is the 1288 utterances of LibriSpeech test-clean between 4 and 10 s long. Our numbers and those of Simple-TTS are means over three sampling seeds, and the interval is their standard deviation. Simple-TTS is run from its released checkpoint with its own sampler, on the same utterances with the same prompt length, and with its guidance scale selected on the same dev-clean utterances.

B.3.3 TEXT-TO-MOTION TRAINING AND INFERENCE

Data and preprocessing Every clip is brought to SMPL body-model parameters: AMASS (Mahmood et al., 2019) provides them natively, and the BONES-SEED captures (Bones Studio, 2026) are converted from their native skeleton with the publicly released NVIDIA conversion code. Native frame rates of 30 to 250 fps are resampled to 12 Hz with an anti-aliased box filter over the native frames that fall inside each output frame. Left–right mirrored copies are generated at the parameter level by swapping the joint triplets and negating the appropriate axes, and the words left and right are swapped in the caption. Body-shape parameters are discarded, so the model sees a neutral-shape skeleton. Captions come from the HumanML3D captions and BABEL action labels of AMASS (Guo et al., 2022; Punnakkal et al., 2021) and from the per-take descriptions of BONES-SEED; 91% of the hours carry some text, and the rest is trained under the null condition described below. The resulting corpus is 276 hours, 100,985 clips including mirrors. Motion Data by Bones Studio (`https://bones.studio/`); the BONES-SEED dataset is distributed at `https://huggingface.co/datasets/bones-studio/seed` and its use is subject to the BONES-SEED Dataset License Agreement.

Pose representation Each 12 Hz frame is a 136-D vector with all rotations in the continuous 6-D form of Zhou et al. (2019): the pelvis translation delta expressed in the previous frame's heading frame (2-D), the ground clearance of the lowest body joint (1-D, from which the absolute pelvis height is recovered exactly at decode time), the heading delta (1-D), the root orientation with heading factored out (6-D), and the 21 SMPL body joints (126-D). The representation is position- and heading-free by construction, so the model cannot memorise where in the capture volume a clip took place; decoding integrates the deltas from an initial heading and position, and the encode–decode round trip is exact.

Causal temporal autoencoder Motion is tokenized with a causal temporal autoencoder that follows the design of Xiao et al. (2025) but is trained on our representation. The encoder is a causal convolution to width 1024 followed by two stages of a stride-2 causal convolution and a three-block dilated residual stack (dilations 1, 3, 9), giving a $4\times$ temporal downsample, and a linear head to the mean and log-variance of a 32-D Gaussian code. One code therefore covers four frames, or 333 ms. The decoder mirrors the encoder with nearest-neighbour upsampling and causal convolutions. The loss is a Gaussian negative log-likelihood with a single batch-wide scale set to the reconstruction RMSE, so its gradient is scale-free, applied to the full feature vector, with weight 7 to the root block, and with weight 1 to forward-kinematics joint positions, plus a per-sample KL term. The forward-kinematics term is ours; it places the reconstruction objective on the quantity the evaluators read, since rotation errors compound down the kinematic chain, and reduced the reconstruction FID of real test motion by a third at equal code size. Gradient clipping at norm 1 and per-sample KL weighting were both required at 32-D to keep the code scale bounded. The autoencoder is trained for 200k steps on 64-frame crops with batch 128, AdamW $(0.9, 0.99)$ without weight decay, learning rate 2×10^{-4} with 1k warm-up steps and cosine decay, in bf16, in about one hour on one H200. On held-out clips it reaches a mean joint error of 3.3° and a joint-position RMSE of 0.9 cm. Causality of both encoder and decoder is verified numerically at every training start-up by perturbing the future half of an input and confirming that earlier outputs are unchanged, so decoding a full sequence at once is identical to streaming decoding.

Latent caches and regularisation The corpus and the HumanML3D training split are encoded to posterior means. Each clip is tokenized at all four chunk offsets, which quadruples the number of distinct code sequences. During autoregressive training, Gaussian noise with a per-dimension standard deviation drawn uniformly from $[0.3, 1.0]$ (in standardised code units) is added to the context

codes of every window while the target codes are kept clean. Both measures were necessary: without them the held-out loss of an autoregressive model over a deterministic code rises from about 5k steps while the training loss keeps falling, and the one-offset, low-noise configuration is worse by 3.5 FID and 0.05 R@1 on the 272-D protocol at every training length.

Text conditioning Captions are encoded once by a frozen flan-T5-XL encoder (Chung et al., 2022), and its contextual token states are stored as a bank with one entry per (caption, token) pair. Each clip carries a 64-token prompt block drawn from this bank, RMS-matched to the motion embeddings and passed through a single trainable linear projection; 13 trainable identifiers cover padding and the duration tokens. The clip duration in seconds is appended to the block as a duration token, replaced by an open-ended token for 20% of the blocks. Where a clip has several captions one is drawn at random per training window. The prompt block is replaced by padding with probability 0.1, which defines the null condition for classifier-free guidance. Prompt tokens are keys for every motion position regardless of the attention window.

Autoregressive model The trunk is the Ladder Transformer at 12 encoder and 12 decoder blocks of width 768, with 12 query and 6 key–value heads and the rotary time encoding of Sec. 3 over real timestamps. Motion positions attend causally within a 2 s window, *i.e.* six codes (Table A8), and the decoder query at each code position reads encoder states strictly from the past, so the head predicts the code at that position from its predecessors and the caption. Every position of a packed training row is both context and target; there is no masking objective. Rows are 2048 tokens of packed variable-length clips, each clip with its own prompt block, and the batch is 16 rows.

Flow-matching head The head models $p(\mathbf{z}_t \mid \mathbf{h}_t)$ where $\mathbf{h}_t$ is the trunk output at the query position. It is a residual MLP of nine adaptive-layer-norm blocks at width 1792: each block applies a non-affine layer norm whose scale and shift, together with a residual gate, are produced from $\mathbf{h}_t$ and a 128-D sinusoidal embedding of the flow time by a zero-initialised linear map, so the head is the identity at initialisation and does not disturb the trunk's gradients; each block's MLP has a $4\times$ hidden expansion with GELU. Codes are mapped to unit scale so that their variance matches the noise source. With $\mathbf{z}_1$ the target code, $\mathbf{z}_0 \sim \mathcal{N}(0, I)$ and $\mathbf{z}_\tau = (1-\tau)\mathbf{z}_0 + \tau\mathbf{z}_1$, the head regresses the constant velocity $\mathbf{z}_1 - \mathbf{z}_0$ from $(\mathbf{z}_\tau, \tau, \mathbf{h}_t)$ (Lipman et al., 2023; Liu et al., 2023b). The head also drops its own condition to a learned null embedding with probability 0.1, independently of the prompt dropout.

Training schedule Training proceeds in three stages, all with AdamW, batch 16 rows, 500 warm-up steps and cosine decay. First, a text-conditioned model is trained for 100k steps at learning rate 10^{-4} directly on the 136-D pose representation of the full corpus, predicting one four-frame chunk per step. Second, this model is continued on the 32-D codes of the same corpus; we take the checkpoint after 5k steps of a 10k-step schedule at peak learning rate 10^{-4}, since longer latent pretraining is flat on FID and costs R-precision. Third, the model is fine-tuned for 6k steps at learning rate 2×10^{-4} on the codes of the 23,384 HumanML3D training sequences, mirrors included, with the same context noise. The second and third stages take roughly 40 and 50 minutes on a single H200.

Inference and evaluation Generation is strictly causal: one code per step, the first from the caption and duration token alone, each subsequent one conditioned on the codes already produced. Each code is sampled by integrating the learned velocity field from noise with 16 Euler steps and classifier-free guidance $w{=}4$, mixing the conditioned and null velocities only for flow time $\tau \geq 0.125$, *i.e.* not in the first two steps (Kynkäänniemi et al., 2024). The causal decoder of the autoencoder converts each code into four frames as soon as it is emitted. For the protocol of Guo et al. (2022) the 12 Hz poses are spline-resampled to 20 fps, passed through SMPL forward kinematics with a gender and body shape drawn per sample from the HumanML3D training subjects, and the resulting 22 joint positions are converted to the standard 263-D features and scored with the released evaluator. The caption's ground-truth length is given as the duration token, as in the standard protocol. Every number in Table 2 is the mean over 20 generations per test prompt, and the interval is $1.96\,\mathrm{sd}/\sqrt{20}$ over generations. With fixed-shape compiled kernels the model emits its first code about 60 ms after the prompt and runs at several times real time at batch size one.

B.4 Attentive probing

We deliberately use two different probe tiers in Sec. 4.1 and Sec. 4.2. The ablation ranks three arms of the same recipe, so it uses a cheaper cached probe. We use caching, where encoder outputs are extracted once, average-pooled to a 2×2 spatial grid per frame, and read by a three-block pooler

without test-time augmentation. The main table is the calibrated V-JEPA 2 recipe (Assran et al., 2025). The frozen encoder is run on every training batch with random augmentation, the four-block pooler attends over the full token set, and the final evaluation averages three views. Our temporal input differs from V-JEPA 2. Assran et al. (2025) encode two 16-frame segments separately and concatenates their tokens, while our encoder reads the whole clip in a single causal pass.

In all cases the backbone is frozen and only the pooler is trained with AdamW. Hyperparameters are selected exactly as in the V-JEPA 2 evaluation protocol (Assran et al., 2025): several poolers with different learning rates and weight decays are trained in parallel on the same encoder pass, each is scored on the evaluation split, and the best one is reported. The evaluation split is the validation set for SSv2 and Jester, whose test labels are not public, and the test set for Diving-48, which has no validation split. This matches the released V-JEPA 2 configuration, which likewise scores its Diving-48 probes on the test split and reports the best of its parallel heads.

Every cell of Table 1 that we measured uses the same base grid of four learning rates: $\{3{\cdot}10^{-3}, 10^{-3}, 3{\cdot}10^{-4}, 10^{-4}\}$ by three weight decays $\{0.03, 0.1, 0.4\}$, with cosine decay to zero over the 20 epochs, no warmup, and one pooler per cell trained in parallel on the same encoder pass. During training, the cells are evaluated each epoch on a fixed random subset of the evaluation split (2,000 clips, or 800 on Diving-48) only to prune diverged cells. The surviving cells are scored on the full split after the final epoch and we report the best, without early stopping. We expand the grid adaptively if the winner sits on one or more grid edge. The extension moved the reported number only for VideoMAE v2 on SSv2. Table A9 lists the selected cell behind each measured number. For reference, V-JEPA 2 selects over $\{5{\cdot}10^{-3}, 3{\cdot}10^{-3}, 10^{-3}, 3{\cdot}10^{-4}, 10^{-4}\} \times \{0.01, 0.1, 0.4, 0.8\}$ on SSv2, and over $\{10^{-3}, 3{\cdot}10^{-4}, 10^{-4}\}$ at a fixed weight decay of 0.8 on Diving-48 and Jester (Assran et al., 2025).

Table A9: **Selected probe cells.** The learning rate and weight decay of the pooler behind each number in Table 1 that we measured ourselves. All other entries in Table 1 are published values.

Model	Benchmark	LR	WD	Top-1
Ladder Transformer (ours)	Diving-48	$3{\cdot}10^{-4}$	0.1	90.3
Ladder Transformer (ours)	SSv2	$3{\cdot}10^{-4}$	0.4	67.9
Ladder Transformer (ours)	Jester	$3{\cdot}10^{-4}$	0.4	97.7
RVM	Diving-48	$3{\cdot}10^{-4}$	0.03	86.0
RVM	SSv2	$3{\cdot}10^{-4}$	0.1	71.7
VideoMAE v2	Diving-48	10^{-4}	0.4	76.7
VideoMAE v2	SSv2	10^{-4}	1.0	61.9

Table A10: **Models and probe configurations.** The model spec, pre-training setup, and probe configurations used in Sec. 4.2 and Sec. 4.1. Params are the frozen encoder the probe reads; the full checkpoint including the decoder is given in parentheses. Frames are the per-clip cap; clips shorter than the cap are used in full.

		Topology ablation	Main table
Model			
Scale		ViT-B	ViT-L
Encoder / decoder blocks		12 / 12	24 / 24
Width		768	1024
Attention heads (query / KV)		12 / 6	16 / 8
Params, encoder (full)		79M (159M)	278M (558M)
Pre-training			
Steps		300k	275k
Sequences per step		64	64
Peak learning rate		2×10^{-4}	2×10^{-4}
Schedule		2k linear warmup, constant	2k warmup, constant, cosine from 250k
Resolution		128 px	128 px
Frames		$\leq$128 at 12–15 fps	$\leq$128 at 12–15 fps
Tokens per sequence		8192	8192
Modalities		video + audio	video only
Probe			
Tier[†]		cached, light	online, heavy
Features		final layer, 2×2 pooled per frame	final layer, all tokens
Encoder pass		once, no augmentation	every epoch, with augmentation
Pooler blocks / heads		3 / 6	4 / 16
Resolution		128 px	128 px
Frames:	SSv2	8–72 at 12 fps	8–72 at 12 fps
	Diving-48	112 at 12.5 fps	232 at 25 fps
	Jester	–	40 at 12 fps
Train clips:	SSv2	42k subset	168,913 (all)
	Diving-48	15,027 (all)	15,027 (all)
	Jester	–	118,562 (all)
Eval clips:	SSv2	12k subset of val	24,777 (all val)
	Diving-48	1,970 (all test)	1,970 (all test)
	Jester	–	14,787 (all val)
LR × WD grid		5 × 8 (SSv2), 5 × 9 (D48)	4 × 3
Epochs (grid / selected cell)		15 / 40	20 / 20
Cell selection		best on full eval split	best on full eval split, final epoch
Test-time augmentation		none	3 views

[†] *Light* runs the frozen encoder once, without augmentation, and caches its final-layer tokens average-pooled to a 2×2 grid per frame; a three-block pooler is then trained on the cache and scored without test-time augmentation. It ranks arms rather than producing absolute numbers. *Heavy* is the V-JEPA 2 attentive-probe recipe (Assran et al., 2025): the encoder is re-run on every training batch under random augmentation, the four-block pooler attends over all tokens, and the reported number averages three views.