SEMANTOK: PREDICTABLE SEMANTIC TOKENS FOR EFFICIENT AUTOREGRESSIVE VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent video-based world models pair the scalability of autoregressive (AR) prediction with the visual quality of diffusion models. The choice of scene tokenizer is paramount for the optimal performance of each of these, both in terms of fidelity and semantics. Flexible-length, coarse-to-fine tokenizers yield exactly that: the first coarse tokens carry the clip's global semantics while later tokens further specify details. Existing flexible tokenizers only apply a representation-alignment (REPA) loss on early decoder hidden states, a target the decoder can partly meet from its noised input instead. We introduce *SemanTok*, a flexible video tokenizer that feeds frozen DINO features into its encoder and adds lightweight heads that reconstruct them from each retained token prefix alone. SemanTok achieves high semantic alignment and video fidelity at every AR model size: a 201M SemanTok AR model matches or beats a VideoFlexTok AR model $3.4\times$ its size, and larger SemanTok AR models further improve fidelity. It keeps semantic alignment on out-of-distribution classes and gives the decoder higher semantic alignment at every noise level, including pure noise. It performs well in both reconstruction and generation, and its short token prefixes are cheaper to predict and give better generation fidelity, with pixel detail deferred to later tokens.

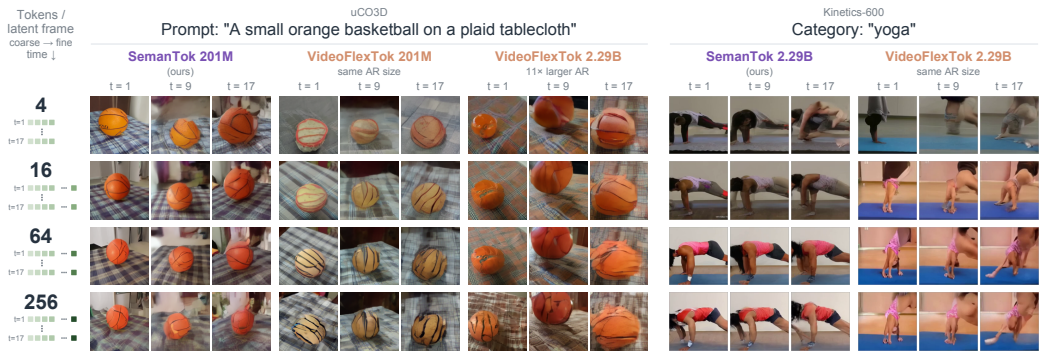


Figure 1: From a prompt with "…orange basketball…" (left, uCO3D), our SemanTok tokenizer lets a 201M autoregressive (AR) model generate a video that keeps the ball's shape and appearance through the orbit with a budget of only k=4 tokens per latent frame. VideoFlexTok: at the same model's size the ball is semantically misaligned, and even an $11\times$ larger AR model (2.29B) leaves its shape unstable up to k=64. So the smaller, faster model on SemanTok tokens matches or beats the larger one of VideoFlexTok. Right (Kinetics-600, class "yoga"): both tokenizers at the same 2.29B AR size, i.e., equal cost. SemanTok keeps a complex body motion stable from k=16, and larger k refines its appearance and motion. Meanwhile VideoFlexTok changes the scene between k=4 and k=16 and poorly simulates body motion. Both VideoFlexTok and SemanTok are variable-length: the AR model generates tokens coarse to fine, and any budget k decodes to a video. SemanTok prioritizes semantic content in its first tokens, so a small k already fixes what the clip contains. Pipeline in fig. 2; videos in the supplementary material.

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.

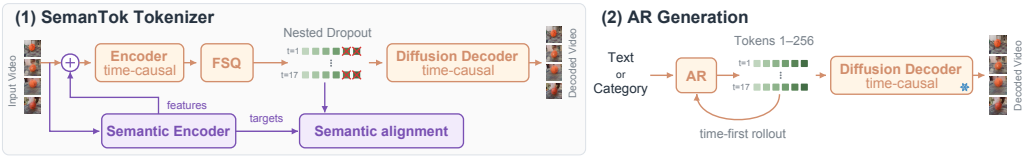


Figure 2: Overview. (1) Tokenizer training: encoder, FSQ, and diffusion decoder are trained jointly, with nested dropout so that the decoder can reconstruct the clip from any token prefix. SemanTok keeps this VideoFlexTok recipe (orange) and adds semantic supervision from a frozen teacher: its features enter the encoder, and every retained prefix is trained to predict them (purple; details in fig. 3). (2) AR training and generation: with the tokenizer frozen, an AR model learns to predict its tokens in time-first, coarse-to-fine order, and the decoder trained in (1) renders any generated prefix as video.

1 Introduction

Video clips vary widely in complexity: a static shot has few entities and little motion, while a car chase has changing viewpoints and detail. To make such clips tractable, video models compress them into learned tokens that an autoregressive (AR) model predicts, as a language model predicts text, and a diffusion decoder renders into pixels (Yan et al., 2021; Yu et al., 2024a). These tokens must therefore be both easy to predict and informative enough to render. Yet standard video tokenizers map every clip to the same fixed-size spatiotemporal grid (NVIDIA, 2025; Tang et al., 2024; Yu et al., 2024a). Because prediction cost grows with sequence length, simple and complex clips then get the same representational budget and the same compute.

Flexible-length, coarse-to-fine tokenizers (Bachmann et al., 2025; Atanov et al., 2026) replace the fixed grid with a linear sequence that can be meaningfully cropped to various lengths. At inference time, the application, not the model, selects a budget k and the autoregressive model predicts only up to that prefix length. Each prefix should support a high-quality rendering, with additional tokens only increasing its specificity. We argue that this approach should aim to carry a clip's semantics in the coarse prefix, so that a small AR model can settle on what the video should contain, before a larger model spends capacity on fixing appearance details. Existing flexible tokenizers use representation alignment (REPA) (Yu et al., 2025) at an early decoder layer, showing that this improves downstream fidelity and semantic alignment (Atanov et al., 2026); this is mostly inspired by image tokenizers (Zhu et al., 2024; Chen et al., 2025; Yao et al., 2025). But that decoder also sees the noised latent, which at low noise meets the target almost regardless of the tokens (fig. 9).

For this, we propose SemanTok, a flexible video tokenizer designed to prioritize semantic content early in an AR-model rollout (fig. 1; pipeline in fig. 2). We keep the original decoder REPA loss and the coarse-to-fine recipe. Frozen DINOv2 features (Oquab et al., 2024) enter the encoder alongside the video latents. Auxiliary heads then read only the retained token prefix and reconstruct the DINO features. We call these two uses of the teacher *semantic supervision*. No loss assigns information to particular tokens; nested dropout concentrates the most relevant information in the earliest ones. The result is a variable-length semantic code: every prefix, at any budget, carries the teacher's view of the clip.

The same prefix that these pathways make more semantic is also easier for an AR model to predict. At the same length, the AR model then needs fewer bits to code it: SemanTok keeps what the clip contains early and defers part of the pixel detail the AR model could not predict, a prefix-level form of the compression–generation trade-off (Ramanujan et al., 2025; Wang et al., 2025).

We evaluate SemanTok (section 3) on fidelity and semantic alignment in the generation and ground-truth-token reconstruction settings for video (section 4). We score seven AR sizes (49M–2.29B) across token budgets, on class-conditioned (Kinetics-600) and text-conditioned (uCO3D). Seven findings emerge (section 5):

1. **SemanTok exhibits high semantic alignment and video fidelity at every AR model size.** A 201M SemanTok AR model matches or beats a VideoFlexTok AR model 3.4× its size (figs. 1, 4, 5 and 13; see exception 7).

2 **Increasing SemanTok model size further improves fidelity.** The smaller SemanTok AR model exceeds the larger VideoFlexTok AR model in Kinetics-600 class accuracy (semantic alignment). Increasing model size further improves fidelity (gFVD) (figs. 5 to 7 and 16).

3 **SemanTok is able to maintain semantic alignment over out-of-distribution classes.** Both tokenizer reconstruction and AR generation show the generalization capability of SemanTok on held-out classes (table 1 and figs. 8 and 17; see exception 7).

4 **SemanTok achieves higher decoder-REPA semantic alignment at all noise levels, including the pure noise setting.** (figs. 9 and 18; see exception 7).

5 **SemanTok has high fidelity on reconstruction as well as generation.** VideoFlexTok performs worse on generation than reconstruction, while SemanTok performs well on both (figs. 10, 12 and 14 and table 2; see exception 7).

6 **SemanTok's has better generation fidelity from its earlier tokens, which are cheaper to predict.** Under the same AR model, generation at k=16 costs about 32% fewer bits per token, and pixel detail is deferred to later tokens, not discarded (figs. 11 and 20).

7 **SemanTok with 1 token per frame struggles to serve every objective.** At k=1, SemanTok trails VideoFlexTok in semantic alignment with a mostly clean latent. Surprisingly, SemanTok still keeps its class-accuracy lead at every budget (fig. 4 and tables 6 to 10).

2 Related Work

Token-based video generation. Video models predict discrete tokens from grid tokenizers, autoregressively (Yan et al., 2021; Kondratyuk et al., 2024) or by masked prediction (Villegas et al., 2023; Yu et al., 2024a), and world models use the same recipe for action-conditioned rollouts (Bruce et al., 2024; NVIDIA, 2025) or predict self-supervised features instead of pixels (Assran et al., 2025; Zhou et al., 2025). In hybrids, AR fixes the content and diffusion renders it (Li et al., 2024; 2025b); VAR instead orders prediction coarse-to-fine across scales (Tian et al., 2024). Every grid location still reaches the generator.

Ordered and flexible tokenizers. TiTok and LARP compress images and videos into 1D token sequences (Yu et al., 2024b; Wang et al., 2025). Nested dropout and Matryoshka losses order such codes by importance (Rippel et al., 2014; Kusupati et al., 2022): ElasticTok drops token suffixes (Yan et al., 2025), Semanticist finds a PCA-like, semantics-first ordering (Wen et al., 2025), and FlexTok and VideoFlexTok make every prefix decodable, with REPA on an early decoder layer (Bachmann et al., 2025; Atanov et al., 2026; Yu et al., 2025). ReToK strengthens decoder alignment for shorter prefixes (Fu et al., 2026), LoST aligns ordered 3D latents with DINO (Dutt et al., 2026), and SpeechTokenizer distills a teacher into its first audio level (Zhang et al., 2024). SemanTok trains every retained prefix of a flexible video code to predict the teacher without the noised latent, which no prior video tokenizer does to our knowledge.

Tokenizers for generation, not reconstruction. Reconstruction alone does not make a latent easy to model. DiGIT, MAETok, VA-VAE, ImageFolder, GigaTok, and UniTok align or predict foundation-model features to improve generation (Zhu et al., 2024; Chen et al., 2025; Yao et al., 2025; Li et al., 2025a; Xiong et al., 2025; Ma et al., 2025); REPA-E trains the VAE through the REPA loss (Leng et al., 2025), and RAE diffuses directly in frozen DINO features (Zheng et al., 2026). LARP and CRT shape the code with an AR prior, trading reconstruction for generation (Wang et al., 2025; Ramanujan et al., 2025), consistent with the perception–distortion trade-off (Blau & Michaeli, 2018). These works use fixed-length, mostly image, codes. We place semantics in the *early prefixes* of a flexible video tokenizer, where a small AR model spends its budget.

3 Method

We review VideoFlexTok (orange in fig. 3), then SemanTok. Our goal is to make the early prefix carry clip semantics without changing the codebook, native decoder, or AR interface.

VideoFlexTok tokenizer. Let x be an RGB clip. A frozen VidTok VAE maps x to $z \in \mathbb{R}^{T\times h\times w\times C_z}$, where T=5, h=w=C_z=16. We write $\mathbf{z}_{t,p} \in \mathbb{R}^{C_z}$ for the latent vector at frame t and

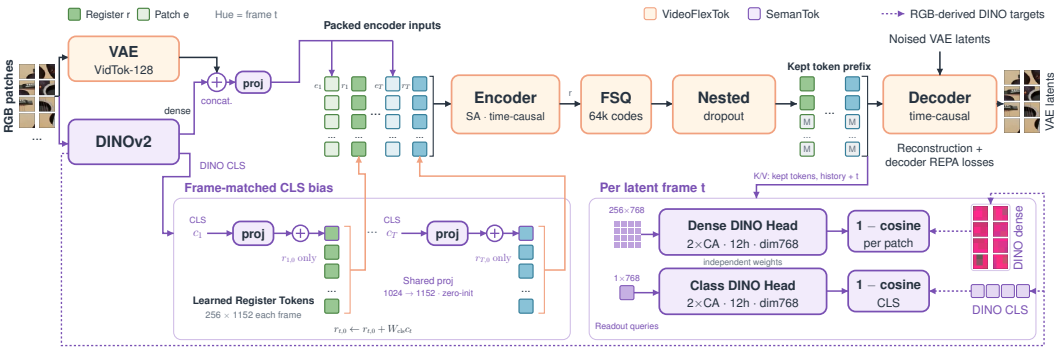


Figure 3: The figure contrasts two tokenizer variants. VideoFlexTok provides the base coarse-to-fine path (orange), while SemanTok changes its encoder input and adds a Dense DINO head and a Class DINO head (purple). A frozen VidTok VAE maps the clip to latents; each frame projects its patches to e and packs them with K learnable register tokens $\mathbf{r}$: $(e_{t,0}, \ldots, e_{t,P-1}, r_{t,0}, \ldots, r_{t,K-1})$, interleaved over time. The time-causal encoder transforms the packed sequence, retaining only the register-token outputs for FSQ; nested dropout forms the kept token prefix that conditions a time-causal decoder reconstructing VAE latents under flow-matching and decoder-REPA losses. SemanTok concatenates frozen DINOv2 patch features with each VAE patch before projection, and adds a zero-initialized projection of the matching DINO class token to $r_{t,0}$. Readout queries in the Dense DINO head and Class DINO head cross-attend to the kept token prefix from frames $t' \leq t$.

spatial position p, with $t \in \{1, \ldots, T\}$ and $p \in \{0, \ldots, P-1\}$, where $P{=}h{\times}w{=}256$. A learned linear map lifts each patch to encoder width $d_e{=}1152$:

$$\mathbf{e}_{t,p} = W_{\text{in}}\mathbf{z}_{t,p} + \mathbf{b}_{\text{in}} \in \mathbb{R}^{d_e}.$$

Independently, the encoder holds $K{=}256$ learnable register tokens $\mathbf{r}_{t,i} \in \mathbb{R}^{d_e}$, $i \in \{0, \ldots, K-1\}$, following VideoFlexTok (Atanov et al., 2026). Index i is the coarse-to-fine token position.

For each frame the encoder packs the P patches, then the K register tokens:

$$\mathcal{S}_t = (\mathbf{e}_{t,0}, \ldots, \mathbf{e}_{t,P-1}, \mathbf{r}_{t,0}, \ldots, \mathbf{r}_{t,K-1}).$$

The encoder input is the time-interleaved sequence $(\mathcal{S}_1, \ldots, \mathcal{S}_T)$. With time-causal attention, frame t sees only past frames; within each frame, patches attend freely and register token i reads all patches and register tokens $j \leq i$. Patch embeddings are discarded; each register-token output, having already read the patches, is linearly projected to $\mathbf{u}_{t,i} \in \mathbb{R}^{D_q}$, $D_q{=}6$, and passed through FSQ (Mentzer et al., 2024), which bounds the six dimensions with $\tanh$ and rounds on the lattice $[8, 8, 8, 5, 5, 5]$ to give

$$\mathbf{q}_{t,i} = \text{FSQ}(\mathbf{u}_{t,i}) \in \mathbb{R}^{D_q}.$$

Nested dropout samples one k uniformly from $\{1, 2, 4, \ldots, 256\}$ per clip and replaces $\mathbf{q}_{t,i}$ for $i \geq k$ in every latent frame with a learned mask token. The kept token prefix conditions a time-causal rectified-flow decoder on noised VAE latents. VideoFlexTok trains encoder, FSQ, nested dropout, and decoder jointly (step (1) in fig. 2) with

$$\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{Flow}} + \lambda\,\mathcal{L}^{\text{dec}}_{\text{REPA}}, \qquad \lambda = 1,$$

where $\mathcal{L}_{\text{Flow}}$ is flow matching and the decoder REPA term aligns an early decoder layer with frozen DINOv2 patches.

Each $\mathbf{q}_{t,i}$ also maps to a codebook index

$$\tau_{t,i} = \text{idx}(\mathbf{q}_{t,i}) \in \{0, \ldots, V-1\}, \qquad V = \prod_j L_j = 64000.$$

A separate AR model (step (2) in fig. 2) is trained on all TK indices in time-first order, $(\tau_{1,0}, \ldots, \tau_{T,0}, \tau_{1,1}, \ldots, \tau_{T,K-1})$, so any prefix is a valid token budget at inference; the frozen tokenizer decoder renders AR-sampled tokens to video.

SemanTok tokenizer. SemanTok keeps the codebook, nested dropout, decoder, and $\mathcal{L}_{\text{base}}$; the purple paths in fig. 3 add semantic supervision, which changes the encoder input and supervises the

tokens. A frozen DINOv2-L teacher (Oquab et al., 2024) supplies, per latent frame, a 16×16 patch grid $\mathbf{d}_{t,p}$ aligned with the P latent positions and a class token $\mathbf{c}_t$, both in $\mathbb{R}^{D_D}$, D_D=1024. Each patch embedding fuses both inputs, $\mathbf{e}_{t,p} = W_{\text{in}}[\mathbf{z}_{t,p}; \mathbf{d}_{t,p}] + \mathbf{b}_{\text{in}}$, and a zero-initialized projection of the class token is added to the first register token, $\widetilde{\mathbf{r}}_{t,0} = \mathbf{r}_{t,0} + W_{\text{cls}}\mathbf{c}_t$. Packing, attention, and FSQ are otherwise as above.

Two independently parameterized cross-attention heads read the kept token prefix. For frame t, their shared context at budget k is

$$\mathcal{C}_{t,k} = \{\mathbf{q}_{s,i} : 1 \le s \le t,\ 0 \le i < k\}.$$

The temporal restriction matches the tokenizer's causal path. Each head has two cross-attention layers of width d_h=768 with 12 attention heads. The Dense DINO head h_ϕ has spatial readout queries $\mathbf{a}_p \in \mathbb{R}^{d_h}$, which each aim to reconstruct one DINO patch from the linear token sequence, and the Class DINO head h_ψ has a single readout query $\mathbf{a}_{\text{cls}}$, distinct from the register token $\mathbf{r}_{t,0}$:

$$\widehat{\mathbf{D}}_t = \{\widehat{\mathbf{d}}_{t,p}\}_{p=0}^{P-1} = h_\phi(\mathcal{C}_{t,k}), \qquad \widehat{\mathbf{c}}_t = h_\psi(\mathcal{C}_{t,k}),$$

$$\mathcal{L}_{\text{dense}} = \frac{1}{TP}\sum_{t=1}^{T}\sum_{p=0}^{P-1}\left(1 - \cos(\widehat{\mathbf{d}}_{t,p}, \mathbf{d}_{t,p})\right), \qquad \mathcal{L}_{\text{cls}} = \frac{1}{T}\sum_{t=1}^{T}\left(1 - \cos(\widehat{\mathbf{c}}_t, \mathbf{c}_t)\right).$$

Both heads supervise the same discrete representation consumed by the decoder and predicted by the AR model; unlike the decoder REPA layer, they never see the noised latent, so only the tokens can lower their losses. Nested dropout varies k, so each sampled prefix must support both DINO predictions.

The full objective is

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \mu_{\text{dense}}\mathcal{L}_{\text{dense}} + \mu_{\text{cls}}\mathcal{L}_{\text{cls}}, \qquad \mu_{\text{dense}} = \mu_{\text{cls}} = 0.5.$$

No loss assigns a particular DINO feature to a particular token. The ordering emerges from the shared prefix constraint, and the decoder target remains $\mathbf{z}$, never DINO features.

4 Experimental Details

We compare SemanTok to its closest prior work VideoFlexTok which differs only in semantic supervision, at the same codebook, sequence length, and AR recipe. We ask how that supervision changes generation. First, we measure how fidelity and semantic alignment change with AR model size (1), how larger and longer-trained AR models improve fidelity further (2), and whether semantic alignment holds on out-of-distribution classes (3). We then examine how the token prefix affects decoder-REPA semantic alignment (4). We measure the difference between generation fidelity and reconstruction fidelity i.e. realization gap (5), and trace SemanTok's generation gain to specific tokens (6). Finally, we report where SemanTok falls behind: at one token per frame (7).

Data We use Kinetics-600 for class-to-video setting with a controlled action label, and uCO3D for text-to-video over objects including an in-distribution (ID) and out-of-distribution (OOD) (never seen in training) class split.

Model Following VideoFlexTok (Atanov et al., 2026), SemanTok shares the backbone, decoder REPA, codebook, sequence length, sampling settings, LLaMA-style AR recipe, and pixel resolution (17 frames at 128×128 on both datasets). SemanTok changes the encoder input and adds the token-level objectives of section 3 using DINOv2.

Evaluation The metrics we evaluate SemanTok on are:

- **Fidelity** i.e. closeness in appearance to real video, measured by gFVD and gFID.
- **Semantic alignment** i.e. whether the video shows the conditioned content: class accuracy, text–video cosine similarity (ViCLIP), video–video cosine similarity in ViCLIP space (ClipV). For the tokenizer we also report decoder semantic alignment (REPA), the DINOv2 cosine similarity of the decoder-REPA readout.

On Kinetics-600, class accuracy is closed-set UMT-L top-1 (Li et al., 2023) over 2,048 generated clips. On uCO3D, it is nearest-class-mean top-1 in InceptionV3 space over 2,560 clips per split,

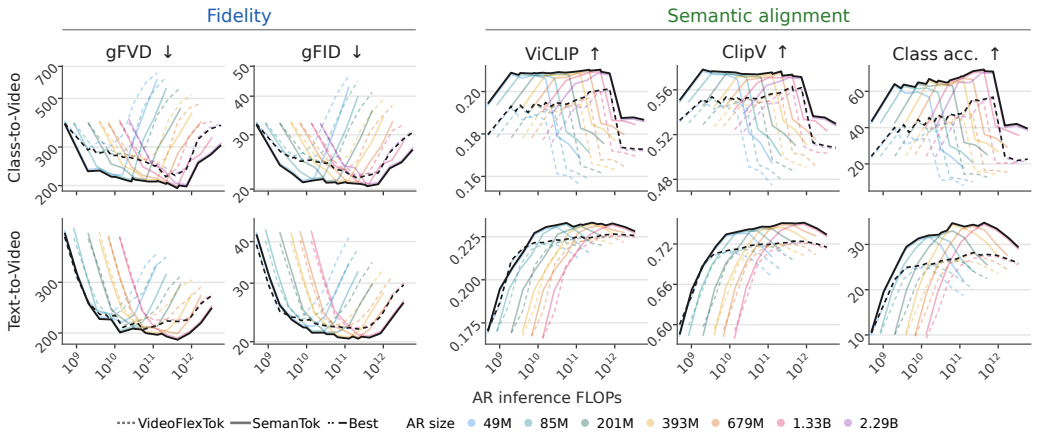


Figure 4: ❶ SemanTok improves generation at every compute budget on Kinetics-600 and uCO3D. Generation versus AR inference FLOPs per clip. Each faded curve is one AR size (colour) sweeping the token budget k from 1 to 256 tokens per frame. Black: the best score each tokenizer reaches at a given compute. SemanTok's envelope is better over most of the compute range.

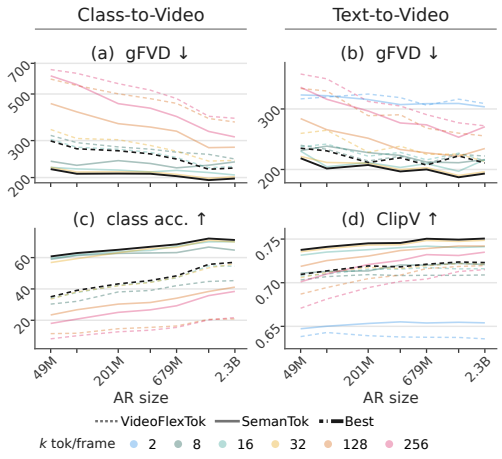


Figure 5: ❷ Fidelity scales with AR model size for both VideoFlexTok and SemanTok, while the semantic-alignment gap between does not close.

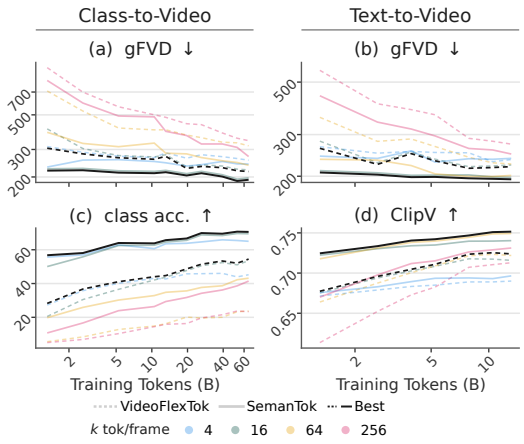


Figure 6: ❷ Semantic alignment gap between SemanTok and VideoFlexTok closes slower than the fidelity gap between them.

which stays defined for held-out classes. Additional details, including bootstrap intervals, the scope of our ablations, and the tokenizer training budget, are in section D.

Scalability and efficiency. We sweep seven AR sizes from 49M to 2.29B parameters, their inference FLOPs, and a 1.33B AR model trained from 1.3B to 65.5B tokens. We measure efficiency at two levels. At the model level, it is the AR size needed to reach a given fidelity and semantic alignment. At the token level, it is how hard the tokens are to predict: the validation cross-entropy (in bits per token). All AR models train on all 256 tokens per latent frame, and the budget $k \in \{1, 2, 4, \ldots, 256\}$ is chosen only at evaluation.

5 Results

❶ SemanTok exhibits high semantic alignment and video fidelity at every AR model size. With each tokenizer at its best k, every SemanTok AR model reaches lower gFVD and gFID and higher class accuracy than the VideoFlexTok AR model of the same size, on both Kinetics-600 and uCO3D (figs. 4 and 5). On Kinetics-600, SemanTok lowers gFVD by 11–24% and gFID by 5–17% and raises class accuracy by 25–61%, with the largest gains for the smallest AR models. On uCO3D, SemanTok improves gFVD, gFID, and class accuracy by 2–13%, 5–8%, and 22–30%, with no clear trend in AR model size. At the single budget k=16, an 85M SemanTok

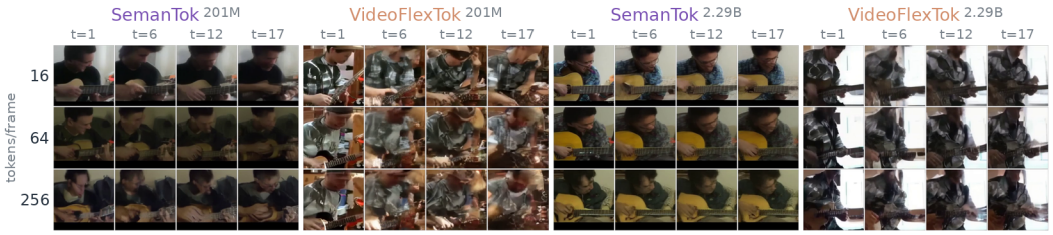


Figure 7: ❷ SemanTok stays ahead at both AR model sizes.Class-to-video rollouts for one Kinetics-600 playing-guitar label. At 201M, SemanTok degrades slowly over time, while VideoFlexTok is sharp only at t=1. Videos in the supplementary material.

AR model beats VideoFlexTok AR model of every size and budget, up to 2.29B, in gFID on both datasets and in gFVD on Kinetics-600. The 85M SemanTok AR model matches that VideoFlexTok AR model in uCO3D gFVD and beats it in class accuracy on both datasets.

In semantic alignment even the smallest SemanTok model (49M) beats the best VideoFlexTok AR model of any size, up to 2.29B, in class accuracy, ClipV, and ViCLIP on both datasets. On Kinetics-600, SemanTok's class accuracy is 0.631 against 0.560 for the 2.29B VideoFlexTok model.

More tokens cost more AR compute and their generations degrade in gFVD beyond a certain point. However, VideoFlexTok gives worse gFVD after only k=16, while SemanTok does not degrade until after k=32.

❷ **Increasing SemanTok model size further improves fidelity.** Larger or longer-trained AR models mostly trim the compounding error of the token rollout, which improves fidelity. The same scaling gains less in semantic alignment, because what the clip shows is largely decided by the first tokens, which come from the tokenizer (❻). Figure 5 separates the two axes by AR size. SemanTok's best-k class accuracy and ClipV lie above VideoFlexTok at every size. Fidelity improves with AR model size for both tokenizers, and SemanTok reaches a given fidelity with fewer parameters. From 49M to 2.29B, SemanTok's best-k gFVD falls from 224 to 202 on Kinetics-600 and from 218 to 197 on uCO3D. SemanTok's benefit is largest where AR capacity is scarce: at k=64, SemanTok's gFID gain over VideoFlexTok halves from 49M to 2.29B. Training budget behaves like model size (fig. 6). For the 1.33B AR model on Kinetics-600, SemanTok's best-k gFVD lead shrinks from 28% to 9% by 26B tokens, then holds at 11–14%. SemanTok's class-accuracy lead on Kinetics-600 persists, at 30% after 65.5B tokens. So 5× more training buys VideoFlexTok some fidelity, but not semantic alignment. On uCO3D, SemanTok's ClipV nearly saturates by 13B tokens and stays above VideoFlexTok's. As one rollout demonstrates (fig. 7), scaling to 2.29B sharpens the videos of both tokenizers, but SemanTok's maintains the lead.

❸ **SemanTok is able to maintain semantic alignment over out-of-distribution classes.** We test this on uCO3D, whose out-of-distribution (OOD) object classes are never seen in training, unlike its in-distribution (ID) classes. SemanTok leads on semantic-alignment in both tokenizer reconstruction and AR generation. SemanTok prefixes recover the object class at lower k better than VideoFlexTok's on both ID and OOD clips (fig. 8). At k=16 on uCO3D, SemanTok's reconstructions score higher ClipV and class accuracy than VideoFlexTok's but about 2.7 dB lower PSNR (table 2). The lead carries over to AR generation: with a 201M AR model at k=16, SemanTok raises generated class accuracy over VideoFlexTok by 24% on ID clips and by 29% on OOD classes, and improves ClipV on both (table 1). ViCLIP favors SemanTok from k=16 on, but not at k=4.

❹ **SemanTok achieves higher decoder-REPA semantic alignment at all noise levels, including the pure noise setting.** Decoder REPA aligns an early decoder layer with DINOv2 features(section 3). The decoder also sees a partly noised latent, which can supply part of this target without the tokens. We therefore read out the REPA projection on both datasets while varying the noise level σ of the decoder input (fig. 9). For generation from pure noise, SemanTok's readout has a higher DINOv2 cosine similarity than VideoFlexTok's at every k. At k=32 on uCO3D, SemanTok's pure-noise readout reaches the similarity that VideoFlexTok reaches only with a 75%-clean latent (0.718 vs. 0.716). At k=256 the latent adds almost nothing to SemanTok's readout, but nearly 9× more to VideoFlexTok's (section C). (The exception is the smallest budgets with a mostly clean latent, see ❼).

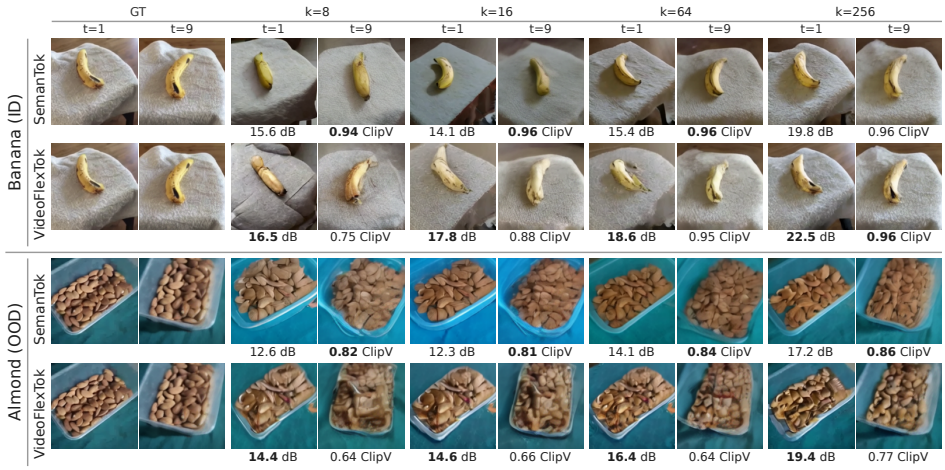


Figure 8: ❸ SemanTok reconstructions on uCO3D from ground-truth tokens recover the original object class at an earlier k than VideoFlexTok; for both in-distribution (ID) clips and out-of-distribution (OOD) clips.

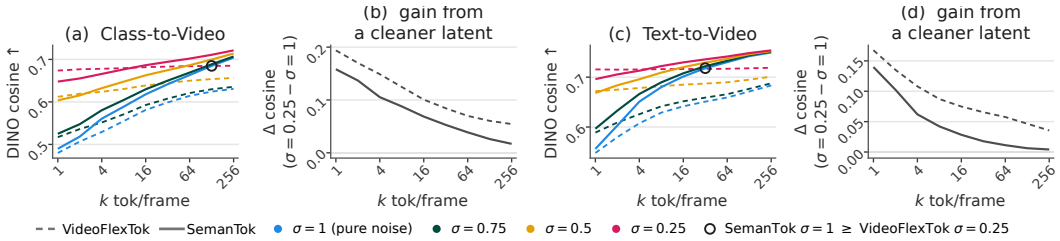


Figure 9: ❹ SemanTok's decoder-REPA readout depends less on the noised latent and achieves higher REPA-alignment than VideoFlexTok's.

❺ **SemanTok has high fidelity on reconstruction as well as generation** (see fig. 10). VideoFlexTok reconstructs somewhat better in PSNR and rFVD, but it loses more fidelity in generation, so its realization gap is wider. SemanTok has the lower gFVD at every AR size, at both k=64 and k=128, and hence lower realization gap. (On Kinetics-600, generated class accuracy can exceed reconstruction because the AR model sees the class label.)

❻ **SemanTok's generation fidelity gain comes from its earlier tokens, which are cheaper to predict.** Two measurements on Kinetics-600 show that SemanTok's gain is concentrated in the earlier tokens. First, we force the first m ground-truth tokens per frame and free-run to k=256 (fig. 11*(a)*). With nothing forced, SemanTok's gFVD is 23% lower than VideoFlexTok's at 201M. SemanTok's lead vanishes once the first 16–64 tokens are forced, after which VideoFlexTok's better-reconstructing tail edges ahead. Second, SemanTok's prefix is cheaper to predict (fig. 11*(b)*).

The bits the AR model needs per predicted token is measured by cross-entropy. At k=16, SemanTok needs 32% fewer bits per token than VideoFlexTok (8.8 vs. 12.9 at 201M). SemanTok's marginal entropy is only about one bit lower, so most of the saving comes from context. It is to be noted that the cost saving does not come from repetition: SemanTok repeats tokens less often overall than VideoFlexTok (section D.4).

SemanTok's cheap prefix keeps the semantics and defers pixel detail. SemanTok's first 4 tokens cost 37 bits per frame, yet nearly match the class accuracy of VideoFlexTok's first 32 tokens (405 bits, table 2). Later tokens restore part of that detail, and the generative decoder fills in the rest.

❼ **SemanTok with one token per frame struggles to serve every objective.** This fact is consistent throughout our experiments (tables 6 to 10). SemanTok's losses to VideoFlexTok that clear the bootstrap intervals are all on uCO3D at the smallest budgets: gFID and ClipV at k=1 for every AR size, gFID at k=4 for 49M and 679M, and gFVD at k=1 for 85M. On Kinetics-600, SemanTok is never significantly worse. Decoder REPA shows the same budget limit (fig. 9): at k=1, SemanTok's pure-noise readout leads by only 0.01 DINOv2 cosine, against 0.04–0.07 from

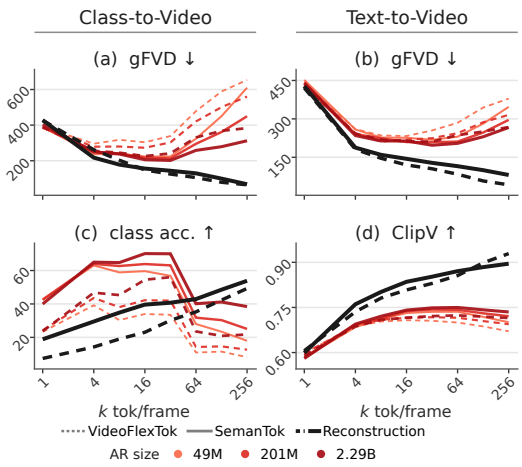


Figure 10: 5 SemanTok loses less fidelity and semantic alignment from reconstruction to generation than VideoFlexTok.

		k=4	k=16	k=256
Acc.	VideoFlexTok	0.23/0.16	0.27/0.18	0.24/0.16
	SemanTok	**0.27/0.18**	**0.34/0.23**	**0.27/0.18**
ViCLIP	VideoFlexTok	**0.21/0.19**	0.23/0.20	0.22/0.19
	SemanTok	0.21/0.18	**0.23/0.20**	**0.23/0.20**
ClipV	VideoFlexTok	0.70/**0.63**	0.73/0.65	0.70/0.63
	SemanTok	**0.70**/0.62	**0.75/0.66**	**0.73/0.65**

Table 1: 3 The semantic gain holds on unseen classes. Semantic alignment of generated videos from a 201M AR model for in-distribution (ID) and out-of-distribution (OOD) samples of uCO3D dataset; bold is better. SemanTok leads class accuracy on both splits at every k.

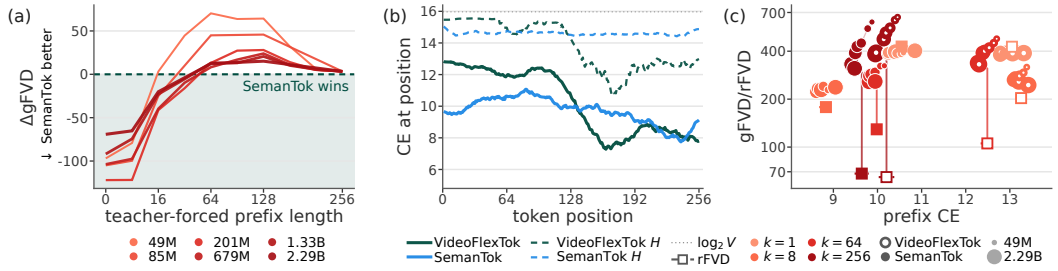


Figure 11: 6 SemanTok's generation gain comes specifically from its first tokens, which are cheaper to predict (Kinetics-600). (a) ΔgFVD after teacher-forcing the first m ground-truth tokens per frame and free-running to k=256: at 201M, SemanTok leads by 23% with nothing forced, and forcing only the first 16–64 tokens removes the lead. (b) Under a 201M AR model, SemanTok's cross-entropy per token position is lower for the first 128 positions and higher over 129–192. (c) Larger AR models narrow the gap between generation and reconstruction for both tokenizers, but none closes it.

k=16, and with a mostly clean latent (σ=0.25) SemanTok's readout is lower up to k=4 on uCO3D and k=8 on Kinetics-600. A token carries at most about 16 bits, too few to satisfy flow matching, decoder REPA, and semantic supervision at the same time. SemanTok still never falls behind in class accuracy, at any budget or AR size, and SemanTok's fidelity lead follows at larger budgets 1.

Limitation : Because SemanTok prioritizes semantic alignment over reconstruction, it struggles to reconstruct the same colors/appearance details at lower token budgets.

6 Conclusion

We introduce SemanTok, an AR video generation tokenizer and prediction module, which emphasizes semantic alignment at flexible token budget. Semantic supervision from a frozen teacher, as encoder input and as a target for every retained prefix, makes token prefixes richer in semantics (1), lends to scaling in size (2), maintains generalization in out-of-distribution setting (3), has better semantic alignment in high noise inputs (4), performs well at both reconstruction and generation (5), and is cheaper to predict (6).

Future directions include optimizing predictability directly, organizing early prefixes with video-native or language-aligned teachers, and using such prefixes as compact states for action-conditioned world models.

Reproducibility Statement

Both tokenizers share the architecture and objective in section 3. SemanTok's changes are the DINO encoder input, the class-token injection, and the two prediction heads with their loss weights. Table 3 lists tokenizer settings and training budgets. Tables 4 and 5 list the AR size ladder, learning rates, and dataset-specific regularization. Section D gives the sampling settings, including the budget-dependent AR guidance. It also describes each metric and its readout model: FVD, FID, ViCLIP, ClipV, UMT-L, and the nearest-class-mean protocol for held-out classes. All datasets and pretrained models are public and cited where they are first used.

AI Use Statement

We used generative AI tools to implement methods, to give feedback on our research methodology and experiments, and to help interpret results. We also used them to write and edit code, including training, evaluation, and plotting scripts, and to draft and revise the text of some parts of the paper. We did not use them to generate synthetic data, develop theoretical models or conceptual frameworks, propose hypotheses, translate, clean datasets, or analyze qualitative data. Mathematical claims and proofs do not apply to this work. The authors checked all code, results, and text, and take full responsibility for the content of this paper.

References

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Andrei Atanov, Jesse Allardice, Roman Bachmann, Oğuzhan Fatih Kar, R. Devon Hjelm, David Griffiths, Peter Fu, Afshin Dehghan, and Amir Zamir. Videoflextok: Flexible-length coarse-to-fine video tokenization. *arXiv preprint arXiv:2604.12887*, 2026.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1D token sequences of flexible length. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 2241–2292. PMLR, 2025.

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning*, 2024.

Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and Sergey Tulyakov. Panda-70M: Captioning 70M videos with multiple cross-modality teachers. In *CVPR*, 2024.

Niladri Shekhar Dutt, Zifan Shi, Paul Guerrero, Chun-Hao Paul Huang, Duygu Ceylan, Niloy J. Mitra, and Xuelin Chen. LoST: Level of semantics tokenization for 3D shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Zixuan Fu, Lanqing Guo, Chong Wang, Binbin Song, Ding Liu, and Bihan Wen. Improving flexible image tokenizers for autoregressive image generation. *arXiv preprint arXiv:2601.01535*, 2026.

Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, et al. VideoPoet: A large language model for zero-shot video generation. In *International Conference on Machine Learning*, 2024.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, et al. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, 2022.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19948–19958, 2023.

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

Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. ImageFolder: Autoregressive image generation with folded tokens. In *International Conference on Learning Representations*, 2025a.

Zongyi Li, Shujie Hu, Shujie Liu, Long Zhou, Jeongsoo Choi, Lingwei Meng, Xun Guo, Jinyu Li, Hefei Ling, and Furu Wei. ARLON: Boosting diffusion transformers with autoregressive models for long video generation. In *International Conference on Learning Representations*, 2025b.

Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. UniTok: A unified tokenizer for visual generation and understanding. In *Advances in Neural Information Processing Systems*, 2025.

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

NVIDIA. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Vivek Ramanujan, Kushal Tirumala, Armen Aghajanyan, Luke Zettlemoyer, and Ali Farhadi. When worse is better: Navigating the compression-generation trade-off in visual tokenization. In *Advances in Neural Information Processing Systems*, pp. 138949–138976, 2025. doi: 10.52202/085713-4178.

Oren Rippel, Michael Gelbart, and Ryan Adams. Learning ordered representations with nested dropout. In *International Conference on Machine Learning*, 2014.

Anni Tang, Tianyu He, Junliang Guo, Xinle Cheng, Li Song, and Jiang Bian. Vidtok: A versatile and open-source video tokenizer. *arXiv preprint arXiv:2412.13061*, 2024.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems*, 2024.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. In *International Conference on Learning Representations*, 2023.

Hanyu Wang, Saksham Suri, Yixuan Ren, Hao Chen, and Abhinav Shrivastava. LARP: Tokenizing videos with a learned autoregressive generative prior. In *International Conference on Learning Representations*, 2025.

Xin Wen, Bingchen Zhao, Ismail Elezi, Jiankang Deng, and Xiaojuan Qi. "Principal Components" enable a new language of images. In *International Conference on Computer Vision*, 2025.

Tianwei Xiong, Jun Hao Liew, Zilong Huang, Jiashi Feng, and Xihui Liu. GigaTok: Scaling visual tokenizers to 3 billion parameters for autoregressive image generation. In *International Conference on Computer Vision*, 2025.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using VQ-VAE and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Wilson Yan, Volodymyr Mnih, Aleksandra Faust, Matei Zaharia, Pieter Abbeel, and Hao Liu. ElasticTok: Adaptive tokenization for image and video. In *International Conference on Learning Representations*, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion—tokenizer is key to visual generation. In *International Conference on Learning Representations*, 2024a.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *Advances in Neural Information Processing Systems*, 2024b.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. SpeechTokenizer: Unified speech tokenizer for speech large language models. In *International Conference on Learning Representations*, 2024.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 79115–79135. PMLR, 2025.

Yongxin Zhu, Bocheng Li, Hang Zhang, Xin Li, Linli Xu, and Lidong Bing. Stabilize the latent space for image autoregressive modeling: A unified perspective. In *Advances in Neural Information Processing Systems*, 2024.

A QUALITATIVE EXAMPLES

Stills from reconstructions (ground-truth tokens through each frozen decoder) and from AR-model rollouts. Every k column is a prefix of the same token sequence: the clip's encoder tokens for reconstruction, one rollout's tokens for generation. Stills cannot show temporal consistency, so we

recommend watching the videos in the supplementary material: each caption names its video file, and the folder holds further examples.



Figure 12: Tokenizer reconstruction on uCO3D from ground-truth tokens: two in-distribution clips and one out-of-distribution clip. PSNR and ClipV against VAE-GT; bold is best per k. Videos: `ReconstructionUCO3D.mp4`, `ReconstructionUCO3D_Flashlight.mp4`.

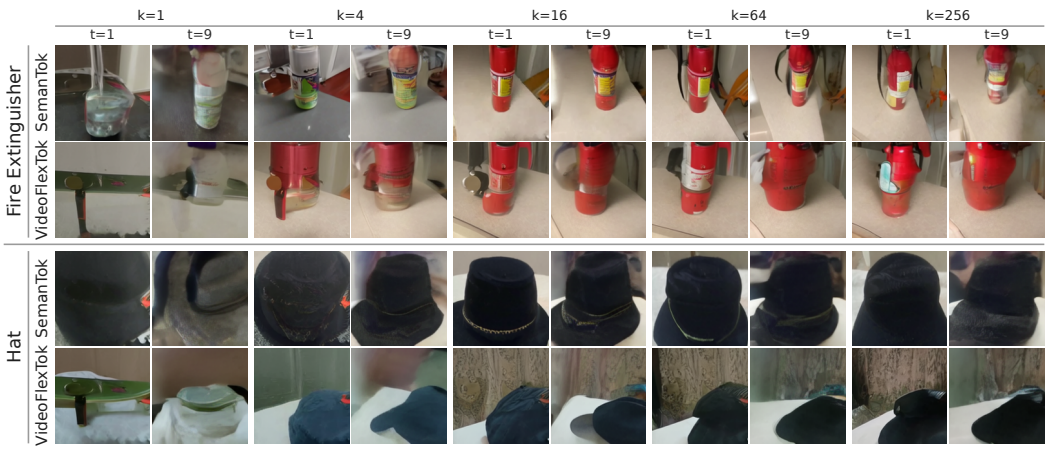


Figure 13: Text-to-video on uCO3D, 201M AR model; one rollout per tokenizer, decoded from its first k tokens per frame. Prompts: "A red fire extinguisher with a black handle and silver nozzle, sits on a countertop. It has a white label on its side and a black strap around its middle." and "A black fedora hat with a short brim and small, round crown sits on a white surface. It has two small holes at the top of the crown." Video: `GenerationUCO3D_FireExtinguisher.mp4`.

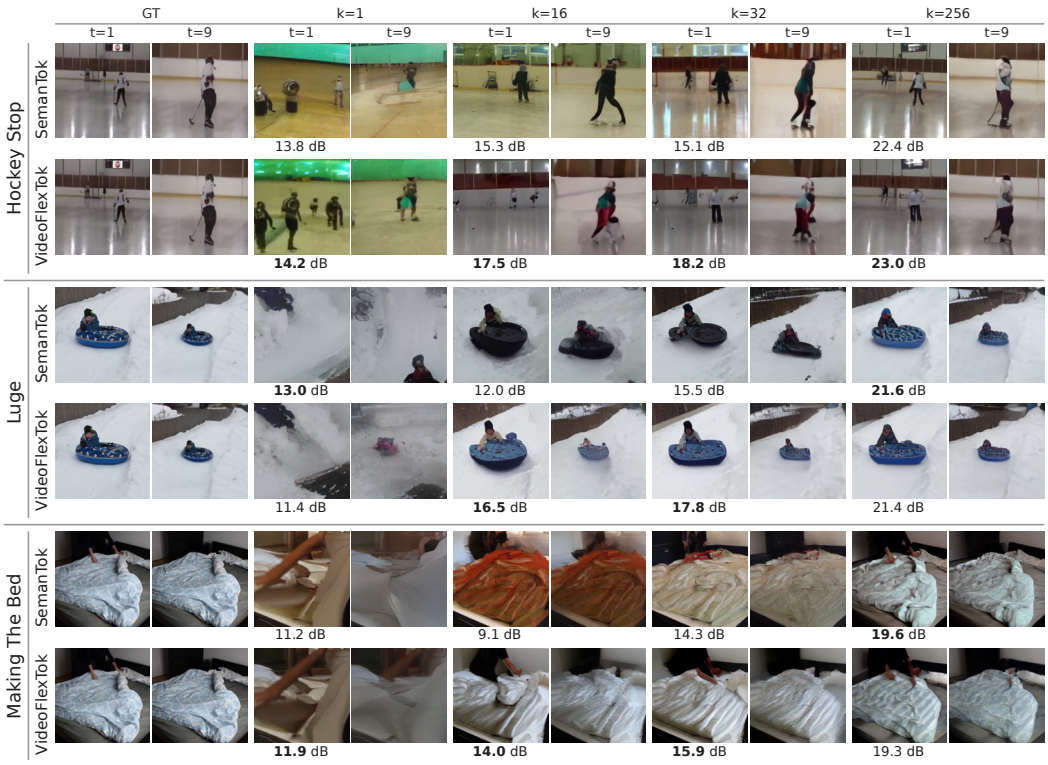


Figure 14: Tokenizer reconstruction on Kinetics-600 from ground-truth tokens. PSNR against VAE-GT; bold is best per k. Videos: `ReconstructionK600_HockeyStop.mp4`, `ReconstructionK600_Luge.mp4`.

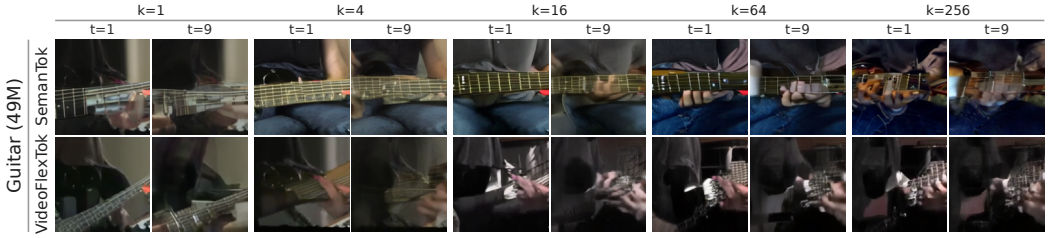


Figure 15: Class-to-video on Kinetics-600 (playing guitar) with a 49M AR model; one rollout per tokenizer (the first evaluation sample of the class), decoded from its first k tokens per frame. Video: `GenerationK600_Guitar_49M.mp4`.



Figure 16: Class-to-video on Kinetics-600 (cooking egg) across AR model sizes at k=16, 32, 64, frame t=9. Each (size, tokenizer) is one rollout decoded from its first k tokens per frame. SemanTok's is the first evaluation sample of the class; VideoFlexTok's is, per size, the one of its rollouts of the class closest to SemanTok's in ViCLIP embedding. Video: `GenerationK600_CookingEgg_k16-64_ARsize.mp4`.



Figure 17: Text-to-video on uCO3D (out-of-distribution category) across AR model sizes at k=8, 32, 256, frame t=9. Each (size, tokenizer) is that model's evaluation rollout for the same prompt, decoded from its first k tokens per frame. Prompt: "A light blue dumbbell with a hexagonal handle and a rounded head sits on a surface. It has white text or numbers printed on its side, such as 'Tone' or '5 LB'. The overall color scheme is dominated by the blue of the dumbbell itself." Video: `GenerationUCO3D_Barbell_ARsize_x_k.mp4`.

B TOKENIZER RECONSTRUCTION VERSUS TOKEN BUDGET

All rows score the same fake side — the frozen decoder applied to ground-truth tokens. These are tokenizer reconstruction measurements, not the AR-model rollouts in fig. 4. Only the reference differs, as each metric requires: rFVD against the reference distribution, ViCLIP against the caption, and PSNR/SSIM per clip against VAE- GT. Compare within a row; the metrics are not comparable across references. Figure 10 overlays the AR-model curves from fig. 4 on these reconstruction numbers.

C WHERE THE TOKENIZERS KEEP DINO SEMANTICS

Both tokenizers train an early decoder layer to predict DINOv2 patch features through decoder REPA. We read that projection out without any training, which asks where the decoder finds its semantics. We keep the first k tokens per frame and fix the decoder noise level σ, with the same noise seed for both tokenizers and every k. At σ=1 the decoder input is pure noise, so any DINO content in the readout comes from the token prefix. Generation starts from this state, with AR-sampled tokens as the only input. At lower σ the noised latent also carries the clip; we measure

Table 2: Tokenizer reconstruction versus k. Bold is better.

		1	4	8	16	32	64	128	256
Kinetics-600									
$rFVD \downarrow$	VideoFlexTok	**426.7**	262.0	203.0	**148.1**	**125.2**	**105.5**	**79.1**	**64.8**
	SemanTok	427.8	**217.4**	**178.3**	156.0	143.9	129.3	99.6	68.2
ViCLIP↑	VideoFlexTok	0.155	0.163	0.168	0.174	0.177	0.182	0.185	0.188
	SemanTok	**0.170**	**0.179**	**0.182**	**0.185**	**0.186**	**0.187**	**0.190**	**0.192**
ClipV↑	VideoFlexTok	0.569	0.631	0.678	0.730	0.767	**0.804**	**0.840**	0.867
	SemanTok	**0.601**	**0.681**	**0.715**	**0.748**	**0.775**	0.803	0.839	**0.873**
class acc.↑	VideoFlexTok	0.074	0.143	0.190	0.230	0.302	0.353	0.423	0.493
	SemanTok	**0.188**	**0.295**	**0.348**	**0.396**	**0.407**	**0.429**	**0.485**	**0.539**
PSNR↑	VideoFlexTok	**10.24**	**11.55**	**13.17**	**14.64**	**15.80**	**17.15**	**18.53**	**19.66**
	SemanTok	10.13	10.51	10.78	11.33	13.02	14.78	16.79	18.53
SSIM↑	VideoFlexTok	**0.278**	**0.319**	**0.370**	**0.417**	**0.466**	**0.529**	**0.607**	**0.668**
	SemanTok	0.277	0.288	0.297	0.316	0.372	0.446	0.543	0.634
uCO3D									
$rFVD \downarrow$	VideoFlexTok	**420.8**	**181.1**	**146.2**	**120.1**	**105.1**	**83.7**	**57.6**	**42.4**
	SemanTok	427.4	187.7	159.6	144.0	128.2	115.2	99.7	80.8
ViCLIP↑	VideoFlexTok	0.157	0.194	0.206	0.213	0.218	0.224	0.234	**0.241**
	SemanTok	**0.166**	**0.206**	**0.214**	**0.227**	**0.231**	**0.237**	**0.237**	0.239
ClipV↑	VideoFlexTok	**0.606**	0.736	0.781	0.808	0.830	0.856	**0.898**	**0.929**
	SemanTok	0.599	**0.761**	**0.801**	**0.835**	**0.852**	**0.871**	0.884	0.895
class acc.↑	VideoFlexTok	0.092	0.315	0.451	0.520	0.582	0.648	**0.733**	**0.780**
	SemanTok	**0.111**	**0.378**	**0.492**	**0.565**	**0.619**	**0.659**	0.694	0.719
PSNR↑	VideoFlexTok	**13.46**	**16.09**	**17.34**	**18.66**	**19.73**	**20.82**	**22.59**	**23.88**
	SemanTok	12.49	14.28	15.09	15.96	17.39	18.86	20.49	21.46
SSIM↑	VideoFlexTok	**0.491**	**0.535**	**0.565**	**0.596**	**0.629**	**0.672**	**0.734**	**0.780**
	SemanTok	0.464	0.509	0.525	0.544	0.577	0.619	0.676	0.717

$\sigma \in \{1, 0.75, 0.5, 0.25\}$. We report the mean cosine to the true DINOv2 features of each latent frame's first RGB frame, the target both tokenizers train against. Each dataset uses 256 validation clips and the tokenizers of section 5 (66B training tokens on uCO3D, 131B on Kinetics-600). As a check, at k=256 and σ=0.25 the readout lies within 0.05 of the alignment each trainer logged for the same checkpoint.

At low noise, VideoFlexTok's decoder takes its semantics from the latent. At σ=0.25, its readout barely depends on k (fig. 18a, d). It moves from 0.716 at k=1 to 0.719 at k=256 on uCO3D, and from 0.674 to 0.685 on Kinetics-600. This suggests that the noised latent meets much of its decoder REPA target, which weakens the pressure on the prefix to hold more semantics. Its curves rise with k mainly at high noise and stay separated by noise level up to k=256. From pure noise its readout rises from 0.548 to 0.683 on uCO3D and from 0.480 to 0.631 on Kinetics-600, so its prefix does carry semantics. SemanTok's σ=0.25 readout keeps rising with k, from 0.696 to 0.754 and from 0.648 to 0.721.

SemanTok's prefix carries them. From pure noise, SemanTok's readout reaches 0.749 vs. 0.683 at k=256 on uCO3D and 0.704 vs. 0.631 on Kinetics-600. Its four noise levels converge as k grows (fig. 18b, e). At k=256 the gap between σ=0.25 and pure noise falls to 0.004 on uCO3D (0.017 on Kinetics-600), against 0.035 (0.054) for VideoFlexTok. SemanTok's prefix alone therefore gives the decoder, by k=256, nearly all the DINO content that the latent adds. From pure noise, SemanTok reaches VideoFlexTok with a half-clean latent (σ=0.5) from k=16 on uCO3D (0.701 vs. 0.685) and k=64 on Kinetics-600 (0.663 vs. 0.650), and with a 75%-clean latent (σ=0.25) from k=32 (0.718 vs. 0.716) and k=128 (0.685 vs. 0.685), always at the same k. Decoder REPA is present in both tokenizers, so the difference comes from the token-side changes. SemanTok's lead grows with the noise level (fig. 18c, f). At $\sigma \geq 0.75$ it leads at every k on both datasets, by 0.008–0.010 at k=1 and 0.063–0.074 at k=256. SemanTok's dense token head reads the prefix without the decoder. It trails every SemanTok decoder readout but also rises monotonically with k. On uCO3D from k=8 to 128

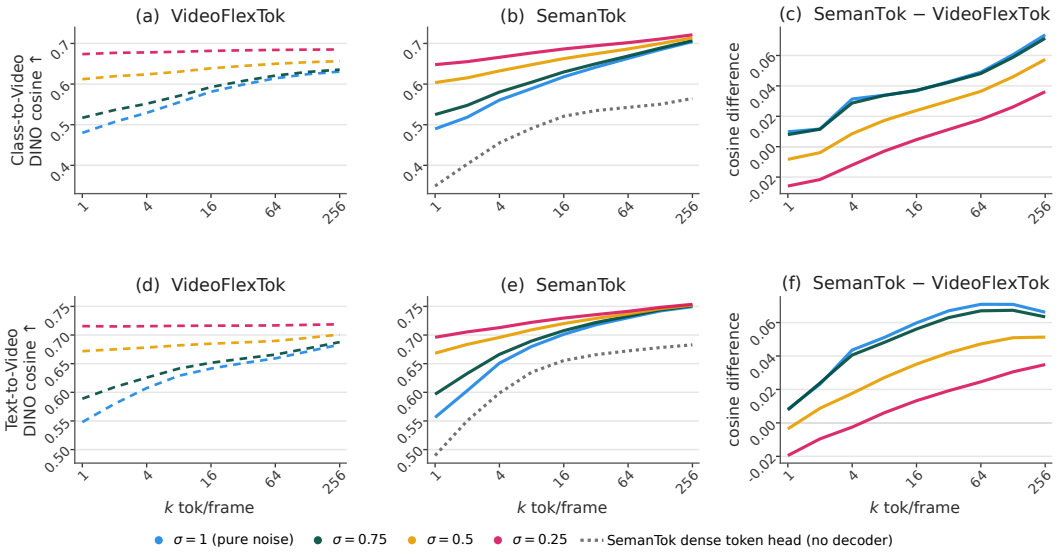


Figure 18: Where the tokenizers keep DINO semantics. Decoder-REPA readout of DINOv2 from the first k tokens per frame, 256 validation clips per dataset. Rows: Class-to-Video (Kinetics-600) and Text-to-Video (uCO3D). Colour: decoder noise level σ; at $\sigma{=}1$ the input is pure noise, so the prefix is the decoder's only source. (a, d) VideoFlexTok. (b, e) SemanTok; dotted: its dense token head, which reads the prefix without the decoder. (c, f) SemanTok minus VideoFlexTok at each σ; positive favors SemanTok. Figure 9 overlays (a, b) and (d, e) and adds the gain from $\sigma{=}1$ to $\sigma{=}0.25$.

this two-layer head even exceeds VideoFlexTok's pure-noise decoder readout (0.656 vs. 0.641 at $k{=}16$).

Limits. At the smallest budgets the tokenizers are on par. The pure-noise difference is at most 0.01 at $k{=}1$. With a cleaner latent, VideoFlexTok is ahead up to $k{=}2$ at $\sigma{=}0.5$ and up to $k{=}8$ at $\sigma{=}0.25$, by at most 0.026. This matches the mixed $k{\leq}4$ regime in section 5. The probe also does not separate the token-side changes from one another: DINO in the encoder input, the class-token injection, and the prefix DINO targets.

D Experimental details

D.1 Tokenizer training

VideoFlexTok and SemanTok share the tokenizer architecture and optimizer in table 3. Both use a frozen VidTok-128 VAE, an 18-layer encoder, an 18-layer time-causal decoder, and a six-dimensional FSQ bottleneck. Nested dropout samples uniformly from powers of two through $k{=}256$. The decoder REPA weight is 1.0 in both arms.

SemanTok additionally concatenates frozen DINOv2-L features to the encoder input with a learned linear projection. Its dense and class-token prediction losses each have weight 0.5. The DINO features are $16 \times 16 \times 1024$ per latent frame, and the class token is projected into register token zero with a zero-initialized layer. These heads are causal over latent frames. The paper results use tokenizers trained on 66B tokens on uCO3D and 131B tokens on Kinetics-600.

Ablations. Early in the project, we ran smaller-scale ablations to narrow down the design. They compared SigLIP 2 (Tschannen et al., 2025) with DINOv2 as the semantic teacher, and we kept DINOv2. They also favoured using both forms of semantic supervision together: frozen DINOv2 features at the encoder input, as dense patch features and as a class-token bias on the first register token, and explicit DINO targets on every retained token prefix.

DINO frames. The VAE compresses time causally by $4\times$, so the 17 RGB frames map to the five latent frames as $\{0\}$, $\{1..4\}$, $\{5..8\}$, $\{9..12\}$, and $\{13..16\}$. SemanTok's dense DINO features for a

latent frame, used as encoder input and as the dense target, average DINOv2 over that frame's RGB group. The first latent frame uses frame 0 alone. The class token $\mathbf{c}_t$ comes from the first RGB frame of each group (frames 0, 1, 5, 9, and 13).

Knob	uCO3D	Kinetics-600
Tokenizer training tokens	66B	131B
RGB frames / latent frames	17 / 5	17 / 5
Latent grid / channels	16×16 / 16	same
Encoder / decoder width	1152 / 1152	same
Tokens per latent frame	256	256
FSQ levels / vocabulary	$[8, 8, 8, 5, 5, 5]$ / 64k	same
Global / micro batch	512 / 8	512 / 8
Peak LR / warmup tokens	2.5×10^{-4} / 1.3B	2.5×10^{-4} / 3.9B
AdamW β / grad clip	$(0.9, 0.99)$ / 1.0	same
Random-resized-crop scale	$[0.6, 1.0]$	$[0.08, 1.0]$
Decoder eval	50 steps, CFG 3.0	same

Table 3: Tokenizer settings used by the reported checkpoints.

D.2 TOKENIZER TRAINING BUDGET

The released VideoFlexTok tokenizer uses roughly 400B training tokens on Kinetics-600. We train both arms for 131B tokens on Kinetics-600 and 66B on uCO3D, less than a third of that budget, and our Kinetics-600 tokenizers were still improving. AR models trained on earlier tokenizer checkpoints show SemanTok's lead at every checkpoint, with no sign of the gap closing (fig. 19), so a longer budget is unlikely to reverse it.

Kinetics-600 still improves. From 66B to 131B tokens, both tokenizers reconstruct better at k=256. These numbers score 2,560 clips against their VAE-decoded ground truth, not table 2's 2,048-clip real reference bank, so absolute values and the k=256 rFVD ordering differ from that table. VideoFlexTok's rFVD falls from 78.4 to 59.2, PSNR rises from 19.0 to 19.7 dB, and class accuracy from 0.418 to 0.491. SemanTok's rFVD falls from 71.0 to 54.9, PSNR rises from 18.1 to 18.5 dB, and class accuracy from 0.466 to 0.516. We therefore use the 131B checkpoints, and a longer budget would likely improve both arms further.

uCO3D overfits after 66B. We also continued both uCO3D tokenizers to 98B tokens. Reconstruction at k=16 got worse for both (256 clips, k=16): PSNR fell from 18.66 to 18.07 dB for VideoFlexTok and from 15.96 to 15.56 dB for SemanTok, and rFVD rose from 351 to 373 and from 425 to 469. For a 49M AR model at k=16, SemanTok's class accuracy and ClipV also dipped slightly (0.309 → 0.302 and 0.731 → 0.725), although its gFVD improved (226 → 205). We attribute the decline to overfitting on the small uCO3D training set and report the 66B checkpoints for both arms.

The uCO3D lead holds throughout training. To test whether SemanTok only converges faster, we took six tokenizer checkpoints between 6.6B and 66B training tokens and trained a fresh 49M AR model on each one's tokens for 8k steps (fig. 19). This recipe is cheaper than section 5's (unaugmented tokens, fewer AR steps), so compare only within the figure. SemanTok has higher ClipV at every checkpoint and both k. At 66B tokens it scores **0.710 vs. 0.680** at k=16 and **0.669 vs. 0.627** at k=256. Its gFVD is lower at every checkpoint at k=256 and from 26B tokens at k=16, reaching **198 vs. 224** and **386 vs. 452** at 66B. ViCLIP and class accuracy show the same ordering (class accuracy 0.295 vs. 0.214 at k=16, 66B). VideoFlexTok gains little after 26B tokens and closes none of these gaps. At k=256 SemanTok's own gFVD rises after 26B, so its lead there narrows from 199 to 66.

The Kinetics-600 lead holds throughout training. We repeat this test on Kinetics-600 with the full recipe of section 5 rather than a cheap probe: for tokenizer checkpoints at 26B, 66B, 98B, and 131B training tokens we train a 201M AR model for 20k steps on each one's tokens and score it with the protocol of fig. 4 (fig. 19, bottom row). SemanTok has higher class accuracy and ClipV at every

checkpoint and every k, and lower gFVD at every checkpoint from k=8. The gaps do not close with training: at k=16, gFVD is **291 vs. 349** at 26B and **217 vs. 273** at 131B, and class accuracy is **0.413 vs. 0.205** and **0.639 vs. 0.422**. At 66B tokens, half the budget, SemanTok already beats VideoFlexTok at 131B on all three metrics at every $k \geq 4$, so its lead is not an artifact of the 131B budget.

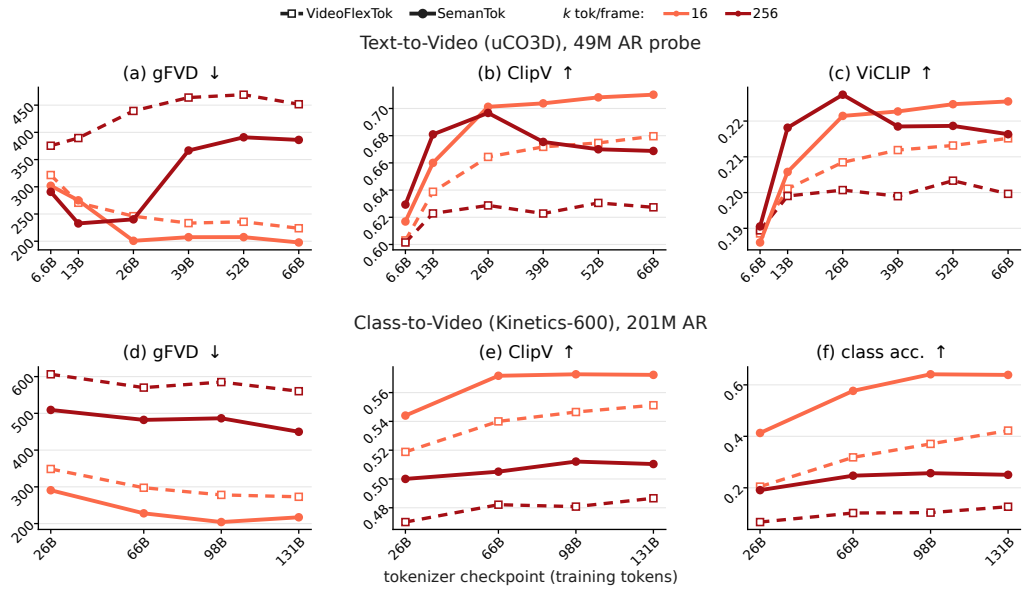


Figure 19: SemanTok's lead holds throughout tokenizer training on both datasets. Each point is an AR model trained on one tokenizer checkpoint's tokens. Top: uCO3D, a 49M AR probe trained for 8k steps, official validation split. Bottom: Kinetics-600, a 201M AR model with the recipe and evaluation of fig. 4. No gap closes as the tokenizer trains longer.

Released checkpoint. We cannot compare against the released VideoFlexTok Kinetics-600 checkpoint. Its decoder is fine-tuned for bidirectional attention, whereas the paper describes such fine-tuning only for Panda-70M (Chen et al., 2024). The reconstruction metrics we measured for this checkpoint do not match those reported in the paper, likely because of this deviation. No time-causal checkpoint is released, so we retrain VideoFlexTok with the same time-causal recipe, data, and budget as SemanTok.

D.3 Autoregressive training

The downstream AR model is a LLaMA-style causal decoder with RMSNorm and SwiGLU. At depth d, its width is $64d$ and it has d attention heads. It trains on the time-first sequence of TK=1,280 scalar indices $\tau_{t,i}$ with global batch 512. A budget k corresponds to the first Tk scalar indices, or k token positions per latent frame. All runs use AdamW with $\beta = (0.9, 0.95)$, weight decay 0.05, gradient clipping 1.0, bf16, a 2.5% warmup, and cosine decay to one percent of peak LR. Head bias uses the log-unigram initialization. Depth-scaled initialization is disabled. We otherwise follow VideoFlexTok. For uCO3D only, we increase trunk dropout and conditioning dropout after observing overfitting.

D.4 Evaluation protocols

AR generation. We follow the VideoFlexTok evaluation pipeline. Sampling uses temperature 1.0 without top-k or top-p truncation. The decoder uses 50 flow steps and guidance 3.0. Kinetics-600 AR guidance is 3.0 for $k \in \{1, 4\}$, 2.0 for $k \in \{8, 16, 32\}$, and 1.0 thereafter, following the budget dependence reported by VideoFlexTok. uCO3D uses AR guidance 3.0 at every budget. We apply the same settings to both tokenizers and did not tune guidance for either.

Non-emb. params	Depth	Width	Heads	uCO3D peak LR	K600 peak LR
49M	10	640	10	8.00×10^{-4}	1.60×10^{-3}
85M	12	768	12	6.67×10^{-4}	1.33×10^{-3}
201M	16	1024	16	5.00×10^{-4}	1.00×10^{-3}
393M	20	1280	20	4.00×10^{-4}	8.00×10^{-4}
679M	24	1536	24	3.33×10^{-4}	6.67×10^{-4}
1.33B	30	1920	30	2.67×10^{-4}	5.33×10^{-4}
2.29B	36	2304	36	2.22×10^{-4}	4.44×10^{-4}

Table 4: AR model size ladder. uCO3D uses $0.512/\text{width}$; Kinetics-600 uses the VideoFlexTok rule $1.024/\text{width}$.

Knob	Dataset-specific value
Conditioning	uCO3D: umT5, 128 tokens; K600: class
Tokenizer crop views	uCO3D: 25; K600: 6
AR training tokens	uCO3D: 13.1B at every size K600: 13.1B through 679M; 26.2B thereafter; 65.5B for fig. 6
Trunk dropout	uCO3D: 0.25; K600: 0.1
Condition drop	uCO3D: 0.2; K600: 0.1
Condition-feature drop	uCO3D: 0.1; K600: 0
Cross-attention drop	0

Table 5: Dataset-specific AR settings. Crop views affect only AR training; validation tokens are unaugmented.

Sample sizes and budget selection. On Kinetics-600, each (AR size, k) cell uses 2,048 generated clips, conditioned on labels drawn from 2,048 validation clips that also form the real FVD and FID reference. On uCO3D, each split (ID and OOD) uses 2,560 generated clips against 1,024 real clips. Figure summaries weight the splits 1,014:152, as in the official validation set; gFVD and gFID are pooled Fréchet distances, and the other metrics are weighted means. Table-level ID/OOD numbers (table 1) are per split. All numbers come from a single training run and sampling seed; section D.5 gives bootstrap intervals for the headline comparisons. Best-k values are selected per metric on the same evaluation set, which favors both arms equally; the fixed-k comparisons in section 5 involve no selection.

Tokenizer reconstruction. Table 2 contains no AR-model rollouts. We decode ground-truth tokens and apply the same rFVD, ViCLIP, ClipV, class accuracy, PSNR, and SSIM evaluation used for generation.

Semantic alignment. ViCLIP scores generated video against the conditioning caption. ClipV scores the same video against the VAE-decoded ground-truth clip in the ViCLIP video encoder, with no text encoding.

Class accuracy. On Kinetics-600, a UMT-L classifier finetuned for Kinetics-600 scores the first 16 of 17 frames, resized to 224×224, as in VideoFlexTok (Atanov et al., 2026; Li et al., 2023). Top-1 is taken against the clip's action class. For AR generation that class is the conditioning label. On uCO3D, each class centroid is the L2-normalized mean of real-frame Inception features in the evaluation pool. A frame is correct if its nearest centroid, by cosine, is the clip's true object class. NCM stays defined for OOD classes, which a closed-set classifier never saw. This readout is coarse, so on uCO3D we also report ViCLIP and ClipV. Tokenizer reconstruction uses the same two readouts on decoded ground-truth tokens.

Cross-entropy and teacher forcing. The cross-entropy probe uses ground-truth previous tokens and reports bits/token. In the hybrid Kinetics-600 experiment, we teacher-force m ground-truth token positions per frame, supplying Tm scalar indices, and then free-run to $k{=}256$ with the same AR model and decoder sampling settings. Cross-entropy is measured on 1,024 validation clips. The marginal entropy H of a position is the plug-in entropy of its code, with the Miller–Madow

correction, over 120k training clips; context bits are H minus cross-entropy. Prefix bits per frame sum the per-position cross-entropy over the first k positions.

Token repeats. On 60k Kinetics-600 training clips, a *duplicate* is a token equal to any earlier token of its frame, and a *copy* is a token equal to the one at the same position in the previous latent frame (fig. 20). Over all 256 positions, SemanTok duplicates 8.5% of tokens against VideoFlexTok's 12.0%, and copies 0.05% against 0.42%. In the first 16 positions, where its cross-entropy advantage is largest, both tokenizers stay near zero (SemanTok: 1.1% duplicates and 0.14% copies; VideoFlexTok: 0.2% and 0.02%).

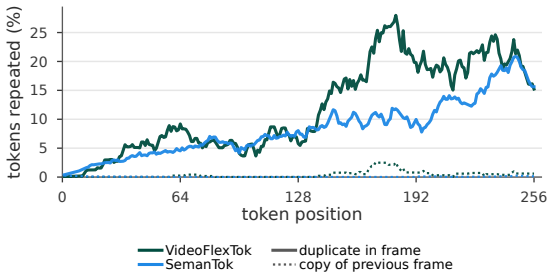


Figure 20: Token repeats by position on Kinetics-600 (running mean over 16 positions). Solid: duplicates of an earlier token in the same frame. Dotted: copies of the previous frame's token at the same position.

D.5 Uncertainty of the generation comparisons

We estimate uncertainty with a paired bootstrap over the evaluation clips, using 1,000 replicates. Both tokenizers condition on the same prompts or labels, so each replicate resamples the two arms jointly. We report SemanTok's advantage, signed so that positive values favour SemanTok, with its 95% interval. At k=16 we also pool six sampling seeds, which change only the AR and flow noise. On uCO3D the ID validation split holds only 1,014 clips, which caps the real reference.

Equal AR size. Figures 21 and 22 compare the tokenizers at equal AR size and budget. Semantic alignment favours SemanTok almost everywhere, with class accuracy from k=2 on uCO3D and at every budget on Kinetics-600, and ClipV from k=8. Fidelity needs longer prefixes. On uCO3D, gFID gains exclude zero at every size from k=32, and gFVD gains at $k \in \{64, 128\}$. On Kinetics-600, with six seeds pooled at k=16, gFVD, gFID, and class accuracy exclude zero at every size. VideoFlexTok is significantly ahead only at $k \leq 4$ on uCO3D, mostly at k=1. With each arm at its own best k, every tested metric excludes zero at every size except uCO3D gFVD, which does so at two of seven sizes.

Across AR sizes. Against larger VideoFlexTok AR models, the small SemanTok AR model's semantic-alignment lead excludes zero in every comparison, for example $+0.053$ class accuracy for 201M vs. 679M on uCO3D. With six seeds pooled, its gFID lead does too, for example $+1.19$ $[0.81, 1.55]$ for 201M vs. 679M on uCO3D, where the pooled estimate also uses every generated clip. On Kinetics-600, the 85M SemanTok AR model's gFVD advantage over the 1.33B VideoFlexTok AR model also excludes zero ($+19.1$ $[7.5, 28.9]$). Only uCO3D gFVD remains at parity, for example $+3.1$ $[-3.1, 9.0]$ for 201M vs. 679M.

E Generation and decoder REPA per AR size and budget

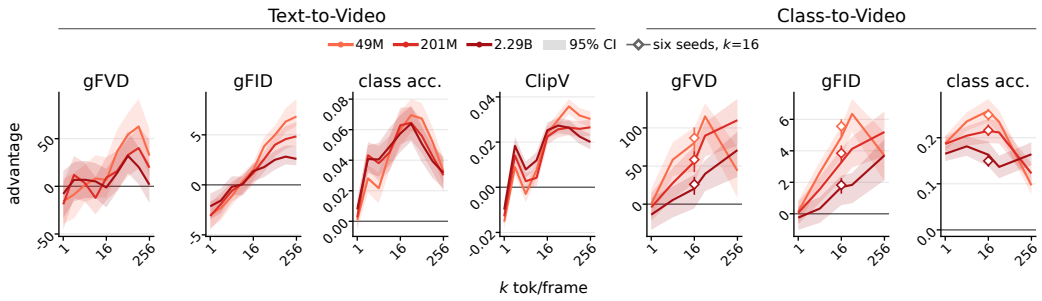


Figure 21: SemanTok's advantage over VideoFlexTok at equal AR size, as a function of the token budget. Each panel shows the paired difference between the two tokenizers under the same AR size, k, prompts, and labels, signed so that positive values favour SemanTok (VideoFlexTok minus SemanTok for gFVD and gFID; SemanTok minus VideoFlexTok for class accuracy and ClipV). Lines are point estimates for three AR sizes; bands are 95% bootstrap intervals over evaluation clips; the grey line marks no difference. SemanTok's semantic-alignment lead appears from $k{=}2$, its fidelity lead from $k{\approx}16$ on uCO3D and from $k{=}4$ (small AR models) to $k{=}32$ (largest) on Kinetics-600, and VideoFlexTok is ahead only at $k{\leq}4$ on uCO3D. Diamonds: Kinetics-600 at $k{=}16$ with six sampling seeds pooled.

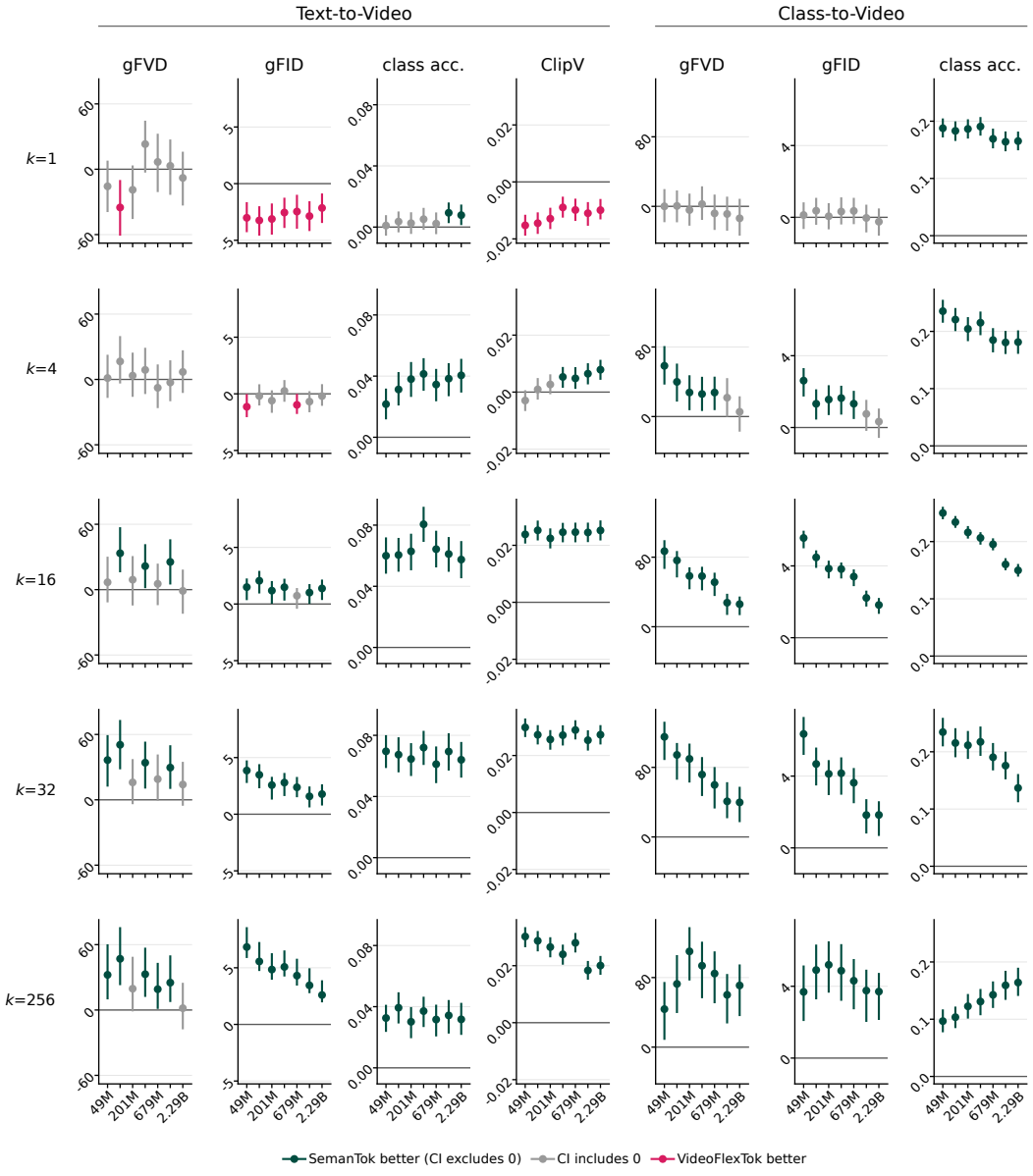


Figure 22: SemanTok's advantage for every AR size at five token budgets. Rows fix k, columns fix the metric, and within each panel the AR size grows from 49M (left) to 2.29B (right). Dots are the paired difference at equal AR size and k, signed as in fig. 21 so that positive values favour SemanTok; whiskers are 95% bootstrap intervals over evaluation clips. Teal marks an interval above zero, grey an interval that includes zero, and magenta an interval below zero. Each column shares its y-range, so the change with k reads down the page. The Kinetics-600 k=16 row pools six sampling seeds; all other cells use one.

Table 6: Kinetics-600 class-to-video generation: fidelity. AR models sampled with k tokens per frame, tokenizers at 200k steps, AR models at 20k steps (1.33B and 2.29B at 40k), n=2048. Bold is better at that k and size; † marks the better tokenizer where the 95% paired-bootstrap CI of the difference (single sampling seed) excludes zero. CIs exist at $k \in \{1, 4, 16, 32, 256\}$ only. With six sampling seeds pooled, the k=16 gFVD and gFID gains of SemanTok exclude zero at every size.

AR		k=1	4	8	16	32	64	128	256
gFVD↓									
49M	VideoFlexTok	390.8	295.7	317.5	303.9	338.8	478.6	588.9	653.7
	SemanTok	**390.4**	**236.8**†	**239.2**	**223.7**†	**223.7**†	**324.0**	**450.2**	**610.1**†
85M	VideoFlexTok	386.3	284.1	293.3	278.9	307.0	450.6	548.9	627.3
	SemanTok	**385.4**	**244.0**†	**228.8**	**219.9**†	**212.5**†	**322.8**	**411.0**	**554.5**†
201M	VideoFlexTok	**384.4**	278.9	279.9	272.9	302.9	420.7	501.0	560.3
	SemanTok	388.3	**251.0**†	**240.8**	**217.2**†	**213.0**†	**295.2**	**361.2**	**449.8**†
393M	VideoFlexTok	394.9	274.1	273.0	263.8	284.3	400.7	475.1	521.6
	SemanTok	**392.1**	**247.9**†	**233.4**	**212.7**†	**212.6**†	**286.7**	**348.0**	**428.0**†
679M	VideoFlexTok	**380.5**	271.8	264.1	249.1	267.0	390.6	451.1	475.6
	SemanTok	388.1	**243.9**†	**222.4**	**216.5**†	**207.1**†	**279.9**	**332.5**	**390.7**†
1.33B	VideoFlexTok	**388.4**	266.0	257.9	222.8	239.5	337.6	382.2	391.6
	SemanTok	397.0	**243.9**	**229.0**	**211.3**	**198.3**†	**256.9**	**277.6**	**331.7**†
2.29B	VideoFlexTok	**389.2**	255.6	245.2	225.9	241.5	330.9	367.7	383.2
	SemanTok	402.6	**249.8**	**237.5**	**205.7**	**201.7**†	**258.2**	**278.9**	**312.2**†
gFID↓									
49M	VideoFlexTok	32.90	26.87	25.51	26.16	28.04	33.97	40.87	46.54
	SemanTok	**32.76**	**24.27**†	**22.93**	**21.31**†	**21.71**†	**26.26**	**32.94**	**42.83**†
85M	VideoFlexTok	32.89	26.37	24.89	25.17	26.54	32.51	38.79	44.80
	SemanTok	**32.52**	**25.05**†	**22.77**	**21.58**†	**21.88**†	**25.99**	**31.51**	**39.85**†
201M	VideoFlexTok	32.73	26.16	24.86	24.51	25.49	31.44	36.80	41.14
	SemanTok	**32.67**	**24.61**†	**23.56**	**21.26**†	**21.38**†	**24.92**	**29.36**	**35.92**†
393M	VideoFlexTok	33.11	25.98	24.55	24.92	25.60	30.88	35.51	39.14
	SemanTok	**32.77**	**24.34**†	**23.09**	**21.08**†	**21.45**†	**24.85**	**28.43**	**34.24**†
679M	VideoFlexTok	32.91	25.71	23.47	23.16	24.73	30.54	34.09	37.37
	SemanTok	**32.55**	**24.39**†	**23.16**	**21.23**†	**21.13**†	**24.38**	**27.49**	**33.05**†
1.33B	VideoFlexTok	**32.70**	24.59	**22.92**	21.94	22.61	26.50	30.92	32.33
	SemanTok	32.74	**23.85**	22.97	**21.15**	**20.80**†	**23.96**	**24.85**	**28.55**†
2.29B	VideoFlexTok	**32.45**	24.09	**22.25**	22.37	22.81	26.61	28.99	30.94
	SemanTok	32.68	**23.77**	23.45	**20.69**†	**21.01**†	**23.66**	**24.67**	**27.21**†

Table 7: Kinetics-600 class-to-video generation: semantic alignment. Same samples as Table 6. Class acc. is UMT-L top-1. Bold is better at that k and size; † marks the better tokenizer where the 95% paired-bootstrap CI of the difference (single sampling seed) excludes zero. CIs exist for class acc. at $k \in \{1, 4, 16, 32, 256\}$ only; ViCLIP and ClipV have none.

AR		k=1	4	8	16	32	64	128	256
ViCLIP↑									
49M	VideoFlexTok	0.179	0.192	0.187	0.189	0.187	0.161	0.160	0.156
	SemanTok	**0.193**	**0.208**	**0.206**	**0.208**	**0.206**	**0.179**	**0.175**	**0.168**
85M	VideoFlexTok	0.178	0.193	0.188	0.191	0.190	0.164	0.161	0.159
	SemanTok	**0.194**	**0.208**	**0.208**	**0.208**	**0.206**	**0.181**	**0.177**	**0.171**
201M	VideoFlexTok	0.178	0.194	0.191	0.194	0.192	0.165	0.163	0.161
	SemanTok	**0.193**	**0.207**	**0.207**	**0.208**	**0.208**	**0.183**	**0.181**	**0.176**
393M	VideoFlexTok	0.178	0.194	0.190	0.194	0.194	0.166	0.165	0.163
	SemanTok	**0.193**	**0.208**	**0.208**	**0.208**	**0.209**	**0.184**	**0.182**	**0.177**
679M	VideoFlexTok	0.178	0.194	0.193	0.196	0.197	0.168	0.168	0.165
	SemanTok	**0.192**	**0.206**	**0.207**	**0.209**	**0.210**	**0.185**	**0.184**	**0.180**
1.33B	VideoFlexTok	0.177	0.193	0.193	0.199	0.200	0.172	0.171	0.170
	SemanTok	**0.191**	**0.205**	**0.207**	**0.209**	**0.209**	**0.186**	**0.186**	**0.184**
2.29B	VideoFlexTok	0.176	0.192	0.193	0.200	0.201	0.173	0.172	0.172
	SemanTok	**0.190**	**0.204**	**0.205**	**0.209**	**0.208**	**0.187**	**0.187**	**0.186**
ClipV↑									
49M	VideoFlexTok	0.531	0.550	0.543	0.546	0.539	0.491	0.487	0.475
	SemanTok	**0.549**	**0.577**	**0.574**	**0.574**	**0.570**	**0.522**	**0.513**	**0.495**
85M	VideoFlexTok	0.531	0.552	0.545	0.548	0.545	0.493	0.489	0.480
	SemanTok	**0.549**	**0.575**	**0.574**	**0.574**	**0.571**	**0.525**	**0.517**	**0.501**
201M	VideoFlexTok	0.529	0.552	0.549	0.551	0.548	0.497	0.492	0.487
	SemanTok	**0.546**	**0.575**	**0.573**	**0.572**	**0.571**	**0.528**	**0.521**	**0.510**
393M	VideoFlexTok	0.527	0.551	0.545	0.549	0.550	0.500	0.494	0.491
	SemanTok	**0.544**	**0.572**	**0.571**	**0.573**	**0.574**	**0.532**	**0.524**	**0.511**
679M	VideoFlexTok	0.527	0.552	0.550	0.555	0.556	0.504	0.502	0.492
	SemanTok	**0.544**	**0.569**	**0.570**	**0.572**	**0.575**	**0.534**	**0.526**	**0.516**
1.33B	VideoFlexTok	0.524	0.548	0.550	0.556	0.561	0.509	0.504	0.503
	SemanTok	**0.542**	**0.566**	**0.571**	**0.571**	**0.572**	**0.533**	**0.532**	**0.523**
2.29B	VideoFlexTok	0.523	0.547	0.548	0.557	0.561	0.511	0.509	0.507
	SemanTok	**0.539**	**0.564**	**0.567**	**0.571**	**0.570**	**0.535**	**0.534**	**0.528**
class acc.↑									
49M	VideoFlexTok	0.234	0.392	0.303	0.340	0.335	0.109	0.115	0.082
	SemanTok	**0.424**†	**0.631**†	**0.589**	**0.596**†	**0.568**†	**0.280**	**0.234**	**0.180**†
85M	VideoFlexTok	0.242	0.408	0.321	0.377	0.381	0.126	0.117	0.101
	SemanTok	**0.426**†	**0.628**†	**0.613**	**0.618**†	**0.593**†	**0.300**	**0.267**	**0.208**†
201M	VideoFlexTok	0.242	0.436	0.381	0.422	0.420	0.143	0.146	0.126
	SemanTok	**0.426**†	**0.640**†	**0.627**	**0.639**†	**0.631**†	**0.318**	**0.303**	**0.250**†
393M	VideoFlexTok	0.244	0.446	0.390	0.444	0.441	0.158	0.155	0.137
	SemanTok	**0.433**†	**0.659**†	**0.629**	**0.646**†	**0.658**†	**0.333**	**0.313**	**0.267**†
679M	VideoFlexTok	0.250	0.461	0.421	0.471	0.474	0.165	0.164	0.154
	SemanTok	**0.419**†	**0.649**†	**0.632**	**0.673**†	**0.665**†	**0.363**	**0.341**	**0.292**†
1.33B	VideoFlexTok	0.242	0.466	0.448	0.547	0.535	0.204	0.206	0.202
	SemanTok	**0.407**†	**0.650**†	**0.667**	**0.701**†	**0.710**†	**0.408**	**0.384**	**0.357**†
2.29B	VideoFlexTok	0.237	0.467	0.453	0.545	0.560	0.235	0.209	0.217
	SemanTok	**0.400**†	**0.650**†	**0.647**	**0.701**†	**0.700**†	**0.401**	**0.412**	**0.384**†

Table 8: uCO3D text-to-video generation: fidelity. Tokenizers at 100k steps, AR models at 20k steps; 1,014 ID and 152 OOD clips pooled, gFVD and gFID as one Fréchet distance. Bold is better at that k and size; † marks the better tokenizer where the 95% paired-bootstrap CI of the difference (single sampling seed) excludes zero. CIs cover every cell.

AR		k=1	4	8	16	32	64	128	256
gFVD↓									
49M	VideoFlexTok	**436.3**	259.9	235.2	232.9	254.7	284.5	344.1	379.1
	SemanTok	452.0	**258.7**	**226.9**	**226.2**	**218.2**†	**230.4**†	**281.2**†	**346.9**†
85M	VideoFlexTok	**426.8**†	255.9	**229.0**	237.1	260.1	275.5	338.4	367.1
	SemanTok	461.9	**239.4**	233.9	**203.7**†	**209.6**†	**218.9**†	**261.1**†	**320.1**†
201M	VideoFlexTok	**424.4**	244.2	**211.8**	218.6	224.5	244.0	286.8	315.5
	SemanTok	443.2	**240.7**	223.9	**209.5**	**208.6**	**211.3**†	**246.6**†	**296.0**
393M	VideoFlexTok	448.1	246.4	**218.5**	224.3	233.8	253.5	288.6	306.0
	SemanTok	**425.0**	**237.7**	219.8	**202.7**†	**199.7**†	**206.1**†	**228.6**†	**273.2**†
679M	VideoFlexTok	435.8	**232.2**	**208.2**	213.3	221.4	245.4	263.8	288.8
	SemanTok	**429.1**	239.9	210.2	**208.0**	**202.4**	**199.1**†	**223.6**†	**269.7**†
1.33B	VideoFlexTok	432.2	**235.2**	221.3	223.1	221.7	225.4	255.1	273.4
	SemanTok	**429.1**	238.0	**209.3**	**197.7**†	**192.1**†	**201.6**†	**218.6**†	**248.3**†
2.29B	VideoFlexTok	**431.1**	241.7	218.9	**212.5**	211.0	236.4	250.4	268.0
	SemanTok	439.0	**234.9**	**213.7**	213.7	**196.9**	**204.3**†	**230.2**†	**266.3**
gFID↓									
49M	VideoFlexTok	**39.56**†	**24.98**†	24.00	23.87	26.12	29.04	35.36	41.13
	SemanTok	42.58	26.12	**23.66**	**22.37**†	**22.27**†	**24.05**†	**29.04**†	**34.30**†
85M	VideoFlexTok	**40.16**†	**25.35**	23.43	23.60	25.02	27.70	33.30	38.02
	SemanTok	43.41	25.54	**23.32**	**21.55**†	**21.55**†	**22.97**†	**27.52**†	**32.46**†
201M	VideoFlexTok	**39.49**†	**24.35**	22.96	22.52	23.96	26.43	30.77	34.63
	SemanTok	42.61	24.95	**22.83**	**21.33**†	**21.39**†	**22.43**†	**26.19**†	**29.78**†
393M	VideoFlexTok	**39.91**†	24.72	22.51	22.38	23.47	25.95	29.72	33.13
	SemanTok	42.46	**24.46**	**22.45**	**20.90**†	**20.68**†	**21.67**†	**24.72**†	**28.05**†
679M	VideoFlexTok	**40.38**†	**23.59**†	22.60	22.16	23.03	24.97	28.28	31.51
	SemanTok	42.85	24.56	**22.18**	**21.43**	**20.66**†	**21.74**†	**24.55**†	**27.21**†
1.33B	VideoFlexTok	**40.10**†	**24.13**	**22.36**	22.06	22.39	24.01	27.72	29.99
	SemanTok	42.97	24.82	22.76	**21.05**†	**20.81**†	**21.46**†	**23.93**†	**26.54**†
2.29B	VideoFlexTok	**40.57**†	**24.35**	22.30	22.07	22.46	24.15	26.71	28.91
	SemanTok	42.72	24.54	**22.25**	**20.69**†	**20.69**†	**21.65**†	**23.89**†	**26.31**†

Table 9: uCO3D text-to-video generation: semantic alignment. Same samples as Table 8. Class acc. is NCM. Bold is better at that k and size; † marks the better tokenizer where the 95% paired-bootstrap CI of the difference (single sampling seed) excludes zero. CIs cover every ClipV and class-acc. cell; ViCLIP has none.

AR		k=1	4	8	16	32	64	128	256
ViCLIP↑									
49M	VideoFlexTok	**0.170**	**0.210**	**0.217**	0.221	0.221	0.220	0.217	0.212
	SemanTok	0.169	0.205	0.214	**0.226**	**0.230**	**0.231**	**0.226**	**0.220**
85M	VideoFlexTok	0.170	**0.208**	**0.218**	0.221	0.222	0.222	0.219	0.216
	SemanTok	0.170	0.206	0.215	**0.227**	**0.230**	**0.232**	**0.228**	**0.222**
201M	VideoFlexTok	0.169	**0.208**	**0.217**	0.222	0.223	0.224	0.221	0.220
	SemanTok	**0.170**	0.205	0.215	**0.226**	**0.229**	**0.231**	**0.228**	**0.225**
393M	VideoFlexTok	0.168	**0.207**	0.216	0.221	0.222	0.223	0.223	0.220
	SemanTok	**0.170**	0.206	0.216	**0.227**	**0.230**	**0.232**	**0.230**	**0.226**
679M	VideoFlexTok	0.167	**0.209**	**0.217**	0.222	0.224	0.224	0.225	0.222
	SemanTok	**0.168**	0.206	0.214	**0.227**	**0.230**	**0.232**	**0.229**	**0.227**
1.33B	VideoFlexTok	0.166	**0.208**	0.217	0.221	0.225	0.226	0.225	0.225
	SemanTok	0.166	0.207	0.217	**0.226**	**0.230**	**0.232**	**0.230**	**0.227**
2.29B	VideoFlexTok	0.166	**0.207**	**0.216**	0.221	0.224	0.226	0.226	0.225
	SemanTok	**0.167**	0.205	0.215	**0.227**	**0.230**	**0.232**	**0.229**	**0.228**
ClipV↑									
49M	VideoFlexTok	**0.598**†	**0.688**	0.703	0.708	0.706	0.700	0.687	0.671
	SemanTok	0.583	0.685	**0.711**†	**0.731**†	**0.736**†	**0.736**†	**0.719**†	**0.701**†
85M	VideoFlexTok	**0.599**†	0.688	0.707	0.710	0.712	0.707	0.694	0.681
	SemanTok	0.584	**0.689**	**0.713**†	**0.735**†	**0.739**†	**0.739**†	**0.725**†	**0.710**†
201M	VideoFlexTok	**0.597**†	0.690	0.709	0.715	0.717	0.716	0.705	0.694
	SemanTok	0.584	**0.692**	**0.713**†	**0.738**†	**0.743**†	**0.742**†	**0.731**†	**0.721**†
393M	VideoFlexTok	**0.594**†	0.687	0.708	0.716	0.716	0.717	0.709	0.702
	SemanTok	0.585	**0.693**†	**0.720**†	**0.740**†	**0.744**†	**0.745**†	**0.737**†	**0.726**†
679M	VideoFlexTok	**0.592**†	0.689	0.709	0.717	0.719	0.719	0.716	0.704
	SemanTok	0.582	**0.694**†	**0.720**†	**0.742**†	**0.748**†	**0.749**†	**0.739**†	**0.732**†
1.33B	VideoFlexTok	**0.590**†	0.690	0.708	0.716	0.722	0.722	0.719	0.713
	SemanTok	0.579	**0.697**†	**0.722**†	**0.740**†	**0.747**†	**0.749**†	**0.742**†	**0.731**†
2.29B	VideoFlexTok	**0.590**†	0.687	0.709	0.716	0.721	0.724	0.720	0.715
	SemanTok	0.580	**0.695**†	**0.721**†	**0.741**†	**0.749**†	**0.750**†	**0.742**†	**0.735**†
class acc.↑									
49M	VideoFlexTok	0.101	0.219	0.246	0.249	0.247	0.231	0.203	0.187
	SemanTok	**0.102**	**0.240**†	**0.290**†	**0.309**†	**0.316**†	**0.298**†	**0.255**†	**0.220**†
85M	VideoFlexTok	0.103	0.223	0.245	0.255	0.250	0.239	0.221	0.202
	SemanTok	**0.106**	**0.254**†	**0.290**†	**0.315**†	**0.317**†	**0.306**†	**0.267**†	**0.241**†
201M	VideoFlexTok	0.101	0.221	0.251	0.259	0.265	0.259	0.237	0.225
	SemanTok	**0.103**	**0.259**†	**0.296**†	**0.322**†	**0.330**†	**0.315**†	**0.279**†	**0.255**†
393M	VideoFlexTok	0.099	0.220	0.260	0.263	0.264	0.260	0.249	0.236
	SemanTok	**0.105**	**0.261**†	**0.306**†	**0.343**†	**0.336**†	**0.324**†	**0.304**†	**0.273**†
679M	VideoFlexTok	0.097	0.223	0.258	0.269	0.279	0.270	0.261	0.244
	SemanTok	**0.099**	**0.258**†	**0.299**†	**0.334**†	**0.340**†	**0.332**†	**0.302**†	**0.276**†
1.33B	VideoFlexTok	0.092	0.222	0.254	0.269	0.275	0.273	0.265	0.255
	SemanTok	**0.102**†	**0.260**†	**0.305**†	**0.330**†	**0.344**†	**0.333**†	**0.314**†	**0.289**†
2.29B	VideoFlexTok	0.092	0.220	0.263	0.280	0.288	0.286	0.282	0.270
	SemanTok	**0.100**†	**0.261**†	**0.312**†	**0.337**†	**0.352**†	**0.337**†	**0.321**†	**0.302**†

Table 10: Decoder-REPA readout versus k and decoder noise σ. Cosine between DINOv2 features of the first frame of each latent group and the decoder-REPA readout from the first k tokens per frame, 256 validation clips. σ=1 is pure noise, as in generation. Bold is better at that k and σ.

σ		k=1	4	8	16	32	64	128	256
Kinetics-600									
1	VideoFlexTok	0.480	0.529	0.555	0.581	0.599	0.614	0.624	0.631
	SemanTok	**0.490**	**0.561**	**0.589**	**0.618**	**0.642**	**0.663**	**0.685**	**0.704**
0.75	VideoFlexTok	0.517	0.552	0.571	0.593	0.608	0.621	0.630	0.636
	SemanTok	**0.525**	**0.580**	**0.605**	**0.630**	**0.650**	**0.669**	**0.689**	**0.707**
0.5	VideoFlexTok	**0.612**	0.624	0.630	0.639	0.645	0.650	0.654	0.656
	SemanTok	0.604	**0.632**	**0.648**	**0.663**	**0.675**	**0.686**	**0.700**	**0.714**
0.25	VideoFlexTok	**0.674**	**0.678**	**0.679**	0.682	0.683	0.684	0.685	0.685
	SemanTok	0.648	0.666	0.677	**0.686**	**0.694**	**0.702**	**0.711**	**0.721**
uCO3D									
1	VideoFlexTok	0.548	0.607	0.629	0.641	0.651	0.659	0.671	0.683
	SemanTok	**0.556**	**0.651**	**0.680**	**0.701**	**0.718**	**0.730**	**0.742**	**0.749**
0.75	VideoFlexTok	0.589	0.626	0.641	0.652	0.659	0.666	0.677	0.688
	SemanTok	**0.597**	**0.666**	**0.690**	**0.708**	**0.722**	**0.733**	**0.744**	**0.751**
0.5	VideoFlexTok	**0.672**	0.678	0.682	0.685	0.687	0.690	0.695	0.701
	SemanTok	0.668	**0.696**	**0.709**	**0.720**	**0.729**	**0.737**	**0.746**	**0.752**
0.25	VideoFlexTok	**0.716**	**0.715**	0.716	0.716	0.716	0.717	0.718	0.719
	SemanTok	0.696	0.713	**0.722**	**0.730**	**0.736**	**0.741**	**0.748**	**0.754**