NIKA: Efficient Neural Video Representation via Structured Latent Diversity

Slater Victoroff
iph.so
slater@iph.so

Madison May
indico
madison@pragmatic.ml

Abstract

Implicit neural video representations offer compact, continuous alternatives to conventional codecs, but higher reconstruction fidelity typically requires more computation per decoded frame. We introduce NIKA, which shifts video-specific capacity from the decoder into a structured latent state, allowing reconstruction quality to scale with latent expressivity while keeping active decoding lightweight. NIKA constructs this state from complementary spatial, spectral, and temporal bases, then decodes it with lightweight ConvNeXt-style convolutional upsampling. On UVG, a 2.91M-parameter NIKA model achieves 33.33 dB PSNR at 462 decoding FPS with 4.5G MACs on an RTX A5000, outperforming comparable single-resolution NeRV-family baselines while using 39–51$\times$ fewer MACs. Ablations show that diversifying latent components improves reconstruction more reliably than reallocating capacity within a single component, and qualitative analysis reveals specialization among components. Together, these results identify structured latent diversity as a practical alternative to scaling decoder complexity for high-fidelity, efficient neural video representation.

1 Introduction

Efficient representations of high-dimensional signals remain a central challenge in machine learning, particularly for video, where spatial detail, temporal coherence, and high data rates impose competing demands on storage efficiency, decoding speed, and reconstruction quality. Conventional video codecs achieve strong rate–distortion performance through decades of hand-engineered design [23, 25], but their specialized structure can limit adaptability beyond narrowly defined signal classes. Neural representations offer a flexible alternative, yet identifying architectures that achieve favorable quality–efficiency trade-offs remains an open problem.

Implicit neural representations (INRs) model signals as continuous functions parameterized by neural networks [7, 16, 22]. For video, early frame-wise INR methods such as NeRV and E-NeRV map a temporal coordinate directly to a reconstructed frame using a learned decoder [5, 11]. This formulation enables fast full-frame reconstruction, but existing improvements in fidelity typically come from increasing decoder capacity, adding hierarchical decoding stages, or introducing more expensive spatial processing [6, 9, 10, 27]. As a result, reconstruction quality and per-frame decoding cost remain tightly coupled.

This work is motivated by a different scaling strategy: shifting video-specific capacity from the decoder into a structured latent state while keeping the active decoder path lightweight. This strategy reflects the heterogeneity of video structure: global layout, localized detail, oscillatory texture, and temporal variation are not equally compact in a single representational form. A purely convolutional, spectral, grid-based, or low-rank representation may capture some aspects of the signal efficiently while representing others inefficiently [2, 4, 12, 14, 17]. This suggests an alternative to scaling

Preprint.

a single decoder: allocate capacity across complementary latent components, each with different inductive biases, and use a lightweight decoder primarily as a shared readout.

We introduce NIKA (Neural Implicit Component Assembly), a neural video representation that constructs frames from a structured latent state composed of multiple complementary components, including low-rank spatial factors, spectral features, grid-based components, and lightweight temporal operators. The latent state serves as the primary video-specific representation, while the decoder remains nearly fixed across model scales. As a result, reconstruction quality can improve through increased latent expressivity with minimal impact on active per-frame decoding cost.

Empirically, NIKA achieves strong quality–speed trade-offs on standard neural video benchmarks. At comparable parameter counts, it outperforms single-resolution NeRV-family baselines while using substantially fewer MACs and maintaining much higher decoding speed. Parameter-matched ablations show that the full mixed representation substantially outperforms variants restricted to real/spatial, complex/spectral, Tucker-factorized, or grid-based components, indicating that the gains are not explained by parameter count alone. Qualitative analysis further shows that these components contribute distinct image structures. Together, these results support structured latent diversity as a practical alternative to scaling decoder complexity.

The primary contributions of this work are summarized below:

- **Structured multi-component video representation.** We introduce NIKA, which constructs a latent representation from complementary low-rank, spectral, grid-based, and temporal structures.
- **Latent-centric scaling.** NIKA shifts video-specific capacity into a structured latent state, allowing reconstruction quality to scale while keeping the active decoder path lightweight.
- **Empirical analysis of representational composition.** Through parameter-matched ablations and visualizations, we show that mixed latent components outperform restricted variants and exhibit qualitative specialization.
- **Open-source implementation.** We release training, evaluation, and visualization code to support reproducibility.

2 Related Work

Neural video representations differ along several architectural axes, including query structure, spatial hierarchy, and allocation of per-video capacity. Pixel-wise methods model a coordinate function $f_\theta(x, y, t)$, recovering each frame by densely evaluating spatial coordinates. Frame-wise methods instead learn a map $F_\theta : [0, 1] \to \mathbb{R}^{H\times W\times 3}$ that decodes an entire frame from a temporal coordinate t. Within frame-wise methods, models further differ in whether they use single-resolution or multi-resolution features, explicit motion or flow aggregation, learned embeddings, or hierarchical decoding stages. We focus on fixed-resolution, frame-wise representations as the closest comparison class for studying how per-video capacity allocation affects the quality–efficiency trade-off.

2.1 Pixel-wise Representations

Pixel-wise approaches instantiate the coordinate-function view $f_\theta(x, y, t)$, representing a video as a continuous mapping from spatiotemporal coordinates to pixel values. Coordinate networks provide a flexible representation for continuous signals [22], while flow-based variants such as CoordFlow adapt this idea to video by learning coordinate transformations that account for motion [21]. This formulation can produce high-fidelity reconstructions, but it requires dense spatial evaluation to recover each frame. Consequently, decoding cost scales with the number of output pixels, creating a different quality–efficiency trade-off from frame-wise methods that generate an entire frame in one forward pass.

2.2 Frame-wise Representations

Frame-wise methods learn $F_\theta(t)$, decoding an entire frame from a temporal coordinate. NeRV introduced this formulation with a convolutional decoder [5], and E-NeRV improved the spatial–temporal factorization of the representation [11]. Subsequent methods improve reconstruction quality

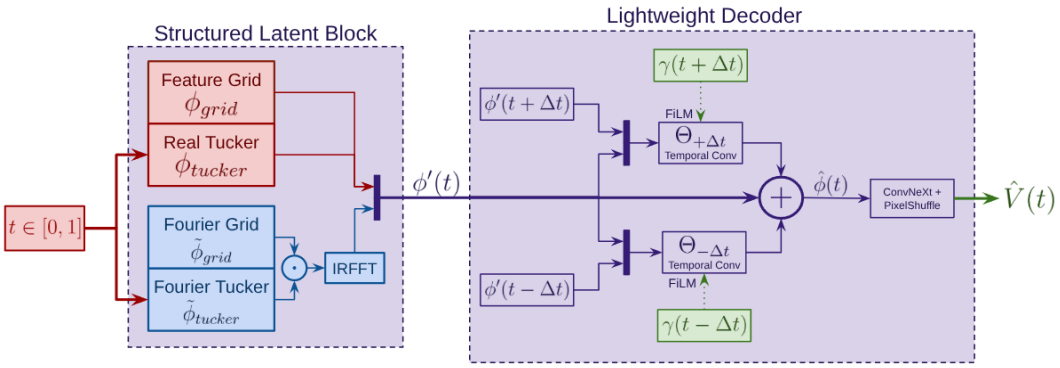

Figure 1: NIKA architecture. Given a normalized time coordinate $t \in [0, 1]$, the Structured Latent Block constructs $\phi'(t)$ from a learned feature grid, a real-valued Tucker factorization, and Fourier-domain components. A lightweight decoder refines $\phi'(t)$ using FiLM-conditioned temporal convolution operators over neighboring latent states, producing $\hat{\phi}(t)$. A ConvNeXt-style PixelShuffle upsampler maps $\hat{\phi}(t)$ to the reconstructed frame $\hat{V}(t)$.

through richer active reconstruction paths: HNeRV and HiNeRV introduce hybrid or hierarchical encodings [6, 9], FFNeRV incorporates flow-guided aggregation [10], and multi-scale variants add additional spatial hierarchy [27]. These approaches substantially improve reconstruction quality over early frame-wise models, but their gains often come from adding decoder capacity, hierarchy, aggregation, or multi-resolution processing that remains active at inference time. Consequently, even frame-wise methods can require tens to hundreds of GMACs per frame at high resolution, and the strongest variants may trade decoding speed for fidelity. This reflects the central trade-off studied in this work: frame-wise methods avoid dense coordinate evaluation, but quality improvements remain closely tied to the cost of the active reconstruction path.

2.3 Structured and Factorized Representations

A complementary line of work studies how representational structure affects approximation efficiency. Fourier-domain representations encode signals through frequency structure, making them natural for smooth, periodic, or globally correlated components [2, 12]. Grid-based encodings store spatially organized features that can be queried efficiently and processed with local operations, making them effective for high-frequency detail and spatially localized variation [17]. Low-rank and tensor-factorized representations compactly model correlations across dimensions by decomposing high-dimensional signals into structured factors [4]. More broadly, classical approximation theory and empirical operator-learning benchmarks both suggest that different representational biases can be effective for different signal classes and dynamics [3, 14, 18].

These approaches impose different inductive biases, but are often adopted as a single dominant representational form or combined serially through conditioning, modulation, or decoder processing [19]. In contrast, NIKA composes multiple structured components inside the per-video latent state, allowing low-rank, Fourier-domain, and grid-based components to contribute complementary capacity before a shared lightweight readout.

3 Method

NIKA represents a video as a continuous function of normalized time $t \in [0, 1]$. Rather than storing video-specific information primarily in a large decoder, NIKA moves most representational capacity into a structured latent state and uses a lightweight shared decoder to reconstruct frames. The architecture has two main stages, shown in Figure 1: a Structured Latent Block that produces a latent representation $\phi'(t)$, and a lightweight decoder that applies temporal latent refinement followed by ConvNeXt-style [13] upsampling to produce the reconstructed frame $\hat{V}(t)$.

3.1 Structured Latent Block

The Structured Latent Block constructs a latent feature field $\phi'(t)$ at latent spatial resolution $H_\phi \times W_\phi$. It combines complementary real-valued and Fourier-domain components, each designed to capture different structure in the video signal. The real branch captures low-rank spatiotemporal structure and static spatial detail, while the Fourier branch captures oscillatory and convolutional structure through learned spectral modulation.

Real-valued branch. The real-valued branch consists of a time-dependent Tucker component and a learned feature grid. We parameterize a latent tensor

$$X \in \mathbb{R}^{T_\phi \times C_\phi \times H_\phi \times W_\phi}$$

using a Tucker factorization [8]:

$$X \approx \mathcal{G} \times_1 U_T \times_2 U_C \times_3 U_H \times_4 U_W,$$

where $\mathcal{G} \in \mathbb{R}^{R_T \times R_C \times R_H \times R_W}$ is a learned core tensor and U_T, U_C, U_H, U_W are learned factor matrices. For a continuous time t, we interpolate the temporal factor U_T and contract the factorization to obtain

$$\phi_{\text{tucker}}(t) \in \mathbb{R}^{C_\phi \times H_\phi \times W_\phi}.$$

To represent spatially localized or persistent content, we also learn a feature grid

$$G \in \mathbb{R}^{C_G \times H_\phi \times W_\phi},$$

which is projected to C_ϕ channels when needed:

$$\phi_{\text{grid}}(h, w) = W_G G(:, h, w) + b_G.$$

In our implementation, this grid is time-independent and therefore provides an efficient representation for static or slowly varying structure.

The real-valued branch output is thus given by:

$$\phi_{\text{real}}(t) = \text{concat}\big(\phi_{\text{grid}}, \phi_{\text{tucker}}(t)\big).$$

Fourier-domain branch. The Fourier branch mirrors this structure in the spectral domain. We parameterize a complex-valued half-plane tensor

$$\tilde{X} \in \mathbb{C}^{T_\phi \times C_\phi \times H_\phi \times W_\phi^+},$$

where $W_\phi^+ = \lfloor W_\phi/2 \rfloor + 1$ is the non-redundant Fourier width used by the real inverse FFT. As in the real branch, $\tilde{X}$ is represented by a complex Tucker factorization. Interpolating its temporal factor at time t and contracting the factorization gives

$$\tilde{\phi}_{\text{tucker}}(t) \in \mathbb{C}^{C_\phi \times H_\phi \times W_\phi^+}.$$

We also learn a complex Fourier grid

$$\tilde{G} \in \mathbb{C}^{C_{\tilde{G}} \times H_\phi \times W_\phi^+},$$

which is projected to C_ϕ channels to produce $\tilde{\phi}_{\text{grid}}$. The Fourier branch combines the time-dependent spectral Tucker component with the learned spectral grid using elementwise multiplication:

$$\tilde{\phi}_{\text{spec}}(t) = \tilde{\phi}_{\text{tucker}}(t) \odot \tilde{\phi}_{\text{grid}}.$$

The corresponding real-valued spatial feature map is recovered with an inverse real FFT:

$$\phi_{\text{spec}}(t) = \text{IRFFT}\big(\tilde{\phi}_{\text{spec}}(t)\big).$$

Elementwise multiplication in the Fourier domain corresponds to convolution in the spatial domain, so the spectral branch acts as a structured mechanism for spatial mixing. This gives the model a complementary basis for representing periodic, oscillatory, and globally coupled structure.

Latent composition. The final output of the Structured Latent Block is obtained by concatenating the real and spectral components followed by normalization:

$$\phi'(t) = \mathrm{GN}\Big(\mathrm{concat}\big(\phi_{\mathrm{real}}(t), \phi_{\mathrm{spec}}(t)\big)\Big),$$

where GN denotes Group Normalization [26]. Since $\phi_{\mathrm{real}}(t)$ contributes $2C_\phi$ channels and $\phi_{\mathrm{spec}}(t)$ contributes C_ϕ channels, the resulting latent has shape

$$\phi'(t) \in \mathbb{R}^{3C_\phi \times H_\phi \times W_\phi}.$$

3.2 Lightweight Decoder

The lightweight decoder maps the structured latent state to a reconstructed frame. It consists of two stages: temporal latent refinement, which applies FiLM-conditioned convolutional operators to neighboring latent states, and a ConvNeXt-style PixelShuffle [20] upsampler, which maps the refined latent to RGB space. In our reported models, the active reconstruction path uses small hidden widths, with the ConvNeXt upsampler and temporal operators typically using hidden dimension 48. This keeps per-frame computation low relative to decoder-heavy neural video representations, while most video-specific capacity resides in the structured latent state and segment-local copies.

Temporal latent refinement. For a temporal offset Δt, we form an operator input by combining the current latent with a neighboring latent:

$$z_{+\Delta t}(t) = \mathrm{concat}\big(\phi'(t), \phi'(t+\Delta t)\big),$$

and analogously

$$z_{-\Delta t}(t) = \mathrm{concat}\big(\phi'(t), \phi'(t-\Delta t)\big).$$

Each offset has a lightweight temporal convolution operator $\Theta_{\pm\Delta t}$, which produces a residual correction in latent space:

$$\delta_{+\Delta t}(t) = \Theta_{+\Delta t}\big(z_{+\Delta t}(t); \gamma(t+\Delta t)\big),$$
$$\delta_{-\Delta t}(t) = \Theta_{-\Delta t}\big(z_{-\Delta t}(t); \gamma(t-\Delta t)\big).$$

Here $\gamma(\cdot)$ is a learned Fourier time encoding used to generate FiLM [19] modulation parameters for the corresponding temporal convolution.

We compute the time encoding as

$$\gamma(\tau) = \Big[\tau,\ \{\cos(\lambda_k \tau)\}_{k=1}^{K},\ \{\sin(\lambda_k \tau)\}_{k=1}^{K}\Big],$$

where the frequencies $\{\lambda_k\}$ are learned. The resulting encoding modulates the convolutional operator through FiLM parameters, allowing each temporal operator to adapt to the queried time.

The temporally refined latent is computed additively:

$$\hat{\phi}(t) = \phi'(t) + \sum_{\Delta t \in \mathcal{S}} \delta_{\Delta t}(t).$$

The offset set $\mathcal{S}$ is configurable. In the reported experiments, we use one forward and one backward offset, $\mathcal{S} = \{+\Delta t, -\Delta t\}$, which provides local temporal context while keeping the active decoder path small.

At video boundaries, we avoid explicit masking by using virtual temporal padding. The internal latent timeline includes additional learned positions beyond the first and last frame, allowing boundary frames to query neighboring latent states without special-case masked operators. These virtual positions are optimized jointly with the rest of the segment-local latent parameters and are only used to provide neighbor context for boundary queries.

ConvNeXt upsampling. The refined latent $\hat{\phi}(t)$ is decoded into the reconstructed frame:

$$\hat{V}(t) = D(\hat{\phi}(t)).$$

The decoder D is a compact ConvNeXt-style upsampling network. It projects the latent features into a hidden channel space, applies depthwise spatial convolution, channel-wise normalization, and pointwise MLP mixing in a residual ConvNeXt block, then projects to subpixel output channels. PixelShuffle converts these subpixel channels into the final RGB frame.

3.3 Segmented Scaling

NIKA scales model capacity by partitioning a video into temporal segments and assigning each segment an independent copy of the same base architecture. A global frame time t is routed to a segment index j and converted to a local normalized time t_j. The reconstructed frame is then produced by the corresponding segment-local model:

$$\hat{V}(t) = \hat{V}_j(t_j).$$

This segmented construction increases stored per-video capacity without increasing the active computation required to decode any individual frame. Only one segment-local model is evaluated for each query, so the active per-frame path remains fixed as the number of segments grows. In our scaling experiments, larger parameter budgets are obtained primarily by adding segment copies rather than by increasing Tucker ranks, hidden widths, or decoder depth. Thus, total stored capacity can grow while the per-frame reconstruction path remains approximately constant.

Segmentation is deliberately simple. Segment-local functions are not constrained to agree as continuous functions at arbitrary boundary times; however, evaluated video frames are routed to exactly one segment-local model. As a result, segmentation does not introduce ambiguity in the reconstructed frame sequence used for training or evaluation. More sophisticated boundary-aware or parameter-sharing segment constructions are possible, but we leave them to future work.

4 Experiments

Our experiments test the central premise of NIKA: that video-specific capacity can be shifted from expensive active decoders into structured latent representations while maintaining or improving reconstruction quality. We evaluate this claim along three axes: quality–efficiency trade-offs against single-resolution frame-wise neural video baselines, scaling behavior as stored capacity increases while active per-frame computation remains fixed, and ablations that distinguish latent component diversity from simply adding more parameters to one representational form.

4.1 Datasets and Evaluation Protocol

We evaluate on two commonly used benchmarks for neural video representations: the UVG dataset (made available by the Ultra Video Group under a CC-BY-NC 4.0 license) [15] and the Big Buck Bunny (Bunny) dataset (made available by the Blender Foundation under the CC-BY-3.0 license) [1]. UVG contains seven 1920×1080 videos, while Bunny contains 132 frames at 1280×720 resolution. Following prior work, we train a separate model for each video sequence.

We compare against representative single-resolution, frame-wise NeRV-family video representations, including NeRV, E-NeRV, HNeRV, and FFNeRV. These methods are the closest comparison class because they decode full frames from temporal inputs rather than evaluating a coordinate network independently at each pixel, and because they avoid multi-resolution or patch-wise reconstruction schemes. We report reconstruction quality using PSNR, model size in parameters, computational cost in MACs, and encoding/decoding throughput in frames per second (FPS).

Following the evaluation setup used by HiNeRV, we evaluate XXS/XS/S scales on Bunny and S/M/L scales on UVG. At each scale, we match model parameter counts as closely as possible. For baseline methods, we use the parameter counts, MACs, and PSNR values reported by HiNeRV. Because throughput is highly hardware- and implementation-dependent, we remeasure encoding and decoding FPS for available public implementations on our hardware using batch size 1. This keeps quality and architecture-level comparisons tied to the published configurations while making throughput comparisons hardware-consistent. For NIKA, parameter counts, MACs, PSNR, and throughput are measured directly from our implementation under the same timing protocol.

4.2 Implementation Details

All NIKA models are implemented in PyTorch. We optimize reconstruction quality directly using a PSNR-based objective, which we find stable in our setting. Optimization is performed using SOAP [24] with a plateau-based learning-rate schedule.

Model	Size	MACs	Encoding FPS	Decoding FPS	PSNR
NeRV	0.83M/1.64M/3.20M	25G/57G/101G	58.9/52.7/35.8	591/276/114	26.82/29.61/32.56
E-NeRV	0.88M/1.65M/3.31M	26G/101G/104G	42.6/36.8/25.6	110/94.6/75.3	29.03/31.75/36.69
HNeRV	0.82M/1.66M/3.28M	23G/48G/94G	38.3/33.8/24.3	279/182/101	31.08/33.68/36.95
FFNeRV	0.91M/1.66M/3.19M	26G/58G/102G	38.9/38.2/37.4	173/172/171	30.37/33.83/37.01
NIKA	0.81M/1.62M/3.24M	2.0G/2.0G/2.0G	**229/229/229**	**1010/1010/1010**	**35.92/37.97/39.61**

Table 1: Video representation results on Bunny at XXS/XS/S scales. NIKA increases stored capacity across scales while keeping active per-frame MACs fixed.

NIKA models are trained for 2000 epochs. Although this is more epochs than the 300 epochs commonly used in prior NeRV-family work, NIKA has substantially higher encoding throughput, so the resulting wall-clock training cost remains comparable. We report encoding FPS alongside reconstruction quality to make this trade-off explicit.

NIKA scales primarily through segmentation rather than through larger active decoders. For Bunny, the XXS/XS/S models use 1, 2, and 4 copies of the same base architecture, respectively. For UVG, the S/M/L models use 2, 4, and 8 copies. Since each frame is routed to a single segment-local model, this increases stored per-video capacity while keeping active per-frame computation fixed.

Encoding and decoding throughput are measured at batch size 1 over full-sequence inference and exclude host-device data transfer overhead. Public implementations and published configurations vary in precision, preprocessing, cropping, and quantization settings; we therefore release timing scripts and Docker containers used for baseline benchmarking to support reproducibility.

4.3 Main Results

Tables 1 and 2 report results on Bunny and UVG. Across both datasets, NIKA improves the quality–efficiency trade-off relative to the evaluated single-resolution, frame-wise NeRV-family baselines: it matches or exceeds their reconstruction quality while using substantially fewer MACs and achieving higher measured decoding throughput.

On Bunny, NIKA improves PSNR at every evaluated scale while keeping active computation fixed at approximately 2.0G MACs per decoded frame. At the largest scale, NIKA reaches 39.61 dB PSNR with 3.24M parameters, compared to 37.01 dB for FFNeRV and 36.95 dB for HNeRV at similar parameter counts. NIKA also decodes at 1010 FPS, exceeding the fastest measured baseline throughput at the same scale.

On UVG, NIKA shows the same behavior at higher resolution. At the 3M-parameter scale, NIKA achieves 33.33 dB average PSNR with 4.5G MACs, compared to 32.68 dB for HNeRV with 175G MACs and 32.63 dB for FFNeRV with 228G MACs. This corresponds to a 39–51$\times$ reduction in MACs relative to these comparable baselines while improving average reconstruction quality. As stored capacity increases, NIKA improves from 33.33 dB to 34.51 dB and 35.72 dB while keeping active per-frame MACs fixed.

These results support the central scaling hypothesis of NIKA: high-quality reconstruction does not require proportionally scaling the active decoder.

4.4 Ablation Study

We next evaluate whether NIKA's performance arises from combining complementary latent components rather than simply increasing parameter count. Table 3 reports approximately parameter-matched ablations on Bunny at the 3M-parameter scale. Each ablation removes or restricts one aspect of the full model while keeping the overall parameter count close to the reference setting.

The full NIKA model achieves the best reconstruction quality. Replacing segmented scaling with a single model copy reduces PSNR from 39.61 dB to 38.54 dB, indicating that increasing stored capacity through segmentation contributes substantially while keeping active per-frame computation fixed. Removing temporal operators reduces PSNR to 38.49 dB, showing that latent-space temporal refinement provides a similarly sized contribution.

Model	Size	MACs	FPS	Beauty	Bosph.	Honey	Jockey	Ready	Shake	Yacht	Avg.
NeRV	3.31M	227G	15.1/49.3	32.83	32.20	38.15	30.30	23.62	33.24	26.43	30.97
E-NeRV	3.29M	230G	16.4/52.2	33.13	33.38	38.87	30.61	24.53	34.26	26.87	31.75
HNeRV	3.26M	175G	11.9/41.5	33.56	35.03	**39.28**	31.58	25.45	34.89	28.98	32.68
FFNeRV	3.40M	228G	18.1/83.1	33.57	35.03	38.95	31.57	25.92	34.41	28.99	32.63
NIKA	2.91M	4.5G	**88.7/462**	**33.58**	**35.46**	39.19	**32.91**	**27.14**	**35.59**	**29.42**	**33.33**
NeRV	6.53M	228G	8.5/29.6	33.67	34.83	39.00	33.34	26.03	34.39	28.23	32.78
E-NeRV	6.54M	245G	11.1/38.7	33.97	35.83	**39.75**	33.56	26.94	35.57	28.79	33.49
HNeRV	6.40M	349G	5.5/17.5	**33.99**	36.45	39.56	33.56	27.38	35.93	30.48	33.91
FFNeRV	6.44M	229G	16.7/75.8	33.98	36.63	39.58	33.58	27.39	35.91	30.51	33.94
NIKA	5.82M	4.5G	**88.7/462**	33.97	**36.87**	39.31	**34.98**	**29.24**	**36.36**	**30.83**	**34.51**
NeRV	13.01M	230G	3.8/12.9	34.15	36.96	39.55	35.80	28.68	35.90	30.39	34.49
E-NeRV	13.02M	285G	8.23/24.7	34.25	37.61	**39.74**	35.45	29.17	36.97	30.76	34.85
HNeRV	12.87M	701G	2.8/9.8	**34.30**	37.96	39.73	35.47	29.67	**37.16**	32.31	35.23
FFNeRV	12.66M	232G	12.3/53.7	34.28	**38.48**	39.74	**36.72**	30.75	37.08	32.36	35.63
NIKA	11.4M	4.5G	**88.7/462**	34.25	38.33	39.41	36.70	**31.66**	37.13	**32.59**	**35.72**

Table 2: Video representation results on UVG at S/M/L scales. FPS denotes encoding/decoding speed. NIKA improves average PSNR across scales while keeping active per-frame MACs fixed.

Variant	Params	PSNR	Δ PSNR
Full model, single segment	2.96M	38.54	-1.07
No temporal operators	3.07M	38.49	-1.12
Real/spatial only	3.17M	37.51	-2.10
Complex/spectral only	2.80M	37.36	-2.25
Grid components only	2.99M	34.76	-4.85
Tucker components only	2.98M	33.61	-6.00
Full NIKA	3.24M	**39.61**	–

Table 3: Ablation study on Bunny using approximately parameter-matched 3M-parameter variants. Restricted variants underperform the full model, indicating that reconstruction quality benefits from combining complementary representational domains, parameterization types, and temporal refinement.

Restricting the latent state to a single representational family causes larger degradation. Real/spatial-only and complex/spectral-only variants lose more than 2 dB relative to the full model, while grid-only and Tucker-only variants lose 4.85 dB and 6.00 dB, respectively. These results indicate that reconstruction quality depends on combining complementary representational domains and parameterization types rather than allocating parameters to a single latent structure.

Together, these ablations support the central design hypothesis of NIKA: structured latent diversity and lightweight temporal refinement both improve reconstruction, while the strongest performance comes from combining multiple complementary latent components within a scalable stored representation.

4.5 Qualitative Component Analysis

Figure 2 visualizes isolated latent components across UVG sequences. Each component family exhibits recurring visual structure across diverse videos, suggesting a stable representational role rather than an arbitrary decomposition. Together with Table 3, these visualizations support the view that NIKA benefits from latent components with complementary inductive biases.

5 Limitations

NIKA is evaluated in the standard per-video neural representation setting, where a separate model is optimized for each video sequence. While this enables high-fidelity reconstruction and controlled study of representation structure, it does not address generalization across videos or streaming adaptation. Extending structured latent representations to shared or amortized settings remains an important direction for future work.

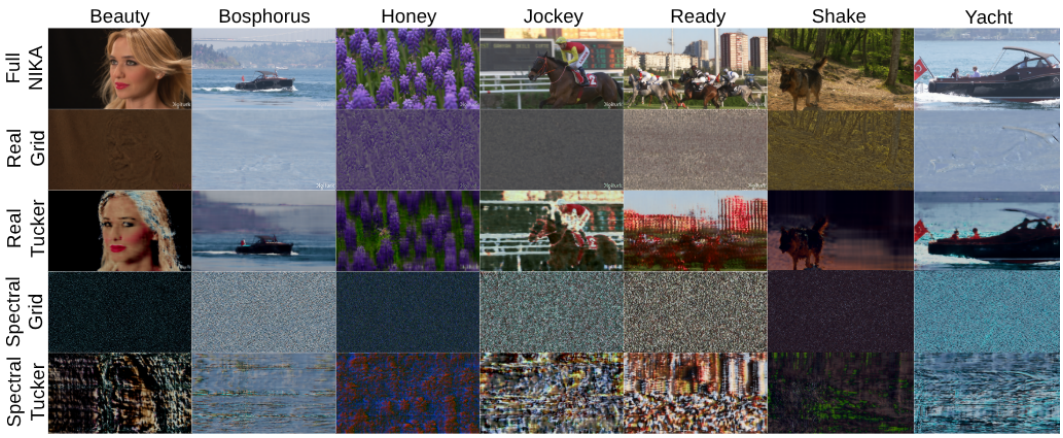

Figure 2: Qualitative latent-component visualizations on UVG. Columns show selected UVG frames; rows show the full NIKA reconstruction followed by isolated real grid, real Tucker, spectral grid, and spectral Tucker components. Components exhibit recurring structural patterns across videos. Frames are selected for visual interpretability.

Our comparisons focus specifically on single-resolution, frame-wise neural video representations. We do not compare directly against pixel-wise INR methods, multi-resolution spatial hierarchies, or hybrid codec architectures, which explore different trade-offs between reconstruction quality, flexibility, and computational cost. Likewise, segmented scaling in NIKA is intentionally simple and does not enforce continuity or parameter sharing across segment boundaries.

Although NIKA achieves low active per-frame MAC counts, practical throughput at these operating points is increasingly influenced by framework and kernel-launch overhead rather than arithmetic cost alone. The current implementation does not use quantization or specialized inference kernels, leaving additional efficiency optimization as future work.

Finally, our experiments are limited to the UVG and Bunny benchmarks. While these datasets are widely used in prior neural video representation work, they cover a relatively narrow range of scene diversity and motion complexity. Evaluating structured latent representations on larger and more varied video distributions remains an important area for future study.

6 Conclusion

We introduced NIKA, a neural video representation that shifts video-specific capacity from the active decoder into a structured latent state composed of complementary components. By combining low-rank, spectral, grid-based, and temporal structures with a lightweight reconstruction path, NIKA improves the quality–efficiency trade-off for single-resolution frame-wise neural video representation.

Across standard benchmarks, NIKA matches or exceeds comparable NeRV-family baselines while using substantially fewer MACs and maintaining high measured decoding throughput. Scaling experiments show that reconstruction quality can improve as stored capacity increases, even when active per-frame computation remains fixed. Ablations further indicate that these gains depend on combining complementary latent components rather than concentrating parameters into a single representational form.

These results suggest that efficient neural video representation need not rely primarily on increasingly expensive decoders. Instead, structured latent diversity provides a practical scaling axis for high-fidelity reconstruction. More broadly, NIKA points toward compositional latent representations as a promising direction for efficient modeling of high-dimensional signals beyond video.

References

[1] Blender Foundation. Big buck bunny. Open Movie, 2008. URL `https://peach.blender.org/`.

[2] Ronald N. Bracewell. *The Fourier Transform and Its Applications*. McGraw–Hill, 1986.

[3] Emmanuel J. Candès and David L. Donoho. New tight frames of curvelets and optimal representations of objects with C^2 singularities. *Communications on Pure and Applied Mathematics*, 57(2):219–266, 2004.

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350, 2022.

[5] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. In *Advances in Neural Information Processing Systems*, 2021.

[6] Hao Chen, Matthew Gwilliam, Ser-Nam Lim, and Abhinav Shrivastava. Hnerv: A hybrid neural representation for videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10270–10279, 2023.

[7] Amer Essakine, Yanqi Cheng, Chun-Wun Cheng, Lipei Zhang, Zhongying Deng, Lei Zhu, Carola-Bibiane Schönlieb, and Angelica I. Aviles-Rivero. Where do we stand with implicit neural representations? a technical and performance survey. *Transactions on Machine Learning Research*, 2025.

[8] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009. doi: 10.1137/07070111X.

[9] Ho Man Kwan, Ge Gao, Fan Zhang, Art Gower, and David Bull. Hinerv: Video compression with hierarchical encoding-based neural representation. In *Advances in Neural Information Processing Systems*, 2023.

[10] Joo Chan Lee, Daniel Rho, Jong Hwan Ko, and Eunbyung Park. Ffnerv: Flow-guided frame-wise neural representations for videos. In *ACM International Conference on Multimedia*, pages 7859–7870, 2023.

[11] Zizhang Li, Mengmeng Wang, Huaijin Pi, Kechun Xu, Jianbiao Mei, and Yong Liu. E-nerv: Expedite neural video representation with disentangled spatial-temporal context. In *European Conference on Computer Vision*, pages 267–284, 2022.

[12] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

[13] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[14] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[15] Alexandre Mercat, Marko Viitanen, and Jarno Vanne. Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. In *ACM Multimedia Systems Conference*, 2020.

[16] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421, 2020.

[17] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):102:1–102:15, 2022.

[18] Ruben Ohana et al. The well: A large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

[19] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[20] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016.

[21] Daniel Silver and Ron Kimmel. Coordflow: Coordinate flow for pixel-wise neural video representation, 2025.

[22] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, 2020.

[23] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.

[24] Nikhil Vyas, Sham Kakade, Jitendra Malik, Mehrdad Mahdavi, Boaz Barak, and Naman Agarwal. Soap: Improving and stabilizing shampoo using adam. *arXiv preprint arXiv:2409.11321*, 2024.

[25] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

[26] Yuxin Wu and Kaiming He. Group normalization. In *European Conference on Computer Vision*, pages 3–19, 2018.

[27] Jun Zhu, Xinfeng Zhang, Lv Tang, and JunHao Jiang. Msnerv: Neural video representation with multi-scale feature fusion, 2025.

A Model Configurations

Table 4 reports the primary NIKA configurations used for scaling experiments. Table 5 reports the restricted variants used in the ablation study. Table 6 reports the shared optimization settings. Tucker ranks are listed as $[R_C, R_H, R_W, R_T]$. Unless otherwise noted, all models use one temporal operator step and decoder/operator hidden dimension 48. For Bunny experiments, grid channels are doubled relative to the listed base configuration to account for the lower spatial resolution and target parameter scale.

Config	Grid ch.	Real Tucker ranks	Spectral Tucker ranks	Segments	Decoder/op. hidden
XXS	2	$[2, 40, 40, 40]$	$[2, 30, 30, 30]$	1	48/48
XS	2	$[3, 50, 50, 50]$	$[3, 40, 40, 40]$	1	48/48
Small	4	$[3, 75, 75, 60]$	$[3, 60, 60, 40]$	1	48/48
Medium	8	$[3, 90, 90, 70]$	$[3, 75, 75, 60]$	1	48/48
Large	16	$[4, 100, 100, 100]$	$[4, 80, 80, 80]$	1	48/48
XXS1	2	$[2, 40, 40, 40]$	$[2, 30, 30, 30]$	1	48/48
XXS2	2	$[2, 40, 40, 40]$	$[2, 30, 30, 30]$	2	48/48
XXS4	2	$[2, 40, 40, 40]$	$[2, 30, 30, 30]$	4	48/48
XS2	2	$[3, 50, 50, 50]$	$[3, 40, 40, 40]$	2	48/48
XS4	2	$[3, 50, 50, 50]$	$[3, 40, 40, 40]$	4	48/48
XS8	2	$[3, 50, 50, 50]$	$[3, 40, 40, 40]$	8	48/48

Table 4: Primary NIKA model configurations. Segmented variants use repeated copies of the same base architecture, with only one segment-local model evaluated per decoded frame.

Config	Grid ch.	Real Tucker ranks	Spectral Tucker ranks	Spectral grid ch.	Segments/op. steps
XXS4-noop	2	$[2, 40, 40, 40]$	$[2, 30, 30, 30]$	2	4 / 0
XXS4-real	4	$[2, 50, 50, 50]$	–	–	4 / 1
XXS4-complex	–	–	$[3, 40, 40, 40]$	4	4 / 1
XXS4-tucker	–	$[2, 50, 50, 50]$	$[3, 40, 40, 40]$	–	4 / 1
XXS4-grid	3	–	–	–	4 / 1

Table 5: Ablation configurations used for the parameter-matched Bunny study. "–" indicates that the component is disabled. Decoder and operator hidden dimensions are 48 for all variants.

Setting	Value
Optimizer	SOAP
Objective	PSNR objective, computed from frame MSE with $\epsilon = 10^{-8}$
Initial learning rate	10^{-2}
Learning-rate schedule	ReduceLROnPlateau on epoch PSNR
Schedule factor / patience	0.5 / 40 epochs
Schedule threshold / cooldown	0.015 / 20 epochs
Minimum learning rate	2×10^{-3}
Training epochs	2000
Batching	Sequential frame batches within each temporal segment
Checkpoint selection	Best epoch PSNR, with checkpoints saved at least 10 epochs apart
OOM handling	Batch size is halved and the epoch is retried

Table 6: Shared training settings used for NIKA experiments.

B Benchmarking and Reproducibility

Throughput benchmarking for neural video representations is sensitive to implementation details, hardware, precision, batch size, input resolution, and preprocessing conventions. Public codebases also vary in whether they provide reference configurations at the exact resolutions and model scales

used in published comparisons. In some cases, reproducing published settings requires adapting implementations to non-default resolutions or reconciling differences in data loading and batching.

For baseline methods, we report PSNR, parameter counts, and MACs from published configurations, then remeasure encoding and decoding throughput for available public implementations on a shared hardware/software setup using batch size 1. For NIKA, all reported values are measured directly from our implementation under the same timing protocol. We release benchmarking scripts and Docker containers for these runs. FPS should therefore be interpreted as an implementation- and hardware-specific measurement, while MACs provide a complementary estimate of active per-frame computation.

C Optimization Variability

Per-video neural representation differs from supervised prediction settings because each model is optimized to reconstruct a fixed video sequence rather than to generalize across held-out examples. The most relevant stochastic variation is therefore optimization variability from initialization and training dynamics. To estimate this effect, we train five NIKA models on Bunny using the same XXS4 configuration for 300 epochs with different random seeds. These runs are shorter than the 2000-epoch models used in the main benchmark tables and are intended to measure short-run optimization stability rather than replace the final reported results.

Seed	Best PSNR	Final PSNR	Best epoch
41	37.16	37.14	296
42	37.22	37.22	300
43	36.91	36.89	296
44	37.45	37.36	294
45	37.25	37.25	300
Mean $\pm$ std.	37.20 ± 0.19	37.17 ± 0.18	–

Table 7: Optimization variability on Bunny over five 300-epoch runs using the same NIKA XXS4 configuration and different random seeds. These runs measure sensitivity to initialization and short-run training dynamics, and are shorter than the 2000-epoch runs used for the main benchmark results.

D Additional Qualitative Examples

Figure 3 shows component visualizations for frames sampled at temporal segment boundaries in the Beauty sequence. The full NIKA reconstructions remain visually coherent across segments, while isolated component views exhibit stronger segment-local variation. At the same time, each component family retains a recognizable structural character, suggesting that segment-local models can adopt different local coding conventions while preserving stable high-level representational roles.

E Broader impacts

NIKA targets efficient reconstruction of existing videos rather than generation, recognition, or decision-making. Its potential benefits include lower storage, bandwidth, and decoding costs for video archives and media delivery. At the same time, more efficient video representation could indirectly reduce the cost of storing or transmitting large video collections, including in privacy-sensitive or surveillance settings. Deployment on private or sensitive video should follow the consent, privacy, and retention constraints appropriate to that data.

Figure 3: Segment-boundary component visualizations for the Beauty sequence. Columns show frame indices sampled at segment boundaries. The top row shows the corresponding full NIKA reconstructions, while the lower rows show isolated Real Tucker and Spectral Tucker components. Although isolated component appearance varies across segments, each component family retains a recognizable structural character.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's core claims (structured latent representation, improved quality/efficiency trade-offs, and benefits from representational diversity) which align with the method and experimental scope described in Sections 1, 3, 4, and 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations section (Section 5) that discusses key constraints, including per-video training, limited comparison scope, and restricted dataset diversity.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper is primarily methodological and empirical, and it does not present formal theorems that would require associated assumptions and proofs.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The architecture is described in detail in Section 3. In addition, an open-source implementation is available as part of the supplemental material and will be made available via GitHub after the review process.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may

be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

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

   Answer: [Yes]

   Justification: An open-source implementation is available as supplemental material and includes scripts to reproduce the main result tables.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

Justification: The paper specifies the datasets used, training length, optimizer, learning rate schedule, module hyperparameters, evaluation metrics, and parameter counts..

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: To provide the reader with an estimate of inter-run variance and sensitivity to initialization, we report results of 5 training runs on the Bunny dataset in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: Experiments were run primarily on a local workstation with two 24GB NVIDIA RTX A5000 GPUs, with throughput measurements reported on a single RTX A5000. A small number of supplementary benchmarking/debugging runs were performed on rented A100 instances, totaling approximately $100 of cloud compute. We report training epochs, encoding/decoding FPS, and model sizes to characterize the practical compute cost.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: To the best of our knowledge, the research conforms to the NeurIPS Code of Ethics. The work uses standard benchmark video datasets and offline experiments without human subjects or sensitive personal data.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix E discusses the main potential impacts: lower storage and transmission costs, and the corresponding privacy risk if efficient representation is applied to sensitive video.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

Justification: As our research is limited to video compression, it has low risk for misuse and is not in the same risk category as research on video generation where synthesis of misleading content is a real concern.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The UVG dataset used for experiments in this paper is made available by the Ultra Video Group under a non-commercial Creative Commons BY-NC 4.0 license which is documented in Section 4.1. The Big Buck Bunny dataset is made available by the Blender Foundation under the Creative Commons Attribution 3.0 license.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The accompanying code includes detailed documentation, including installation guidelines, dataset access, and steps for reproducing the results in the paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: Our work does not involve crowdsourcing or experiments with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Our work does not involve human subjects or crowdsourced data, so IRB approval or equivalent review is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: While LLM-based tools were used for coding assistance, LLM use did not impact the core methodology and was not central to this work.

    Guidelines:

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