GIViC: Generative Implicit Video Compression

Ge Gao Siyue Teng Tianhao Peng Fan Zhang David Bull
Visual Information Lab, University of Bristol
{ge1.gao, siyue.teng, tianhao.peng, fan.zhang, dave.bull}@bristol.ac.uk
https://ge1-gao.github.io/GIViC

Abstract

*While video compression based on implicit neural representations (INRs) has recently demonstrated great potential, existing INR-based video codecs still cannot achieve state-of-the-art (SOTA) performance compared to their conventional or autoencoder-based counterparts given the same coding configuration. In this context, we propose a **G**enerative **I**mplicit **Vi**deo **C**ompression framework, **GIViC**, aiming at advancing the performance limits of this type of coding methods. GIViC draws inspiration from the remarkable ability of large language and diffusion models to capture long-range dependencies, a characteristic also inherent to Implicit Neural Representations (INRs). Through the newly designed **implicit diffusion** process, GIViC performs diffusive sampling across coarse-to-fine spatiotemporal decompositions, gradually progressing from coarser-grained full-sequence diffusion to finer-grained per-token diffusion. A novel **Hierarchical Gated Linear Attention-based transformer** (HGLA), is also integrated into the framework, which dual-factorizes global dependency modeling along scale and sequential axes. The proposed GIViC model has been benchmarked against SOTA conventional and neural codecs using a Random Access (RA) configuration (YUV 4:2:0, GOPSize=32), and yields BD-rate savings of 15.94%, 22.46% and 8.52% over VVC VTM, DCVC-FM and NVRC, respectively, on the UVG test set. As far as we are aware, GIViC is the first INR-based video codec that outperforms VTM, in terms of coding performance, based on the RA coding configuration.*

1. Introduction

The ubiquitous consumer demand for high-quality digital video has accelerated the development of increasingly powerful compression techniques [12]. While the latest video standards, such as MPEG H.266/VVC [10] and AOM (Alliance for Open Media) AV1 [29], offer impressive coding efficiency and architectural compatibility with previous

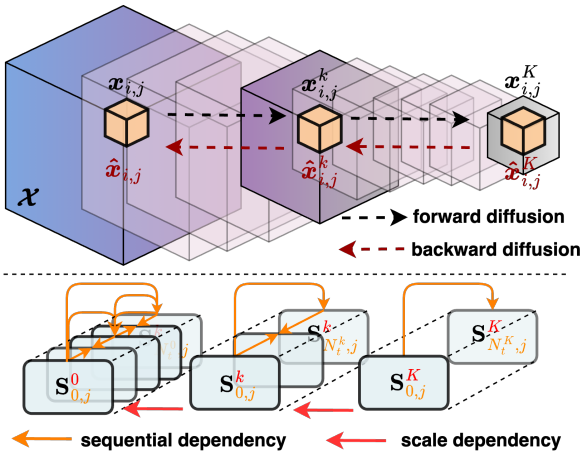

Figure 1. (**Top**) Illustration of the implicit diffusion framework based on spatiotemporal downsampling of a GOP $\mathcal{X}$ with additive noise, interlinking independent diffusion within constant-sized tokens $\{\boldsymbol{x}^k_{i,j}\}$ across $k = 1, \ldots, K$ levels of abstractions. (**Bottom**) The global spatiotemporal dependencies are captured by the 2D hidden states $\mathbf{S}^k_{i,j}$ of the HGLA transformer, recurrently updated along both scale and sequence axes.

standards, their coding gains are achieved through the use of increasingly sophisticated tools built upon the conventional hybrid video coding framework. In contrast, neural video compression [48, 50, 53] has emerged in recent years as a data-driven framework, leveraging end-to-end optimization to achieve a performance level that rivals, or in some cases, surpasses [38, 50, 80] that of standard video codecs.

More recently, implicit neural representation (INR) based solutions [13] have provided a more flexible, and potentially lightweight, alternative to these 'generic' neural video coding backbones. By adaptively *overfitting* a neural network to a specific (input) video sequence, INR-based video codecs [14, 45, 47] exploit long-term spatiotemporal dependencies through sequence-level parameter sharing and stochastic optimization, showing the potential to achieve competitive coding performance [24, 46].

However, the application scenarios, and more critically,

the compression performance of existing INR-based codecs are generally limited by their encoding latency, i.e., the number of consecutive frames that can be represented with a single set of learnable parameters. When the system latency is constrained to be compatible with the Low Delay or Random Access configurations [9] typically used in standard video codecs, INR-based methods [24] are outperformed by SOTA conventional codecs such as VVC VTM [11] and generic neural codecs such as the recently improved DCVC models [50, 59].

In this paper, we enhance the INR framework by architecturally scaling its capacity for long-term dependency modeling, employing diffusion models (DMs) and transformer backbones capable of modeling *full-GOP-level spatiotemporal dynamics*. The resulting video compression framework, GIViC (**G**enerative **I**mplicit **Vi**deo **C**ompression), is built on a novel conditional implicit diffusion model, as shown in Figure 1 (top). This decomposes a joint diffusion process into cascaded spatiotemporal pyramids, where each stage is extrapolated from denoised representations at coarser scales and previously denoised reference tokens, accelerating denoising while preserving representation quality. GIViC also integrates HGLA (Hierarchical Gated Linear Attention), a novel linear transformer that harmonizes efficiency and effectiveness by dual-factorizing long-term dependency modeling along both scale and sequence axes. Leveraging hierarchically gated recurrence, HGLA scales linearly with long context length spanning the entire GOP, as illustrated in Figure 1 (bottom). The main contributions of this paper are summarized as follows:

- This is the **first time** diffusion models and transformers have been jointly integrated into an INR-based framework for video compression, resulting in a highly expressive architecture for full-GOP-level distribution modeling that enables SOTA compression performance.
- We propose a novel **implicit diffusion framework** that reformulates the standard diffusion method into an equivalent multi-resolution approach. It decomposes the diffusion process into spatiotemporal pyramidal stages, interlinking coarser-grained global variations with finer-grained local details during the per-token diffusion denoising process.
- We further develop a gated linear transformer backbone, **HGLA**, tailored specifically for our diffusion formulation, that captures long-term dependencies jointly along scale and sequence axes. HGLA achieves linear complexity w.r.t (long) context lengths while maintaining competitive performance compared to vanilla transformers that have quadratic complexity.

We have benchmarked GIViC against SOTA conventional and neural video codecs on the UVG, MCL-JCV, and JVET-B datasets under the Random Access (RA) configuration (YUV colorspace). Results demonstrate significant coding gains, with GIViC outperforming VTM 20.0, DCVC-FM, and NVRC by 15.94%, 22.46%, and 8.52%, respectively, on the UVG test set, and by 7.71%, 22.34%, and 16.13%, respectively, on the JVET-B test set. To the best of our knowledge, GIViC is the **first INR-based video codec to surpass VTM performance** in the RA coding mode.

2. Related Work

Neural video compression. The focus of video compression research is progressively shifting from conventional hand-crafted codecs [10, 69, 79] to those that incorporate learning-based enhancement of individual coding tools [1, 88, 88] often within end-to-end optimized coding frameworks. Recent contributions have been based on various innovations including: improving sub-components [34, 36, 38, 49, 54, 80], optimizing rate control [82, 92], leveraging instance-specific overfitting [56, 74] and accelerating inference [35, 58]. Currently, the best-performing model [59] has been reported [72] to offer improved performance over ECM [68] under the Low-Delay configuration.

Implicit neural representations (INRs), which use neural networks to map multimedia signals into coordinate-based representations [22, 39, 65, 81], offer an efficient and elegant (albeit unconventional) alternative for video compression. INR-based methods [3, 13, 14, 31, 47, 84] exploit sequence-level spatiotemporal redundancy by encoding video sequences within a compact set of network parameters, reformulating visual data compression into a model compression task that leverages pruning, quantization, and entropy penalization techniques [25, 30, 91]. While recent advances have improved the compression efficiency of INRs through hierarchical encoding [45] and more advanced compression methods [24, 46], they are still outperformed by SOTA conventional [11, 68] and neural video codecs [50, 59] under the same latency constraints.

Long sequence modeling. Recently, the success of Large Language Models (LLMs) [19, 37] has been driven by a core principle in information theory - *jointly modeling long token sequences can maximize compression efficiency* [15]. However, unlike natural languages, visual signals are associated with bidirectional dependencies that defy simple unidirectional structures, resulting in poorer performance with decoder-only architectures compared to diffusion and non-autoregressive methods [70, 77]. Additionally, the quadratic complexity of self-attention in LLMs poses challenges for scaling to long contexts, motivating the development of linear attention [27, 57, 86], which enables parallelized training, linear complexity inference, and performance comparable to standard transformers. This design has been recently adopted in some neural compression methods [38, 60].

Diffusion models. Diffusion models (DMs) [16, 32, 66]

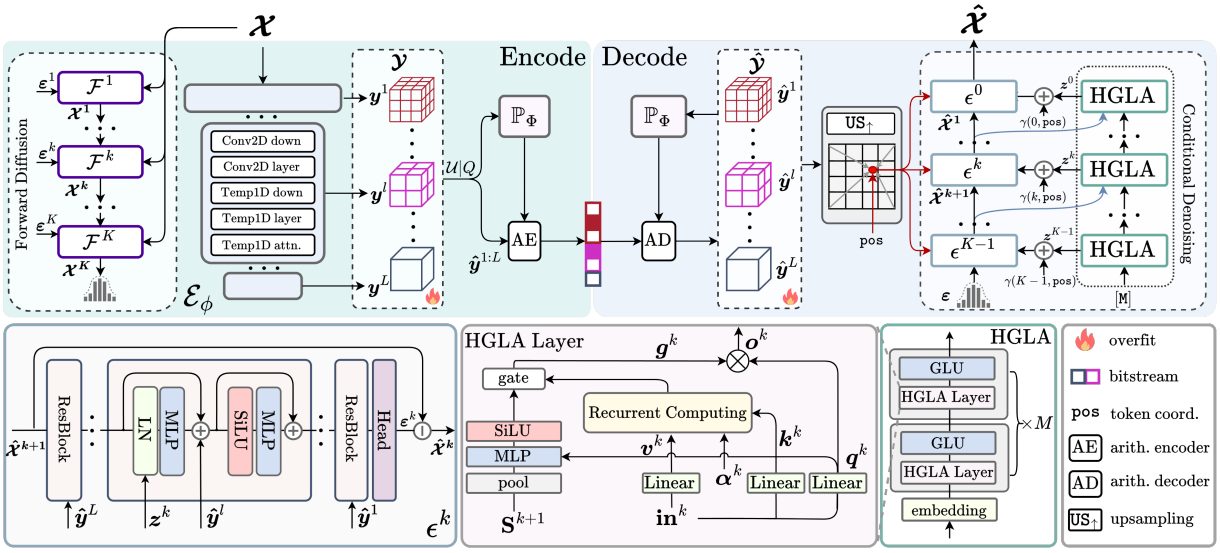

Figure 2. Illustration of the GIViC network architecture.

have proved to be more reliable and expressive compared to other types of generative models, e.g., VAEs [42] and GANs [26]. They have contributed to (generative) visual compression by unconditionally communicating lossy Gaussian samples [73] or by generating photorealistic images conditioned on entropy-encoded information [62, 85, 89, 93]. Although most diffusion models enforce a fixed forward corruption process and operate at a single resolution [66], recent studies [5, 18, 33, 71] demonstrate a more generalized and efficient alternative, i.e., performing diffusion across multiple resolutions and incorporating *arbitrary* degradations such as blurring and vector quantization.

3. Methods

Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{T\times H\times W\times 3}$ be a GOP (Group of Pictures) with T consecutive video frames with spatial resolution $H \times W$. As shown in Figure 2, a set of latents $\boldsymbol{\mathcal{Y}} = \{\boldsymbol{y}^l\}_{l=1}^{L}, \boldsymbol{y}^l \in \mathbb{R}^{S_t^l \times S_h^l \times S_w^l \times D^l}$, embedded within a compact local grid structure [45], is initialized by a spatiotemporal encoder $\mathcal{E}_\phi$, i.e., $\boldsymbol{\mathcal{Y}} = \mathcal{E}_\phi(\boldsymbol{\mathcal{X}})$ and stochastically overfitted to $\boldsymbol{\mathcal{X}}$ during encoding. Here (S_t^l, S_h^l, S_w^l) denotes the local grid size at level l, and D^l is the channel dimension. The complete representation $\boldsymbol{\mathcal{Y}}$ is quantized into $\hat{\boldsymbol{\mathcal{Y}}} = \{\hat{\boldsymbol{y}}^l\}_{l=1}^{L}$ by $Q(\cdot)$, which we relax as $q_\phi(\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}) = \mathcal{U}(\boldsymbol{\mathcal{Y}} - 0.5, \boldsymbol{\mathcal{Y}} + 0.5)$ during encoding to address the non-differentiability issue [4]. A context model $\mathbb{P}_\Phi(\cdot)$ is then used to evaluate the probability mass function (PMF) of $\hat{\boldsymbol{\mathcal{Y}}}$, with which $\hat{\boldsymbol{\mathcal{Y}}}$ could be losslessly entropy encoded into the bitstream.

At the decoder, the quantized hierarchical latents $\hat{\boldsymbol{\mathcal{Y}}}$ are entropy decoded from the bitstream based on the same context model $\mathbb{P}_\Phi(\cdot)$, which is recurrently updated by the previously decoded latents from the previous spatiotemporal *subgroups* - see the description in **Tokenization and shuffling** for detailed definitions. $\hat{\boldsymbol{\mathcal{Y}}}$ contains per-token visual priors that could be extracted via coordinate-based interpolation and used to steer the denoising process towards faithfully reconstructing $\hat{\boldsymbol{\mathcal{X}}}$. The conditional denoising processing is based on a *per-token* denoising diffusion variational autoencoder, implemented by unrolling a small, L-layer (plus a head layer mapping to the pixel space) INR-based denoiser over K representation levels, i.e., $\boldsymbol{\epsilon}_\theta := \{\epsilon^k\}_{k=0}^{K-1}$. Each ϵ^k predicts the noise ε^k at step k, from which the denoised output is produced as $\hat{\boldsymbol{\mathcal{X}}}^k$, conditioned by $\gamma(k, \texttt{pos}) + \boldsymbol{z}^k$. Here $\gamma(\cdot)$ denotes positional embedding [67], pos denotes the token's global 3D coordinate, and $\boldsymbol{\mathcal{Z}}^k$ is the conditioning vector sampled by the HGLA transformer based on the previously denoised output $\hat{\boldsymbol{\mathcal{X}}}^{k+1}$ (or a mask token [M] if $k = K$) as input, and its parameters are optimized offline.

3.1. Spatiotemporal Encoder

The spatiotemporal encoder $\mathcal{E}_\phi$ relies on large spatiotemporal receptive fields to generate consistent and compact latents. To avoid pretraining $\mathcal{E}_\phi$ entirely from scratch, we instead 'inflate' a pretrained image autoencoder [21] to handle the additional temporal dimension, inspired by recent image-to-video generation methods [7, 87], by inserting temporal downsampling, convolutional, and attention layers in between 2D spatial operations, as illustrated in Figure 2.

3.2. Implicit diffusion

The proposed implicit diffusion is a generalized [5, 28, 63] spatiotemporal pyramidal framework, which subsumes traditional explicit multi-resolution diffusion models into a

single, continuous forward-reverse chain. Instead of treating sub-band and resolution transitions separately, we embed them implicitly within a unified process, ensuring end-to-end consistency across continuous representation scales. We perform per-token diffusion while maintaining a constant token size across scales, eliminating the need to chain separate models at progressively higher resolutions. This design significantly enhances training and inference efficiency by leveraging **(i)** distributed computations across multiple spatiotemporal resolutions, and **(ii)** a novel dual-factorized conditional denoising strategy which enables finer-grained subspaces to be extrapolated from those at coarser scales and fully decoded/denoised reference frames.

Tokenization and shuffling. We start off by defining a *tokenization-and-shuffle* operation[1] that specifies the order by which tokens are denoised. We first partition the input GOP $\boldsymbol{\mathcal{X}}$ into $N_t = \lceil T/r_t \rceil$ subgroups along the temporal dimension and reorder these subgroups according to the hierarchical frame structure in the RA configuration used by modern standard video codecs [9]. Here r_t is the resolution of the token in the temporal domain. Within each subgroup (indexed by i), the frames are further patchified into continuous-valued, non-overlapping 3D tokens with spatial resolution (r_h, r_w), yielding $\{\boldsymbol{x}_{i,j}\}_{j=1}^{N_s}$, where $N_s = \lceil H/r_h \rceil \times \lceil W/r_w \rceil$ and $\boldsymbol{x}_{i,j} \in \mathbb{R}^{r_t \times r_h \times r_w \times 3}$. The partitioned tokens are further grouped spatially according to the Quincunx pattern [23], resulting in $\{\boldsymbol{x}_{i,j}\}_{j \in G_d}, d = 1, \ldots, 5$. This defines the (causal) order in which tokens are decoded. Here d stands for the spatial decoding step, G_d denotes the group of tokens in this temporal subgroup (i) that are decoded at step d, and $|G_1| = |G_2| = \lceil N_s/16 \rceil$ and $|G_d| = 2 \times |G_{d-1}|$ for $d = 3, 4, 5$. Based on this grouping method, the number of tokens decoded per step is **doubled** along both spatial and temporal axes, which reduces the number of decoding steps[2] to $5K \cdot (\log_2(N_t - 1) + 2)$.

Forward diffusion. We define a forward diffusion process that entails a sequence of transforms $\boldsymbol{\mathcal{F}} = \{\mathcal{F}^k\}_{k=1}^{K}$ that progressively "corrupt" $\boldsymbol{\mathcal{X}}$:

$$\boldsymbol{\mathcal{X}}^k = \mathcal{F}^k(\boldsymbol{\mathcal{X}}, \boldsymbol{\varepsilon}^k) = \mathtt{DS}\big(\boldsymbol{\mathcal{X}}, \boldsymbol{R}^k\big) + \bar{\beta}^k \boldsymbol{\varepsilon}^k, \qquad (1)$$

in which $\boldsymbol{\mathcal{X}}^k \in \mathbb{R}^{T^k \times H^k \times W^k \times 3}$, $T^k = \lceil T/R_t^k \rceil$, $H^k = \lceil H/R_s^k \rceil$, and $W^k = \lceil W/R_s^k \rceil$. $\boldsymbol{\varepsilon}^k \sim q(\boldsymbol{\varepsilon}^k)$ represents the Gaussian noise with a normal distribution. $\mathtt{DS}(\cdot, \boldsymbol{R}^k)$ denotes the frequency decomposition operation that downsamples the input temporally and spatially by a factor of $\boldsymbol{R}^k = (R_t^k, R_s^k)$ at scale k. $q(\cdot)$ corresponds to a normal distribution. When $k = 1$, $\boldsymbol{\mathcal{X}}^0 \equiv \boldsymbol{\mathcal{X}}$. $\bar{\beta}^k$ is the noise scheduling parameter controlling the strength of noise at step k.

[1]For better clarity, a visual illustration of this process is available in the Supplementary.

[2]Here, we assume $\log_2(N_t - 1) \in \mathbb{Z}$.

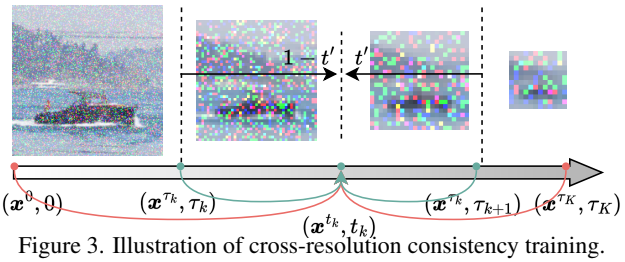

Figure 3. Illustration of cross-resolution consistency training.

Tokenization-and-shuffle is also applied to each $\boldsymbol{\mathcal{X}}^k$, producing re-ordered 3D tokens $\{\{\boldsymbol{x}_{i,j}^k\}_{j \in G_d^k}\}_{i=1}^{N_t^k}$ with the constant token size (r_t, r_s, r_s), where $N_t^k = \lceil T^k/r_t \rceil$ and $N_s^k = \lceil H^k/r_s \rceil \times \lceil W^k/r_s \rceil$. In this way, the number of tokens is reduced with the scales and each token $\boldsymbol{x}_{i,j}^k$ encapsulates the visual contents of multiple corresponding tokens in $\boldsymbol{\mathcal{X}}^{k-1}$ at a coarser scale. The transforms $\mathcal{F}^1 \cdots \mathcal{F}^k$ gradually destroy the original information, i.e., $\mathbb{I}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}}^k) \leq \mathbb{I}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}}^{k-1}), \forall k \in \{1, \ldots, K\}$ and $\mathbb{I}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}}^K) \approx 0$. $\mathbb{I}(\cdot, \cdot)$ denotes the mutual information.

Conditional denoising. At step k (with the corresponding denoiser ϵ^k), the per-token denoised output[3] $\hat{\boldsymbol{x}}^{k-1}$ is derived by:

$$\begin{aligned}\hat{\boldsymbol{x}}^{k-1} &= \hat{\boldsymbol{x}}^k - \bar{\beta}^k \boldsymbol{\varepsilon}^k \\ &= \hat{\boldsymbol{x}}^k - \bar{\beta}^k \epsilon^k \left(\hat{\boldsymbol{x}}^k; \boldsymbol{z}^k + \gamma(k, \mathsf{pos}), \{\hat{\boldsymbol{y}}^l\}_{l=1}^{L}\right), \end{aligned} \qquad (2)$$

where $\boldsymbol{z}^k$ is the corresponding conditioning token of $\boldsymbol{x}^k$ produced by the HGLA transformer ($\boldsymbol{\mathcal{Z}} = \{\boldsymbol{z}^k\}$). $\gamma(\cdot)$ denotes the positional embedding [67].

We then define a set of uniform intervals $0 = \tau_0 < \tau_1 < \cdots < \tau_K = 1$, as shown in Figure 3, along the continuum of spatiotemporal resolutions, which partitions the diffusion time interval $[0, 1]$ into K sub-intervals. Here, we allow $K \to \infty$ during training. Considering a random diffusion step t_k falling into the sub-interval $[\tau_k, \tau_{k+1})$, we calculate the normalized position of t_k within the interval as $t' = (t_k - \tau_k)/(\tau_{k+1} - \tau_k)$. The corresponding denoised token $\hat{\boldsymbol{x}}^{t_k}$ could be yielded via two interpolation paths "blending" the higher-resolution clean and lower-resolution noisy counterparts:

$$\hat{\boldsymbol{x}}_{(1)}^{t_k} = t_k \mathtt{DS}(\boldsymbol{x}, \boldsymbol{R}^{t_k}) + (1 - t_k) \mathtt{US}(\boldsymbol{x}^K, \boldsymbol{R}^{\tau_k} / \boldsymbol{R}^{t_K}), \qquad (3)$$

$$\hat{\boldsymbol{x}}_{(2)}^{t_k} = t' \mathtt{US}(\boldsymbol{x}^{\tau_{k+1}}, \boldsymbol{R}^{t_k} / \boldsymbol{R}^{\tau_{k+1}}) + (1 - t') \mathtt{DS}(\boldsymbol{x}^{\tau_k}, \boldsymbol{R}^{\tau_k} / \boldsymbol{R}^{t_k}), \qquad (4)$$

where $\mathtt{US}(\cdot, \cdot)$ stands for the upsampling operation, similar to $\mathtt{DS}$. With this formulation, the K stages at inference time could be viewed as a discretization of continuous, densely sampled downsampling stages over timesteps $\tau \in [0, 1]$

[3]We omit the subscripts i, j in subsection 3.2 for simplicity.

during training. The denoising objective $\mathcal{L}_{\text{consistency}}(\theta)$ is defined as:

$$\mathcal{L}_{\text{consistency}}(\theta) = \mathbb{E}\left[\|\boldsymbol{\epsilon}_\theta(\hat{\boldsymbol{x}}^{t_k}_{(1)}, t^k) - \boldsymbol{\epsilon}_\theta(\hat{\boldsymbol{x}}^{t_k}_{(2)}, t^k)\|^2\right], \quad (5)$$

which enforces that the output at time t^k for arbitrary k is similar regardless of the path taken.

3.3. HGLA Transformer

We propose a linear attention based transformer [27, 40, 57, 86] dubbed HGLA (Hierarchically Gated Linear Attention), which leverages fixed-size 2D hidden states to store historical contexts, enabling recurrent updates that are parallel during training and of linear complexity w.r.t context length during inference. HGLA[4] is the backbone in our framework for denoising conditioning, i.e., $\boldsymbol{\mathcal{Z}}^k \sim \prod_{k=K}^{k} p_\psi(\boldsymbol{\mathcal{Z}}^k|\hat{\mathcal{X}}^{K:k+1})$, and for modeling the hierarchical prior $\prod_{l=L}^{1} \mathbb{P}_\Phi(\boldsymbol{\mathcal{Y}}^l)$. It is inspired by HGRN2 [61] and tailored for the GIViC's multi-scale design by maintaining matrix-valued state $\mathbf{S}^k_{i,j}$ per representation level k that is updated per spatiotemporal subgroup for scale k and interacts across scales, which facilitates long-term dependency modeling along both scale and sequence axes.

Architecture. Each HGLA module comprises M HGLA layers (as shown in Figure 2). The dynamics of $\mathbf{S}^k_{i,j}$ are modulated by data-dependent decays, based on cumulative softmaxing $\texttt{cumax}$ [61] along the sequence dimension, that specify a lower bound $\boldsymbol{\alpha} \in \mathbb{R}^{M\times h}$ on how rapidly the historical contexts are updated, thus guiding lower and upper layers of each HGLA module to focus on short- and long-term dependencies, respectively. Specifically, we maintain a set of $\boldsymbol{\Gamma}^k \in \mathbb{R}^{M\times h}$ per representation scale k, where h stands for the hidden state dimension, and introduce a bias term $\Delta^k = \log(\frac{R^k_t \cdot R^k_s}{R^K_t \cdot R^K_s}) \cdot \mathbf{1}_{M\times h}$, resulting in the gating $\boldsymbol{\alpha}^k \in \mathbb{R}^{M\times h}$:

$$\boldsymbol{\alpha}^k = \texttt{cumsum}\left(\texttt{softmax}\left(\tilde{\boldsymbol{\Gamma}}^k, \text{dim} = 0\right), \text{dim} = 0\right), \quad \text{where} \quad \tilde{\boldsymbol{\Gamma}}^k = \boldsymbol{\Gamma}^k + \Delta^k. \quad (6)$$

Further, to support cross-scale information propagation, we attend the query $\boldsymbol{q}^k_{i,j}$ to a *learned mixture* of the current scale's hidden state $\mathbf{S}^k_{i,j}$ and the *updated* hidden state from the coarser scale $\mathbf{S}^{k+1}_{\prec(i,j)}$, based on the gating value $\boldsymbol{g}^k_{\prec(i,j)} \in \mathbb{R}^h$, where we use $\prec (i,j)$ to denote token indices that are smaller than i spatially and j temporally[5].

[4] In subsection 3.3 we use the notation for the case of diffusion conditioning (i.e., $k = 1, \ldots, K$), however the idea is easily generalizable (as used for the entropy model $\mathbb{P}_\Phi$).

[5] We have $\mathbf{S}^k_{i,j} \equiv \boldsymbol{S}^k_{\prec(i',j')}$ where i', j' entail the indices for all tokens to be decoded the next step.

The aforementioned HGLA operation at the scale k is formalized as:

$$\boldsymbol{g}^k_{i,j} = \sigma\left(\boldsymbol{W_g}\left([\boldsymbol{q}^k_{i,j}; \text{pool}\left(\mathbf{S}^{k+1}_{i,j}\right)]\right) + \boldsymbol{b_g}\right), \quad (7)$$

$$\mathbf{S}^k_{i,j} = \mathbf{S}^{k+1}_{i,j} \cdot \text{Diag}(\boldsymbol{\alpha}^k_{i,j}) + \sum \boldsymbol{v}^k_{i,j} \otimes \boldsymbol{k}^k_{i,j}, \quad (8)$$

$$\boldsymbol{o}^k_{i,j} = \left(\boldsymbol{g}^k_{i,j}\mathbf{S}^{k+1}_{i,j} + \left(\mathbf{1} - \boldsymbol{g}^k_{i,j}\right)\mathbf{S}^k_{\prec(i,j)}\right) \cdot \boldsymbol{q}^k_{i,j}, \quad (9)$$

where $\text{pool}(\cdot)$ denotes the average pooling operation and $\boldsymbol{o}^k_{i,\forall j}$ is the output of the HGLA module. Here, the queries $\boldsymbol{q}^k_{i,j}$, keys $\boldsymbol{k}^k_{i,j}$, and values $\boldsymbol{v}^k_{i,j}$ for the module's input $\mathbf{in}^k_{i,j} \in \mathbb{R}^h$ are generated as,

$$\boldsymbol{q}^k_{i,j} = \boldsymbol{W_q}\mathbf{in}^k_{i,j}, \boldsymbol{k}^k_{i,j} = \boldsymbol{W_k}\mathbf{in}^k_{i,j}, \boldsymbol{v}^k_{i,j} = \boldsymbol{W_v}\mathbf{in}^k_{i,j}, \quad (10)$$

in which $\{\boldsymbol{W_q}, \boldsymbol{W_k}, \boldsymbol{W_v} \in \mathbb{R}^{h\times h}\}$ are the learnable linear embedding matrices. It is noted that, for the initial decoding step where the first group of $\hat{\boldsymbol{x}}^K_{i,j}$ (or, equivalently, $\hat{\boldsymbol{y}}^L$ in the case of context modeling) has not been decoded, we feed the transformer(s) with a set of learned mask tokens $[\texttt{M}]$.

3.4. Optimization

Loss function. The above-described per-token generative framework $p(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^{0:K}, \hat{\boldsymbol{y}}^{1:L})$ could be formalized as:

$$p(\hat{\boldsymbol{x}}^K)\prod_{k=K-1}^{0} \underbrace{p_\theta(\hat{\boldsymbol{x}}^k|\boldsymbol{z}^k, \hat{\boldsymbol{y}}^{1:L})}_{\text{denoising obj.}} \underbrace{p_\psi(\boldsymbol{z}^k|\hat{\boldsymbol{x}}^{K:k+1})}_{\text{recurrent hidden state}} \quad (11a)$$

$$p(\hat{\boldsymbol{y}}^L)\underbrace{\prod_{l=L-1}^{1}\mathbb{P}_\Phi(\hat{\boldsymbol{y}}^l|\hat{\boldsymbol{y}}^{>l})}_{\text{hierarchical prior}}. \quad (11b)$$

This can be re-written using negative log likelihood over all tokens with a GOP-level Lagrange multiplier λ, yielding the rate-distortion (RD) objective of GIViC:

$$\mathcal{L}_{\text{RD}} = \mathbb{E}[-\log p(\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{Y}}) - \lambda \log p(\boldsymbol{\mathcal{Y}})]. \quad (12)$$

It is noted that when $\hat{\boldsymbol{\mathcal{X}}}$ is stochastically generated during decoding, it may result in unstable reconstructions and potentially fail the distortion metric. Given that the majority of open-sourced video compression baselines remain distortion-oriented, we modify the diffusion loss in Equation 5 using a *post-hoc* guiding mechanism [20] to optimize GIViC for the MSE loss.

Pretraining. To improve training efficiency, we employ a multi-stage pre-training procedure [49, 54], in which we first fix $(\boldsymbol{\mathcal{F}}, \boldsymbol{\epsilon})$ of the implicit diffusion and optimize $\mathcal{E}_\phi$ and $\mathbb{P}_\Phi$ with a pre-trained quadtree entropy model [24] for entropy modeling, which we discard in later stages. We then fix $\mathcal{E}_\phi$, swap the HGLA-based backbone with $\mathbb{P}_\Phi$ for the context model, and only update $\mathbb{P}_\Phi$ to optimize the rate loss. Finally, we jointly optimize all components with the RD loss defined by Equation 12.

BD-rate (%)	UVG		MCL-JCV		JVET-B		Model Complexity			
Codec	PSNR	MS-SSIM	PSNR	MS-SSIM	PSNR	MS-SSIM	Enc. FPS	Dec. FPS	Params (M)	kMACs/px
HM 18.0 (RA) [64]	-45.65	-40.67	-44.41	-40.01	-43.88	-46.94	0.06	39.5	N/A	N/A
VTM 20.0 (RA) [11]	-15.94	-16.19	-11.44	-8.57	-7.71	-6.32	0.02	23.1	N/A	N/A
AV1 libaom v3.0.2 (RA) [29]	-26.80	-23.77	-	-	-10.31	-9.98	0.03	24.5	N/A	N/A
DCVC-DC [49]	-36.68	-40.12	-37.03	-31.13	-36.42	-30.63	0.99	1.39	50.8	1274
DCVC-FM [49]	-22.46	-27.23	-20.31	-21.01	-22.34	-24.28	0.93	4.87	44.9	1073
PNVC (RA) [24]	-34.87	-25.74	-30.24	-31.98	-25.86	-23.03	0.01	23.6	21.8	102
NVRC [46]	-8.52	-4.92	-33.59	-30.62	-16.13	-10.66	4.4 ± 2.1	16.5 ± 6.7	16.8 ± 14.5	582.1 ± 396.9
GIViC w/o Overfit.	3.37	3.55	4.13	4.27	3.65	2.99	0.45	9.79	225.9	2399
GIViC w/ Overfit.	0.00	0.00	0.00	0.00	0.00	0.00	0.03	9.79	225.9	2399

Table 1. Compression performance results of the proposed GIViC framework. Here each BD-rate value is calculated when the corresponding benchmark codec is used as the anchor. Complexity figures for all benchmarked methods have also been provided for comparison.

Encoding. The latent grids are initialized by running $\mathcal{E}_\phi$ and treated as learnable parameters to be iteratively updated during the encoding process. Here we follow [41] to estimate the gradients of $\hat{\mathcal{Y}}$ based on soft-rounding [2] instead of STE, and replace the additive uniform noise with samples from the Kumaraswamy distribution [44] with progressive annealing. Other components in the GIViC framework are fixed during encoding.

4. Experiment Configuration

Implementation. We pre-trained five baseline models with $\lambda = \{85, 170, 380, 840, 1024\}$. By default, the number of diffusive sampling steps is set to 500 at training time and 8 at inference time. The 3D token size is set to $(4, 8, 8)$. Both the MLP-based denoiser ϵ and the HGLA transformer M have a depth of 4 (i.e., $L = M = 4$). All submodules are optimized using the ADAM optimizer [43] and with an initial learning rate set to 10^{-4} that is progressively annealed following a cosine scheduling.

Datasets. GIViC was pretrained on Vimeo-90k [83], and fine-tuned on additional 3,024 videos extracted from raw Vimeo footage, each of which consists of 32 frames, following the practices by [24, 50]. For a more comprehensive assessment of GIViC under different training conditions, we have ablated its performance by instead fine-tuning GIViC on the 7-frame Vimeo-90k sequences (V1.1 in Table 2). For testing, we evaluated all models on the UVG [55], MCL-JCV [76], and JVET-B [8] test sets.

Baselines. GIViC is compared against seven SOTA baselines, including (i) three **conventional codecs** - H.265/HEVC Test Model HM 18.0 [64], H.266/VVC Test Model VTM 20.0 [11] and AV1 libaom v3.0.2 [29]; (ii) two **neural video codecs** - DCVC-DC [49] and DCVC-FM [50], and (iii) two **INR-based codecs** - PNVC [24] and NVRC [46]. Additional comparisons are available in the Supplementary.

Test conditions. All experiments on conventional codecs use the Random Access mode defined in JVET common test conditions [9]. For each rate point, we calculate the bitrate (bit/pixel, bpp) and quantitatively assess the quality of lossy reconstructions using PSNR and MS-SSIM [78] in the YUV colorspace. In the Supplementary, we further report VMAF[51] and LPIPS results which align better with human visual preference. The Bjøntegaard Delta Rate (BD-rate) [6] is then used to measure the relative compression efficiency between codecs. We configured PNVC, GIViC, and the conventional codecs in RA mode with a GoPsize=32 and IntraPeriod=32. For HM 18.0 and VTM 20.0, we employed QP = $\{16, 20, 34, 38, 32, 36\}$ to cover a broader range of bitrates. We note unlike the other selected baselines, NVRC incurs a system delay equal to the full sequence length.

5. Results and Discussion

5.1. Overall Performance

Quantitative results. The rate-distortion performance of the proposed GIViC codec, compared to conventional and neural video codecs, is summarized in Table 1. Notably, GIViC outperforms all tested neural video codecs in terms of compression performance. Specifically, GIViC achieves 15.94%, 36.68%, and 8.52% BD-rate reductions compared to VTM 20.0 (RA), DCVC-FM, and NVRC, respectively. While we cannot directly benchmark GIViC against DCVC-LCG [59] (the source code of the latter is not available), we have compared GIViC with the results reported in its original literature, which confirms the superior performance of GIViC. Notably, GIViC remains dominant in coding performance even without sequence-specific overfitting involved, a result presumably attributed to its powerful 3D autoencoder and diffusion-transformer backbone, which accurately capture global dependencies. Figure 4 illustrates the rate-distortion curves of GIViC alongside a selected subset of baseline methods from each category - conventional,

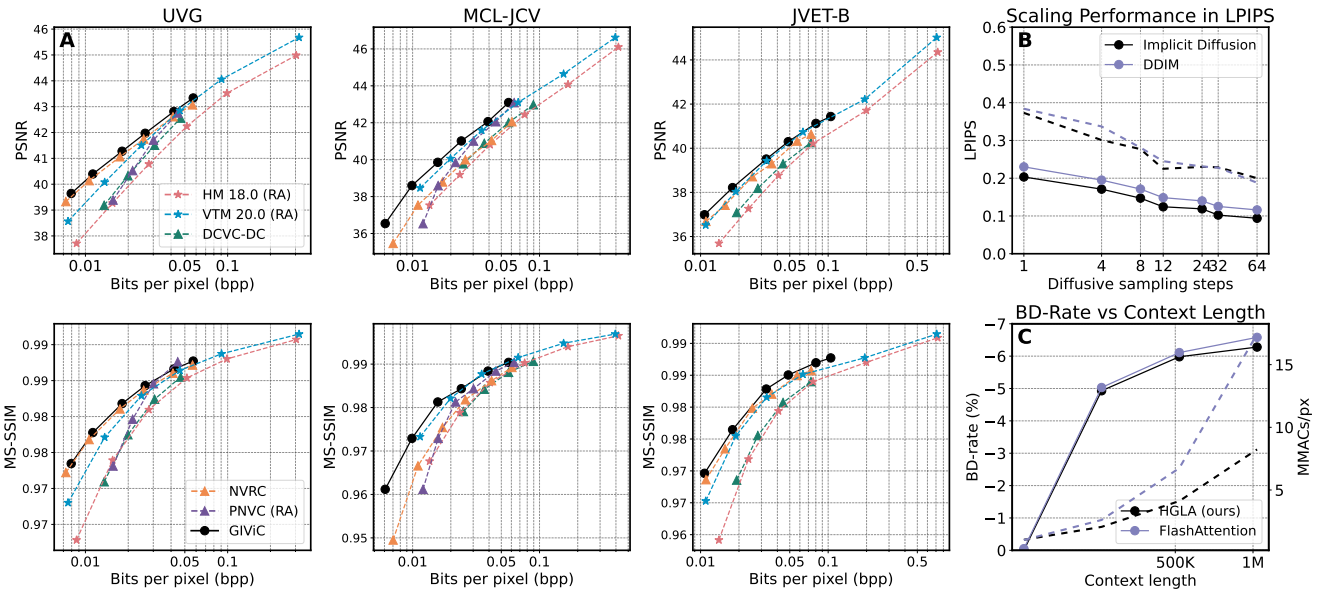

Figure 4. (**A**) Rate-distortion curves on UVG, MCL-JCV, and JVET-B datasets. (**B**) Reconstruction quality PSNR w.r.t diffusive sampling steps for low bitrate range (solid lines) and high bitrate range (dashed line) respectively. (**C**) BD-rate (PSNR, solid lines) and decoding complexity (dashed lines) w.r.t context length.

autoencoder-based, and INR-based - for three different test sets. GIViC demonstrates consistently strong performance across diverse datasets and the entire tested bitrate range, in terms of both PSNR and MS-SSIM.

Qualitative results. We further demonstrate the superior performance of our method in terms of subjective visual quality. Figure 5 compares frame reconstructions from the proposed GIViC with those from VTM, DCVC-FM, PNVC, and NVRC, showcasing GIViC's improved perceptual quality and reduction of compression artifacts. Please refer to the Supplementary for more examples at various distortion-perception trade-offs.

Complexity. A complexity profiling is summarized in Table 1. The average encoding and decoding speeds (FPS) for each model are measured on a single NVIDIA A100 GPU. For INR-/overfitting-based methods, the encoding FPS is measured for one full forward and backward pass [45]. On average, the training of GIViC takes 1.78 hours on one GOP with 32 frames at the 1920 $\times$ 1080 resolution. While GIViC exhibits longer encoding and decoding times than some benchmark codecs, this latency can be substantially reduced without incurring noticeable performance drop through three strategies: 1) removing or decreasing the number of overfitting steps, 2) initializing the model with states and latent grids from the preceding GOP, and 3) reducing the model size and the number of diffusive sampling steps. More results can be found in the Supplementary.

Scaling performance. We assess GIViC's scaling behaviors by reporting the variation in reconstruction or compression performance w.r.t number of diffusive sampling steps and context lengths, respectively, in comparison with other diffusion and transformer models. In Figure 4 **(B)** we compare the generation fidelity (measured by LPIPS [90] in the RGB colorspace) of the proposed implicit diffusion against the vanilla DDIM [20] in a single resolution for three lower bitrates (solid lines) and three higher bitrates (dashed lines), respectively. It can be observed that our model scales more favorably and converges at a faster rate for both bitrate ranges. A similar trend is also seen from scaling HGLA on the UVG dataset, shown in Figure 4 **(C)**, where we configure the GoPSize to 16, 32, 64, and 128, respectively, with the corresponding context length equal to $\frac{\text{GoPSize} \times HW}{r_t \times r_s^2}$. It can be observed that the context length of HGLA leads to a steady, non-trivial, and comparable gain in compression performance with FlashAttention [17], despite its linear (instead of quadratic) complexity w.r.t the context length.

5.2. Ablation Study

We analyze the impact of our methodological contributions and design choices by systematically removing or replacing sub-components and measuring the resulting change in BD-rate and model complexity on two datasets: the UVG and MCL-JCV datasets. The ablative variants include:

Effectiveness of pre-training is verified by replacing the 32-frame Vimeo raw sequences with the original Vimeo-90k dataset for fine-tuning (V1.1) and removing the overfitting process in the encoding pipeline (V1.2).

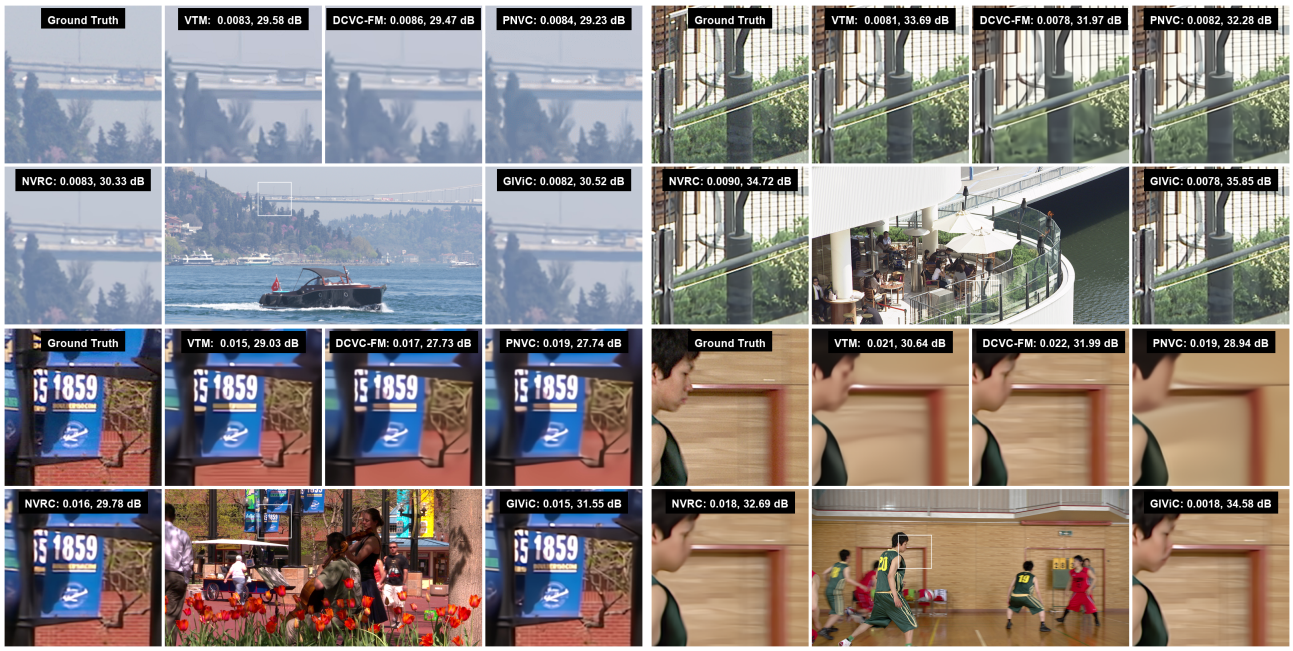

Figure 5. Visual comparison of reconstructions by different video codec baselines, where we report the average sequence bpp and the corresponding frame's PSNR.

Effectiveness of temporal inflation is tested by instead pretraining a 3D autoencoder entirely from scratch (V2.1) for the same number of optimization steps.

Effectiveness of implicit diffusion model is confirmed by respectively replacing the implicit diffusion with single-resolution per-token diffusion (V3.1), employing RelayDiffusion [71] (V3.2), and replacing the proposed consistency objective with simple flow matching [52] (V3.3).

Effectiveness of HGLA is verified by keeping the original lower bound formulation (V4.1), removing the learned gating across layers (V4.2), and replacing it with a vanilla transformer [75] that performs next-scale prediction (V4.3). Further, we ablate the context model $\mathbb{P}_\Phi$ by replacing it with a convolution-based model [24] with a comparable size (V5.1) and allowing for instance-specific overfitting Φ following [24] (V5.2).

The ablation study results are reported in Table 2, where all these ablative variants result in compression loss when compared to the original GIViC, indicating that each contribution in this work does improve the overall performance.

	BD-rate (%)		Model Complexity	
Version	UVG	MCL	params.(M)	kMACs/px
V1.1	5.87	5.51	256.1 (0.00%↓)	2399 (0.00%↓)
V1.2	3.66	3.19	256.1 (0.00%↓)	2399 (0.00%↓)
V2.1	0.98	0.97	303.5 (18.6%↑)	2806 (16.9%↑)
V3.1	2.35	2.69	256.1 (0.00%↓)	3590 (49.6%↑)
V3.2	2.53	3.23	256.1 (0.00%↓)	2399 (0.00%↓)
V3.3	3.09	3.41	256.1 (0.00%↓)	2399 (0.00%↓)
V4.1	1.99	2.43	256.1 (0.00%↓)	2399 (0.00%↓)
V4.2	3.17	3.20	223.8 (0.09%↓)	2371 (1.27%↓)
V4.3	0.19	0.25	219.7 (2.74%↓)	2987 (24.5%↑)
V5.1	6.77	6.25	221.2 (2.17%↓)	2380 (0.79%↓)
V5.2	3.20	2.65	221.2 (2.17%↓)	2380 (0.79%↓)

Table 2. Ablation study results on the UVG and MCL-JCV datasets in terms of BD-rates (measured in PSNR), and the entire model's size and kMACs/pixel, measured against GIViC. Here, the underlined ablative variants are irrelevant to architectural modifications and thus incur no changes in model complexity.

6. Conclusion

This paper proposes GIViC, an INR-based video coding framework using generalized diffusion and a novel transformer architecture. GIViC achieves superior compression performance, significantly outperforming state-of-the-art codecs like VTM 20.0 and NVRC on various datasets. To our knowledge, GIViC is the INR-based video codec that has surpassed VTM under the same Random Access (RA) constraint. However, it is also noted that GIViC does exhibit relatively high computational complexity due to its reliance on diffusion and transformer backbones, limiting its adoption for applications that require real-time decoding speeds.

Acknowledgment

This work was funded by the UKRI MyWorld Strength in Places Programme (SIPF00006/1). We would like to thank Mr. Ho Man Kwan for providing the NVRC compression results. We also acknowledge the Advanced Computing Research Centre at the University of Bristol for providing computational facilities.

References

[1] Mariana Afonso, Fan Zhang, and David R Bull. Video compression based on spatio-temporal resolution adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(1):275–280, 2018. 2

[2] Eirikur Agustsson and Lucas Theis. Universally quantized neural compression. *Advances in Neural Information Processing Systems*, 33:12367–12376, 2020. 6

[3] Yunpeng Bai, Chao Dong, Cairong Wang, and Chun Yuan. PS-NeRV: Patch-wise stylized neural representations for videos. In *IEEE International Conference on Image Processing*, pages 41–45. IEEE, 2023. 2

[4] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018. 3

[5] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *Advances in Neural Information Processing Systems*, 36:41259–41282, 2023. 3

[6] Gisle Bjontegaard. Calculation of average PSNR differences between rd-curves. *ITU SG16 Doc. VCEG-M33*, 2001. 6

[7] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 3

[8] Jill Boyce, Karsten Suehring, Xiang Li, and Vadim Seregin. JVET-J1010: JVET common test conditions and software reference configurations. In *10th Meeting of the Joint Video Experts Team*, pages JVET–J1010, 2018. 6

[9] Jill Boyce, Karsten Suehring, Xiang Li, and Vadim Seregin. JVET-J1010: JVET common test conditions and software reference configurations. In *10th Meeting of the Joint Video Experts Team*, pages JVET–J1010, 2018. 2, 4, 6

[10] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the Versatile Video Coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. 1, 2

[11] Adrian Browne, Yan Ye, and Seung Hwan Kim. Algorithm description for Versatile Video Coding and Test Model 19 (VTM 19). In *the JVET meeting*. ITU-T and ISO/IEC, 2023. 2, 6

[12] David Bull and Fan Zhang. *Intelligent Image and Video Compression: Communicating Pictures*. Academic Press, 2021. 1

[13] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. NeRV: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021. 1, 2

[14] Hao Chen, Matthew Gwilliam, Ser-Nam Lim, and Abhinav Shrivastava. HNeRV: A hybrid neural representation for videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10270–10279, 2023. 1, 2

[15] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, 2006. 2

[16] Duolikun Danier, Fan Zhang, and David Bull. LDMVFI: Video frame interpolation with latent diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1472–1480, 2024. 2

[17] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022. 7

[18] Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alex Dimakis, and Peyman Milanfar. Soft diffusion: Score matching with general corruptions. *Transactions on Machine Learning Research*, 2023. 3

[19] Gregoire Deletang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. In *International Conference on Learning Representations*, 2024. 2

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 5, 7

[21] Zhihao Duan, Ming Lu, Zhan Ma, and Fengqing Zhu. Lossy image compression with quantized hierarchical vaes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 198–207, 2023. 3

[22] Emilien Dupont, Adam Golinski, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. COIN: Compression with implicit neural representations. In *International Conference on Learning Representations Workshop on Neural Compression: From Information Theory to Applications*, 2021. 2

[23] Alaaeldin El-Nouby, Matthew J. Muckley, Karen Ullrich, Ivan Laptev, Jakob Verbeek, and Herve Jegou. Image compression with product quantized masked image modeling. *Transactions on Machine Learning Research*, 2023. 4

[24] Ge Gao, Ho Man Kwan, Fan Zhang, and David Bull. PNVC: Towards practical inr-based video compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3068–3076, 2025. 1, 2, 5, 6, 8

[25] Carlos Gomes, Roberto Azevedo, and Christopher Schroers. Video compression with entropy-constrained neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18497–18506, 2023. 2

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014. 3

[27] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2, 5

[28] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Miguel Ángel Bautista, and Joshua M. Susskind. f-DM: A multi-stage diffusion model via progressive signal transformation. In *International Conference on Learning Representations*, 2023. 3

[29] Jingning Han, Bohan Li, Debargha Mukherjee, Ching-Han Chiang, Adrian Grange, Cheng Chen, Hui Su, Sarah Parker, Sai Deng, Urvang Joshi, et al. A technical overview of AV1. *Proceedings of the IEEE*, 109(9):1435–1462, 2021. 1, 6

[30] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 2

[31] Bo He, Xitong Yang, Hanyu Wang, Zuxuan Wu, Hao Chen, Shuaiyi Huang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. Towards scalable neural representation for diverse videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6132–6142, 2023. 2

[32] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[33] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. 3

[34] Yung-Han Ho, Chih-Peng Chang, Peng-Yu Chen, Alessandro Gnutti, and Wen-Hsiao Peng. CANF-VC: Conditional augmented normalizing flows for video compression. In *European Conference on Computer Vision*, pages 207–223. Springer, 2022. 2

[35] Zhihao Hu and Dong Xu. Complexity-guided slimmable decoder for efficient deep video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14358–14367, 2023. 2

[36] Zhihao Hu, Guo Lu, and Dong Xu. FVC: A new framework towards deep video compression in feature space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1502–1511, 2021. 2

[37] Yuzhen Huang, Jinghan Zhang, Zifei Shan, and Junxian He. Compression represents intelligence linearly. *arXiv preprint arXiv:2404.09937*, 2024. 2

[38] Wei Jiang, Junru Li, Kai Zhang, and Li Zhang. Ecvc: Exploiting non-local correlations in multiple frames for contextual video compression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7331–7341, 2025. 1, 2

[39] Yuxuan Jiang, Ho Man Kwan, Tianhao Peng, Ge Gao, Fan Zhang, Xiaoqing Zhu, Joel Sole, and David Bull. HIIF: Hierarchical encoding based implicit image function for continuous super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[40] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 5

[41] Hyunjik Kim, Matthias Bauer, Lucas Theis, Jonathan Richard Schwarz, and Emilien Dupont. C3: High-performance and low-complexity neural compression from a single image or video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9347–9358, 2024. 6

[42] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[43] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[44] Ponnambalam Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2):79–88, 1980. 6

[45] Ho Man Kwan, Ge Gao, Fan Zhang, Andrew Gower, and David Bull. HiNeRV: Video compression with hierarchical encoding-based neural representation. *Advances in Neural Information Processing Systems*, 36:72692–72704, 2024. 1, 2, 3, 7

[46] Ho Man Kwan, Ge Gao, Fan Zhang, Andrew Gower, and David Bull. NVRC: Neural video representation compression. In *Advances in Neural Information Processing Systems*, pages 132440–132462. Curran Associates, Inc., 2024. 1, 2, 6

[47] Joo Chan Lee, Daniel Rho, Jong Hwan Ko, and Eunbyung Park. FFNeRV: Flow-guided frame-wise neural representations for videos. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7859–7870, 2023. 1, 2

[48] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34:18114–18125, 2021. 1

[49] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22616–22626, 2023. 2, 5, 6

[50] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with feature modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26099–26108, 2024. 1, 2, 6

[51] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara. Toward A Practical Perceptual Video Quality Metric. Netflix TechBlog, 2016. Accessed: 2025-07-31. 6

[52] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 8

[53] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. DVC: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 11006–11015, 2019. 1

[54] Fabian Mentzer, George D Toderici, David Minnen, Sergi Caelles, Sung Jin Hwang, Mario Lucic, and Eirikur Agustsson. VCT: A video compression transformer. *Advances in Neural Information Processing Systems*, 35:13091–13103, 2022. 2, 5

[55] Alexandre Mercat, Marko Viitanen, and Jarno Vanne. UVG dataset: 50/120fps 4K sequences for video codec analysis and development. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 297–302, 2020. 6

[56] Seungjun Oh, Hyunmo Yang, and Eunbyung Park. Parameter-efficient instance-adaptive neural video compression. In *Proceedings of the Asian Conference on Computer Vision*, pages 250–267, 2024. 2

[57] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. RWKV: Reinventing RNNs for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023. 2, 5

[58] Tianhao Peng, Ge Gao, Heming Sun, Fan Zhang, and David Bull. Accelerating learnt video codecs with gradient decay and layer-wise distillation. In *Picture Coding Symposium*, 2024. 2

[59] Linfeng Qi, Zhaoyang Jia, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Long-term temporal context gathering for neural video compression. In *European Conference on Computer Vision*, pages 305–322. Springer, 2025. 2, 6

[60] Shiyu Qin, Jinpeng Wang, Yimin Zhou, Bin Chen, Tianci Luo, Baoyi An, Tao Dai, Shutao Xia, and Yaowei Wang. Mambavc: Learned visual compression with selective state spaces. *arXiv preprint arXiv:2405.15413*, 2024. 2

[61] Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. HGRN2: Gated linear RNNs with state expansion. In *First Conference on Language Modeling*, 2024. 5

[62] Lucas Relic, Roberto Azevedo, Markus Gross, and Christopher Schroers. Lossy image compression with foundation diffusion models. In *European Conference on Computer Vision*, pages 303–319. Springer, 2024. 3

[63] Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *International Conference on Learning Representations*, 2023. 3

[64] Chris Rosewarne, Karl Sharman, Rickard Sjoberg, and Gary Sullivan. High efficiency video coding (HEVC) test model 16 (HM 16) improved encoder description update 16. In *the JVET meeting*. ITU-T and ISO/IEC, 2022. 6

[65] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020. 2

[66] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 3

[67] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 3, 4

[68] Karsten Suehring. ECM: Enhanced compression model. https://vcgit.hhi.fraunhofer.de/ecm/ECM, 2021. 2

[69] Gary J. Sullivan, Jens-Rainer Ohm, Woojin 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. 2

[70] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 2

[71] Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. *arXiv preprint arXiv:2309.03350*, 2023. 3, 8

[72] Siyue Teng, Yuxuan Jiang, Ge Gao, Fan Zhang, Thomas Davis, Zoe Liu, and David Bull. Benchmarking conventional and learned video codecs with a low-delay configuration. *arXiv preprint arXiv:2408.05042*, 2024. 2

[73] Lucas Theis, Tim Salimans, Matthew D Hoffman, and Fabian Mentzer. Lossy compression with gaussian diffusion. *arXiv preprint arXiv:2206.08889*, 2022. 3

[74] Ties van Rozendaal, Iris Huijben, and Taco S Cohen. Overfitting for fun and profit: Instance-adaptive data compression. In *International Conference on Learning Representations*, 2021. 2

[75] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 8

[76] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C-C Jay Kuo. MCL-JCV: a JND-based H.264/AVC video quality assessment dataset. In *IEEE International Conference on Image Processing*, pages 1509–1513, 2016. 6

[77] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 2

[78] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, pages 1398–1402. IEEE, 2003. 6

[79] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, 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. 2

[80] Jinxi Xiang, Kuan Tian, and Jun Zhang. MIMT: Masked image modeling transformer for video compression. In *International Conference on Learning Representations*, 2022. 1, 2

[81] Dejia Xu, Peihao Wang, Yifan Jiang, Zhiwen Fan, and Zhangyang Wang. Signal processing for implicit neural representations. *Advances in Neural Information Processing Systems*, 35:13404–13418, 2022. 2

[82] Tongda Xu, Han Gao, Chenjian Gao, Yuanyuan Wang, Dailan He, Jinyong Pi, Jixiang Luo, Ziyu Zhu, Mao Ye, Hongwei Qin, et al. Bit allocation using optimization. In *International Conference on Learning Representations*, pages 38377–38399. PMLR, 2023. 2

[83] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8): 1106–1125, 2019. 6

[84] Hao Yan, Zhihui Ke, Xiaobo Zhou, Tie Qiu, Xidong Shi, and Dadong Jiang. DS-NeRV: Implicit neural video representation with decomposed static and dynamic codes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23019–23029, 2024. 2

[85] Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[86] Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023. 2, 5

[87] Sihyun Yu, Weili Nie, De-An Huang, Boyi Li, Jinwoo Shin, and Anima Anandkumar. Efficient video diffusion models via content-frame motion-latent decomposition. In *International Conference on Learning Representations*, 2024. 3

[88] Fan Zhang, Di Ma, Chen Feng, and David R Bull. Video compression with CNN-based postprocessing. *IEEE MultiMedia*, 28(4):74–83, 2021. 2

[89] Pingping Zhang, Jinlong Li, Meng Wang, Nicu Sebe, Sam Kwong, and Shiqi Wang. When Video Coding Meets Multimodal Large Language Models: A Unified Paradigm for Video Coding. *arXiv preprint arXiv:2408.08093*, 2024. 3

[90] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 7

[91] Xinjie Zhang, Ren Yang, Dailan He, Xingtong Ge, Tongda Xu, Yan Wang, Hongwei Qin, and Jun Zhang. Boosting neural representations for videos with a conditional decoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2556–2566, 2024. 2

[92] Yiwei Zhang, Guo Lu, Yunuo Chen, Shen Wang, Yibo Shi, Jing Wang, and Li Song. Neural rate control for learned video compression. In *International Conference on Learning Representations*, 2024. 2

[93] Chuqin Zhou, Guo Lu, Jiangchuan Li, Xiangyu Chen, Zhengxue Cheng, Li Song, and Wenjun Zhang. Controllable distortion-perception tradeoff through latent diffusion for neural image compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10725–10733, 2025. 3