

001           **Latent-INR: A Flexible Framework for Implicit**  
002           **Representations of Videos with Discriminative**  
003           **Semantics**

004           Anonymous ECCV 2024 Submission

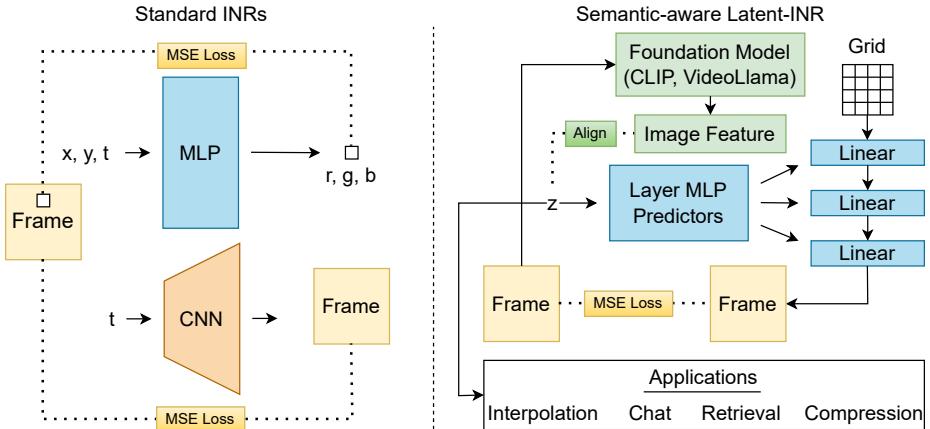
005           Paper ID #2356

006           **Abstract.** Implicit Neural Networks (INRs) have emerged as powerful  
007 representations to encode all forms of data, including images, videos, au-  
008 dios, and scenes. With video, many INRs for video have been proposed  
009 for the compression task, and recent methods feature significant improve-  
010 ments with respect to encoding time, storage, and reconstruction quality.  
011 However, these encoded representations lack semantic meaning, so they  
012 cannot be used for any downstream tasks that require such properties,  
013 such as retrieval. This can act as a barrier for adoption of video INRs  
014 over traditional codecs as they do not offer any significant edge apart  
015 from compression. To alleviate this, we propose a flexible framework  
016 that decouples the spatial and temporal aspects of the video INR. We  
017 accomplish this with a dictionary of per-frame latents that are learned  
018 jointly with a set of video specific hypernetworks, such that given a la-  
019 tent, these hypernetworks can predict the INR weights to reconstruct the  
020 given frame. This framework not only retains the compression efficiency,  
021 but the learned latents can be aligned with features from large vision  
022 models, which grants them discriminative properties. We align these la-  
023 tents with CLIP and show good performance for both compression and  
024 video retrieval tasks. By aligning with VideoLlama, we are able to per-  
025 form open-ended chat with our learned latents as the visual inputs. Ad-  
026 ditionally, the learned latents serve as a proxy for the underlying weights,  
027 allowing us perform tasks like video interpolation. These semantic prop-  
028 erties and applications, existing simultaneously with ability to perform  
029 compression, interpolation, and superresolution properties, are a first in  
030 this field of work.

031           **Keywords:** Implicit Neural Representation · Video Compression · Video  
032           Understanding

033           1 Introduction

034 In today’s age of content explosion, large quantities of data are created every  
035 second, and storing them reliably and efficiently is of utmost importance for  
036 many applications. A scalable compression technique enables companies to pro-  
037 vide better services at reduced cost and helps the end consumer by improving  
038 their access to high-fidelity data in addition to decongesting the network. Since



**Fig. 1:** Existing INRs for video (left) typically take some time-coordinate, or time and positional coordinates and train a single network to reconstruct a video. In contrast to these, we propose an INR system where a dictionary of implicit latent codes is learned for a video, one latent per frame. The latents are aligned to the image features of a large vision model, while simultaneously an INR system is learned which, given these latent codes, generates a positional INR which can reconstruct the frame. With this framework, we successfully develop an INR which performs both reconstructive tasks like compression, and semantic downstream tasks like retrieval and interactive chat.

the early 90s, several compression techniques have been created and widely deployed for this exact purpose. Out of these, JPEG [53] for images, HEVC [47], AV1 [12], and H.264 [54] for videos have emerged as the most popular choices, owing to their simple design and scalable performance.

In the past decade, the rise of deep learning led to a renaissance in computer vision, eventually impacting the visual data compression landscape [4, 15, 30]. Despite their success, these ML-based codecs have not seen widespread adoption like traditional codecs. This is in part due to failure to generalize, since ML codecs trained on large datasets can give sub-optimal compression for data points that differ significantly from their training set [7, 57]. Implicit Neural Representations (INR) attempt to avoid the generalization issue by operating internally. Instead of training large models that learn to identify *general* patterns in training data and apply them to specific out-of-distribution data, implicit techniques involve training a small model to exploit the *specific* patterns for the given data point. That is, for video compression, this approach would train one network per video, and for image compression, it would train one network per image. The resulting model is essentially a function that represents the underlying signal in spatial/temporal space.

Despite these advances, neural video compression remains unsolved. Various methods address issues of compression quality [9, 22], but two crucial questions remain unanswered – (i) how to scale for longer videos given architectural rigidity and (ii) how to reduce long encoding time due to training a network for every

video. Although recent works make some progress for these [28], the training time is still quite long, and INR behavior for lossy compression is not well-understood, limiting potential for practical adoption.

Furthermore, these approaches for INR tackle only one axis of the problem, i.e., how to formulate video INRs with the primary goal of compression. These aim to solve problems like long encoding time directly, by reducing it. In contrast to these works, we instead aim to justify the compute and time needed to train implicit representations. So, as a step towards ML-based codecs with compelling real-world potential, we present Latent-INR – a new flexible framework for formulating video INRs, where in addition to compression, the INR enables downstream tasks like retrieval and video question answering, without the need to decode the video. Our framework consists of two parts: (i) a dictionary of learnable latents, one for each frame, and (ii) a set of hypernetworks learned on the entire video which, given a latent as input, predict frame-specific weight modulations on the shared base network. This shared base takes a spatial coordinate grid as input and outputs the specific frame

This design allows us to separate the spatial and temporal aspects of the video by modeling them separately. We can view the set of hypernetworks as a base model that learns the general structure and style of the video, while each learned latent conditions it to output a specific frame. The latent here acts as a proxy for the weights of the frame-specific INR. This property is apparent from the video interpolation ability of our model - a task that other video INR representations struggle to perform. Like other video INRs, our method is competitive for compression, but uniquely retains the properties of original coordinate-based INR. That is, our continuous representations of frames allows for spatial interpolation, which can be leveraged for superresolution and a decoding paradigm we refer to as “any-resolution inference.” That is, at inference/decoding time, our same model, with no changes to latents or architecture, can decode a video at any resolution - a key feature missing from traditional codecs. This latent is also quite flexible, and according to the procedure shown in Figure 1, we can align it with the features from a large vision model, such as CLIP [33] to encode the visual semantics of the frame while retaining nice properties such as alignment with CLIP text embeddings. This allows for a whole spectrum of applications, including frame, concept, and whole video retrieval with text queries.

In summary, our framework gives that extra edge apart from compression to ML-based codecs, paving the way for their widespread adoption. Concretely,

- We propose an auto-decoder latent-based framework with spatio-temporal decoupling for implicit video representations. Compared to other video INR methods, this is a new way of formulating the problem.
- Our system has good compression performance, competing well with other ML-based codecs for PSNR, BPP, and decoding speed while also enabling any-resolution inference.
- The learnt latent embeddings from our framework demonstrate internal generalization from the encoded dataset, achieving video interpolation, a task that other INR based methods struggle to achieve.

- 106 – We align our latents with large foundational models like CLIP [33], thus  
107 making our representations useful for retrieval tasks.
- 108 – We align our entire dictionary with video features for VideoLlama [56] to  
109 enable chat-style applications, including video question answering and cap-  
110 tioning.

## 111 2 Related Work

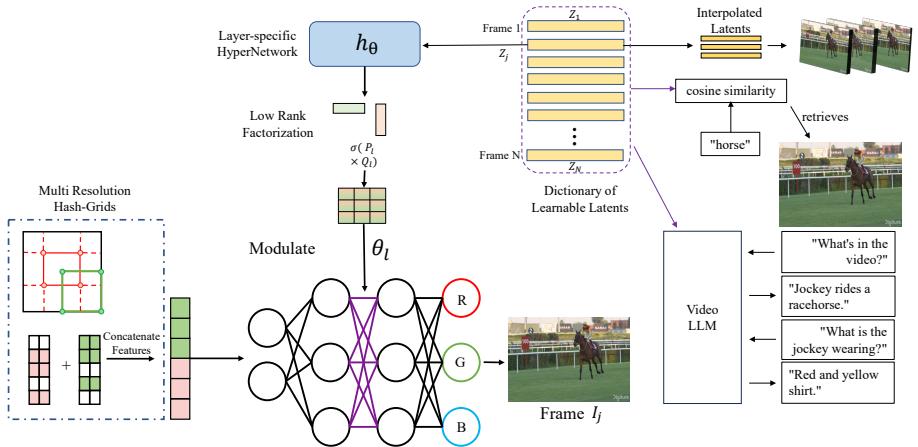
112 **Implicit Neural Representations (INR’s)** are a class of neural networks  
113 designed with the intention of representing a given data point or dataset perfectly  
114 rather than exploiting general patterns and generalizing for unseen data. SIREN  
115 [42] pioneered the use of periodic activations to train simple MLP’s that worked  
116 well across images, SDF and audio. This was followed by a host of works that  
117 improved the training process of INR’s by making them faster [32, 35, 50] work  
118 across multiple scales [36] and encode multiple data points [14]. Models that used  
119 meta learning [46, 49] started gaining ground as they offered the advantages of  
120 compression along with generalization. [38, 48] further made improvements to this  
121 line of work by directly learning sparse-INR’s leading to improved compression  
122 and improved optimization by dataset selection respectively.

123 **Hypernetworks** are a class of networks optimized for predicting parameters  
124 of another network, with the aim of generalizing across unseen tasks [16]. Some  
125 utilized these for scenes [13, 43, 44]. Trans-INR [11] introduced the paradigm  
126 of using a transformer based hypernet to convert data directly from image-  
127 space to INR’s. [21] improved upon this idea and made the important observation  
128 that it is sufficient to modulate only the first hidden layer of an INR to represent  
129 a dataset of points. Unfortunately, these hypernetworks act on input data points  
130 which require test-time optimizations, making them unsuitable for compression  
131 tasks. [39] try to overcome this with an “auto-decoder” framework, where learn-  
132 able latents represent a dataset of videos, with each latent corresponding to  
133 a single video, such that no encoder is needed. Others have investigated this  
134 paradigm for a variety of modalities [5, 37, 40]. Still, the lack of decoupling space  
135 from time prohibits the method from scaling to real-world videos.

136 **Video INRs** have recently gained popularity for compression. [9] was the  
137 first implicit representation which modelled a video as a function mapping the  
138 temporal coordinates to the corresponding frames. Later works [2, 8, 18, 25] it-  
139 erated on this method, providing improvements in performance. [22] enhanced  
140 this concept by incorporating hash-grid [32] representations to speed up encoding  
141 times. NIRVANA [28] represented a video using a series of smaller INR models  
142 trained in an autoregressive manner to scale for longer videos.

143 **Video Interpolation** has been a fundamental task in computer vision,  
144 helping in creating smoother visual experiences. Over the past few years, deep  
145 learning based methods have vastly improved the quality of these interpola-  
146 tions [19, 41]. However, current INR-based video encoders lack this feature (see  
147 discussion in [8, 10], for example), hindering their widespread usage.

148 **Video Retrieval** is an essential process in the digital media landscape,  
149 where the objective is to efficiently search and extract specific video content from



**Fig. 2:** We propose a new framework for video INR models by decoupling the spatial and temporal aspects of modeling. Our framework consists of auto-decoder based learnable latents that modulate the base network using a hypernetwork, via low-rank modulation. Once encoded, the resulting latents act as a proxy for the underlying weights of the representation. On the right, we show the use of these latents for additional tasks like video interpolation. By aligning these latents to the embedding space of foundational models like CLIP and VideoLlama, we also perform retrieval and chat.

expansive datasets. The complexity of understanding and indexing diverse video content has traditionally posed significant challenges. However, with the advent of machine learning-based methods, there has been a remarkable improvement in both the accuracy and efficiency of video retrieval systems [1, 3, 26]. These advances are limited to systems requiring an additional model, which can act as a burden on the system as they do not compress the data.

### 3 Approach

#### 3.1 Background

Implicit Neural Representations parameterize a function,

$$f_\theta : X \rightarrow Y \quad \text{where} \quad X = \{(x_i, y_i) | 0 \leq x_i \leq W, 0 \leq y_i \leq H\}$$

represents the coordinate space, with *height* H and *width* W, and Y represents the underlying signal. In the standard case of an RGB signal, Y can be represented as

$$Y = \{(R, G, B) | R, G, B \in [0, 255]\}$$

This parametrization is usually trained with a standard MSE-loss, where we try to minimize the MSE-loss  $\|f_\theta(X) - Y\|_2$ . For a given video  $V \in R^{N \times H \times W \times 3}$  containing  $N$  frames, [42] represents them as pixels moving across time, i.e.,

$$f_\theta(x, y, t) = Y_t$$

where  $T_i$  denotes the boundary of a step. Other formulations exist which learn frame-based [9] or patch-based [28] representation, yet in each of these formulations, the focus is on representing the underlying data, with the added motivation of compressing it. However, none of these systems are designed with the goal of making these representations,  $f_\theta$ , useful for downstream tasks. Instead, we utilize a learnable latent,  $z$ , as a part of an auto-decoder framework, along with a hypernet  $h$  to not only compress but to create useful representations.

$$f_\theta((x, y) | \theta_t) = Y_t \quad \theta_t = h(z_t) \quad (1)$$

The resulting latent  $z$  can be used for various downstream tasks like interpolation and retrieval, as we show in our work.

### 3.2 Latent-INR

Directly predicting the weights  $\theta$  of the base network  $f$ , using the hypernet  $h$ , is expensive, parameter-heavy, and unsuitable for compression. Hence, we follow [45] [37] and instead predict low-rank matrices, which are then applied to the base network weights. This type of modulation acts as a form of subnetwork selection, analogous to systems proposed in [17] [34]. For a base network  $f$  with  $L$  layers, our formulation now looks like

$$\begin{aligned} f_\theta((x, y) | \theta_t^{l_1}, \theta_t^{l_2} \dots \theta_t^{l_L}) &= Y_t \\ \theta_t^l &= \sigma(P^l \times Q^l) \circ \theta^l \quad h_l(z_t) = [P^l, Q^l] \end{aligned} \quad (2)$$

where  $\theta^l$  represents the weights of the  $l$ -th layer and  $\theta_t^l$  denotes the modulated weights for frame  $t$ . Here,  $\sigma$  signifies an activation function on the matrix-product of low rank matrices  $P^l, Q^l$ , which are of dimensions  $R^{K \times r}$  and  $R^{r \times K}$  where  $K$  is the width of the base network  $f_\theta$  and rank  $r \ll K$ . These matrices are responsible for adjusting the weights  $\theta_l$  as dictated by the corresponding hypernetwork  $h_l$ . Note that all hypernetworks use the same latent  $z_t \in R^D$  as input. The rank  $r$  and the number of modulated layers essentially act as hyperparameters that control the compression-performance trade-off.

### 3.3 Model architecture

In our experiments, both the base network  $f_\theta$  and hypernetworks  $h_l$  are feedforward MLP's that take in a coordinate input. Following [28], we also propose a variation to the base network with an additional convolutional up-sample block, which accepts coordinates of centroids as input and gives frame patches as output. We use the standard ReLU for base network and tanh for the hypernetwork as the respective non-linearities. The latents  $Z$  are initialized to be a standard normal with small variance, as we found empirically that this made the convergence faster. The complete model architecture is presented in Figure 2. For more details, see Appendix.

### 206 3.4 Model Compression

207 We train this entire system end-to-end with MSE-loss as the objective function.  
 208 Once trained, we apply a standard quantization to all network parameters, fur-  
 209 ther reducing the required storage. Given  $\phi$ , a flattened parameter tensor, we  
 210 transform it according to the following equations

$$211 \quad \phi_i = \left\lceil \frac{\phi_i - \phi_{min}}{2^b} \right\rceil \quad \text{scale} = \frac{\phi_{max} - \phi_{min}}{2^b} \quad (3)$$

212 where the  $\lceil \cdot \rceil$  (round) operation converts its argument to the nearest integer  
 213 as dictated by bit width  $b$  of the quantization process. We also store the scale,  
 214  $\phi_{max}$ ,  $\phi_{min}$  and the parameter shapes. These quantized values for all parameters  
 215 are concatenated and further compressed using Huffman encoding.

### 216 3.5 Interpolation

217 Given a video of  $N$  frames and a scale  $\alpha$ , the task of interpolation involves  
 218 creating  $\alpha \cdot N$  coherent frames that essentially increase the FPS of the video.  
 219 Once we encode a video using our framework, we do linear interpolation on  
 220 the frame latents  $\{z_t\}$  and pass the resulting latent through the hypernetwork.  
 221 This gives us the weight modulation required in the INR, and the updated base  
 222 network is used to obtain the interpolated frames.

$$223 \quad z_{inter} = \beta_i \cdot z_t + (1 - \beta_i) \cdot z_{t-1} \quad Y_{inter} = f_\theta(X; h(z_{inter})) \quad (4)$$

224 where,

$$225 \quad \beta_i \in \left[ \frac{1}{\alpha}, \frac{2}{\alpha}, \dots, \frac{\alpha-1}{\alpha} \right]$$

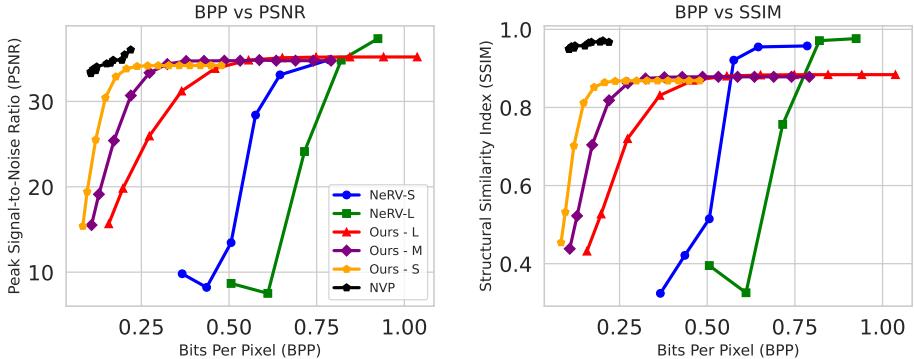
226 essentially generating  $\alpha - 1$  frames between any two given frames. We train with  
 227 held out frames and show results for  $\alpha \in \{2, 4, 8\}$ .

### 228 3.6 Downstream Tasks

229 **Retrieval.** Video retrieval involves searching and retrieving videos or clips from  
 230 a large database based on similarity to given user search queries that are usually  
 231 in the form of text. This can be viewed as a function  $R$  mapping query  $q$  to a  
 232 set of corresponding videos  $V$ .

$$233 \quad R : q \rightarrow V \quad (5)$$

234 The function  $R$  can use a similarity measure like cosine, euclidean, or nearest  
 235 neighbors to retrieve closest matches. We encode a dataset of videos using our  
 236 Latent-INR framework and use the resulting trained latents as our frame level  
 237 representation. To ensure that the latents share the same space as the text  
 238 queries, we add a cosine similarity loss between the latents and the CLIP image



**Fig. 3:** We plot the rate distortion curves on PSNR and SSIM to compare compression with other methods. We observe that our large model achieves comparable PSNR to current SOTA [22]. Note that, while not plotted here, our decoding FPS is superior. Additional per-video results are available in the Supplementary.

embeddings of the corresponding frames. Our loss function during encoding now becomes:

$$L = L_{MSE} + \lambda \cdot L_{clip}(Z_t, Z_t^{clip}) \quad (6)$$

where  $Z_t^{clip}$  is the CLIP Image embedding of the input frame and  $\lambda$  controls the strength of this loss. We set the value of  $\lambda$  to be 0.01 in all our experiments.

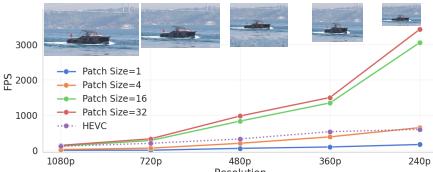
**Chat.** We modify the formulation from retrieval slightly, aligning our dictionary of features to VideoLlama [56] instead of CLIP. Since the shapes are not compatible, we treat our latents as tokens and project the dimension to match the VideoLlama space. With this, we are able to integrate with the powerful LLM, substituting our latents for the raw video input tokens. We can then perform any task that VideoLlama can, in particular question answering and captioning. We wish to emphasize that our latents are flexible – we can align well with any large model, such that to improve the VQA or captioning performance, one would only need to align with a more powerful or efficient model.

## 4 Experiments

### 4.1 Video Compression

We perform comparative analysis for video compression on the standard Ultra Video Group (UVG) dataset [31].

This dataset comprises seven high-quality videos, each featuring diverse scenes shot at 120fps over a duration of five seconds. While most videos contain 600 frames, the ‘shakendry’ video is an exception with 300 frames, all at a resolution of 1080x1920. To assess the visual quality, we use standard metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity index (SSIM). For measuring the storage efficiency of these methods, we use the standard bits



**Fig. 4:** With the same model, we can perform inference at any resolution, with speeds competitive or beating HEVC. We show sample frames for each resolution.



**Fig. 5:** We achieve high quality reconstruction and are able to reproduce even the finer details like water dountains and the hair on the horse.

per pixels (BPP) as our metric. As mentioned earlier, we use feedforward MLPs for both the base network  $f_\theta$  and hypernetworks  $h_l$ . The base network consists of 6 layers with layer size of 512 and each hypernetwork that modulates a selected layer has one hidden layer of size 128 with tanh non-linearity, followed by the output layer. In the case where we use patch centroids as inputs, we add a convolutional layer followed by a pixel-shuffle [27] for upsampling.

We use hash-grids [32] for positional encoding due to their high quality reconstruction, although it should be noted we can use other schemes, such as Fourier features [50] to exchange some quality for faster training (see Appendix). We compare our method against NeRV [9] and NVP [22], with each of them encoding a video per model, and the results are presented in Figure 3. We observe that compression from our framework is comparable to baselines at similar bpp ranges, in addition to the other downstream benefits it offers.

Due to our architecture, we are also able to operate in a novel paradigm, “**any-resolution inference**.” Without changing the network architecture at all, we can decode the video at arbitrary smaller resolutions, as well as at higher resolutions (super-resolution) by leveraging the continuous resolution property of our hash grids and MLPs. We show our FPS decoding at various resolutions in Figure 4, although it should be noted that HEVC, the standard codec we compare to, must encode separately for every resolution while we can store all in the same model. We show some sample reconstructions in Figure 5 to showcase our method’s fidelity.

## 4.2 Video Interpolation

In our framework, learned latents serve as a mapping for the model weights, enabling valid frame outputs upon interpolation. We conduct experiments on two datasets: the “big buck bunny sequence” and a selection of ten videos from the Taichi test set. Frames are held out at a scale stride  $\alpha$  during encoding. During testing, we interpolate the resulting latents on the held out frames and evaluate their performance.

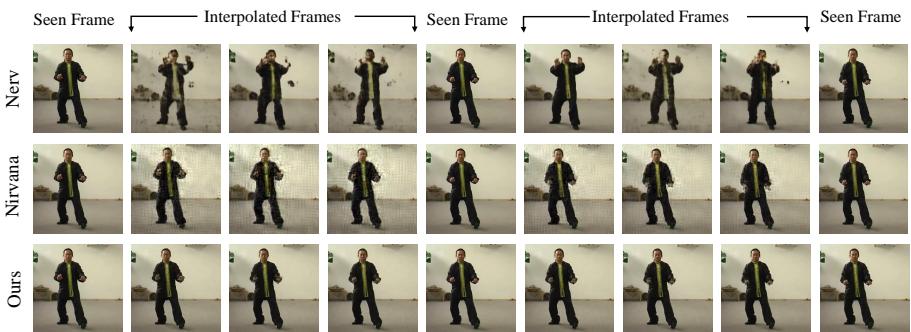
We use the same INR models utilized for compression as our baselines, with a reduction in network layer size and modulating mask rank. While NeRV [9] and NVP [22] interpolate time positions used as input, NIRVANA interpolates

**Table 1:** Interpolation Performance (PSNR), for different scale strides ( $\alpha$ ).

| Dataset | $\alpha$ | NeRV  | NIRVANA | NVP   | Ours         |
|---------|----------|-------|---------|-------|--------------|
| Bunny   | 2        | 15.92 | 19.14   | 20.10 | <b>33.17</b> |
|         | 4        | 15.43 | 18.90   | 19.11 | <b>28.08</b> |
|         | 8        | 13.68 | 18.67   | 18.08 | <b>25.88</b> |
| TaiChi  | 2        | 16.91 | 18.19   | 19.33 | <b>35.13</b> |
|         | 4        | 17.14 | 17.71   | 18.52 | <b>31.84</b> |
|         | 8        | 15.72 | 16.21   | 17.7  | <b>27.72</b> |

**Table 2:** Reconstruction and retrieval ablations of CLIP on MSR-VTT.

| CLIP $\lambda$ | Reconstruction |      | Retrieval (T2V) |      |  |
|----------------|----------------|------|-----------------|------|--|
|                | PSNR           | R@1  | R@5             | R@10 |  |
| 0.0            | 30.03          | 0.1  | 0.3             | 0.8  |  |
| 1e-3           | 29.83          | 28.4 | 50.8            | 60.6 |  |
| 1e-2           | 29.46          | 30.2 | 52.4            | 61.0 |  |
| 1e-1           | 28.93          | 29.7 | 51.5            | 61.8 |  |
| 1.0            | 28.61          | 30.2 | 51.4            | 61.3 |  |

**Fig. 6:** We compare interpolation with Latent-INR to NVP and NIRVANA. We find that our method has less artifacts and smoother motion in the interpolated frames.

the weights. In Table 1, we observe that while other INR methods fail to produce perceptual frames at scale of 2, our model can give reasonable interpolations even at a scale of 8. We confirm this qualitatively also, by inspecting interpolated frames such as those shown in Figure 6. Our outputs have noticeably fewer artifacts, and while imperfect, handle the motion better. Compared to other video INR methods, our approach of using learnt latents facilitates the model to have an internal representation of the video content.

### 4.3 Downstream Tasks

#### Retrieval

To showcase the flexibility of our latents, we align them with CLIP and evaluate their performance on standard retrieval tasks. We utilize the validation set of COIN dataset [51] and a subset of Howto100m dataset to evaluate performance. We first encode each video in our split using our Latent-INR framework with a loss that encourages the latents to be closer to the CLIP-Image embeddings of the frames, in addition to the standard reconstruction loss. We consider two distinct problems – retrieval of the correct class across all videos and retrieval

**Table 3:** Class and segment retrieval. Our method often exceeds CLIP performance.

| Dataset    | Method | Class Level  |              |              | Segment Level |              |              |
|------------|--------|--------------|--------------|--------------|---------------|--------------|--------------|
|            |        | R@1          | R@5          | R@10         | R@1           | R@5          | R@10         |
| COIN       | CLIP   | 31.60        | 44.70        | <b>50.70</b> | <b>6.60</b>   | 13.10        | 16.50        |
|            | Ours   | <b>34.40</b> | <b>45.10</b> | 50.50        | 6.40          | <b>13.30</b> | <b>17.00</b> |
| HowTo100m* | CLIP   | <b>31.58</b> | 36.84        | 47.37        | 21.13         | 37.32        | 40.85        |
|            | Ours   | 31.58        | <b>42.11</b> | <b>47.36</b> | <b>23.24</b>  | <b>43.67</b> | <b>48.60</b> |

**Table 4:** Whole video retrieval. Our method matches CLIP performance.

| Dataset      | Method | Text to Video |              |              | Video to Text |              |              |
|--------------|--------|---------------|--------------|--------------|---------------|--------------|--------------|
|              |        | R@1           | R@5          | R@10         | R@1           | R@5          | R@10         |
| MSR-VTT      | CLIP   | 30.10         | 51.50        | <b>61.50</b> | 24.70         | 49.30        | <b>61.90</b> |
|              | Ours   | <b>30.20</b>  | <b>52.40</b> | 61.10        | <b>25.40</b>  | <b>49.90</b> | 61.70        |
| ActivityNet* | CLIP   | 38.4          | <b>74.8</b>  | <b>86.6</b>  | <b>36.2</b>   | <b>73.6</b>  | <b>84.8</b>  |
|              | Ours   | <b>38.5</b>   | 73.9         | 86.4         | 36.1          | 73.5         | 84.7         |



**Fig. 7:** Nearest Neighbours for segment-level matching of sample queries from COIN validation set. The green boxes denote the true positives and the red ones are false positives. We show the inner product similarity between the image and the corresponding query inside the green boxes at the bottom of each image

of the correct segment within a video. These two use cases cover both ends of the spectrum, from localizing an event in a given video to searching for similar events across videos. We utilize the standard recall at K, where we have selected  $k \in [1, 5, 10]$  to evaluate the efficacy of our method. The results are presented in Table 3. We can see that our method matches CLIP in its retrieval performance and even exceeds it in some cases. The qualitative results are presented in Figure 7, where we visualize the top 5 nearest neighbours of the text query that map to trained latents across all videos. Further results can be found in the supplementary.

We even find that our method can perform whole-video retrieval on MSR-VTT [55] and a custom 1,000 video sample from the ActivityNet Captions [23] ‘val-1’ split. We average-pool both our features and CLIP features (similar to [6]) and use CLIP features computed on video captions. In Table 4 we find that our retrieval is quite competitive to retrieval using the CLIP features themselves, showing that the learnt latents have similarly good averaging and summarizing



**Fig. 8:** Latent-INR LLM. We show results for aligning our learned latents to a VideoLlama model, which allows for interactive chat. We show a success case (left) and failure case (right) each for a generic (top) and more targeted prompt (bottom).

326 properties even over longer (180 seconds) videos, as well as alignment even to  
 327 the paragraph-length captions used in ActivityNet.

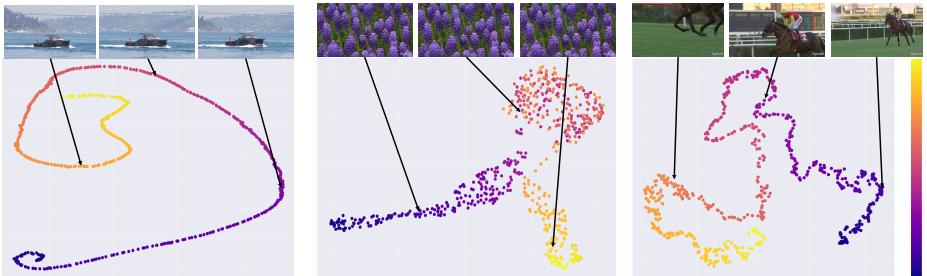
### 328 Video-based Chat

329 We evaluate the performance of our trained latents, when aligned to inter-  
 330 mediate VideoLlama features. This alignment enables access to the full scope of  
 331 text chat with video understanding. We show a sample of such results, in the  
 332 form of text and video prompts with text response, in Figure 8. These results  
 333 show the LLM is able to understand video inputs when provided in the form  
 334 of INR latents rather than raw video tokens. While not perfect, we infer the  
 335 majority of the shortcomings of this system are primarily the fault of the LLM  
 336 we align to.

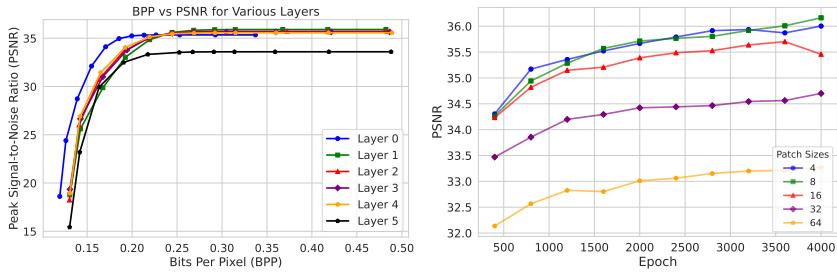
337 Furthermore, on the basis of our success in aligning with CLIP and now  
 338 VideoLlama, we believe our latents can be aligned to any representation. So, for  
 339 more powerful chat, one simply needs to align to a more powerful chatbot. We  
 340 thus provide these results two purposes. First, we show our model's capability  
 341 to power efficient open-ended captioning and question answering, while still re-  
 342 taining reconstruction capabilities. Second, we point to the immense potential of  
 343 our model (or a similar paradigm) to continue to be leveraged with such models  
 344 as they expand in their size and performance.

### 345 4.4 Visualizing Trained Latents

346 The trained latents, representing the modulated frames, offer intriguing insights  
 347 when visualized in a reduced dimensional space. Utilizing Uniform Manifold Ap-  
 348 proximation and Projection (UMAP) [29] we project the embeddings  $Z_t$  into  
 349 a 2D space, allowing for an intuitive interpretation of their relationships. In  
 350 Figure 9, we plot the UMAP for three distinct videos from the UVG dataset:  
 351 'Bosphore,' 'Honeybee,' and 'Jockey,' each offering unique characteristics for  
 352 examination. 'Bosphore', characterized by its slow-moving object and relatively



**Fig. 9:** We visualize the trained latents  $Z_t$  projected to 2D using UMAP. We show that the trained latents from our framework capture meaningful semantics of the underlying data. Left to right: Latents for Bosphore, Honeybee and Jockey from UVG dataset. Dark to Light color indicates frame numbers ranging from 0 to 600.



**Fig. 10:** Ablations to study the effect of layer modulations in the hypernetwork and the effect of patch size on reconstruction quality (PSNR).

static foreground, exhibits a smooth latent trajectory in the 2D space. This smoothness reflects the minimal variance in frame content, suggesting that our method effectively captures the subtle dynamics of the scene. In contrast, the ‘Honeybee’ video, with its repetitive frames, results in latents that cluster tightly together, signifying our model’s ability to recognize and encode repetitive patterns efficiently. The most dynamic of the three, ‘Jockey’, presents a more complex scenario with rapid changes in both the foreground and background. Here, the latents form clusters around similar scenes, yet maintain a discernible trajectory through the 2D space. These visualizations illustrate the semantic richness embedded within the latents obtained from our framework even when trained only for compression.

## 5 Ablation Studies

**CLIP  $\lambda$ .** We investigate the impact of the large model alignment weighting term on both reconstruction and retrieval for MSR-VTT. In Table 2, we find

367 that PSNR decreases slightly as  $\lambda$  increases. However, the retrieval performance  
 368 seems to saturate at  $\lambda = 0.01$ . So, we suggest not tuning the  $\lambda$  too high for any  
 369 application, given the diminishing returns.

370 **Layer Modulations.** In our approach, we have separate hypernetworks that  
 371 modulated the selected layers. To evaluate the importance of each, we design an  
 372 experiment where they are modulated in isolation. We use the same setup as  
 373 the compression experiments with the modulating mask rank fixed at 20 for all  
 374 models. In Figure 10, we can clearly see that the first few layers have a significant  
 375 impact on the encoding performance. This matches the observations from [21]  
 376 about the out sized impact of first few layers while modulating INRs.

377 **Patch Size.** Scaling to higher-resolution videos can be memory-intensive. This  
 378 is particularly true when employing memory-demanding positional encoding  
 379 schemes such as hash-grids [32]. To investigate this aspect further, we experiment  
 380 with models that process centroids of fixed-size patches, directly predicting the  
 381 corresponding frame patches, to save memory. From Figure 10 we observe that  
 382 smaller patches have similar performance, with a steep drop observed for higher  
 383 patch sizes.

## 384 6 Conclusion

385 **Limitations.** Our latents are somewhat restricted by the quality of the em-  
 386 beddings they are aligned to. Additionally, more work is still required to match  
 387 standard codecs in terms of storage and encoding time, in spite of impressive  
 388 gains in terms of quality and decoding speed. Future work could both improve  
 389 the compression, and leverage more powerful vision models.

390 **Broader Impacts.** Our method for simultaneously compressing and learning  
 391 useful features for recognition could reduce the need to decode videos for these  
 392 tasks and thus save computational resources, cutting costs and helping the envi-  
 393 ronment. However, work that advances performance for compression and recogni-  
 394 tion also has applications in surveillance and warfare.

395 In this work, we propose a new framework, Latent-INR, where we decou-  
 396 ple the temporal aspect from the spatial into a dictionary of learnable latents.  
 397 These auto-decoder based learnable latents modulate the layers of the base INR  
 398 network via low-rank modulation using hypernetworks. Latent-INR is not only  
 399 well-suited to video compression, but the resulting latents learn an internal rep-  
 400 resentation of the data they encode that lends itself to SOTA interpolation for  
 401 video INRs. Additionally, we also augment these latents by training them to  
 402 be aligned with CLIP and VideoLlama, which allows us to bring the power of  
 403 foundational models to compressed representations, and perform retrieval and  
 404 chat-based applications like captioning and question answering. Our work thus  
 405 opens up new possibilities of research in the implicit neural space where down-  
 406 stream tasks can be performed by these model without the need for decoding.

# 407 Latent-INR: A Flexible Framework for Implicit 408 Representations of Videos with Discriminative 409 Semantics

## 410 Supplementary Material

### 411 A Network Architecture

412 **Base Network:** We use an MLP with 10 layers, width of 512 and ReLU non-  
 413 linearity as our base network  $f_\theta$ .

414 **Hypernetwork:** All hypernetworks  $h^l$  used to modulate a layer  $l$  of the base  
 415 network have 3 layers with a hidden dimension of 512 and tanh as non-linearity.  
 416 Unless specified, we only modulate the first hidden layer of the base network.

417 **Latents:** Each latent  $Z_t$  corresponding to a frame has a dimension of 512 and  
 418 is initialized to be standard Gaussian before training. We set our learning rate  
 419 as 5e-4 and used the standard Adam optimizer without any weight decay.

### 420 B Compression

#### 421 B.1 Fourier Features

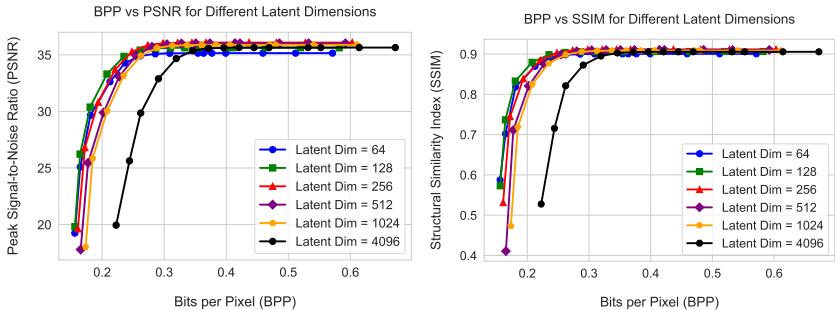
422 We use the multiresolution hash grid for positional encoding in all our models.  
 423 In table 5 we show results for full coordinate resolution using fourier features for  
 424 positional encoding. Due to lack of a hash grid, the resulting models train upto  
 425 30% faster, but at the cost of inferior reconstruction.

#### 426 B.2 Quantization

427 Instead of quantizing all components equally, we notice that retaining the latents  
 428 and the base network at full precision provides better reconstruction at negligible  
 429 additional storage.

#### 430 B.3 Effect of latent dimension

431 To study the effect of latent dimension on compression, we train models by  
 432 varying it and encode the “bosphore” video from UVG dataset. The results are  
 433 presented in Figure 11. We notice that there is positive gains till dimension 512  
 434 and diminishing returns thereafter. Hence we choose that as our default latent  
 435 size in all our experiments.



**Fig. 11:** Effect of varying latent dimension across different bitrates.

## C Video Retrieval

We perform two retrieval tasks on the COIN dataset [51] - *class-level*, and *segment-level*. In both settings, we use the standard val set as the database. For *class-level*, we use the distinct video-level task names in COIN as our query set. For *segment-level*, we use the set of distinct clip-level captions in COIN as our query set. We get the CLIP ViT-B/32 text embeddings of each of these captions, and these become our query vectors. For database vectors, we use the per-frame learned latents for each video in the database. For comparison with CLIP, we replace these database vectors with the CLIP ViT-B/32 image embeddings for each frame. For *class-level* retrieval, we consider a result frame a positive match if it belongs to a video with the same class label as the queried caption. On the other hand, for *segment-level* retrieval, we consider a result frame a positive match only if it belongs to a segment with the same caption as the query. Further, this search is done over all videos. We use FAISS [20] as our retrieval implementation and use *coseine similarity* as the distance metric.

We perform whole-level video retrieval as described in the main paper. For text, we use CLIP to compute a feature for the paragraph caption. For the video, we compute a per-frame feature for CLIP, or use the learnt latents from Latent-INR. For a single video feature, we then average these per-frame features. We normalize all features, and perform retrieval by finding the closest embeddings using dot product similarity. Both text-to-video and video-to-text are performed in the same manner, the only difference being which features are used as query and key.

Fig.12, shows the retrieval results on the COIN data in the *segment-level* setting. It can be seen that a majority of failure cases could be attributed to visual similarity across different tasks when seen at an individual frame level.

## D Video Chat

We interface our latents with learned features from Video-Llama [56] to enable interactive chat with the compressed videos. In [56], the  $N$  video frames are



**Fig. 12:** Nearest Neighbours for segment-level matching of sample queries from COIN validation set. The green boxes denote the true positives and the red ones are false positives. We show the inner product similarity between the image and the corresponding query inside the green boxes at the bottom of each image



**Fig. 13:** Additional results for Latent-INR interface with Video-LLM.

passed through a ViT based visual encoder to extract features of size  $k \times d$  per frame. These are then passed through a Query Former [24] to obtain a unified video representation of size  $k_v \times d_v$ . This tensor is then passed to a trainable MLP layer before aligning with an LLM of our choice (LLama-2 [52] in our models).

We align our latents  $Z$  with these per-video features of size  $k_v \times d_v$  using a linear projection layer which is trained end to end. The loss function is slightly modified to incorporate a cosine similarity loss between the terms.

$$L = L_{MSE} + \lambda \cdot L_{cos}(F_t, F_t^{\text{V-LLM}}) \quad (7)$$

where  $F_t$  is the predicted feature and  $F_t^{\text{V-LLM}}$  is the corresponding Video-LLama extracted features. We show additional results of the interactive chat in Figure 13.

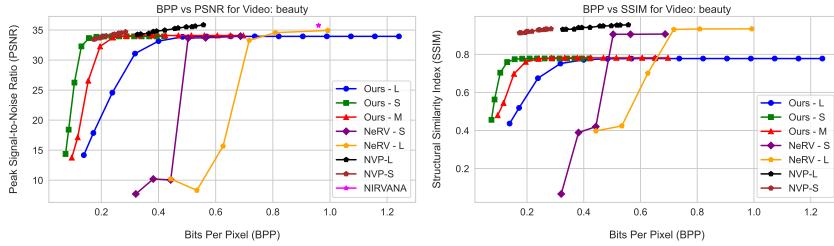
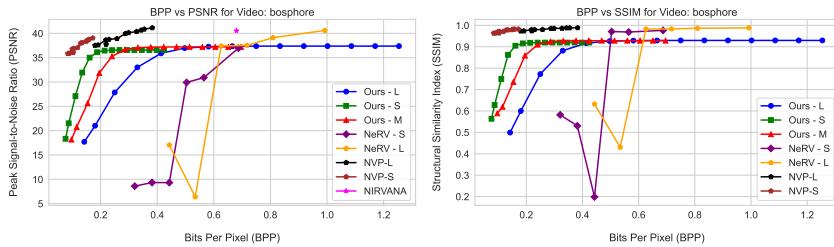
**Table 5:** Fourier Feature Models

| Method                   | PSNR  | BPP  |
|--------------------------|-------|------|
| Ours- Fourier - <i>S</i> | 31.99 | 0.31 |
| Ours- Fourier - <i>M</i> | 33.69 | 0.62 |
| Ours- Fourier - <i>L</i> | 33.19 | 0.84 |

477

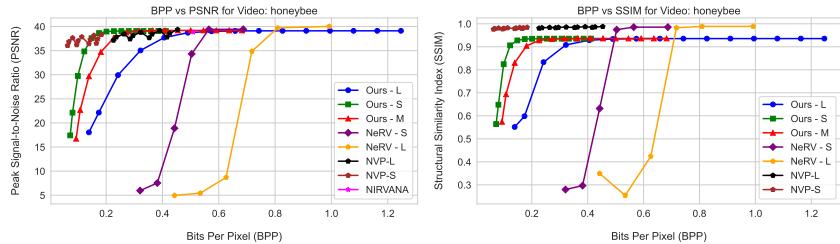
## E Video-wise results

477

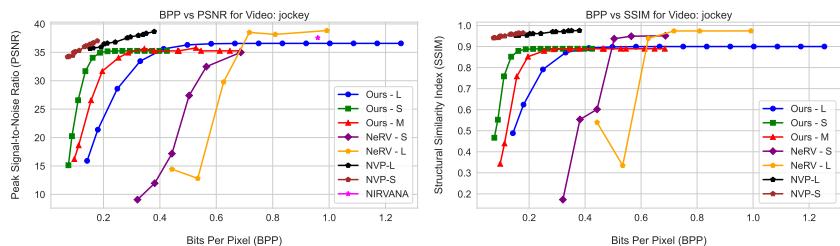
**Fig. 14:** BPP vs. PSNR, SSIM for beauty.**Fig. 15:** BPP vs. PSNR, SSIM for bosphore.478  
479  
480  
481  
482  
483  
484

We plot the results for each video from UVG dataset [31] in Figures 14, 15, 16, 17, 18, 19, and 20. We show three versions of our model based on the dimension of the *low-rank* modulating matrix. The *Ours-s*, *Ours-m*, and *Ours-l* correspond respectively to *size* = 50, 100, 200. The *Ours-m* model achieves reasonable performance when compared to other methods, and at the same time can do the downstream tasks of interpolation and retrieval which none of the compared methods can.

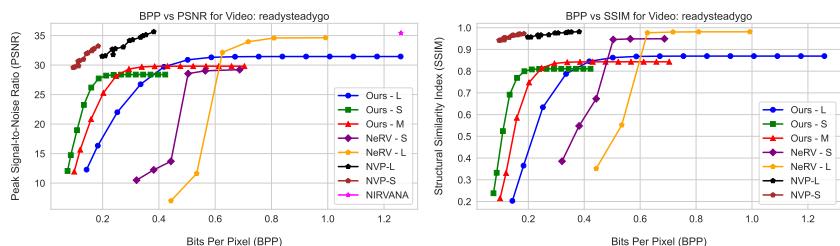
478  
479  
480  
481  
482  
483  
484



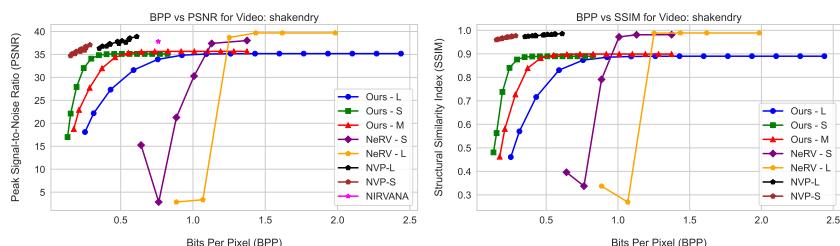
**Fig. 16:** BPP vs. PSNR, SSIM for honeybee.



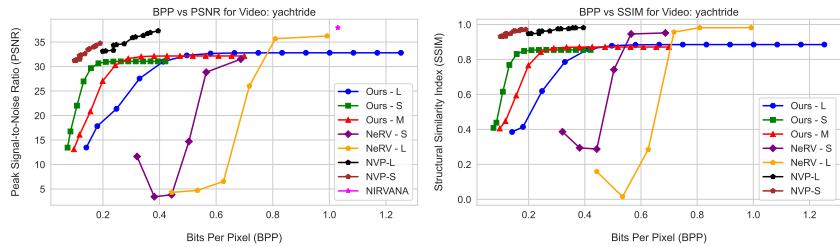
**Fig. 17:** BPP vs. PSNR, SSIM for jockey.



**Fig. 18:** BPP vs. PSNR, SSIM for readysteadygo.



**Fig. 19:** BPP vs. PSNR, SSIM for shakendry.



**Fig. 20:** BPP vs. PSNR, SSIM for yachtride.

485

## References

485

- 486 1. A scheme for shot detection and video retrieval using spatio temporal features.  
487 International Journal of Recent Technology and Engineering (2019), <https://api.semanticscholar.org/CorpusID:241499090> 5  
488
- 489 2. Bai, Y., Dong, C., Wang, C., Yuan, C.: Ps-nerv: Patch-wise stylized neural repre-  
490 sentations for videos. In: 2023 IEEE International Conference on Image Processing  
491 (ICIP). pp. 41–45. IEEE (2023) 4  
492
- 493 3. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video  
494 and image encoder for end-to-end retrieval. 2021 IEEE/CVF International Con-  
495 ference on Computer Vision (ICCV) pp. 1708–1718 (2021), <https://api.semanticscholar.org/CorpusID:232478955> 5  
496
- 497 4. Ballé, J., Minnen, D.C., Singh, S., Hwang, S.J., Johnston, N.: Variational image  
498 compression with a scale hyperprior. ArXiv **abs/1802.01436** (2018), <https://api.semanticscholar.org/CorpusID:3611540> 2  
499
- 500 5. Bauer, M., Dupont, E., Brock, A., Rosenbaum, D., Schwarz, J., Kim, H.: Spatial  
501 functa: Scaling functa to imagenet classification and generation. arXiv preprint  
502 arXiv:2302.03130 (2023) 4  
503
- 504 6. Buch, S., Eyzaguirre, C., Gaidon, A., Wu, J., Fei-Fei, L., Niebles, J.C.: Revisiting  
505 the "video" in video-language understanding (2022) 11  
506
- 507 7. Cao, L., Jiang, A., Li, W., Wu, H., Ye, N.: Oodhdhr-codec: Out-of-distribution  
508 generalization for HDR image compression. In: Thirty-Sixth AAAI Conference on  
509 Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Appli-  
510 cations of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educa-  
511 tional Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22  
512 - March 1, 2022. pp. 158–166. AAAI Press (2022). <https://doi.org/10.1609/AAAI.V36I1.19890>, <https://doi.org/10.1609/aaai.v36i1.19890> 2  
513
- 514 8. Chen, H., Gwilliam, M., Lim, S.N., Shrivastava, A.: Hnerv: A hybrid neural rep-  
515 resentation for videos. In: Proceedings of the IEEE/CVF Conference on Computer  
516 Vision and Pattern Recognition. pp. 10270–10279 (2023) 4  
517
- 518 9. Chen, H., He, B., Wang, H., Ren, Y., Lim, S.N., Shrivastava, A.: Nerv: Neural  
519 representations for videos. Advances in Neural Information Processing Systems  
520 **34**, 21557–21568 (2021) 2, 4, 6, 9  
521
- 522 10. Chen, H., Matthew, G., He, B., Lim, S.N., Shrivastava, A.: Cnerv: Content-  
523 adaptive neural representation for visual data. In: BMVC (2022) 4  
524
- 525 11. Chen, Y., Wang, X.: Transformers as meta-learners for implicit neural representa-  
526 tions. In: European Conference on Computer Vision (2022) 4  
527
- 528 12. Chen, Y., Murherjee, D., Han, J., Grange, A., Xu, Y., Liu, Z., Parker, S., Chen, C.,  
529 Su, H., Joshi, U., et al.: An overview of core coding tools in the av1 video codec.  
530 In: 2018 picture coding symposium (PCS). pp. 41–45. IEEE (2018) 2  
531
- 532 13. Chiang, P.Z., Tsai, M.S., Tseng, H.Y., Lai, W.S., Chiu, W.C.: Stylizing 3d scene  
533 via implicit representation and hypernetwork. In: Proceedings of the IEEE/CVF  
534 Winter Conference on Applications of Computer Vision. pp. 1475–1484 (2022) 4  
535
- 536 14. Dupont, E., Kim, H., Eslami, S.M.A., Rezende, D.J., Rosenbaum, D.: From data to  
537 functa: Your data point is a function and you can treat it like one. In: Interna-  
538 tional Conference on Machine Learning (2022), <https://api.semanticscholar.org/CorpusID:249395684> 4  
539
- 540 15. Ehrlich, M., Davis, L.S.: Deep residual learning in the jpeg transform domain.  
541 In: Proceedings of the IEEE/CVF International Conference on Computer Vision  
542 (ICCV) (October 2019) 2  
543

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

- 534 16. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation  
535 of deep networks. In: International Conference on Machine Learning (2017), <https://api.semanticscholar.org/CorpusID:6719686> 4 534  
536 17. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable  
537 neural networks. arXiv preprint arXiv:1803.03635 (2018) 6 535  
538 18. He, B., Yang, X., Wang, H., Wu, Z., Chen, H., Huang, S., Ren, Y., Lim, S.N., Shri-  
539 vastava, A.: Towards scalable neural representation for diverse videos. In: Proceed-  
540 ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.  
541 pp. 6132–6142 (2023) 4 540  
542 19. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E.G., Kautz, J.: Super  
543 slomo: High quality estimation of multiple intermediate frames for video inter-  
544 polation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recog-  
545 nition pp. 9000–9008 (2017), <https://api.semanticscholar.org/CorpusID:10817557> 4 545  
546 20. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE  
547 Transactions on Big Data 7(3), 535–547 (2019) 2 546  
548 21. Kim, C., Lee, D., Kim, S., Cho, M., Han, W.S.: Generalizable implicit neural  
549 representations via instance pattern composers. In: Proceedings of the IEEE/CVF  
550 Conference on Computer Vision and Pattern Recognition. pp. 11808–11817 (2023)  
551 4, 14 550  
552 22. Kim, S., Yu, S., Lee, J., Shin, J.: Scalable neural video representations with learn-  
553 able positional features. Advances in Neural Information Processing Systems 35,  
554 12718–12731 (2022) 2, 4, 8, 9 553  
555 23. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C.: Dense-captioning events  
556 in videos (2017) 11 555  
557 24. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: bootstrapping language-image pre-  
558 training with frozen image encoders and large language models. In: Proceedings  
559 of the 40th International Conference on Machine Learning. ICML’23, JMLR.org  
560 (2023) 3 559  
561 25. Li, Z., Wang, M., Pi, H., Xu, K., Mei, J., Liu, Y.: E-nerv: Expedite neural video  
562 representation with disentangled spatial-temporal context. In: European Confer-  
563 ence on Computer Vision. pp. 267–284. Springer (2022) 4 562  
564 26. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: Clip4clip: An  
565 empirical study of clip for end to end video clip retrieval. Neurocomputing 508,  
566 293–304 (2021), <https://api.semanticscholar.org/CorpusID:233296206> 5 565  
567 27. Luo, H., Chen, Y., Zhou, Y.: An extremely effective spatial pyramid and pixel shuf-  
568 fle upsampling decoder for multiscale monocular depth estimation. Computational  
569 Intelligence and Neuroscience 2022 (2022), <https://api.semanticscholar.org/CorpusID:251272212> 9 569  
570 28. Maiya, S.R., Girish, S., Ehrlich, M., Wang, H., Lee, K.S., Poirson, P., Wu, P.,  
571 Wang, C., Shrivastava, A.: Nirvana: Neural implicit representations of videos with  
572 adaptive networks and autoregressive patch-wise modeling. In: Proceedings of the  
573 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14378–  
574 14387 (2023) 3, 4, 6 573  
575 29. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and  
576 projection for dimension reduction (2020) 12 574  
577 30. Mentzer, F., Toderici, G., Tschannen, M., Agustsson, E.: High-fidelity gen-  
578 erative image compression. ArXiv [abs/2006.09965](https://arxiv.org/abs/2006.09965) (2020), <https://api.semanticscholar.org/CorpusID:219721015> 2 578  
579 580 581 582

- 583 31. Mercat, A., Viitanen, M., Vanne, J.: Uvg dataset: 50/120fps 4k sequences for video  
584 codec analysis and development. In: Proceedings of the 11th ACM Multimedia  
585 Systems Conference. p. 297–302. MMSys ’20, Association for Computing Machinery,  
586 New York, NY, USA (2020). <https://doi.org/10.1145/3339825.3394937>,  
587 <https://doi.org/10.1145/3339825.3394937> 8, 4
- 588 32. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives  
589 with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15  
590 (Jul 2022). <https://doi.org/10.1145/3528223.3530127>, <https://doi.org/10.1145/3528223.3530127> 4, 9, 14
- 591 33. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry,  
592 G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning trans-  
593 ferable visual models from natural language supervision. In: International Confer-  
594 ence on Machine Learning (2021), <https://api.semanticscholar.org/CorpusID:231591445> 3, 4
- 595 34. Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A., Rastegari, M.: What’s  
596 hidden in a randomly weighted neural network? In: Proceedings of the IEEE/CVF  
597 conference on computer vision and pattern recognition. pp. 11893–11902 (2020) 6
- 598 35. Saragadam, V., LeJeune, D., Tan, J., Balakrishnan, G., Veeraraghavan, A., Bara-  
599 niuk, R.: Wire: Wavelet implicit neural representations. 2023 IEEE/CVF Con-  
600 ference on Computer Vision and Pattern Recognition (CVPR) pp. 18507–18516  
601 (2023), <https://api.semanticscholar.org/CorpusID:255749557> 4
- 602 36. Saragadam, V., Tan, J., Balakrishnan, G., Baraniuk, R.G., Veeraraghavan,  
603 A.: MINER: multiscale implicit neural representations. CoRR **abs/2202.03532**  
604 (2022), <https://arxiv.org/abs/2202.03532> 4
- 605 37. Schwarz, J.R., Tack, J., Teh, Y.W., Lee, J., Shin, J.: Modality-agnostic variational  
606 compression of implicit neural representations. arXiv preprint arXiv:2301.09479  
607 (2023) 4, 6
- 608 38. Schwarz, J.R., Teh, Y.W.: Meta-learning sparse compression networks (2022) 4
- 609 39. Sen, B., Agarwal, A., Namboodiri, V.P., Jawahar, C.: Inr-v: A continuous repre-  
610 sentation space for video-based generative tasks. arXiv preprint arXiv:2210.16579  
611 (2022) 4
- 612 40. Sen, B., Singh, G., Agarwal, A., Agaram, R., Krishna, K.M., Sridhar, S.: Hyp-nerf:  
613 Learning improved nerf priors using a hypernetwork. arXiv preprint arXiv:2306.06093  
614 (2023) 4
- 615 41. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert,  
616 D., Wang, Z.: Real-time single image and video super-resolution using an effi-  
617 cient sub-pixel convolutional neural network. 2016 IEEE Conference on Com-  
618 puter Vision and Pattern Recognition (CVPR) pp. 1874–1883 (2016), <https://api.semanticscholar.org/CorpusID:7037846> 4
- 619 42. Sitzmann, V., Martel, J.N.P., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Im-  
620 plicit neural representations with periodic activation functions (2020) 4, 5
- 621 43. Sitzmann, V., Rezhikov, S., Freeman, B., Tenenbaum, J., Durand, F.: Light field  
622 networks: Neural scene representations with single-evaluation rendering. Advances  
623 in Neural Information Processing Systems **34**, 19313–19325 (2021) 4
- 624 44. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Con-  
625 tinuous 3d-structure-aware neural scene representations. Advances in Neural In-  
626 formation Processing Systems **32** (2019) 4
- 627 45. Skorokhodov, I., Ignatyev, S., Elhoseiny, M.: Adversarial generation of continu-  
628 ous images. In: Proceedings of the IEEE/CVF conference on computer vision and  
629 pattern recognition. pp. 10753–10764 (2021) 6

- 633 46. Strümpler, Y., Postels, J., Yang, R., Gool, L.V., Tombari, F.: Implicit neural rep- 633  
634 resentations for image compression. In: European Conference on Computer Vision 634  
635 (2021), <https://api.semanticscholar.org/CorpusID:244954443> 4 635  
636 47. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency 636  
637 video coding (hevc) standard. IEEE Transactions on circuits and systems for video 637  
638 technology **22**(12), 1649–1668 (2012) 2 638  
639 48. Tack, J., Kim, S., Yu, S., Lee, J., Shin, J., Schwarz, J.R.: Learning large-scale 639  
640 neural fields via context pruned meta-learning (2023) 4 640  
641 49. Tancik, M., Mildenhall, B., Wang, T., Schmidt, D., Srinivasan, P.P., Barron, J.T., 641  
642 Ng, R.: Learned initializations for optimizing coordinate-based neural representa- 642  
643 tions. In: CVPR (2021) 4 643  
644 50. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Sing- 644  
645 hal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn 645  
646 high frequency functions in low dimensional domains. NeurIPS (2020) 4, 9 646  
647 51. Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J.: Coin: 647  
648 A large-scale dataset for comprehensive instructional video analysis (2019) 10, 2 648  
649 52. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bash- 649  
650 lykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation 650  
651 and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023) 3 651  
652 53. Wallace, G.K.: The jpeg still picture compression standard. Communications of 652  
653 the ACM **34**(4), 30–44 (1991) 2 653  
654 54. Wiegand, T., Sullivan, G., Bjontegaard, G., Luthra, A.: Overview of the h.264/avc 654  
655 video coding standard. IEEE Transactions on Circuits and Systems for Video Tech- 655  
656 nology (2003) 2 656  
657 55. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for 657  
658 bridging video and language. In: Proceedings of the IEEE Conference on Computer 658  
659 Vision and Pattern Recognition (CVPR) (June 2016) 11 659  
660 56. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual lan- 660  
661 guage model for video understanding. arXiv preprint arXiv:2306.02858 (2023), 661  
662 <https://arxiv.org/abs/2306.02858> 4, 8, 2 662  
663 57. Zhang, M., Zhang, A., McDonagh, S.G.: On the out-of-distribution generaliza- 663  
664 tion of probabilistic image modelling. In: Neural Information Processing Systems 664  
665 (2021), <https://api.semanticscholar.org/CorpusID:237431305> 2 665