Implicit Neural Representation Facilitates Unified Universal Vision Encoding

Matthew Gwilliam, Xiao Wang, Xuefeng Hu, and Zhenheng Yang

TikTok

Abstract. Models for image representation learning are typically designed for either recognition or generation. Various forms of contrastive learning help models learn to convert images to embeddings that are useful for classification, detection, and segmentation. On the other hand, models can be trained to reconstruct images with pixel-wise, perceptual, and adversarial losses in order to learn a latent space that is compatible with image generation. We seek to unify these two directions with a first-of-its-kind model that learns representations which are simultaneously well-suited for recognition and generation. We train our model as a hyper-network for implicit neural representation, which learns to map images to model weights for fast, accurate reconstruction. We further integrate our INR hyper-network with knowledge distillation to improve its generalization and performance. Beyond the novel training design, the model also learns an unprecedented compressed embedding space with outstanding performance for various visual tasks. The complete model competes with state-of-the-art results for image representation learning, enables downstream generative capabilities, and produces high-quality tiny embeddings.

Keywords: Image Representation Learning · Implicit Neural Representation · Knowledge Distillation

1 Introduction

Vision encoders, which learn to compute useful image and video representations, are foundational components of modern computer vision systems. Although such encoders may be able to solve many different tasks, in practice, we typically use them in a specialized manner according to their strengths. Models trained with text-image contrastive learning [79] lend themselves well to tasks such as retrieval and visual question answering. Models trained with image-only contrastive learning [16, 68] or self-distillation [11, 75] perform well for fine-grained image understanding, such as semantic segmentation. Variational autoencoders, which reconstruct pixels, enable image and video synthesis with diffusion and auto-regressive models. Some prior and concurrent works attempt to unify these recognition-focused and generative-focused models post-hoc [26, 58, 69, 119]. These works point to a very promising synergy between pretraining methods for recognition and generative models. A natively unified encoder's features should have good

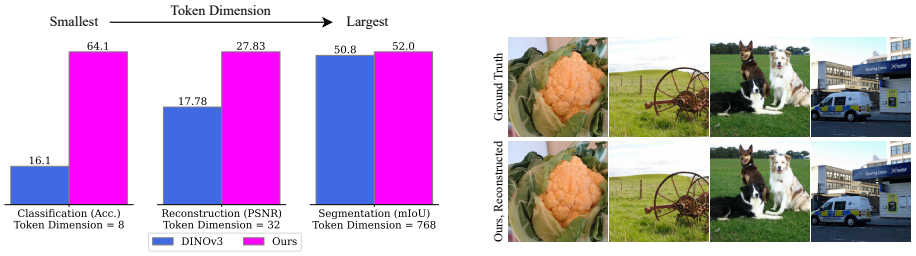

Fig. 1: (Left) Unified modeling, in terms of both tasks and token dimensions. We propose implicit neural **H**yper-networks for **U**nified **V**isual **R**epresentation, **HUVR**, with good **classification**, **reconstruction**, and **segmentation** (shown for ViT-B/16 on ImageNet, ImageNet, and ADE20K, respectively). We design our model to generate not only standard-sized tokens, but also **Tin**y **Tok**ens (TinToks). Here, the tiny embeddings of DINOv3 are generated via principal component analysis (PCA). **(Right)** Reconstruction. We unify recognition and generative task families.

high level (image classification), mid level (semantic segmentation), low level (depth estimation), and pixel level (reconstruction) information out-of-the-box.

These requirements point to hyper-networks for implicit neural representation (INR) [22, 45] as a natural candidate. An INR hyper-network is a network that takes image inputs, and predicts neural network weights (INR) as output. These output INRs take pixel coordinates as inputs, and give pixel color channel values as outputs. INR hyper-networks are similar to traditional autoencoders in the sense that their latents have good pixel representation. However, unlike convolutional VAEs, transformer-based INR hyper-networks can easily borrow encoder design, patch sizes, and latent dimensions from state-of-the-art Vision Transformer (ViT) encoders [27]. Furthermore, compared to other representation learning methods, INR hyper-networks perform compression along multiple axes, first by converting a specific image to latents, and second by learning a shared "base" INR to represent all possible images. We hypothesize this is helpful for learning high quality embeddings not only at the pixel-level, but also for low-, middle-, and high-level image information [35, 55, 85, 91].

Unified representation requires more than simply training a model whose representations have suitable semantics for any task. In practice, tasks differ from each other not just in terms of the level of information or semantics they need, but also in terms of the amount of computational resources available to solve them. Tasks like retrieval become very difficult as the amount of data increases at scale, and reducing the embedding size introduces massive savings. Thus, we propose to unify representations not just in terms of tasks, but also by designing a model that can generate both compressed and non-compressed representations.

We design our "**H**yper-network for **U**nified **V**isual **R**epresentation" (HUVR) with a novel INR hyper-network design to natively unify disparate representa-

tion learning families, particularly recognition and reconstruction, in terms of both architectural design and pretraining. HUVR encodes images both in standard size representations, as well as a smaller representation that we refer to as "**Tiny Tok**ens" (TinToks). Figure 1 shows their superior recognition and novel reconstruction abilities.

We build on the strengths of INR hyper-networks to design HUVR and its TinToks. INR hyper-networks do not natively learn good high level semantics, and the design of the latent tokens makes it difficult to recover patch-level information. We solve the gap in semantics via knowledge distillation from a pretrained vision encoder. We refactor the design of the latent tokens to naturally support both a mapping between input patches and patch tokens, as well as a global "cls" token. To create the compressed representation, the TinToks, we introduce learnable feature downsampling and upsampling layers in between the transformer backbone and the layers that produce the final INR prediction.

In summary, our key contributions are:

1. We design and train the first INR hyper-network that can match or outperform DINOv3 [88] – with ViT-B/16 HUVR achieves +0.4% for ImageNet [24] classification and +1.2 mIoU for ADE20K [120] semantic segmentation, and +4.84 PSNR for reconstruction.
2. The compressed representation TinToks, at 96x compression, can offer +41% ImageNet classification and +8.2 mIoU for ADE20K segmentation compared to a DINOv3 PCA baseline, and +1.26 PSNR compared to the Stable Diffusion VAE [81] at equal embedding size.
3. An improved design for hyper-networks for image implicit neural representations yields +2.34 PSNR even with only 10% training time on ImageNette.

2 Related Work

2.1 Implicit Neural Representation

Implicit Neural Representation. An implicit neural representation (INR) is a neural network that maps coordinates to a signal (image, video, 3D, audio, etc.) [14, 67, 70, 84, 89, 95, 103]. Due to their small size, a significant volume of research focuses on trying to leverage INRs for image and video compression [14, 29,30,46,48,51,52,65,92,116]. Methods in the NeRV [14] family use a combination of linear layers, convolutional layers, and PixelShuffle [87] operations to map frame embeddings to the RGB values of the frame for video compression [12,13, 37, 39, 47, 50, 54, 59, 83, 101, 104, 106, 115, 117, 118]. Some INR methods use grid features for positional embedding [34, 48, 50, 54, 65].

Hyper-Networks. INRs require costly per-sample training, which limits their applicability. Some try to avoid the need to train per-sample by instead training a hyper-network. The hyper-network learns to predict INR network weights that can reconstruct the input [15,22,45,53,114] or generate novel outputs [38,90,109]. Latent-INR redesigns INRs as *per-video* hyper-networks, and aligns an learnable latent for each frame with its corresponding CLIP embedding [66]. While we

also pursue INR with semantics, unlike Latent-INR, we learn a general hypernetwork that can predict INR network weights even for inputs not seen during training.

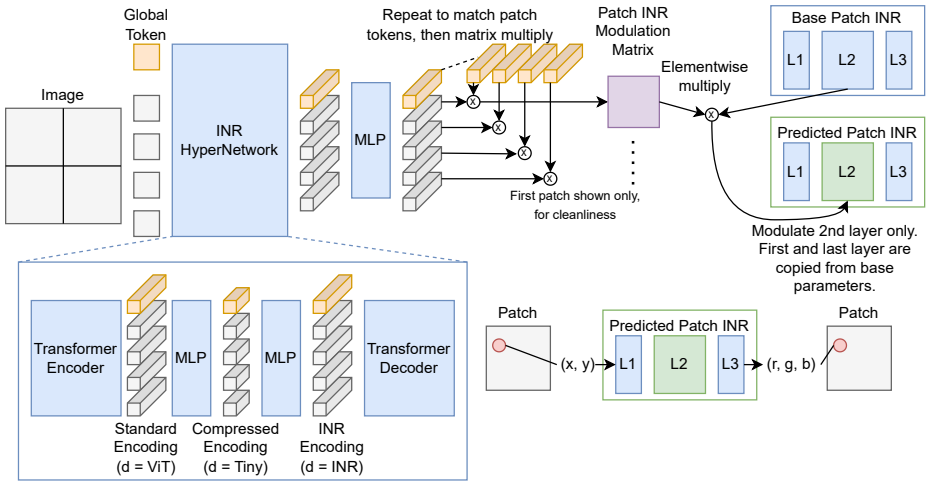

Fig. 2: Our system first patchifies an image and prepends a global (cls) token, like a standard ViT. However, unlike a standard ViT, we compress the standard encoding outputs to yield compressed encodings, which we can then upsample to predict an image INR and reconstruct the image. Both the standard and compressed encodings from our model have powerful recognition and reconstruction capabilities.

2.2 Image Representation Learning

Early methods for unsupervised image representation learning focused on restoring withheld or corrupted information [68, 74, 77, 112], but their performance lagged behind supervised learning. Contrastive learning and clustering were able to compete with supervised learning [6, 9–11, 16, 17, 19–21, 36, 57, 76, 96, 110, 122]. iBOT [121] was able to outperform these with masked image modeling [1, 4, 40, 44], and recent entries in the DINO family fuse these with a self-distillation paradigm based on knowledge distillation [42] for state-of-the-art results [75, 88]. Some methods train an image representation jointly with text modeling [41, 79, 80, 94, 111], and these perform best as vision encoders for multimodal large language models [3, 61, 62, 99, 105].

2.3 Unified Representation Learning

Some methods train models for both recognition and generation [18, 25, 26, 28, 72, 102], while other recent works rely on a pre-trained generative model, which

Table 1: Comparison with unified methods. Unlike these methods, while HUVR only provides latents for downstream generation, it can also support dense prediction and extreme compression (TinToks).

Method	INet	FID↓	Capabilities
BigBiGAN (ResNet)	56.6	21.61	Generation + Recognition
MAGE (ViT-B)	74.7	11.11	Generation + Recognition
Sorcen (ViT-B)	75.1	**9.61**	Generation + Recognition
HUVR (ViT-B)	**85.0**	24.72	Recon + Recog + Dense + Compress

is then adapted post-hoc for recognition [56,69,102]. REPA [108] provides a way to train a generative model with guidance from a recognition model, whereas we train a reconstruction model with guidance from a recognition model. However, while a generative model trained with REPA has a significant gap in recognition, our ViT-B outperforms the DINOv3 ViT-B on many tasks. MAGE [58] and Sorcen [32] focus more on end-to-end generation, while we aim to support generation by replacing the VAE. To that end, we outperform them for classification (Table 1), and prior works do not show results for reconstruction, segmentation, and depth estimation.

3 Methods

3.1 Hyper-Network for Implicit Neural Representation

Background. Implicit neural representations store signals as neural network weights. A function $f_\theta : X \to I$, which maps coordinates X to an image (or more generally, a signal) I. In the case of images, we can represent these coordinates as $X = \{(x_i, y_j | 0 \leq x_i < W, 0 \leq y_j < H\}$ and images as color tuples $I = \{(r_{x_i,y_j}, g_{x_i,y_j}, b_{x_i,y_j}) | 0 \leq x_i < W, 0 \leq y_j < H\}$. In practice, these networks require long per-sample training time, with multiple iterations for every pixel, for f_θ to "memorize" I. After training for hundreds of iterations, an implicit neural network can nearly perfectly reconstruct a chosen image, video, or scene. However, there is no 'generalizability' – for every new I, we must train a new corresponding f_θ.

To solve this, a hyper-network approximates a function, h, for mapping input signals to their corresponding instance-specific INR, $h : I \to f_{\theta'}$ [22]. Instead of a fitting f_θ separately for each sample, h is trained on a large dataset, and learns a general mapping between data samples and θ'. After this pre-training, encoding I as θ' requires only one forward pass of h.

Prior INR hyper-networks are initialized with 4 learnable components:

1. E, a transformer encoder
2. θ_b, "shared" or "base" INR weights with n layers, $\theta_b = \{W_i\}_{i=1}^n$
3. θ_0, input weight tokens corresponding to the number of layers n
4. fully-connected (FC) linear layers to project from d_E (transformer encoder output dimension) to d_{in} of W_i

The transformer encoder, E, takes the image and weight tokens as input, and gives output image and weight tokens. The output image tokens are discarded. The output weight tokens corresponding to some W_i in θ_b are fed through an FC layer to change their token dimension to d_{in}, and then repeated $k = \frac{d_{\text{out}}}{n}$ times to form a modulation matrix M_i of shape $d_{\text{in}} \times d_{\text{out}}$. Each M_i is multiplied elementwise with W_i to modulate it, forming the final weights θ' of f. In other words, for a sample-specific $f_{\theta'}$, we have $\theta' = \{W_i \odot M_i\}_{i=1}^{n}$. For more details, we include a tutorial in the Appendix.

Overview. Figure 2 depicts our method. Unlike prior works, there are no learnable weight tokens. Instead of predicting image INRs, our h predicts $f_{\theta'}$ for each patch separately. Patch tokens are fed to the INR hyper-network, along with a novel learnable global token. The INR hyper-network itself is composed of 3 parts, which generate 3 sets of tokens in sequence:

1. Standard Vision Transformer (ViT) output tokens, which we can use for classification, reconstruction, segmentation, generation, etc.
2. Compressed encoding (TinToks), which we use for the same tasks as the ViT tokens. These are up to 96× smaller than the ViT tokens.
3. An INR modulation encoding, which we use only for reconstruction.

The INR modulation encoding for each patch is projected and then multiplied by a projection of the global token, which yields a matrix M_i to modulate the corresponding W_i of θ_b. We use all modulated, per-patch INRs for a given image to reconstruct the input patch. We concatenate the reconstructed patches to reconstruct the full I.

Key Innovation #1: Patch Tokens as Weight Tokens. Existing INR hyper-networks discard the output image tokens, using them neither to calculate loss nor perform inference. This makes dense tasks like semantic segmentation very challenging, since I is represented mainly in the weight tokens, but these lack clear correlation with spatial locations in the input. To resolve the inefficiency and lack of capability, we use the data tokens themselves as weight tokens. This presents a challenge – in the standard hyper-network formulation, the number of weight tokens must be a factor of the d_{in} or d_{out} for W_i in θ_b. This makes it difficult to formulate good hyper-network and INR settings. To solve this, we refactor the framework, predicting instead an $f_{\theta'}$ per-patch, rather than per-image. So, the output tokens of h are also patch tokens.

This leads to another issue. The output token has dimension d_{ViT}, but W_i has dimension $d_{\text{in}} \times d_{\text{out}}$. Following the ideas from prior works [15, 22] we could simply make sure $d_{\text{ViT}} = d_{\text{in}}$ and, and copy the output token d_{out} times to yield an M_i to multiply with W_i. However, in practice this is suboptimal (see Section 4.5).

Key Innovation #2: Global Tokens to Modulate and Summarize. Now, recall that the original INR hyper-network formulation does not have a class token. We solve both of these problems simultaneously by adding a global token, g, which can act as a class (cls) token for recognition tasks. We can project g to yield $g_{\text{proj}} \in \mathbb{R}^{d_{\text{out}}}$, and patch tokens p to yield $p_{\text{proj}} \in \mathbb{R}^{d_{\text{in}}}$. We then compute the outer product to obtain a modulation matrix for each layer:

$$M_i = g\, p^\top \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}, \tag{1}$$

where M_i is used to modulate the weights W_i of the hyper-network. With this modification, the architecture now includes a class token, patch tokens, and no wasted tokens, making it suitable for both INR prediction and image recognition.

Key Innovation #3: Tiny tokens (TinToks). In practice, we want to be able to set the dimension of the transformer encoder, d_{ViT} separately from d_{in} and d_{out} for any W_i. Additionally, compute-constrained applications require smaller tokens. To satisfy this, we introduce an intermediate representation, TinToks, which have their own dimension d_t. To accomplish this, we use a linear layer to downsample from d_{ViT} to d_t, and another to upsample for decoding. We then process the tokens with a transformer decoder to allow for better reconstruction. We use final linear projections to convert the tokens from the decoding dimension to d_{in} and d_{out} for the patch and global tokens, respectively.

Training. We train the model with distillation losses (described in Section 3.2) and visual quality losses. For the visual quality, we compute mean-squared error on pixels between the input image and the concatenated patch predictions of each patch $f_{\theta'}$. We can also optionally train with SSIM [100] and LPIPS [113] losses to further improve the reconstruction.

3.2 Distillation to INR Hyper-Networks

Key Innovation #4: Distillation for Unified Reprsentation. We align our representations to the pre-trained model (for most of our experiments, DINOv3) by computing a loss between DINOv3 features and a linear projection of our features. While we could hypothetically apply this to the outputs of any of our layers, in practice we apply this loss to the outputs of the last encoder (enc) and decoder (dec) blocks. We learn separate transformations for each block, and within each block we learn separate transformations depending on whether a token is global (type_g or patch (type_p. So, considering token types $\text{type}_g, \text{type}_p \in T$ and outputs of the final encoder and decoder blocks $o_{\text{enc}}, o_{\text{dec}} \in O$, we get 4 components of our loss, which we weight individually with some α. We can compute an L_2 distillation loss with features F from some teacher as follows:

$$L_{distillation} = \sum_{t}^{T} \sum_{o}^{O} L_2(\theta_{t,o}(F_{t,o}) - F_{t,\text{teacher}}) * \alpha_{t,o} \tag{2}$$

Note that we do not perform distillation on the compressed tokens directly. Combined with the INR reconstruction objective, we find that distillation to the encoder and decoder is sufficient to imbue the compressed tokens with good semantics for downstream reconstruction tasks.

Table 2: Tiny token results. We compare our compressed tokens to a principal component analysis (PCA) baseline. We compute the PCA transforms using the ImageNet-1k training set. We report linear probing classification accuracies for ImageNet (original and ReaL labels), ObjectNet, and some fine-grained datasets. We train a decoder on frozen DINOv3 for a reconstruction baseline.

Settings			Classification								Reconstruction	
Enc.	Dim.	Pre-Training	INet	ReaL	ONet	Cars	CUB	DTD	Flowers	Food	PSNR	SSIM
VAE	-	SD VAE	-	-	-	-	-	-	-	-	24.99	0.7078
ViT-B	8	DINOv3, PCA	16.1	18.0	8.2	31.3	44.9	28.8	57.8	31.7	15.51	0.4975
		HUVR (ours)	**57.1**	**64.2**	**36.0**	**68.9**	**67.2**	**43.9**	**98.0**	**77.5**	**24.66**	**0.6918**
	16	DINOv3, PCA	40.6	45.1	19.5	58.5	67.6	45.5	86.7	59.2	16.75	0.5228
		HUVR (ours)	**71.4**	**78.6**	**44.6**	**82.1**	**79.4**	**64.1**	**99.1**	**87.3**	**26.25**	**0.7361**
	32	C-RADIOv3, PCA	71.8	79.2	42.1	73.9	64.6	71.0	93.7	83.3	-	-
		SigLIP 2, PCA	62.3	67.8	31.3	86.8	75.4	70.2	98.6	85.9	-	-
		DINOv3, PCA	64.1	70.3	35.7	80.0	80.7	64.5	97.3	78.9	17.68	0.5398
		HUVR (ours)	**76.7**	**83.2**	**49.4**	**87.3**	**84.4**	**73.3**	**99.5**	**90.8**	**27.83**	**0.7845**
ViT-L	32	C-RADIOv3, PCA	77.9	**83.8**	47.5	82.0	73.2	**73.3**	98.6	89.4	-	-
		SigLIP 2, PCA	63.4	67.9	33.9	**90.6**	77.8	67.0	99.3	89.2	-	-
		DINOv3, PCA	72.2	77.5	45.0	84.2	**84.2**	64.8	99.1	88.1	17.27	0.5311
		HUVR (ours)	**78.1**	82.9	**53.9**	88.1	81.4	66.8	**99.5**	**91.2**	**27.70**	**0.7802**

4 Results

4.1 Settings

Training. For our models, we use a modern version of the Vision Transformer (ViT) [27], with Rotary Positional Embeddings (RoPE) [93]. We focus on ViT-B/16 and ViT-L/16, due to the prevalence of practical adoption for these sizes, and to reduce our training cost. Unless otherwise indicated, we pre-train our models on a mix of DataComp [33] (image-only) and ImageNet22k [82] data (without labels). We ensure at least 10% of all samples are from ImageNet22k (inspired by DINOv3 [88]) for the equivalent of 50 epochs on ImageNet22k. We parallelize according to resource availability by adjusting batch size and scaling the learning rate by a linear factor. For data augmentation, we only use random resized cropping. For details on model and training settings, see the Appendix.

Evaluation (Classification). We compute recognition evaluations with linear probing classification on ImageNet1k [24] using both the original labels and ReaL labels [7], and ObjectNet [5]. We do not perform any finetuning. We also perform linear probing with L-BFGS [60] for 5 FGVC datasets – Caltech-UCSD Birds 200 [98], Stanford Cars [49], Describable Textures Dataset [23], Oxford 102 Flower [73], and Food-101 [8].

Evaluation (Dense prediction). We perform linear probing on ADE20K [120] for semantic segmentation following the setup from AM-RADIO [80] and use a

Table 3: Learnable compression alternatives. We compare our compressed tokens to learnable baselines as well. AE indicates a simple autoencoder baseline, and L2B is AE on ViT-B features distilled with ViT-L teacher.

Settings			Classification							
Enc.	Dim.	Pre-Training	INet	ReaL	ONet	Cars	CUB	DTD	Flowers	Food
ViT-B	32	DINOv3, PCA	64.1	70.3	35.7	80.0	80.7	64.5	97.3	78.9
		DINOv3, AE	53.8	59.9	32.7	76.5	68.3	60.7	92.1	78.0
		DINOv3, L2B	67.3	74.1	41.3	81.6	77.1	68.2	98.8	84.8
		HUVR (ours)	**76.7**	**83.2**	**49.4**	**87.3**	**84.4**	**73.3**	**99.5**	**90.8**

Table 4: Diffusion results. We train a class-conditional DiT-XL on ImageNet using the original VAE latents, and other DiT-XLs using our compressed tokens. We compare the DiTs in terms of FID, Inception Score, Precision and Recall.

Autoencoder	Latents	FID↓	sFID↓	IS↑	P↑	R↑
SD VAE	32×32×4	**23.05**	68.65	**70.34**	**0.4318**	**0.4775**
Ours	16×16×16	24.72	76.09	60.17	0.3850	0.4645
Ours	16×256×256	24.53	**68.37**	66.13	0.4307	0.4367

similar setup for a depth probe [31] on NYUv2 [71]. For reconstruction, we report PSNR, SSIM, and LPIPS on the 50,000 images from the ImageNet-1k validation set (which is never seen during training).

Evaluation (Generation). For generation, we follow the protocol from DiT [78]. We also perform some comparison to prior INR hyper-networks by training and evaluating our reconstruction capability on comparable settings for the ImageNette subset of ImageNet [43], Celeb-A [63], and LSUN Churches [107] to prove that HUVR is the state-of-the-art INR hyper-network for images. For more details, see the Appendix.

4.2 Tiny Tokens are a Unified Representation

We show in Table 2 that our compressed vector representations simultaneously achieve good recognition and reconstruction. To our knowledge, our method is the first to attempt classification, segmentation, and depth estimation with such compressed features. So, for a baseline, we fit a principal component analysis (PCA) transform for DINOv3 [88], C-RADIOv3 [41], and SigLIP2 [97] at the corresponding compressed token dimension on the training set of ImageNet1k. We consider PCA a fair comparison, but we design specialized learnable baselines and outperform these also (Table 3). We then use this transform for ImageNet, and all other datasets, and refer to it as "PCA" for each method. We also try baselines with learnable weights, but find the PCA to be more effective (details in

Table 5: Standard classification results. Our model is competitive with state-of-the-arts for classification, including on fine-grained classification (FGVC) datasets. We set our TinTok $d = 32$ but evaluate the performance of the full, standard-sized tokens.

Settings		Classification							
Encoder	Pre-Training	INet	ReaL	ONet	Cars	CUB	DTD	Flowers	Food
ViT-B, Patch Size 16, $d = 768$	C-RADIOv3	82.4	87.6	54.5	88.6	79.8	83.5	99.1	91.5
	SigLIP 2	84.5	89.0	**68.4**	92.8	82.6	83.7	99.3	94.2
	DINOv3	84.6	88.9	59.4	**93.4**	**89.7**	83.9	**99.7**	93.7
	HUVR (ours)	**85.0**	**89.2**	62.0	93.1	89.4	**84.3**	**99.7**	**94.3**
ViT-L, Patch size 16, $d = 1024$	C-RADIOv3	86.0	89.6	66.0	92.8	85.4	86.2	99.6	94.4
	SigLIP 2	**87.1**	**90.1**	**75.5**	**94.9**	85.0	85.1	99.6	**96.1**
	DINOv3	**87.1**	90.0	71.0	93.7	**91.1**	**86.6**	**99.7**	95.6
	HUVR (ours)	86.9	**90.1**	70.7	91.7	85.7	84.5	**99.7**	95.4

the Appendix). Note that none of these methods perform reconstruction natively, so to have a baseline we train a decoder with roughly the same number of parameters as ours, on the frozen DINOv3 PCA features. Our compressed tokens offer the best unified modeling capability. The gap between ours and the baseline increases as the compression ratio increases – notice the difference between ours and DINOv3 for 8-dim tokens.

To show the promising generative potential of our latents, we train a DiT [78] on the latents of our model instead of the SD VAE [81] latents. We show results in Table 4. While we fail to beat the performance of the existing SD VAE or achieve generative state-of-the-art results, we consider these results promising and hope they highlight the potential our method as a unified representation. Following insights from concurrent work [119], we try increasing the TinTok dimension to 256, and observe that increasing the TinTok dimension seems beneficial for generation. Applying further insights from the concurrent work, which uses frozen pre-trained vision encoders, would likely help further improve the performance, especially given that our model is natively superior for reconstruction.

4.3 HUVR beats DINOv3 for Classification

Even HUVR's standard-size embeddings are quite useful. Table 5 shows that HUVR can match or beat prior works on ImageNet (both original and ReaL labels), and achieve good results on ObjectNet and the fine-grained datasets as well. Comparing our ObjectNet to SigLIP 2, it is worth keeping in mind that SigLIP 2 trains on 25 times more data. For FGVC datasets, DINOv3 mines pretraining data that is similar to many of these, whereas we use pre-training data that is not specifically curated for these [33].

4.4 HUVR can Perform Dense Recogntion Tasks

For fair comparison, since these tasks tend to evaluate at higher resolution (512×512 for ADE20K semantic segmentation, 480×480 for NYUv2 depth es-

Table 6: Dense recognition results. We compare HUVR to RADIOv3, SigLIP 2, and DINOv3 for semantic segmentation on ADE20K and depth estimation on NYUv2.

ViT-B					ViT-L				
		ADE20K		NYUv2			ADE20K		NYUv2
Size	Method	mIoU	mAcc	RMSE↓	Size	Method	mIoU	mAcc	RMSE↓
768	SigLIP 2	40.0	53.6	0.5317	1024	SigLIP 2	42.0	55.8	0.5107
	C-RADIOv3	49.5	62.4	0.3359		C-RADIOv3	51.6	63.8	0.3239
	DINOv3	50.8	62.6	0.3305		DINOv3	**54.2**	**66.2**	**0.3235**
	HUVR	**52.0**	**63.4**	**0.3263**		**HUVR**	53.5	65.2	0.3287
32	SigLIP 2	17.5	25.0	0.8155	32	SigLIP 2	13.5	19.3	0.7740
	C-RADIOv3	28.9	40.3	0.6017		C-RADIOv3	27.3	37.0	**0.5430**
	DINOv3	29.7	38.6	0.7056		DINOv3	29.4	39.2	0.6685
	HUVR	**37.6**	**48.4**	**0.5440**		**HUVR**	**30.9**	**41.1**	0.5726

Table 7: Hyper-network results. We compare our INR hyper-network to prior works on ImageNette (178×178), LSUN (256×256), and CelebA (178×178), in terms of PSNR.

Method	Epochs	ImageNette	LSUN	CelebA
TransINR [22]	4000/12.67/300	29.01	24.21	31.96
IPC [45]	4000/-/300	38.46	-	35.93
LA-IPC [53]	4000/-/300	46.10	-	50.74
ANR [114]	-/12.67/-	-	28.30	-
Ours	**400/12/100**	**48.44**	**34.00**	**56.91**

Table 8: Hyper-network design. We demonstrate how our changes to the original TransINR hyper-network impact the reconstruction performance on ImageNette. For each row, we keep all changes from prior rows, such that the last row contains our full method.

Method	# Params	PSNR↑	SSIM↑	LPIPS↓
TransINR + RoPE	44.00M	23.78	0.7041	0.3866
+ only second layer [45]	43.21M	27.15	0.8000	0.2424
+ patchwise	43.21M	51.96	0.9974	0.0026
+ global token	43.41M	**53.36**	**0.9985**	**0.0007**
+ compression	43.41M	48.58	0.9962	0.0019
+ decoder	43.40M	48.44	0.9954	0.0014

timation), after we pre-train at 256×256, we have a second training stage. We further train for 3 epochs (following DINOv3) on a mix of 256×256 and 512×512 resolution images, from ImageNet22k only. We achieve strong results, in Table 6, for both semantic segmentation and depth estimation, with both standard tokens and TinToks. We can further improve the dense task performance of the TinToks at the expense of reconstruction performance, and we investigate these trade-offs in Section 4.6 and Section 4.7.

4.5 Our Formulation is Ideal for Predicting Image INRs

We show that our novel INR hyper-network design achieves state-of-the-art results compared to prior arts in Table 7. We set our hypernetwork here to have the same number of parameters or fewer, in terms of total encoder parameters, shared parameters, and predicted latents (details in the Appendix). Since our network generally requires more time to train, given equal epochs, we generally

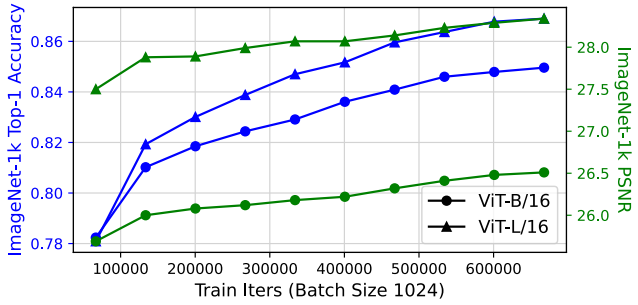

Fig. 3: Training time improves both classification (blue, left axis) and reconstruction performance (green, right axis). Longer training yields incrementally smaller gains for both ViT-B/16 and ViT-L/16 backbones.

use significantly fewer epochs that prior works (400 on ImageNette instead of 4000). We thus achieve the best PSNR (the standard metric for these methods) with equal or shorter training time.

We show how we achieve this in Table 8. The major driver of our good performance is our novel patch-wise design, which improves the performance at the cost of additional memory for forward computation. The global token, which we introduce mainly to facilitate tasks such as classification, also improves reconstruction performance. The compression and decoder are not necessary for reconstruction, but they are essential for good TinToks and well-behaved distillation, respectively.

4.6 Recognition and Reconstruction can Improve Together

In many ways the recognition and reconstruction performance can scale together. One of the most obvious ways is in terms of training time, shown in Figure 3, although the reconstruction performance saturates more quickly than classification. Another way is in terms of distillation teacher selection, which we explore via ablation an ablation in Table 9 where we train for 5 epochs on ImageNet22k with different teachers. For example, the RADIOv3 ViT-L improves all 5 metrics compared to the RADIOv3 ViT-B. In fact, for cases where the ViT-L seems worse, we find in practice that this is a result of limited training time. Given sufficient training time (15 ImageNet22k epochs for DINOv3), there is a crossover point where distilling from the larger teacher becomes superior (see Section 4.7), so in our main results we distill from a ViT-L for our ViT-B, and from a ViT-H for our ViT-L.

4.7 Some Design Decisions Involve Trade-offs

Distillation Teacher Selection. We can revisit Table 9 in terms of some trade-offs between performance for different tasks. For example, the ViT-B DINOv2

Table 9: Distillation teachers. We show the effect of distilling from different teachers in terms of main token classification (d=768), segmentation (d=768), tiny token classification (d=32), and reconstruction (PSNR and SSIM).

Teacher		Classification		Segmentation	Reconstruction	
Enc.	Method	d=768	d=32	mIoU	PSNR	SSIM
ViT-B	DINOv2	82.7	**73.5**	42.20	24.87	0.6950
	C-RADIOv3	78.3	47.1	43.32	25.83	0.7240
	SigLIP 2	81.3	58.2	37.87	**26.61**	**0.7458**
	DINOv3	**83.2**	69.2	**44.01**	25.33	0.7129
ViT-L	DINOv2	80.9	74.1	38.37	25.89	0.7281
	C-RADIOv3	78.8	52.9	**45.67**	**26.49**	0.7433
	SigLIP 2	**82.6**	68.5	38.70	24.29	0.6754
	DINOv3	81.3	**74.2**	40.31	26.30	**0.7439**

Table 10: Distillation token selection. We show the effect of distilling for the global (cls) token only, patch tokens only, and both in terms of main token classificiation (d=768), tiny token classification (d=32), segmentation (d=768), and reconstruction (PSNR and SSIM).

Block Selection		Classification		Segmentation	Reconstruction	
Global	Patch	d=768	d=32	mIoU	PSNR	SSIM
		11.4	1.9	4.18	**28.42**	**0.7973**
✓		82.9	**71.3**	39.29	26.42	0.7468
	✓	81.9	65.5	**45.23**	26.95	0.7580
✓	✓	**83.2**	69.3	43.68	25.33	0.7133

teacher seems to give a better tiny token accuracy, but this is counterbalanced by a worse performance on other metrics. RADIOv3 gives the best overall segmentation performance, but the weakest ImageNet classification, by far. A truly optimal method would probably distill from a mixture of teachers, but we consider such engineering efforts out of scope.

Distillation Token Selection. In Table 10 we compare our chosen distillation approach (where we compute distillation losses to both the global token and patch tokens) to alternatives where we distill only to the global token, only to patch tokens, or neither. For settings, we use a DINOv3 ViT-B teacher and train for 5 ImageNet22k epochs only. In general, the impact of the distillation target tokens on patch performance makes sense. Distillation to the global token is helpful for classification, and distillation to patches is helpful for segmentation. Interestingly, the overall $d = 768$ classification is best when we distill to all tokens. However, the reconstruction suffers substantially. We ultimately choose this setting anyway, and counterbalance the falling reconstruction metrics with longer training and distillation from a larger teacher model.

Distillation Teacher Size "Crossover". When we train the model with a larger teacher, it requires more epochs before it outperforms the smaller teacher. However, given sufficiently long training time, larger teachers are superior. We illustrate this crossover effect for our ViT-B/16 in Table 11. We use the attention-free decoder for this experiment (hence the numbers do not exactly match other tables in the paper). Note that the crossover effect impacts standard ViT tokens ($d = 768$) more than TinToks ($d = 32$). The reconstruction and classification of the TinToks is better using large teachers, regardless of the number of training epochs. However, the standard tokens require many training epochs to justify using the DINOv3 ViT-L teacher instead of the DINOv3 ViT-B teacher (up to 40 epochs for classification, 10 epochs for segmentation).

Table 11: Distillation Teacher Size Crossover Effect. When distilling from DINOv3, training with a teacher larger than the student initially yields poorer results than training with a similar-size teacher. However, after training for enough time, training with the larger teacher is superior. For each metric, we highlight the value for the earliest epoch where the ViT-L teacher gives a better result than the ViT-B at the same epoch. We give TinTok results ($d = 32$) for ImageNet only.

Settings		Classification									Segment. (ADE20K)		Recon. (INet)	
Teacher	Epochs	INet ($d = 32$)	INet	ReaL	ONet	Cars	CUB	DTD	Flowers	Food	mIoU	mAcc	PSNR	SSIM
ViT-B	10	70.6	82.5	87.7	53.3	92.8	88.3	82.6	99.7	91.9	45.74	58.28	25.08	0.7001
	20	71.4	83.2	88.2	54.6	92.9	88.3	83.6	99.6	92.4	46.11	58.52	25.18	0.7027
	30	71.6	83.6	88.3	56.0	93.1	88.9	84.1	99.6	92.7	46.85	59.44	25.22	0.7036
	40	72.5	84.0	88.5	57.8	93.3	**89.3**	84.3	99.7	93.2	46.87	58.69	25.43	0.7086
	50	72.8	84.1	88.6	58.3	**93.4**	**89.3**	**84.9**	99.6	93.2	47.17	58.72	25.50	0.7108
ViT-L	10	70.6	80.3	86.2	47.3	91.1	87.3	80.8	99.5	90.4	44.82	56.74	25.65	0.7173
	20	72.8	82.1	87.4	51.4	91.9	88.1	82.6	99.6	91.6	46.33	59.06	25.70	0.7194
	30	74.5	83.1	88.0	55.2	92.4	88.9	82.8	99.7	92.7	47.35	58.86	25.68	0.7189
	40	75.9	84.1	88.7	58.9	92.9	89.2	84.8	**99.8**	93.7	48.41	60.04	25.95	0.7250
	50	**76.6**	**84.6**	**88.9**	**60.8**	93.1	**89.3**	84.7	99.7	**94.2**	**48.71**	**60.51**	**26.02**	0.7275

Table 12: Distillation block selection. We ablate selection of which block output we choose for the distillation. We try 2 blocks each for the Encoder and the Decoder and report performance of the standard token (d=768) and tiny token (d=32) for ImageNet linear probing classification, and of the tiny token for reconstruction (PSNR, SSIM). Recall that encoder and decoder have 12 and 4 blocks, respectively.

Block Selection		Classification		Reconstruction	
Encoder	Decoder	d=768	d=32	PSNR	SSIM
11	1	82.5	69.7	26.06	0.7370
11	4	82.9	**72.6**	25.40	0.7143
12	1	**83.2**	68.8	**26.18**	**0.7454**
12	4	83.1	69.4	25.36	0.7140

Table 13: Distillation design. We show the impact of increasing the tiny token size to match the original tokens, chaning the INR hidden dimension from 256 to 512, changing the number of INR layers from 4 to 3 or 5, reducing from 4 transformer decoder layers to 1, or removing the attention from the transformer decoder.

	Classification		Segmentation	Reconstruction	
Setting	d=768	d=32	mIoU, d=768	PSNR	SSIM
Default	**83.1**	69.7	43.85	25.35	0.7135
Tiny dim=768	**83.1**	69.1	44.00	**25.65**	**0.7243**
INR w/ dim=512	82.3	70.8	23.36	25.30	0.7119
INR w/ 3 layers	**83.1**	69.6	43.36	25.08	0.7015
INR w/ 5 layers	**83.1**	70.2	**44.44**	25.48	0.7188
Trans. w/ 1 layer	**83.1**	69.6	43.65	24.78	0.6962
Trans. w/o att.	**83.1**	**73.5**	43.83	24.90	0.6943

Distillation Block Selection. We can select any combination of blocks for distillation, and we perform a limited investigation in Table 12, where we train for 5 ImageNet22k epochs, using a DINOv3 ViT-B teacher. The best setting for reconstruction and standard classification (12, 1) is the worst for tiny token classification. Distilling to (11, 4) leads to better TinTok classification, and (12, 1) yields the best reconstruction. However, we ultimately opt to distill to the last block of each network (12, 4), as a good middle ground.

Decoder Design. We examine decoder design choices in Table 13. Some choices, such as increasing the INR hidden dimension from 256 to 512, both increase

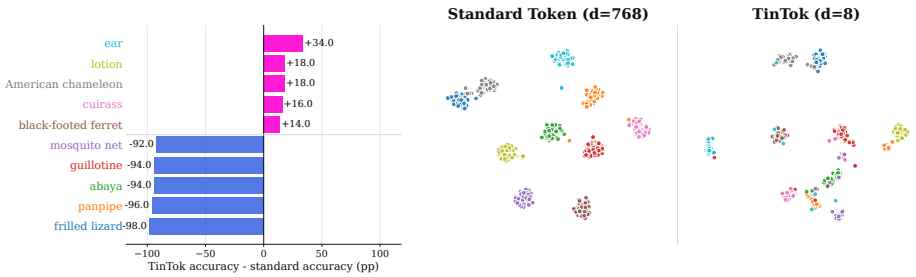

Fig. 4: (Left) Accuracy difference between standard tokens and TinToks (dim=8) on ImageNet, ViT-B/16, 5 most positive and 5 most negative. **(Middle)** t-SNE visualization for these 10 classes, standard tokens. **(Right)** t-SNE for TinToks.

computational cost and decrease accuracy. Increasing the token dimension in the decoder improves reconstruction, but hurts speed and TinTok classification. Converts the hyper-network decoder from a transformer, to a multi-layer perceptron with residual connections, improves the TinTok classification without affecting the standard-sized token classification or segmentation. However, it has a significant negative impact on reconstruction performance. In spite of this, for our TinTok experiments (such as in Table 2), we opt to use this setting, and we mainly to counter-balance the negative effect with more training time.

4.8 Analysis

While in general TinToks exchange performance for size, they can actually perform better for individual classes. We show this in Figure 4. In the same figure, we show that while the TinToks can have a degraded feature structure with more easily confused classes, some classes remain in good condition.

5 Conclusion

We propose HUVR, an INR hyper-network for unified universal image representation. Not only does HUVR compare favorably with prior works for unsupervised learning for image recognition, it also yields compressed TinToks which support recognition and reconstruction. We demonstrate the utility of the method for various datasets and embeddings sizes for tasks including classification, segmentation, and generation. We hope to enable further exploration in unified representation learning, compressed representations, and implicit neural representation as a viable strategy for universal vision encoding.

Limitations. Our pre-training does not operate at the same scale or scope as prior image representation methods. Furthermore, our method does not handle the generation end-to-end; instead, in its current state it would be an alternative to a VAE, but even then the results are not SOTA. Potential application to tasks with Vision Language Models (VLMs) would require text-aligned pre-training.

References

1. Assran, M., Caron, M., Misra, I., Bojanowski, P., Bordes, F., Vincent, P., Joulin, A., Rabbat, M., Ballas, N.: Masked siamese networks for label-efficient learning (2022) 4
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization (2016), https://arxiv.org/abs/1607.06450 5
3. Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023) 4
4. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers (2022) 4
5. Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., Katz, B.: Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. Advances in neural information processing systems **32** (2019) 8
6. Bardes, A., Ponce, J., LeCun, Y.: Vicreg: Variance-invariance-covariance regularization for self-supervised learning. ArXiv **abs/2105.04906** (2021) 4
7. Beyer, L., Hénaff, O.J., Kolesnikov, A., Zhai, X., van den Oord, A.: Are we done with imagenet? (2020), https://arxiv.org/abs/2006.07159 8
8. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: European Conference on Computer Vision (2014) 8
9. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 132–149 (2018) 4
10. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in Neural Information Processing Systems **33**, 9912–9924 (2020) 4
11. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021) 1, 4
12. Chen, H., Gwilliam, M., He, B., Lim, S.N., Shrivastava, A.: Cnerv: Content-adaptive neural representation for visual data (2022) 3
13. Chen, H., Gwilliam, M., Lim, S.N., Shrivastava, A.: Hnerv: A hybrid neural representation for videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10270–10279 (2023) 3
14. Chen, H., He, B., Wang, H., Ren, Y., Lim, S.N., Shrivastava, A.: Nerv: Neural representations for videos. Advances in Neural Information Processing Systems **34**, 21557–21568 (2021) 3, 5
15. Chen, H., Xie, S., Lim, S.N., Shrivastava, A.: Fast encoding and decoding for implicit video representation (2024), https://arxiv.org/abs/2409.19429 3, 6
16. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020) 1, 4
17. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.E.: Big self-supervised models are strong semi-supervised learners. Advances in neural information processing systems **33**, 22243–22255 (2020) 4
18. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Advances in neural information processing systems **29** (2016) 4

19. Chen, X., Fan, H., Girshick, R.B., He, K.: Improved baselines with momentum contrastive learning. CoRR **abs/2003.04297** (2020), https://arxiv.org/abs/2003.04297 4
20. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021) 4
21. Chen*, X., Xie*, S., He, K.: An empirical study of training self-supervised vision transformers. arXiv preprint arXiv:2104.02057 (2021) 4
22. Chen, Y., Wang, X.: Transformers as meta-learners for implicit neural representations (2022) 2, 3, 5, 6, 11
23. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2014) 8
24. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848 3, 8
25. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. arXiv preprint arXiv:1605.09782 (2016) 4
26. Donahue, J., Simonyan, K.: Large scale adversarial representation learning. Advances in neural information processing systems **32** (2019) 1, 4
27. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 2, 8
28. Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., Courville, A.: Adversarially learned inference. arXiv preprint arXiv:1606.00704 (2016) 4
29. Dupont, E., Goliński, A., Alizadeh, M., Teh, Y.W., Doucet, A.: Coin: Compression with implicit neural representations. arXiv preprint arXiv:2103.03123 (2021) 3
30. Dupont, E., Loya, H., Alizadeh, M., Golinski, A., Teh, Y.W., Doucet, A.: Coin++: Data agnostic neural compression. arXiv preprint arXiv:2201.12904 **1**(2), 4 (2022) 3
31. El Banani, M., Raj, A., Maninis, K.K., Kar, A., Li, Y., Rubinstein, M., Sun, D., Guibas, L., Johnson, J., Jampani, V.: Probing the 3d awareness of visual foundation models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 21795–21806 (June 2024) 9
32. Estepa, I.G., de Vera, J.M.R., Sarasúa, I., Nagarajan, B., Radeva, P.: Conjuring positive pairs for efficient unification of representation learning and image synthesis (2025), https://arxiv.org/abs/2503.15060 5
33. Gadre, S.Y., Ilharco, G., Fang, A., Hayase, J., Smyrnis, G., Nguyen, T., Marten, R., Wortsman, M., Ghosh, D., Zhang, J., Orgad, E., Entezari, R., Daras, G., Pratt, S., Ramanujan, V., Bitton, Y., Marathe, K., Mussmann, S., Vencu, R., Cherti, M., Krishna, R., Koh, P.W., Saukh, O., Ratner, A., Song, S., Hajishirzi, H., Farhadi, A., Beaumont, R., Oh, S., Dimakis, A., Jitsev, J., Carmon, Y., Shankar, V., Schmidt, L.: Datacomp: In search of the next generation of multimodal datasets. arXiv preprint arXiv:2304.14108 (2023) 8, 10
34. Girish, S., Shrivastava, A., Gupta, K.: Shacira: Scalable hash-grid compression for implicit neural representations (2023), https://arxiv.org/abs/2309.15848 3

35. Gregor, K., Besse, F., Rezende, D.J., Danihelka, I., Wierstra, D.: Towards conceptual compression (2016), https://arxiv.org/abs/1604.08772 2
36. Grill, J., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.Á., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent: A new approach to self-supervised learning. CoRR **abs/2006.07733** (2020), https://arxiv.org/abs/2006.07733 4
37. Gwilliam, M., Zhang, R., Padmanabhan, N., Du, H., Shrivastava, A.: How to design and train your implicit neural representation for video compression. arXiv preprint arXiv:2506.24127 (2025) 3
38. Haydarov, K., Muhamed, A., Shen, X., Lazarevic, J., Skorokhodov, I., Galappaththige, C.J., Elhoseiny, M.: Adversarial text to continuous image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6316–6326 (June 2024) 3
39. He, B., Yang, X., Wang, H., Wu, Z., Chen, H., Huang, S., Ren, Y., Lim, S.N., Shrivastava, A.: Towards scalable neural representation for diverse videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6132–6142 (June 2023) 3
40. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.B.: Masked autoencoders are scalable vision learners. CoRR **abs/2111.06377** (2021), https://arxiv.org/abs/2111.06377 4
41. Heinrich, G., Ranzinger, M., Hongxu, Yin, Lu, Y., Kautz, J., Tao, A., Catanzaro, B., Molchanov, P.: Radiov2.5: Improved baselines for agglomerative vision foundation models (2024), https://arxiv.org/abs/2412.07679 4, 9
42. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 4
43. Howard, J.: Imagenette, https://github.com/fastai/imagenette/ 9
44. Huang, Z., Jin, X., Lu, C., Hou, Q., Cheng, M.M., Fu, D., Shen, X., Feng, J.: Contrastive masked autoencoders are stronger vision learners (2022) 4
45. Kim, C., Lee, D., Kim, S., Cho, M., Han, W.S.: Generalizable implicit neural representations via instance pattern composers. arXiv preprint arXiv:2211.13223 (2022) 2, 3, 11
46. Kim, H., Bauer, M., Theis, L., Schwarz, J.R., Dupont, E.: C3: High-performance and low-complexity neural compression from a single image or video (2023), https://arxiv.org/abs/2312.02753 3
47. Kim, J., Lee, J., Kang, J.W.: SNeRV: Spectra-Preserving Neural Representation for Video, p. 332–348. Springer Nature Switzerland (Nov 2024). https://doi.org/10.1007/978-3-031-73001-6_19, http://dx.doi.org/10.1007/978-3-031-73001-6_19 3
48. Kim, S., Yu, S., Lee, J., Shin, J.: Scalable neural video representations with learnable positional features. In: Advances in Neural Information Processing Systems (2022) 3
49. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13). Sydney, Australia (2013) 8
50. Kwan, H.M., Gao, G., Zhang, F., Gower, A., Bull, D.: Hinerv: Video compression with hierarchical encoding-based neural representation. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 72692–72704. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/e5dc475c370ff42f2f96dddf8191a40c-Paper-Conference.pdf 3

51. Kwan, H.M., Gao, G., Zhang, F., Gower, A., Bull, D.: Nvrc: Neural video representation compression (2024), https://arxiv.org/abs/2409.07414 3
52. Ladune, T., Philippe, P., Henry, F., Clare, G., Leguay, T.: Cool-chic: Coordinate-based low complexity hierarchical image codec (2023), https://arxiv.org/abs/2212.05458 3
53. Lee, D., Kim, C., Cho, M., Han, W.S.: Locality-aware generalizable implicit neural representation (2023), https://arxiv.org/abs/2310.05624 3, 11, 6
54. Lee, J.C., Rho, D., Ko, J.H., Park, E.: Ffnerv: Flow-guided frame-wise neural representations for videos. In: Proceedings of the 31st ACM International Conference on Multimedia. p. 7859–7870. MM '23, ACM (Oct 2023). https://doi.org/10.1145/3581783.3612444, http://dx.doi.org/10.1145/3581783.3612444 3
55. Lee, K.H., Arnab, A., Guadarrama, S., Canny, J., Fischer, I.: Compressive visual representations (2021), https://arxiv.org/abs/2109.12909 2
56. Li, A.C., Prabhudesai, M., Duggal, S., Brown, E.L., Pathak, D.: Your diffusion model is secretly a zero-shot classifier. In: ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling (2023), https://openreview.net/forum?id=Ck3yXRdQXD 5
57. Li, C., Yang, J., Zhang, P., Gao, M., Xiao, B., Dai, X., Yuan, L., Gao, J.: Efficient self-supervised vision transformers for representation learning (2022) 4
58. Li, T., Chang, H., Mishra, S.K., Zhang, H., Katabi, D., Krishnan, D.: Mage: Masked generative encoder to unify representation learning and image synthesis (2022) 1, 5
59. Li, Z., Wang, M., Pi, H., Xu, K., Mei, J., Liu, Y.: E-nerv: Expedite neural video representation with disentangled spatial-temporal context (2022), https://arxiv.org/abs/2207.08132 3
60. Liu, D.C., Nocedal, J.: On the limited memory bfgs method for large scale optimization. Mathematical programming **45**(1), 503–528 (1989) 8
61. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 26296–26306 (2024) 4
62. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023) 4
63. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015) 9
64. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2019), https://arxiv.org/abs/1711.05101 5
65. Maiya, S.R., Girish, S., Ehrlich, M., Wang, H., Lee, K.S., Poirson, P., Wu, P., Wang, C., Shrivastava, A.: Nirvana: Neural implicit representations of videos with adaptive networks and autoregressive patch-wise modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14378–14387 (2023) 3, 5
66. Maiya, S.R., Gupta, A., Gwilliam, M., Ehrlich, M., Shrivastava, A.: Latent-inr: A flexible framework for implicit representations of videos with discriminative semantics. In: European Conference on Computer Vision. pp. 285–302. Springer (2024) 3
67. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis (2020) 3

68. Misra, I., Maaten, L.v.d.: Self-supervised learning of pretext-invariant representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6707–6717 (2020) 1, 4
69. Mukhopadhyay, S., Gwilliam, M., Yamaguchi, Y., Agarwal, V., Padmanabhan, N., Swaminathan, A., Zhou, T., Ohya, J., Shrivastava, A.: Do text-free diffusion models learn discriminative visual representations? In: European Conference on Computer Vision. pp. 253–272. Springer (2024) 1, 5
70. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics **41**(4), 1–15 (Jul 2022). https://doi.org/10.1145/3528223.3530127, http://dx.doi.org/10.1145/3528223.3530127 3
71. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012) 9
72. Nie, W., Karras, T., Garg, A., Debnath, S., Patney, A., Patel, A.B., Anandkumar, A.: Semi-supervised stylegan for disentanglement learning. In: Proceedings of the 37th International Conference on Machine Learning. pp. 7360–7369 (2020) 4
73. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian Conference on Computer Vision, Graphics and Image Processing (Dec 2008) 8
74. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision. pp. 69–84. Springer (2016) 4
75. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023) 1, 4
76. Pang, B., Zhang, Y., Li, Y., Cai, J., Lu, C.: Unsupervised visual representation learning by synchronous momentum grouping (2022) 4
77. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016) 4
78. Peebles, W., Xie, S.: Scalable diffusion models with transformers (2023), https://arxiv.org/abs/2212.09748 9, 10
79. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021) 1, 4
80. Ranzinger, M., Heinrich, G., Kautz, J., Molchanov, P.: Am-radio: Agglomerative vision foundation model reduce all domains into one. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12490–12500 (June 2024) 4, 8
81. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2022), https://arxiv.org/abs/2112.10752 3, 10
82. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y 8

83. Saethre, J.E., Azevedo, R., Schroers, C.: Combining frame and gop embeddings for neural video representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9253–9263 (June 2024) 3
84. Saragadam, V., LeJeune, D., Tan, J., Balakrishnan, G., Veeraraghavan, A., Baraniuk, R.G.: Wire: Wavelet implicit neural representations (2023) 3
85. Schmidhuber, J.: A computer scientist's view of life, the universe, and everything. In: Foundations of computer science: Potential—theory—cognition, pp. 201–208. Springer (2006) 2
86. Shazeer, N.: Glu variants improve transformer (2020), https://arxiv.org/abs/2002.05202 4
87. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient subpixel convolutional neural network (2016), https://arxiv.org/abs/1609.05158 3, 5
88. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: Dinov3. arXiv preprint arXiv:2508.10104 (2025) 3, 4, 8, 9
89. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Advances in neural information processing systems **33**, 7462–7473 (2020) 3, 1
90. Skorokhodov, I., Ignatyev, S., Elhoseiny, M.: Adversarial generation of continuous images (2021), https://arxiv.org/abs/2011.12026 3
91. Solomonoff, R.J.: A formal theory of inductive inference. part i. Information and control **7**(1), 1–22 (1964) 2
92. Strümpler, Y., Postels, J., Yang, R., van Gool, L., Tombari, F.: Implicit neural representations for image compression (2022), https://arxiv.org/abs/2112.04267 3
93. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024) 8
94. Sun, Q., Wang, J., Yu, Q., Cui, Y., Zhang, F., Zhang, X., Wang, X.: Eva-clip-18b: Scaling clip to 18 billion parameters. arXiv preprint arXiv:2402.04252 (2024) 4
95. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. NeurIPS (2020) 3, 1
96. Tomasev, N., Bica, I., McWilliams, B., Buesing, L., Pascanu, R., Blundell, C., Mitrovic, J.: Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? (2022) 4
97. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., Hénaff, O., Harmsen, J., Steiner, A., Zhai, X.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features (2025), https://arxiv.org/abs/2502.14786 9
98. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011) 8
99. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024) 4

100. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004) 7
101. Wu, C., Quan, G., He, G., Lai, X.Q., Li, Y., Yu, W., Lin, X., Yang, C.: Qs-nerv: Real-time quality-scalable decoding with neural representation for videos. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 2584–2592 (2024) 3
102. Xiang, W., Yang, H., Huang, D., Wang, Y.: Denoising diffusion autoencoders are unified self-supervised learners. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15802–15812 (October 2023) 4, 5
103. Xu, D., Wang, P., Jiang, Y., Fan, Z., Wang, Z.: Signal processing for implicit neural representations (2022) 3
104. Xu, Y., Feng, X., Qin, F., Ge, R., Peng, Y., Wang, C.: Vq-nerv: A vector quantized neural representation for videos (2024), https://arxiv.org/abs/2403.12401 3
105. Xue, L., Shu, M., Awadalla, A., Wang, J., Yan, A., Purushwalkam, S., Zhou, H., Prabhu, V., Dai, Y., Ryoo, M.S., et al.: xgen-mm (blip-3): A family of open large multimodal models. arXiv preprint arXiv:2408.08872 (2024) 4
106. Yan, H., Ke, Z., Zhou, X., Qiu, T., Shi, X., Jiang, D.: 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 (CVPR). pp. 23019–23029 (June 2024) 3
107. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015) 9
108. Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., Xie, S.: Representation alignment for generation: Training diffusion transformers is easier than you think (2025), https://arxiv.org/abs/2410.06940 5
109. Yu, S., Tack, J., Mo, S., Kim, H., Kim, J., Ha, J.W., Shin, J.: Generating videos with dynamics-aware implicit generative adversarial networks (2022), https://arxiv.org/abs/2202.10571 3
110. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 12310–12320. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/zbontar21a.html 4
111. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11975–11986 (2023) 4
112. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European conference on computer vision. pp. 649–666. Springer (2016) 4
113. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 7
114. Zhang, S., Liu, K., Gu, J., Cai, X., Wang, Z., Bu, J., Wang, H.: Attention beats linear for fast implicit neural representation generation. In: European Conference on Computer Vision. pp. 1–18. Springer (2024) 3, 11, 6
115. Zhang, X., Yang, R., He, D., Ge, X., Xu, T., Wang, Y., Qin, H., Zhang, J.: Boosting neural representations for videos with a conditional decoder. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2556–2566 (June 2024) 3
116. Zhang, Y., van Rozendaal, T., Brehmer, J., Nagel, M., Cohen, T.: Implicit neural video compression (2021), https://arxiv.org/abs/2112.11312 3

117. Zhao, Q., Asif, M.S., Ma, Z.: Dnerv: Modeling inherent dynamics via difference neural representation for videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2031–2040 (June 2023) 3
118. Zhao, Q., Asif, M.S., Ma, Z.: Pnerv: Enhancing spatial consistency via pyramidal neural representation for videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19103–19112 (June 2024) 3
119. Zheng, B., Ma, N., Tong, S., Xie, S.: Diffusion transformers with representation autoencoders (2025), https://arxiv.org/abs/2510.11690 1, 10, 8
120. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017) 3, 8
121. Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: ibot: Image bert pre-training with online tokenizer (2022) 4
122. Zhou, P., Zhou, Y., Si, C., Yu, W., Ng, T.K., Yan, S.: Mugs: A multi-granular self-supervised learning framework (2022) 4