6 Supplementary Material

6.1 Data generation

To simulate the training and testing data available to chicks, we created a virtual animal chamber in the Unity Game Engine. The virtual chamber was equipped with two 19" LCD monitors situated on opposite sides and accompanied by two alternating white walls. The LCD monitors were used to display the virtual objects of size 8 cm (length) x 7 cm (height), positioned at the center on a white display background. The floor of the virtual chamber was constructed with wire mesh and had transparent holes to hold food and water on one side of the chamber. The virtual chamber measured 66 cm (length), 42 cm (width), and 69 cm (height).

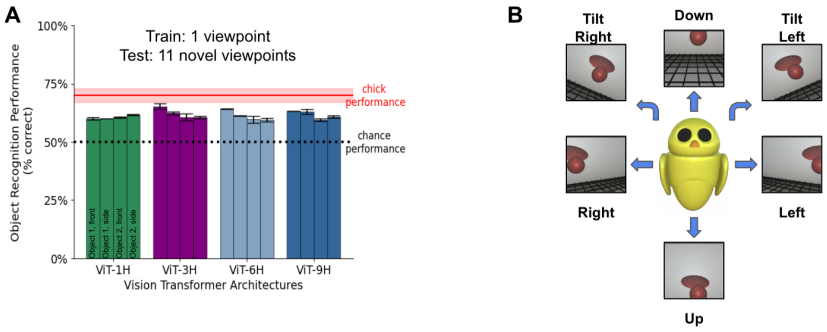
We developed a virtual chick agent to simulate the visual experiences of newborn chicks. The virtual agent had a height of 3.5 units and a length of 1.2 units. To monitor the agent's behavior, the virtual chamber was equipped with two invisible cameras. The first camera, positioned in the chamber's ceiling, captured the top-view of the agent, while the second camera, placed in the agent's head, recorded first-person RGB images (64 x 64 resolution) as the agent moved throughout the chamber.

The agent picked a random location inside the chamber and moved to that position at the rate of 1.5 units/s. During this movement, the agent directed its visual attention towards the object projected on the LCD monitor, ensuring a centered focus. Once the agent reached the destination point, it randomly moved its head along the three axes, rotating 60° on each axis, to naturally augment the data (Figure 1B). This complete random sequence of head rotations lasted for 9.5 seconds. The agent repeated this cycle until they had captured 80,000 first-person images in the rearing condition. We repeated this data simulation cycle for each of the 4 rearing conditions.

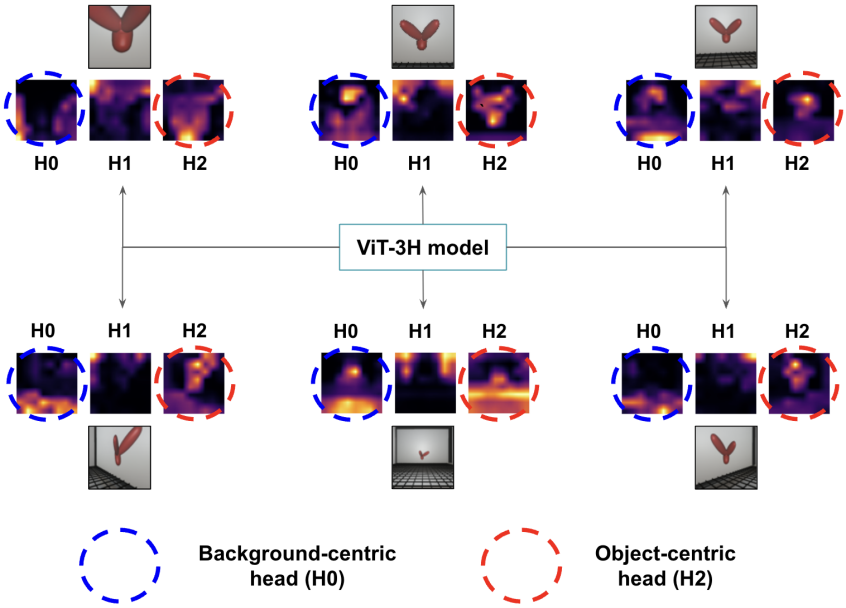
The same procedure was repeated to gather the test samples; however, only 11,000 images were captured for each of the 12 viewpoint ranges. These test images were used to train and evaluate the linear classifiers.

6.2 Heatmap generation

To visualize the internal working of the attention heads, we generated saliency heatmaps and heatmap videos from the last block of the transformer. Section 3.2 describes the steps required to generate a heatmap for a sample image. To create heatmap videos, we followed these steps. First, the agent moved inside the chamber and captured images at 10 frames/s. Second, we used those images and created saliency heatmaps in a sequential order. Third, we concatenated these heatmaps in a series



Supplementary Figure 1: (A) Comparing the view-invariant recognition performance of newborn chicks and ViTs in the sparse linear classifier condition (1 training viewpoint). The red horizontal line shows the chicks' performance, with the ribbon representing standard error. The bars show the view-invariant recognition performance for the four ViT architecture sizes, across the four rearing conditions presented to the chicks. Error bars denote standard error. In this sparse linear classifier condition, the linear classifiers are trained on the same viewpoint as the ViT encoder and tested on the remaining 11 novel viewpoints. (B) The virtual agent's head rotations, which provided natural data augmentations. Upon reaching the destination point inside the chamber, the agent randomly rotated its head across the three axes (tilt, left/right, and up/down).



Supplementary Figure 2: Visualizing the attention patterns of individual attention heads using saliency heatmaps derived from the ViT-3H model. The visualizations were generated by using the simulated first-person images captured by the virtual agent. In the ViT-3H model, (H0-H2) correspond to the attention filters of each of the three heads. The blue dotted line highlights that head (H0) largely focuses on background features, while the red dotted line demonstrates that head (H2) largely attends to the object features. This indicates that some attention heads become specialized to different parts of the environment, such as objects versus backgrounds.

Table 1: ViT-CoT with different architecture sizes.

Model	Parameters	Attention Heads	Transformer Blocks	Batch Size	Epochs
ViT-1H	5.8M	1	1	128	100
ViT-3H	16.9M	3	3	128	100
ViT-6H	36.4M	6	6	128	100
ViT-9H	59.4M	9	9	128	100

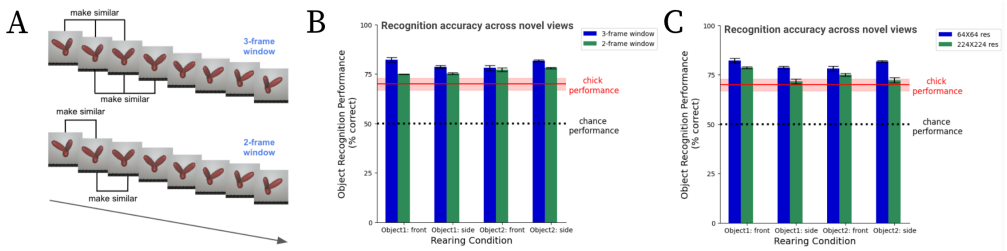
using the OpenCV library to produce a heatmap video. We repeated these steps for all the attention heads of all the different architectures. Figure 2 displays the heatmaps generated by the ViT-3H model, showcasing the visual representations of the stimulus captured from various angles. Some heads become specialized to the object while other become specialized to the background features.

6.3 ViT-CoT Architecture

We systematically increased the number of attention heads and layers to create different architecture sizes ranging from smallest (ViT-1H) to largest (ViT-9H) model (Table 1). The ViT-CoT model contains two primary components: the transformer layers and a SimCLR projection head. The projection head consists of two linear layers and a ReLU activation function. Both of these components are sequentially attached to each other. The projection head receives the output from the final transformer layer as its input and generates an embedding z with a shape of 128. The embedding z is used to adjust the loss function when training the model. During testing, the output is obtained directly from the final transformer layer, without passing through the projection head.

6.4 VideoMAE Training

To train the VideoMAEs, we first sampled 16 frames from a batch with a temporal stride of 1. Each frame had a dimension of 64x64 and 3 channels (16x64x64). Next, we randomly masked 90% of the



Supplementary Figure 3: (A) Diagram showing the 2-frame versus 3-frame learning windows. (B) View-invariant recognition performance of ViT-CoT across the four rearing conditions and two temporal learning windows (2 frames or 3 frames). Error bars denote standard error. We used $N_{\text{train}} = 11$ for the linear probe. The models with a 3-frame learning window performed marginally better than those with a 2-frame learning window. (C) View-invariant recognition performance of ViT-CoT (3-frame learning window) across the four rearing conditions and two image resolutions (64x64 or 224x224). We used $N_{\text{train}} = 11$ for the linear probe. ViT-CoT performs well with both low and high-resolution images.

frames using cube masks. Each mask had a spatial dimension of 8x8 and a temporal dimension of 2 (2x8x8). The VideoMAE encoder then encoded the visible (non-masked patches) into a latent space with a dimension of 512. Finally, the VideoMAE decoder combined the encoded features with the masked patches to reconstruct the entire sequence of frames. Each VideoMAE model was initialized with three different seeds and trained for 100 epochs using a batch size of 64. We used multi-GPU training across 8 NVIDIA A10 GPUs.

To ensure that our findings with Video MAE would generalize across different image sizes and masking sizes, we repeated this process with a higher image (16x224x224) and larger masks (2x16x16).

6.5 CNN Training

We utilized the default ResNet-18 architecture but made it more compact by removing the last two residual blocks, resulting in a 10-layer CNN architecture. We employed 'Contrastive Learning through Time' as the teaching signal, similar to ViT-CoT, for training these CNN models. Each model was initialized with three different seeds and trained for 100 epochs with a batch size of 512.

6.6 Two-alternative forced-choice task

In our main experiments, the ViT (encoder) was trained in an unsupervised manner, but we used a supervised linear classifier (decoder) to evaluate the features learned by the ViT. We also tested whether we would obtain similar results when we used an unsupervised, rather than supervised, decoder to evaluate the ViTs. Specifically, we used a modification of the unsupervised technique described by [2], initially developed to test human babies.

The chicks' preferences in [57] can be conceived as a measure of alignment between the test stimuli and the chick's internal representation of their imprinted object. Chicks will approach a stimulus they perceive to be the most similar to their internal representation of their imprinted object (i.e., the stimulus that produces less mismatch, or 'error,' between the stimulus and their representation). To approximate this in silica, we converted each trained ViT model into an autoencoder that was "imprinted" to the same stimulus as the chicks and tested on the same stimuli as the chicks. An autoencoder is trained to reconstruct the original stimulus from a lower dimensional representation, and the reconstruction loss is higher when there is a larger mismatch between a given stimulus and the internal representation.

We converted the models into autoencoders by attaching a simple fully connected downstream decoder to the (trained and frozen) ViT; then we performed unsupervised training on the decoder, using the same images that were used to train the ViT encoder. Consequently, both the encoder and decoder were only trained on images of a single object shown from a single viewpoint range, akin to the chicks. Once the decoder was trained, we used the output from the decoder to quantify how similar each test stimulus was to the ViT's internal representation.

The chicks were tested using a two-alternative forced-choice test (2AFC). To approximate the 2AFC task, we fed two object images into the autoencoder (i.e., ViT + decoder), then measured the error signal for each image. If the error signal was smaller for the imprinted object than the novel object, the model was scored as 'correct.' If the error signal was larger for the imprinted object than the novel object, the model was scored as 'incorrect.' We evaluated the models across all 12 of the viewpoint ranges.

The model scored 62.1%, which is significantly higher than chance level (50%): $X^2(1, N = 576) = 34.03$, $p = .000000005$. This result shows that a purely unsupervised learning model, in which both the encoder (ViT) and decoder are trained without any supervised signals, can learn to solve the same view-invariant recognition task as newborn chicks when trained 'through the eyes' of chicks.

6.7 Data Availability

The code and data needed to reproduce these findings can be found at the following link:https://github.com/buildingamind/ViT-CoT.