Tracking Without Re-recognition in Humans and Machines

Drew Linsley[*1], **Girik Malik**[*2], **Junkyung Kim**[3],

Lakshmi N Govindarajan[1], **Ennio Mingolla**[†2], **Thomas Serre**[†1]
{drew_linsley,lakshmi_govindarajan,thomas_serre}@brown.edu
{malik.gi,e.mingolla}@northeastern.edu
junkyung@deepmind.com

Abstract

Imagine trying to track one particular fruitfly in a swarm of hundreds. Higher biological visual systems have evolved to track moving objects by relying on both appearance and motion features. We investigate if state-of-the-art deep neural networks for visual tracking are capable of the same. For this, we introduce *PathTracker*, a synthetic visual challenge that asks human observers and machines to track a target object in the midst of identical-looking "distractor" objects. While humans effortlessly learn *PathTracker* and generalize to systematic variations in task design, state-of-the-art deep networks struggle. To address this limitation, we identify and model circuit mechanisms in biological brains that are implicated in tracking objects based on motion cues. When instantiated as a recurrent network, our circuit model learns to solve *PathTracker* with a robust visual strategy that rivals human performance and explains a significant proportion of their decision-making on the challenge. We also show that the success of this circuit model extends to object tracking in natural videos. Adding it to a transformer-based architecture for object tracking builds tolerance to visual nuisances that affect object appearance, resulting in a new state-of-the-art performance on the large-scale TrackingNet object tracking challenge. Our work highlights the importance of building artificial vision models that can help us better understand human vision and improve computer vision.

1 Introduction

Lettvin and colleagues [1] presciently noted, "The frog does not seem to see or, at any rate, is not concerned with the detail of stationary parts of the world around him. He will starve to death surrounded by food if it is not moving." Object tracking is fundamental to survival, and higher biological visual systems have evolved the capacity for two distinct and complementary strategies to do it. Consider Figure 1: can you track the object labeled by the yellow arrow from left-to-right? The task is trivial when there are "bottom-up" cues for object appearance, like color, which make it possible to "re-recognize" the target in each frame (Fig. 1a). On the other hand, the task is more challenging when objects cannot be discriminated by their appearance. In this case integration of object motion over time is necessary for tracking (Fig. 1b). Humans are capable of tracking objects by their motion when appearance is uninformative [2, 3], but it is unclear if the current generation

[*†]These authors contributed equally to this work.
[1]Carney Institute for Brain Science, Brown University, Providence, RI
[2]Northeastern University, Boston, MA
[3]DeepMind, London, UK

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

of neural networks for video analysis and tracking can do the same. To address this question we introduce *PathTracker*, a synthetic challenge for object tracking without re-recognition (Fig. 1c).

Leading models for video analysis rely on object classification pre-training. This gives them access to rich semantic representations that have supported state-of-the-art performance on a host of tasks, from action recognition to object tracking [**?**, **?**, 4]. As object classification models have improved, so too have the video analysis models that depend on them. This trend in model development has made it unclear if video analysis models are effective at learning tasks when appearance is uninformative. The importance of diverse visual strategies has been highlighted by synthetic challenges like *Pathfinder*, a visual reasoning task that asks observers to trace long paths embedded in a static cluttered display [**?**, 5]. *Pathfinder* tests object segmentation when appearance cues like category or shape are missing. While humans can easily solve it [5], feedforward neural networks struggle, including state-of-the-art vision transformers [**?**, **?**, 5]. Importantly, models that learn an appropriate visual strategy for *Pathfinder* are also quicker learners and better at generalization for object segmentation in natural images [**?**, 6]. Our *PathTracker* challenge extends this line of work into video by posing an object tracking problem where the target can be tracked by motion and spatiotemporal continuity, not category or appearance.

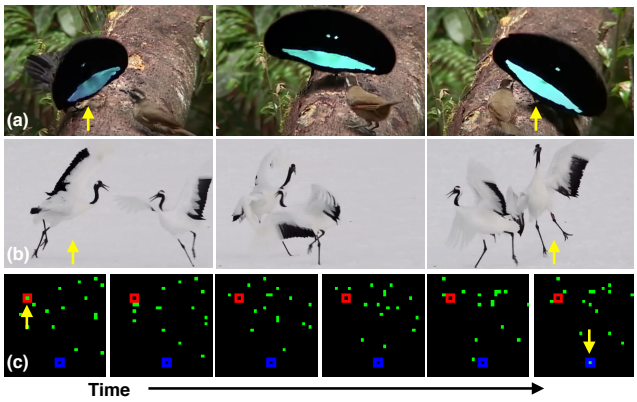

Figure 1: The appearance of objects makes them (*a*) easy or (*b*) hard to track. We introduce the *PathTracker* Challenge (*c*), which asks observers to track a particular green dot as it travels from the red-to-blue markers, testing object tracking when re-recognition is impossible.

Contributions. Humans effortlessly solve our novel *PathTracker* challenge. A variety of state-of-the-art models for object tracking and video analysis do not.

- We find that neural architectures including R3D [**?**] and state-of-the-art transformer-based TimeSformers [**?**] are strained by long *PathTracker* videos. Humans, on the other hand, are far more effective at solving these long *PathTracker* videos.
- We describe a solution to *PathTracker*: a recurrent model inspired by primate neural circuitry involved in object tracking, whose decisions that are strongly correlated with those of humans.
- These same circuit mechanisms improve object tracking in natural videos through a motion-based strategy that builds tolerance to changes in target object appearance, resulting in the certified top score on TrackingNet [7] at the time of this submission.
- We release all *PathTracker* data, code, and human psychophysics at `http://bit.ly/InTcircuit` to spur interest in the challenge of tracking without re-recognition.

2 Related Work

Shortcut learning and synthetic datasets A byproduct of the great power of deep neural network architectures is their vulnerability to learning spurious correlations between inputs and labels. Perhaps because of this tendency, object classification models have trouble generalizing to novel contexts [8,9], and render idiosyncratic decisions that are inconsistent with humans [**?**, 10, 11]. Synthetic datasets are effective at probing this vulnerability because they make it possible to control spurious image/label correlations and fairly test the computational abilities of models. For example, the *Pathfinder* challenge was designed to test if neural architectures can trace long curves despite gaps – a visual computation associated with the earliest stages of visual processing in primates. That challenge identified diverging visual strategies between humans and transformers that are otherwise state of the art in natural image object recognition [**?**, **?**]. Other challenges like Bongard-LOGO [**?**], cABC [5], and PSVRT [12] have highlighted limitations of leading neural network architectures that would have been difficult to identify using natural image benchmarks like ImageNet [13]. These limitations have inspired algorithmic solutions based on neural circuits discussed in SI §A.

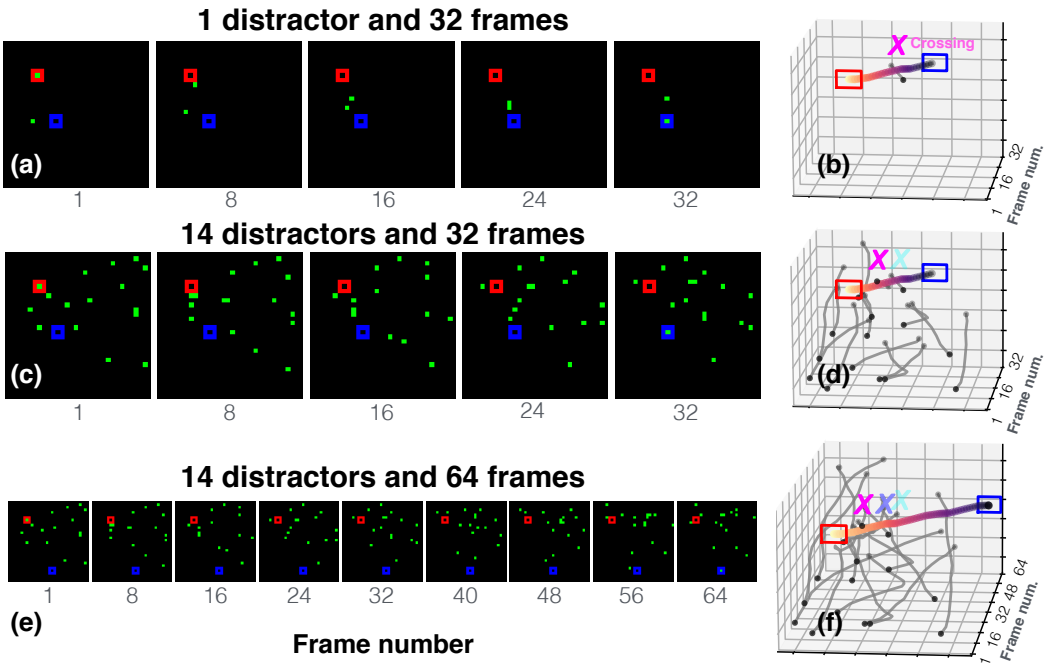

Figure 2: *PathTracker* is a synthetic visual challenge that asks observers to watch a video clip and answer if a target dot starting in a red marker travels to a blue marker. The target dot is surrounded by identical "distractor" dots, each of which travels in a randomly generated and curved path. In positive examples, the target dot's path ends in the blue square. In negative examples, a "distractor" dot ends in the blue square. The challenge of the task is due to the identical appearance of target and distractor dots, which makes appearance-based tracking strategies ineffective. Moreover, the target dot can momentarily occupy the same location as a distractor when they cross each other's paths, making them impossible to individuate in that frame and compelling strategies like motion trajectory extrapolation or working memory to recover the target track. (*b*) A 3D visualization of the video in (*a*) depicts the trajectory of the target dot, traveling from red-to-blue markers. The target and distractor cross approximately half-way through the video. (*c,d*) We develop versions of *PathTracker* that test observer sensitivity to the number distractors and length of videos (*e,f*). The number of distractors and video length interact to make it more likely for the target dot to cross a distractor in a video (compare the one X in *b* vs. two in *d* vs. three in *f*; see SI §B for details).

Models for video analysis A major leap in the performance of models for video analysis came from using networks which are pre-trained for object recognition on large image datasets [**?**]. The recently introduced TimeSformer [**?**] achieved state-of-the-art performance with weights initialized from an image categorization transformer (ViT; [**?**]) that was pre-trained on ImageNet-21K. The story is similar in object tracking [14], where successful models rely on "backbone" feature extraction networks trained on ImageNet or Microsoft COCO [**?**] for object recognition or segmentation [4, 15].

3 The *PathTracker* Challenge

Overview *PathTracker* asks observers to decide whether or not a target dot reaches a goal location (Fig. 2). The target dot travels in the midst of a pre-specified number of distractors. All dots are identical, and the task is difficult because of this: (*i*) observers cannot rely on appearance to track the target, and (*ii*) the paths of the target and distractors can momentarily "cross" and occupy the same space, making them impossible to individuate in that frame and meaning that observers cannot only rely on target location to solve the task. This challenge is inspired by object tracking paradigms of cognitive psychology [2, 3], which suggest that humans might rely on mechanisms for motion perception, attention and working memory to solve a task like *PathTracker*.

The trajectories of target and distractor dots are randomly generated, and the target occasionally crosses distractors (Fig. 2). These object trajectories are smooth by design, giving the appearance of objects meandering through a scene, and the difference between the coordinates of any dot on

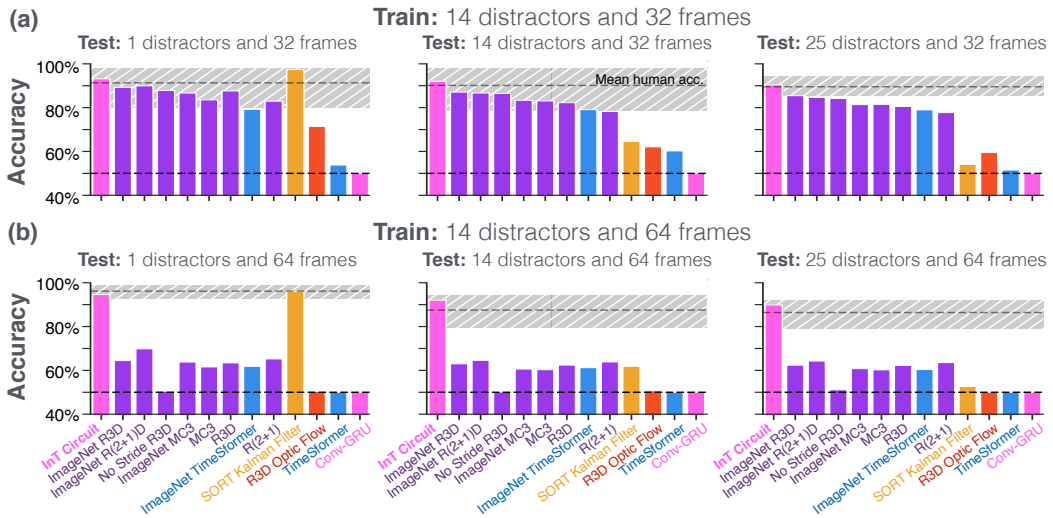

Figure 3: Model accuracy on the *PathTracker* challenge. Video analysis models were trained to solve 32 (*a*) and 64 frame (*b*) versions of challenge, which featured the target object and 14 identical distractors. Models were tested on *PathTracker* datasets with the same number of frames but 1, 14, or 25 distractors (left/middle/right). Grey hatched boxes denote 95% bootstrapped confidence intervals for humans. Only our InT Circuit rivaled humans on each dataset.

successive frames is no more than 2 pixels with less than 20° of angular displacement. In other words, dots never turn at acute angles. We develop different versions of *Pathtracker* which we expect to be more or less difficult by adjusting the number of distractors and/or the length of videos. These variables change the expected number of times that distractors cross the target and the amount of time that observers must track the target (Fig. 2). To make the task as visually simple as possible and maximize contrast between dots and markers, the dots, start, and goal markers are placed on different channels in 32×32 pixel three-channel images. Markers are stationary throughout each video and placed at random locations. Examples videos can be viewed at `http://bit.ly/InTcircuit`.

Human benchmark We began by testing if humans can solve *PathTracker*. We recruited 180 individuals using Amazon Mechanical Turk to participate in this study. Participants viewed *PathTracker* videos and pressed a button on their keyboard to indicate if the target object or a distractor reached the goal. These videos were played in web browsers at 256×256 pixels using HTML5, which helped ensure consistent framerates [16]. The experiment began with an 8 trial "training" stage, which familiarized participants with the goal of *PathTracker*. Next, participants were tested on 72 videos. The experiment was not paced and lasted approximately 25 minutes, and participants were paid $8 for their time. See `http://bit.ly/InTcircuit` and SI §B for an example and more details.

Participants were randomly entered into one of two experiments. In the first experiment, they were trained on the 32 frame and 14 distractor *PathTracker*, and tested on 32 frame versions with 1, 14, or 25 distractors. In the second experiment, they were trained on the 64 frame and 14 distractor *PathTracker*, and tested on 64 frame versions with 1, 14, or 25 distractors. All participants viewed unique videos to maximize our sampling over the different versions of *PathTracker*. Participants were significantly above chance on all tested conditions of *PathTracker* ($p < 0.001$, test details in SI §B). They also exhibited a significant negative trend in performance on the 64 frame datasets as the number of distractors increased ($t = -2.74$, $p < 0.01$). There was no such trend on the 32 frame datasets, and average accuracy between the two datasets was not significantly different. These results validate our initial design assumptions: humans can solve *PathTracker*, and manipulating distractors and video length increases difficulty.

4 Solving the *PathTracker* challenge

Can state-of-the-art models for video analysis match humans on *PathTracker*? To test this question we surveyed a variety of architectures that are the basis for leading approaches to many video analysis tasks, from object tracking to action classification. We restricted our survey to models that

could be trained end-to-end to solve *PathTracker* without any additional pre- or post-processing steps. The selected models fall into three groups: (*i*) deep convolutional networks (CNNs), (*ii*) transformers, and (*iii*) recurrent neural networks (RNNs). The deep convolutional networks include a 3D ResNet (R3D [**?**]), a space/time separated ResNet with "2D-spatial + 1D-temporal" convolutions (R(2+1)D [**?**]), and a ResNet with 3D convolutions in early residual blocks and 2D convolutions in later blocks (MC3 [**?**]). We trained versions of these models with random weight initializations and weights pretrained on ImageNet. We included an R3D trained from scratch without any downsampling, in case the small size of *PathTracker* videos caused learning problems (see SI §C for details). We also trained a version of the R3D on optic flow encodings of *PathTracker* (SI §C). For transformers, we turned to the TimeSformer [**?**]. We test two of its instances: (*i*) attention is jointly computed for all locations across space and time in videos, and (*ii*) temporal attention is applied before spatial attention, which results in massive computational savings. We found similar *PathTracker* performance with both models. We report the latter version here as it was marginally better (see SI §C for performance of the other, joint space-time attention TimeSformer). We include a version of the TimeSformer trained from scratch, and a version pre-trained on ImageNet-20K. Note that state-of-the-art transformers for object tracking in natural videos feature similar deep and multi-headed designs [4] but use additional post-processing steps that are beyond the scope of *PathTracker*. Finally, we include a convolutional-gated recurrent unit (Conv-GRU) [17].

Method The visual simplicity of *PathTracker* cuts two ways: it makes it possible to compare human and model strategies for tracking without re-recognition as long as the task is not too easy. Prior synthetic challenges like *Pathfinder* constrain sample sizes for training to probe specific computations [**?**, **?**, 5]. We adopt the following strategy to select a training set size that would help us test tracking strategies that do not depend on re-recognition. We took Inception 3D (I3D) networks [**?**], which have been a strong baseline architecture in video analysis over the past several years, and tested their ability to learn *PathTracker* as we adjusted the number of videos for training. As we discuss in SI §A, when this model was trained with 20K examples of the 32 frame and 14 distractor version of *PathTracker* it achieved good performance on the task without signs of overfitting to its simple visual statistics. We therefore train all models in subsequent experiments with 20K examples.

We measure the ability of models to learn *PathTracker* and systematically generalize to novel versions of the challenge when trained on 20K samples. We trained models using a similar approach as in our human psychophyics. Models were trained on one version of pathfinder, and tested on other versions with the same number of frames, and the same or different number of distractors. In the first experiment, models were trained on the 32 frame and 14 distractor *PathTracker*, then tested on the 32 frame *PathTracker* datasets with 1, 14, or 25 distractors (Fig. 3a). In the second experiment, models were trained on the 64 frame and 14 distractor *PathTracker*, then tested on the 64 frame *PathTracker* datasets with 1, 14, or 25 distractors (Fig. 3a). Models were trained to detect if the target dot reached the blue goal marker using binary crossentropy and the Adam optimizer [18] until performance on a test set of 20K videos with 14 distractors decreased for 200 straight epochs. In each experiment, we selected model weights that performed best on the 14 distractor dataset. Models were retrained three times on learning rates $\in \{1\text{e-}2, 1\text{e-}3, 1\text{e-}4, 3\text{e-}4, 1\text{e-}5\}$ to optimize performance. The best performing model was then tested on the remaining 1 and 25 distractor datasets in the experiment. We used four NVIDIA GTX GPUs and a batch size 180 for training.

Results We treat human performance as the benchmark for models on *PathTracker*. Nearly all CNNs and the ImageNet-initialized TimeSformer performed well enough to reach the 95% human confidence interval on the 32 frame and 14 distractor *PathTracker*. However, all models performed worse when systematically generalizing to *PathTracker* datasets with a different number of distractors, even when that number decreased (Fig. 3a, 1 distractor). Model performance on the 32 frame *PathTracker* datasets was worst for 25 distractors. No CNN or transformer reached the 95% confidence interval of humans on this dataset (Fig. 3a). The optic flow R3D and the TimeSformer trained from scratch were even less successful but still above chance, while the Conv-GRU performed at chance. Model performance plummeted across the board on 64 frame *PathTracker* datasets. The drop in model performance from 32 to 64 frames reflects a combination of the following features of *PathTracker*. (*i*) The target becomes more likely to cross a distractor when length and the number of distractors increase (Fig. 2; Fig. 2c). This makes the task difficult because the target is momentarily impossible to distinguish from a distractor. (*ii*) The target object must be tracked from start-to-end to solve the

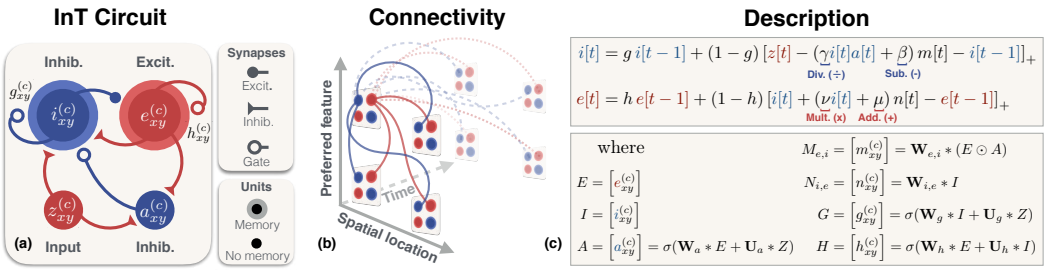

Figure 4: The Index-and-Track (InT) circuit model is inspired by Neuroscience models of motion perception [19] and executive cognitive function [20]. (*a*) The circuit receives input encodings from a video (z), which are processed by interacting recurrent inhibitory and excitatory units (i, e), and a mechanism for selective "attention" (a) that tracks the target location. (*b*) InT units have spatiotemporal receptive fields. Spatial connections are formed by convolution with weight kernels (W_e, W_i). Temporal connections are controlled by gates (g, h). (*c*) Model parameters are fit with gradient descent. Softplus= $[.]$, sigmoid= σ, convolution= $*$, elementwise product = $\odot$.

task, which can incur a memory cost that is monotonic w.r.t. video length. (*iii*) The prior two features interact to non-linearly increase task difficulty (Fig. 2c).

Neural circuits for tracking without re-recognition *PathTracker* is inspired by object tracking paradigms from Psychology, which tested theories of working memory and attention in human observers [2, 3]. *PathTracker* may draw upon similar mechanisms of visual cognition in humans. However, the video analysis models that we include in our benchmark (Fig. 3) do not have inductive biases for working memory, and while the TimeSformer uses a form of attention, it is insufficient for learning *PathTracker* and only reached human performance on one version of the challenge (Fig. 3).

Neural circuits for motion perception, working memory, and attention have been the subject of intense study in Neuroscience for decades. Knowledge synthesized from several computational, electrophysiological and imaging studies point to canonical features and computations that are carried out by these circuits. (*i*) Spatiotemporal feature selectivity emerges from non-linear and time-delayed interactions between neuronal subpopulations [21, 22]. (*ii*) Recurrently connected neuronal clusters can maintain task information in working memory [20, 23]. (*iii*) Synaptic gating, inhibitory modulation, and disinhibitory circuits are neural substrates of working memory and attention [24–28]. (*iv*) Mechanisms for gain control may aid motion-based object tracking by building tolerance to visual nuisances, such as illumination [29, 30]. We draw from these principles to construct the "Index-and-Track" circuit (InT, Fig. 4).

InT circuit description The InT circuit takes an input z at location x, y and feature channel c from video frame $t \in T$ (Fig. 4a). This input is passed to an inhibitory unit i, which interacts with an excitatory unit e, both of which have persistent states that store memories with the help of gates g, h. The inhibitory unit is also gated by another inhibitory unit, a, which is a non-linear function of e, and can either decrease or increase (i.e., through disinhibition) the inhibitory drive. In principle, the sigmoidal nonlinearity of a means that it can selectively attend, and hence, we refer to a as "attention". Moreover, since a is a function of e, which lags in time behind $z[t]$, its activity reflects the displacement (or motion) of an object in $z[t]$ versus the current memory of e. InT units have spatiotemporal receptive fields (Fig. 4b). Interactions between units at different locations are computed by convolution with weight kernels $\mathbf{W}_{e,i}, \mathbf{W}_{i,e} \in \mathbb{R}^{5,5,c,c}$ and attention is computed by $\mathbf{W}_a \in \mathbb{R}^{1,1,c,c}$. Gate activities that control InT dynamics and temporal receptive fields are similarly calculated by kernels, $\mathbf{W}_g, \mathbf{W}_h, \mathbf{U}_g, \mathbf{U}_h \in \mathbb{R}^{1,1,c,c}$. Recurrent units in the InT support non-linear (gain) control. Inhibitory units can perform divisive and subtractive computations, controlled by γ, β. Excitatory units can perform multiplicative and additive computations, controlled by ν, μ. Parameters $\gamma, \beta, \nu, \mu \in \mathbb{R}^c$. "SoftPlus" rectifications denoted by $[.]_+$ enforce inhibitory and excitatory function and competition (Fig. 4c). The final e state is passed to a readout for *PathTracker* (SI §D).

InT *PathTracker* performance We trained the InT on *PathTracker* following the procedure in §4. It was the only model that rivaled humans on each version of *PathTracker* (Fig. 3). The gap in performance between InT and the field is greatest on the 64 frame version of the challenge.

How does the InT solve *PathTracker*? There are at least two strategies that it could choose from. One is to maintain a perfect track of the target throughout its trajectory, and extrapolate the momentum of

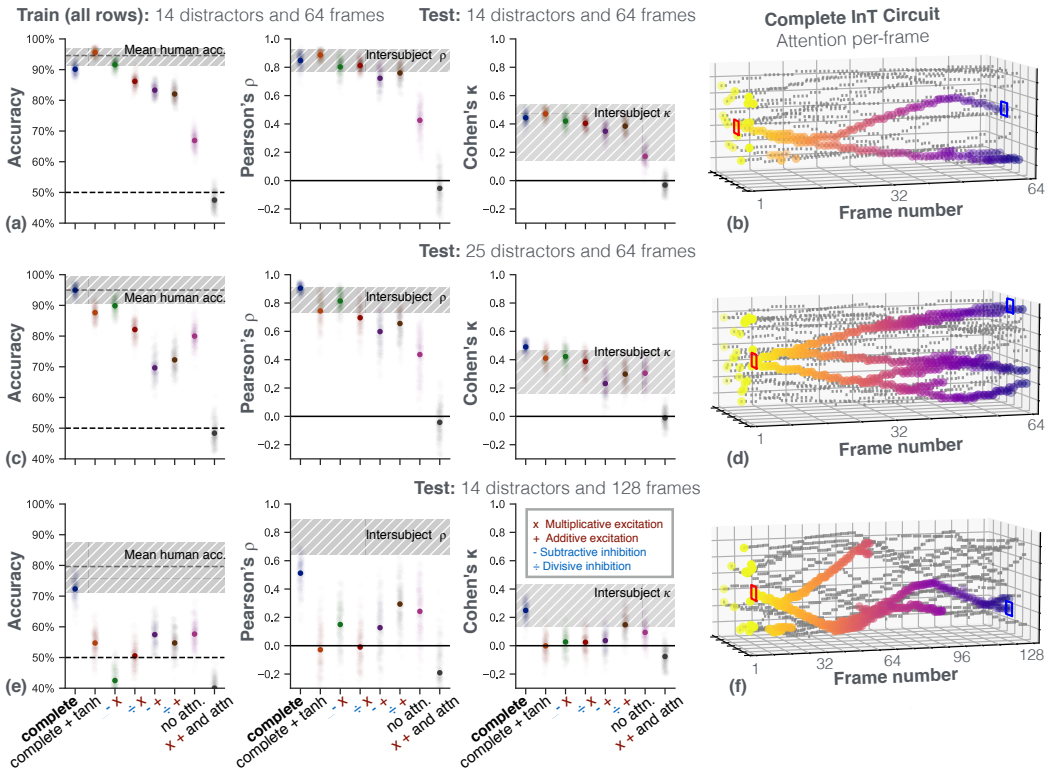

Figure 5: Performance, decision correlations, and error consistency between models and humans on *PathTracker*. In a new set of psychopysics experiments, humans and models were trained on 64 frame *PathTracker* datasets with 14 distractors, and rendered decisions on a variety of challenging versions. Decision correlations are computed with Pearson's ρ, and error consistency with Cohen's κ [**?**]. Only the Complete InT circuit rivals human performance and explains the majority of their decision and error variance on each test dataset (*a,c,e*). Visualizing InT attention (a) reveals that it has learned to solve *PathTracker* by multi-object tracking (*b,d,f*; color denotes time). The consistency between InT and human decisions raises the possibility that humans rely on a similar strategy.

its motion to resolve crossings with distractors. Another is to track all objects that cross the target and check if any of them reach the goal marker by the end of the video. To investigate the type of strategy learned by the InT for *PathTracker* and to compare this strategy to humans, we ran additional psychophysics with a new group of 90 participants using the same setup detailed in §3. Participants were trained on 8 videos from the 14 distractor and 64 frame *PathTracker* and tested on 72 images from either the (*i*) 14 distractor and 64 frame dataset, (*ii*) 25 distractor and 64 frame dataset, or (*iii*) 14 distractor and 128 frame dataset. Unlike the psychophysics in §3, all participants viewing a given test set saw the same videos, which made it possible to compare their decision strategies with the InT.

InT performance reached the 95% confidence intervals of humans on each test dataset. The InT also produced errors that were extremely consistent with humans and explained nearly all variance in Pearson's ρ and Cohen's κ on each dataset (Fig. 5, middle and right columns). This result means that humans and InT rely on similar strategies for solving *PathTracker*.

What is the underlying strategy? We visualized activity of A units in the InT as they processed *PathTracker* videos and found that they had learned a multi-object tracking strategy to solve the task (Fig. 5; see SI §E for the method, and `http://bit.ly/InTcircuit` for animations). The A units track the target object until it crosses a distractor and ambiguity arises, at which point attention splits and it tracks both objects. This strategy indexes a limited number of objects at once, consistent with studies of object tracking in humans [2]. Since the InT is not explicitly constrained for this tracking strategy, we next investigated the minimal circuit for learning it and explaining human behavior.

We developed versions of the InT with lesions applied to different combinations of its divisive/subtractive and multiplicative/additive computations, a version without attention units A, and a version that does not make a distinction of inhibition vs. excitation ("complete + tanh"), in which

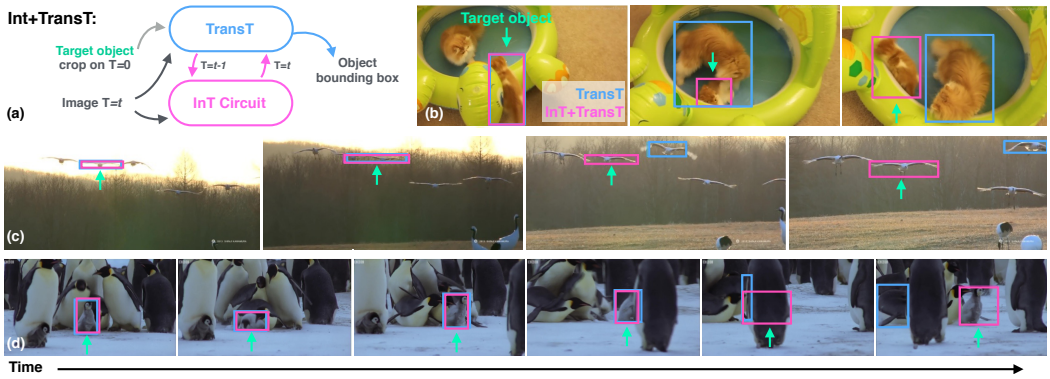

Figure 6: Circuit mechanisms for tracking without re-recognition build tolerance to visual nuissances that affect object appearance. (*a*) The TransT [31] is a transformer architecture for object tracking. We develop an extension, the InT+TransT, in which our InT circuit model recurrently modulates TransT activity. Unlike the TransT, the InT+TransT is trained on sequences to promote tracking strategies that do not rely on re-recognition. (*b-d*) The InT+TransT excels when the target object is visually similar to other depicted objects, undergoes changes in illumination, or is occluded.

rectifications were replaced with hyperbolic tangents that squash activities into $[-1, 1]$. While some of these models marginally outperformed the Complete InT on the 14 distractor and 64 frame dataset, their performance dropped precipitously on the 25 distractor and 64 frame dataset, and especially the very long 14 distractor and 128 frame dataset (Fig. 5e). Attention units in the complete InT's nearest rival (complete + tanh) were non-selective, potentially contributing to its struggles. InT performance also dropped when we forced it to attend to fewer objects (SI §F).

5 Appearance-free mechanisms for object tracking in the wild

The InT solves *PathTracker* by learning to track multiple objects at once, without relying on the re-recognition strategy that has been central to progress in video analysis challenges in computer vision. However, it is not clear if tracking without re-recognition is useful in the natural world. We test this question by turning to object tracking in natural videos. At the time of writing, the state-of-the-art object tracker is the TransT [31], a deep multihead transformer [32]. The TransT finds pixels in a video frame that match the appearance of an image crop depicting a target object. During training, the TransT receives a tuple of inputs, consisting of this target object image and a random additional "search frame" from the same video. These images are encoded with a modified ResNet50 [33], passed to separate transformers, and finally combined by a "cross-feature attention" (CFA) module, which compares the two encodings via a transformer key/query/value computation. The target frame is used for key and value operations, and the search frame is used for the query operation. Through its pure appearance-based approach to tracking, the TransT has achieved top performance on TrackingNet [7], LaSOT [**?**], and GOT-10K [**?**].

InT+TransT We tested whether or not the InT circuit can improve TransT performance by learning a complementary object strategy that does not depend on appearance, or re-recognition. We reasoned that this strategy might help TransT tracking in cases where objects are difficult to discern by their appearance, such as when they are subject to changing lighting, color, or occlusion. We thus developed the InT+TransT, which involves the following modifications of the original TransT (Fig. 6a). (*i*) We introduce two InT circuits to form a bottom-up and top-down feedback loop with the TransT [6, 34], which in principle will help the model select the appropriate tracking strategy depending on the video – re-recognition or not. One InT receives ResNet50 search image encodings and modulates the TransT's CFA encoding of this search image. The other receives the output of the TransT and uses this information to update memory in the first InT. (*ii*) The TransT is trained with pairs of target and search video frames, separated in time by up to 100 frames. We introduce the intervening frames to the InT circuits. See SI §F for extended methods.

Training InT+TransT training and evaluation hews close to the TransT procedure. This includes training on the latest object tracking challenges in computer vision: TrackingNet [7], LaSOT [**?**], and GOT-10K [**?**]. All three challenges depict diverse classes of objects, moving in natural scenes

Model	TrackingNet [7]	LaSOT [?]	GOT [?]	GOT		
				Color	rColor	Occl.
InT+TransT$_{T=8}$	**87.5**	74.0	72.2	43.1	62.5	56.9
InT+TransT$_{T=1}$	87.3	73.6	70.0	36.2	37.8	25.4
TransT [31]	86.7	73.8	72.3	40.7	57.5	55.2

Table 1: Model performance on TrackingNet (P$_{norm}$), LaSot (P$_{norm}$), GOT-10K (AO), and perturbations applied to the GOT-10K (AO). Best performance is in black, and certified state of the art is bolded. Perturbations on the GOT-10K are color inversions on every frame ($Color$) or random frames ($rColor$), and random occluders created from scrambling image pixels ($Occl.$). InT+TransT$_{T=8}$ was trained on sequences of 8 frames, and InT+TransT$_{T=1}$ was trained on 1-frame sequences.

that range from simplistic and barren to complex and cluttered. TrackingNet (30,132 train and 511 test videos) and GOT-10K (10,000 train and 180 test) evaluation is performed on official challenge servers, whereas LaSOT (1,120 train and 280 test) is evaluated with a Matlab toolbox. While the TransT is also trained with static images from Microsoft COCO [**?**], in which the search image is an augmented version of the target, we do not include COCO in InT+TransT since we expect object motion to be an essential feature for our model [31]. The InT+TransT is initialized with TransT weights and trained with AdamW [35] and a learning rate of $1e{-}4$ for InT parameters, and $1e{-}6$ for parameters in the TransT readout and CFA module. Other TransT parameters are frozen and not trained. The InT+TransT is trained with the same objective functions as the TransT for target object bounding box prediction in the search frame, and an additional objective function for bounding box prediction using InT circuit activity in intervening frames. The complete model was trained with batches of 24 videos on 8 NVIDIA GTX GPUs for 150 epochs (2 days). We selected the weights that performed best on GOT-10K validation. A hyperparameter controls the number of frames between the target and search that are introduced into the InT during training. We relied on coarse sampling (1 or 8 frames) due to memory issues associated with recurrent network training on long sequences [**?**].

Results An InT+TransT trained on sequences of 8 frames performed inference around 30FPS on a single NVIDIA GTX and beat the TransT on nearly all benchmarks. It is in first place on the TrackingNet leaderboard (`http://eval.tracking-net.org/`), better than the TransT on LaSOT, and rivals the TransT on the GOT-10K challenge (Table 5). The InT+TransT performed better when trained with longer sequences (compare $T = 8$ and $T = 1$, Table 5). Consistent with InT success on *PathTracker*, the InT+TransT was qualitatively better than the TransT on challenging videos where the target interacted with other similar looking objects (Fig. 6; `http://bit.ly/InTcircuit`).

We also found that the InT+TransT excelled in other challenging tracking conditions. The LaSOT challenge provides annotations for challenging video features, which reveal that the InT+TransT is especially effective for tracking objects with "deformable" parts, such as moving wings or tails (SI §F). We further test if introducing object appearance perturbations to the GOT-10K might distinguish performance between the TransT and InT+TransT. We evaluate these models on the GOT-10K test set with one of three perturbations: inverting the color of all search frames (Color), inverting the color of random search frames (rColor), or introducing random occlusions (Occl.). The InT+TransT outperformed the TransT on each of these tests (Table 5).

6 Discussion

A key inspiration for our study is the centrality of visual motion and tracking across a broad phylogenetic range, via three premises: (*i*) Object motion integration over time *per se* is essential for ecological vision and survival [1]. (*ii*) Object motion perception cannot be completely reduced to recognizing similar appearance features at two different moments in time. In perceptual phenomena like *phi* motion, the object that is tracked is described as "formless" with no distinct appearance [36]. (*iii*) Motion integration over space and time is a basic operation of neural circuits in biological brains, which can be independent of appearance [37]. These three premises form the basis for our work.

We developed *PathTracker* to test whether state-of-the-art models for video analysis can solve a visual task when object appearance is ambiguous. Prior visual reasoning challenges like *Pathfinder* [**?**, **?**, 5], indicate that this is a problem for object recognition models, which further serve as a backbone for many video analysis models. While no existing model was able to contend with humans on *Pathfinder*, our InT circuit was. Through lesioning experiments, we discovered that the InT's ability to explain human behavior depends on its full array of inductive biases, helping it learn a visual

strategy that indexes and tracks a limited number of the objects at once, echoing classic theories on the role of attention and working memory in object tracking [2, 3].

We further demonstrate that the capacity for video analysis without relying on re-recognition helps in natural scenes. Our InT+TransT model is more capable than the TransT at tracking objects when their appearance changes, and is the state of the art on the TrackingNet challenge. Together, our findings demonstrate that object appearance is a necessary element for for video analysis, but it is not sufficient for modeling biological vision and rivaling human performance.

Acknowledgments and Disclosure of Funding

We are grateful to Daniel Bear and Matthew Ricci for their suggestions to improve this work. We would also like to thank Rajan Girsa for help on Mechanical Turk experiments. GM is affiliated with Labrynthe Pvt. Ltd., New Delhi, India. Funding provided by ONR grant #N00014-19-1-2029, the ANR-3IA Artificial and Natural Intelligence Toulouse Institute, and ANITI (ANR-19-PI3A-0004). Additional support from the Brown University Carney Institute for Brain Science and grant S10OD025181, the Center for Computation in Brain and Mind (CCBM), and the Center for Computation and Visualization (CCV).

References

[1] Lettvin, J.Y., Maturana, H.R., McCulloch, W.S., Pitts, W.H.: What the frog's eye tells the frog's brain. Proceedings of the IRE **47**(11) (November 1959) 1940–1951

[2] Pylyshyn, Z.W., Storm, R.W.: Tracking multiple independent targets: Evidence for a parallel tracking mechanism. Spat. Vis. **3**(3) (1988) 179–197

[3] Blaser, E., Pylyshyn, Z.W., Holcombe, A.O.: Tracking an object through feature space. Nature **408**(6809) (November 2000) 196–199

[4] Wang, N., Zhou, W., Wang, J., Li, H.: Transformer meets tracker: Exploiting temporal context for robust visual tracking. (March 2021)

[5] Kim*, J., Linsley*, D., Thakkar, K., Serre, T.: Disentangling neural mechanisms for perceptual grouping. International Conference on Representation Learning (2020)

[6] Linsley, D., Kim, J., Ashok, A., Serre, T.: Recurrent neural circuits for contour detection. International Conference on Learning Representations (2020)

[7] Müller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: TrackingNet: A Large-Scale dataset and benchmark for object tracking in the wild. In: Computer Vision – ECCV 2018, Springer International Publishing (2018) 310–327

[8] 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. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., Garnett, R., eds.: Advances in Neural Information Processing Systems 32. Curran Associates, Inc. (2019) 9453–9463

[9] Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11) (November 2020) 665–673

[10] Ullman, S., Assif, L., Fetaya, E., Harari, D.: Atoms of recognition in human and computer vision. Proc. Natl. Acad. Sci. U. S. A. **113**(10) (March 2016) 2744–2749

[11] Linsley, D., Eberhardt, S., Sharma, T., Gupta, P., Serre, T.: What are the visual features underlying human versus machine vision? In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). (October 2017) 2706–2714

[12] Kim, J., Ricci, M., Serre, T.: Not-So-CLEVR: learning same-different relations strains feedforward neural networks. Interface Focus **8**(4) (August 2018) 20180011

[13] 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. (June 2009) 248–255

[14] Fiaz, M., Mahmood, A., Jung, S.K.: Tracking noisy targets: A review of recent object tracking approaches. (February 2018)

[15] Bertasius, G., Torresani, L.: Classifying, segmenting, and tracking object instances in video with mask propagation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (June 2020) 9736–9745

[16] Eberhardt, S., Cader, J.G., Serre, T.: How deep is the feature analysis underlying rapid visual categorization? In Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., eds.: Advances in Neural Information Processing Systems 29. Curran Associates, Inc. (2016) 1100–1108

[17] Bhat, G., Danelljan, M., Van Gool, L., Timofte, R.: Know your surroundings: Exploiting scene information for object tracking. In: Computer Vision – ECCV 2020, Springer International Publishing (2020) 205–221

[18] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. (December 2014)

[19] Berzhanskaya, J., Grossberg, S., Mingolla, E.: Laminar cortical dynamics of visual form and motion interactions during coherent object motion perception. Spat. Vis. **20**(4) (2007) 337–395

[20] Wong, K.F., Wang, X.J.: A recurrent network mechanism of time integration in perceptual decisions. J. Neurosci. **26**(4) (January 2006) 1314–1328

[21] Takemura, S.Y., Bharioke, A., Lu, Z., Nern, A., Vitaladevuni, S., Rivlin, P.K., Katz, W.T., Olbris, D.J., Plaza, S.M., Winston, P., Zhao, T., Horne, J.A., Fetter, R.D., Takemura, S., Blazek, K., Chang, L.A., Ogundeyi, O., Saunders, M.A., Shapiro, V., Sigmund, C., Rubin, G.M., Scheffer, L.K., Meinertzhagen, I.A., Chklovskii, D.B.: A visual motion detection circuit suggested by drosophila connectomics. Nature **500**(7461) (August 2013) 175–181

[22] Kim, J.S., Greene, M.J., Zlateski, A., Lee, K., Richardson, M., Turaga, S.C., Purcaro, M., Balkam, M., Robinson, A., Behabadi, B.F., Campos, M., Denk, W., Seung, H.S., the EyeWirers: Space–time wiring specificity supports direction selectivity in the retina. Nature **509** (May 2014) 331

[23] Elman, J.L.: Finding structure in time. Cogn. Sci. **14**(2) (April 1990) 179–211

[24] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8) (November 1997) 1735–1780

[25] O'Reilly, R.C., Frank, M.J.: Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. Neural Comput. **18**(2) (February 2006) 283–328

[26] Badre, D.: Opening the gate to working memory. Proc. Natl. Acad. Sci. U. S. A. **109**(49) (December 2012) 19878–19879

[27] D'Ardenne, K., Eshel, N., Luka, J., Lenartowicz, A., Nystrom, L.E., Cohen, J.D.: Role of prefrontal cortex and the midbrain dopamine system in working memory updating. Proc. Natl. Acad. Sci. U. S. A. **109**(49) (December 2012) 19900–19909

[28] Mitchell, J.F., Sundberg, K.A., Reynolds, J.H.: Differential attention-dependent response modulation across cell classes in macaque visual area V4. Neuron **55**(1) (July 2007) 131–141

[29] Berzhanskaya, J., Grossberg, S., Mingolla, E.: Laminar cortical dynamics of visual form and motion interactions during coherent object motion perception. Spat. Vis. **20**(4) (2007) 337–395

[30] Mély, D.A., Linsley, D., Serre, T.: Complementary surrounds explain diverse contextual phenomena across visual modalities. Psychol. Rev. **125**(5) (October 2018) 769–784

[31] Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. (March 2021)

[32] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł.U., Polosukhin, I.: Attention is all you need. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., eds.: Advances in Neural Information Processing Systems. Volume 30., Curran Associates, Inc. (2017)

[33] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. (December 2015)

[34] Gilbert, C.D., Li, W.: Top-down influences on visual processing. Nat. Rev. Neurosci. **14**(5) (May 2013) 350–363

[35] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. (November 2017)

[36] Steinman, R.M., Pizlo, Z., Pizlo, F.J.: Phi is not beta, and why wertheimer's discovery launched the gestalt revolution. Vision Res. **40**(17) (August 2000) 2257–2264

[37] Huk, A.C., Shadlen, M.N.: Neural activity in macaque parietal cortex reflects temporal integration of visual motion signals during perceptual decision making. J. Neurosci. **25**(45) (November 2005) 10420–10436

[38] Kosiorek, A.R., Bewley, A., Posner, I.: Hierarchical attentive recurrent tracking. (June 2017)

[39] Tschopp, F.D., Reiser, M.B., Turaga, S.C.: A connectome based hexagonal lattice convolutional network model of the drosophila visual system. (June 2018)

[40] Fortmann, T.E., Bar-Shalom, Y., Scheffe, M.: Multi-target tracking using joint probabilistic data association. In: 1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes. (December 1980) 807–812

[41] Milan, A., Leal-Taixe, L., Reid, I., Roth, S., Schindler, K.: MOT16: A benchmark for Multi-Object tracking. (March 2016)

[42] Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. (June 2008) 1–8

[43] Dicle, C., Camps, O.I., Sznaier, M.: The way they move: Tracking multiple targets with similar appearance. In: 2013 IEEE International Conference on Computer Vision. (December 2013) 2304–2311

[44] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Kim, T.K.: Multiple object tracking: A literature review. Artif. Intell. **293** (April 2021) 103448

[45] Shimamura, A.P.: Toward a cognitive neuroscience of metacognition. Conscious. Cogn. **9**(2 Pt 1) (June 2000) 313–23; discussion 324–6

[46] Rousseeuw, P.J., Croux, C.: Alternatives to the median absolute deviation. J. Am. Stat. Assoc. **88**(424) (December 1993) 1273–1283

[47] Edgington, E.S.: RANDOMIZATION TESTS. J. Psychol. **57** (April 1964) 445–449

[48] Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: An improved algorithm for TV-L1 optical flow. In: Statistical and Geometrical Approaches to Visual Motion Analysis, Springer Berlin Heidelberg (2009) 23–45

[49] Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, JMLR.org (July 2015) 448–456

[50] Grossberg, S., Mingolla, E.: Neural dynamics of perceptual grouping: textures, boundaries, and emergent segmentations. Percept. Psychophys. **38**(2) (August 1985) 141–171

[51] Adelson, E.H., Bergen, J.R.: Spatiotemporal energy models for the perception of motion. J. Opt. Soc. Am. A, JOSAA **2**(2) (February 1985) 284–299

[52] Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. (July 2016)

A Extended related work

Translating circuits for biological vision into artificial neural networks While the *Pathfinder* challenge of [**?**] presents immense challenges for transformers and deep convolutional networks [5], the authors found that it can be solved by a simple model of intrinsic connectivity in visual cortex, with orders-of-magnitude fewer parameters than standard models for image categorization. This model was developed by translating descriptive models of neural mechanisms from Neuroscience into an architecture that can be fit to data using gradient descent [**?**, **?**]. Others have found success in modeling object tracking by drawing inspiration from "dual stream" theories of appearance and motion processing in visual cortex [**?**, 38], or basing the architecture off of a partial connectome of the drosophila visual system [39]. We adopt a similar approach in the current work, identifying mechanisms for object tracking without re-recognition in Neuroscience, and developing those into differentiable operations with parameters that can be optimized by gradient descent. This approach has the dual purpose of introducing task-relevant inductive biases into computer vision models, and developing theory on their relative utility for biological vision.

Multi-object tracking in computer vision The classic psychological paradigms of multi-object tracking [2] motivated the application of models, like Kalman filters, which had tolerance to object occlusion when they relied on momentum models [40]. However, these models are computationally expensive, hand-tuned, and because of this, not commonly used in computer vision anymore [41]. More recent approaches include flow tracking on graphs [42] and motion tracking models that are relatively computationally efficient [43]. However, even current approaches to multi-object tracking are not learned, instead relying on extensive hand tuning [44]. In contrast, the point of *PathTracker* is to understand the extent to which state-of-the-art neural networks are capable of tracking a single object in an array of distractors. Moreover, the solution discovered by the InT distinctly contrasts with these multi-object tracking models: it learns to index and track potential targets, rather than rely on momentum or other features that multi-object tracking models hard code. This is especially notable, since the InT is a model of neural circuit mechanisms, and its learned strategy predicts how humans solve the same task, and what neural systems are involved.

Limitations In this work we tested a relatively small number of *PathTracker* versions. We mostly focused on small variations to the number of distractors and video length, but in future work we hope to incorporate other variations like speed and velocity manipulations, and generalization across temporal variations. Another limitation is that appearance-free strategies confer relatively modest gains over the state of the art. One potential issue is determining when a visual system should rely on appearance-based vs. appearance-free features for tracking. Our solution is two-pronged and potentially insufficient. The first strategy is for top-down feedback from the TransT into the InT, which we aligns tracks between the two models. The second strategy is potentially naive, in that we gate the InT modulation to the TransT based on its agreement with the prior TransT query, and the confidence of the TransT query. Additional work is needed to identify better approaches. Meta-cognition work from Cognitive Neuroscience is one possible resource [45].

Societal impacts The basic goal of our study is for understanding how biological brains work. *PathTracker* helps us screen models against humans on a simple visual task which tests visual strategies for tracking without "re-recognition", or appearance cues. The fact that we developed a circuit that explains human performance is primarily important because it makes predictions about the types of neural circuit mechanisms that we might ultimately find in the brain in future Neuroscience work. Our extension to natural videos achieves new state-of-the-art because it is able to implement visual strategies that build tolerance to visual nuisances in way that resembles humans. It must be recognized the further development of this model has potential for misuse. One possible nefarious application is for surveillance. On the other hand, such a technology could be essential for ecology, sports, self-driving cars, robotics, and other real-world applications of machine vision. We open source our code and data to promote research towards such beneficial applications.

B Human benchmark

For our benchmark experiments we recruited 120 participants. Every participant was compensated with $8 through MTurk on successful completion of all test trials by pasting a unique code generated

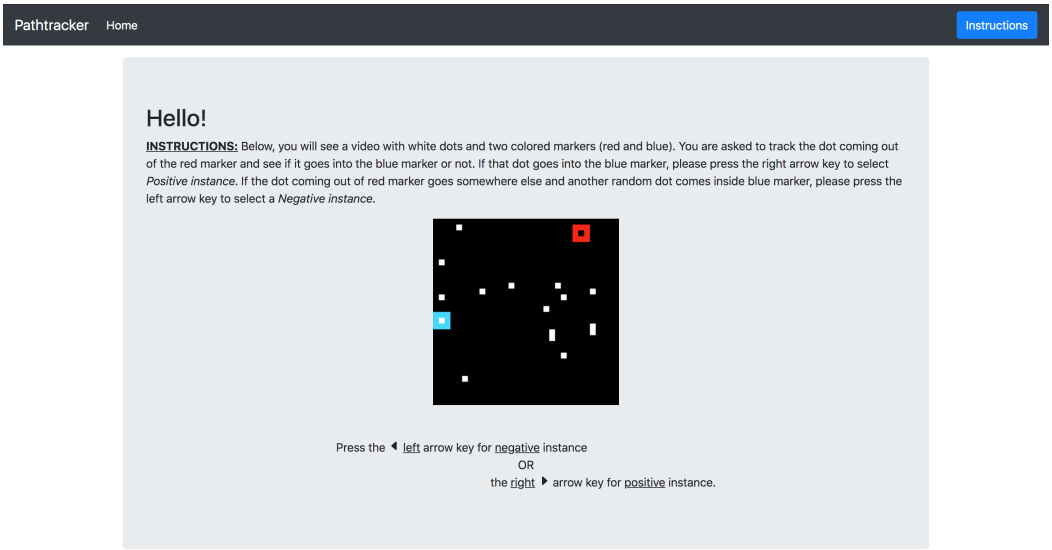

Figure S1: An experimental trial screen.

by the system into their MTurk account. The decision regarding this amount was reached upon by prorating the minimum wage. An additional overhead fee of 40% per participant was paid to MTurk. Collectively, we spent $960 on these benchmark experiments.

The experiment was not time bound and participants could complete it at their own pace, taking around 25 minutes to complete. Videos with 32-, 64- and 128-frames were of duration 4, 8 and 14 seconds respectively. The videos played at 10 frames per second. Participant reaction times were also recorded on every trial and we include these in our data release. After every trial participants were redirected to a screen confirming successful submission of their response. They could start the next trial by clicking the "Continue" button or by pressing spacebar. If not, they were automatically redirected to the next trial after 3000 ms. Participants were also shown a "rest screen" with a progress bar after every 10 trials where they could take additional and longer breaks if needed. The timer was turned off for the rest screen.

Experiment design At the beginning of the experiment, we collected participant consent using a consent form approved by a University's Institutional Review Board (IRB). Our experiment was completed on a computer via Chrome browser. Once consented, we provided a demonstration clearly stating the instructions with an example video to the participants. We also provided them with an option to revisit the instructions, if needed, from the top right corner of the navigation bar at any point during the experiment.

Participants were asked to classify the video as "positive" (the dot leaving the red marker entered the blue marker) or "negative" (the dot leaving the red marker did not enter the blue marker) using the right and left arrow keys respectively. The choice for keys and their corresponding instances were mentioned below the video on every screen, along with a small instruction paragraph above the video. See Fig. S1. Participants were given feedback on their response (correct/incorrect) after every practice trial, but not after the test trials.

Setup The experiment was written in Python Flask, including the server side script and logic. The frontend templates were written in HTML with Bootstrap CSS framework. We used javascript for form submission with keys and redirections, done on the end-user side. The server was run with nginx on 1 Intel(R) Xeon(R) CPU E5-2695 v3 at 2.30GHz, 4GB RAM, Red Hat Enterprise Linux Server.

Video frames for each experiment were generated at 32×32 resolution. Before writing them to the mp4 videos displayed to human participants in the experiment, the frames were resized through nearest-neighbor interpolation to 256×256. In order to allow time for users to prepare for each trial, the first frame of each video was repeated 10 times before the rest of the video played.

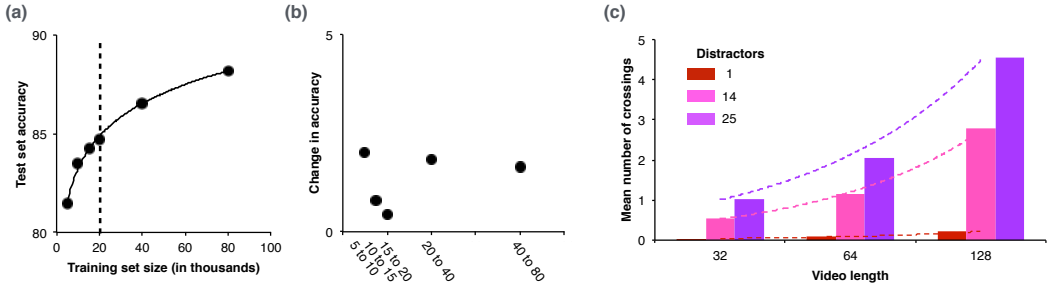

Figure S2: Our approach for selecting training set size on *PathTracker*, and a proxy for difficulty across the versions of the challenge. (a) We plot I3D performance as a function of training set size. The dotted line denotes the point at which the derivative of accuracy w.r.t. training set size is smallest (b). We take this change performance as a function of training set size as evidence that I3D has learned a strategy that is sufficient for the task. We suspected this size would make the *PathTracker* challenging but still solvable for the models we discuss in the main text. (c) The number of average crossings in *PathTracker* videos as a function of distractors and video length. Lines depict exponential fits for each number of distractors across lengths.

Filtering criteria Amazon Mechanical Turk data is notoriously noisy. Because of this, we adopted a simple and bias-free approach to filter participants who were inattentive or did not understand the task (these users were still paid for their time). For the main benchmark described in §3 in the main text, participants completed one of two experiments, where they were trained and tested on videos with 32 or 64 frames. No participant viewed both lengths of *PathTracker*. Participants were trained with 14 distractor videos, then tested on videos with 1, 14, or 25 distractors. We filtered participants according to their performance on the *training videos* for a particular experiment, which were otherwise not used for any analysis in this study. We removed participants who did not exceed 2 median absolute deviations below the median $median(X) - 2 * MAD(X)$ (MAD = median absolute deviation [46]; this is a robust alternative to using the mean and standard deviation to find outliers). The threshold was approximately 40% *training* accuracy for each experiment (chance is 50%). This procedure filtered 74/180 participants in the benchmark.

Statistical testing We assessed the difference between human performance and chance using randomization tests [47]. We computed human accuracy on each test dataset, then over 10,000 steps, we shuffled video labels, and then recomputed and stored the resulting accuracy. We computed $p-$values as the proportion of shuffled accuracies that exceed the real accuracy. We also used linear models for significance testing of trends in human accuracy as we increased the number of distractors. From these models we computed t-tests and p-values.

Using an I3D [?] to select *PathTracker* training set sizes As mentioned in the main text, we selected *PathTracker* training set size for models reported in the main text by investigating sample efficiency of the standard but not state-of-the-art I3D [**?**]. We were specifically interested in identifying a "pareto principle" in learning dynamics where additional training samples began to yield smaller gains in accuracy, potentially signifying a point at which I3D had learned a viable strategy (SI Fig. S2). At this point, we suspected that the task would remain challenging – but still solvable – across the variety of *PathTracker* conditions we discuss in the main text. We focus on basic 32 frame and 14 distractor training and find an inflection point at 20K examples. We plot I3D performance on this condition in SI Fig. S2a and performance slopes in SI Fig. S2b. The first and lowest slope corresponds to 20K samples, and hence may reflect an inflection in the model's visual strategy. Our experiments in the main text demonstrate that this strategy is a viable one for calibrating the difficulty of synthetic challenges.

Target-distractor crossings We compute the number of average crossings between the target object and distractors in *PathTracker*. Increasing video length monotonically increases the number of crossings. Length further interacts with the number of distractors to yield more crossings (SI Fig. S2c).

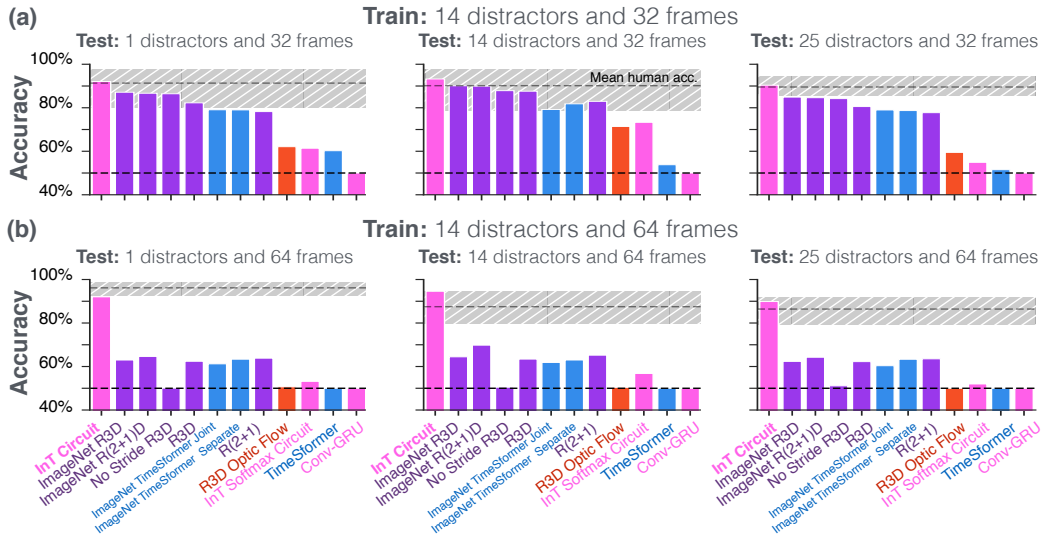

Figure S3: An extended benchmark of state-of-the-art models on *PathTracker* with (*a*) 32 and (*b*) 64 frame versions of the task.

C Solving the Pathtracker challenge

State-of-the-art model details We trained a variety of models on our benchmark. This included an R3D without any strides or downsampling. Because this manipulation caused an explosion in memory usage, we reduced the number of features per-residual block of this "No Stride R3D" from 64/128/256/512 to 32/32/32/32. We also included two forms of TimeSformers [**?**], one with distinct applications of temporal and spatial attention that we include in our main analyses, and another with join temporal and spatial attention (SI Fig. S3).

Optic Flow We followed the method of [**?**] to compute optic flow encodings of *PathTracker* datasets. We used OpenCV's implementation of the TV-L1 algorithm [48]. We extracted two channels from the output given by the algorithm, and appended a channel-averaged version of the corresponding *PathTracker* image, similar to the approach of [**?**].

D InT circuit description

Our InT circuit has two recurrent neural populations, I and E. These populations evolve over time and receive a dynamic "feedforward" drive via Z. This feedforward drive is derived from a convolution between each frame of the *PathTracker* videos a kernel $W_z \in \mathbb{R}^{1,1,3,32}$. This activity is then rectified by a softplus pointwise nonlinearity. InT hidden states are initialized with $0.6931 = \text{softplus}(0)$. The InT circuit also includes Batch Normalization [49] applied to the outputs of its recurrent kernels W_ie, W_ei, with scales (α) and intercepts (η) shared across timesteps of processing. We initialize the scale parameters to 0.1 following prior work [**?**]. We do not store Batch Normalization moments during training. InT gain control (i.e., its divisive normalization) is expected to emerge at steady state [30, 50] in similar dynamical systems formulations, although our formulation relaxes some of these constraints.

The final activity of $E[T]$ in the InT for a *PathTracker* video is passed to a readout that renders a binary decisions for the task. This readout begins by convolving $E[T]$ with a kernel $W_{r1} \in \mathbb{R}^{1,1,32,1}$. The output is channel-wise concatenated with the channel of the first frame containing the location of the goal marker. This activity is then convolved with another kernel $W_{r2} \in \mathbb{R}^{5,5,2,1}$, which is designed to capture overlap between the goal marker and the putative target object/dot. The resulting activity is "global" average pooled and entered into binary crossentropy for model optimization. On *PathTracker*, all versions of the InT and the ConvGRU used this input transformation. All versions of the InT, the ConvGRU, and the "No Stride R3D' used this readout.

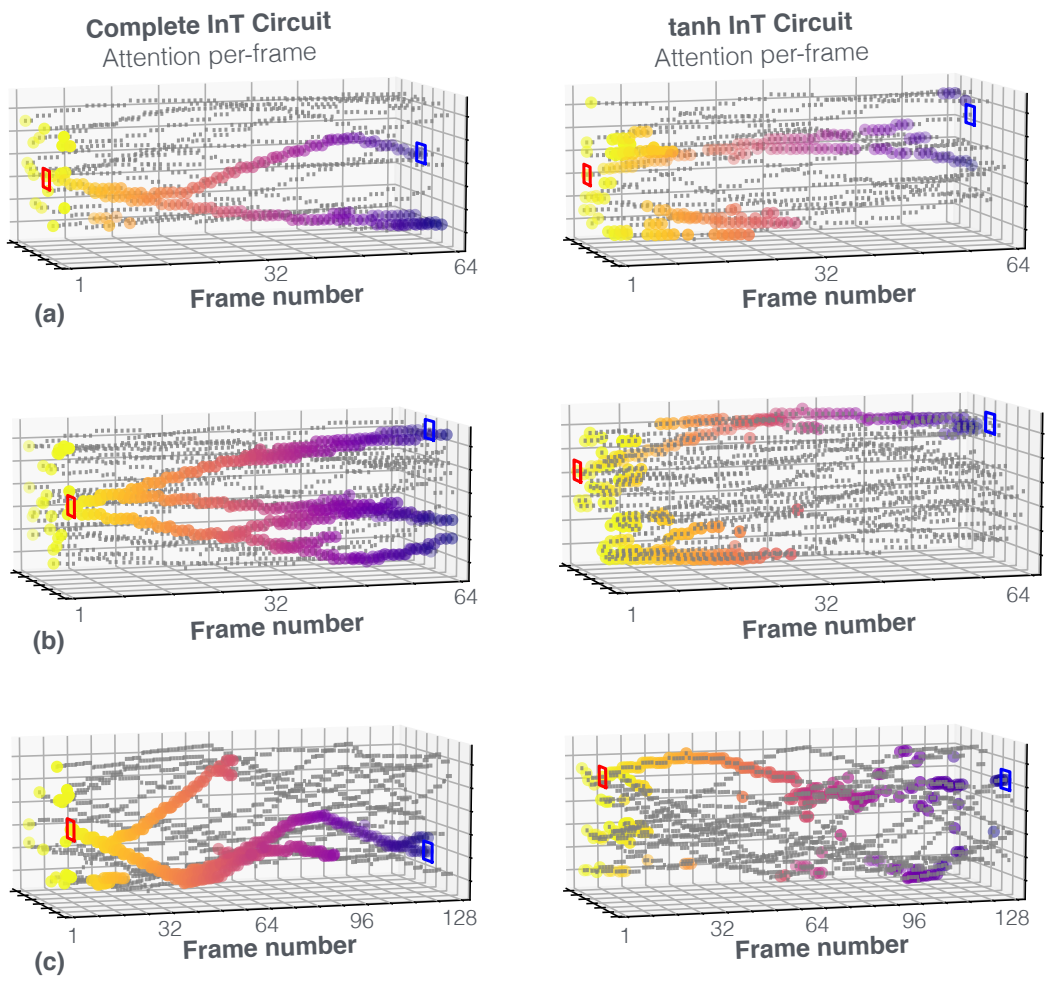

Figure S4: A comparison of attention between the complete InT and one where its softplus rectifications are replaced by tanh.

Spatiotemporal filtering through recurrent connections An open question is whether recurrent neural networks with convolutional connections are capable of learning tuned spatiotemporal feature selectivity. That is, the ability to learn to detect a specific visual feature moving in a certain direction. Adelson and Bergen [51] laid out a plausible solution, in which spatial filters offset by phase are combined over time through positive or negative weights. The success of our InT on *PathTracker* indicates that it might have adopted a similar solution, using its horizontal connection kernels $W_{i,e}, W_{e,i}$ to learn spatial filters offset in phase (e.g., an on-center off-surround and an off-center on-surround filter), which are combined via learned gates to yield spatiotemporal tuning. We leave an analysis of the InT "connectome" as it relates to spatiotemporal feature learning to future work.

Deriving the InT For the sake of clarity and succinctness, we focus the derivation of the InT circuit's update equations to reflect that of generic single neurons, which without loss of generality applies to the each spatial/feature dimension. The InT circuit model is built on top of two recurrent populations (E/I) of neurons (serving excitatory/ inhibitory roles respectively), and a state-less population of neurons (A) that serves as an attentional controller. We denote these populations as follows:

$$E = \left[e_{xy}^{(c)}\right]; I = \left[i_{xy}^{(c)}\right]; A = \left[a_{xy}^{(c)}\right] \tag{1}$$

Here, the x, y subscripts denote spatial tuning, and the c superscript denotes feature tuning. Moving forward, we reference generic units from these populations with e, i, and a respectively. In essence, the circuit can be expressed as a continuous first-order coupled differential system of this form.

$$
\begin{aligned}
\tau_{inh}\frac{di}{dt} &= -i + [z - (\gamma i a + \beta)m]_+ \\
\tau_{exc}\frac{de}{dt} &= -e + [i + (\nu i + \mu)n]_+
\end{aligned} \tag{2}
$$

In Eq. 2, γ, β, ν, and μ are model hyperparameters, while m and n are themselves functions of e, i, and a. The exact functional form of m and n is detailed in Fig. 4b in the main text. z is an external input to the system.

For the purposes of simulating and training this model with gradient descent, we use a first-order Euler approximation with time step Δt. Assuming we choose $g = \frac{\Delta t}{\tau_{inb}}$ and $h = \frac{\Delta t}{\tau_{exc}}$, the discretized version of Eq. 2 can be written as follows.

$$
\begin{aligned}
i_t &= (1-g)\, i_{t-1} + g\,[z_t - (\gamma i_t a_t + \beta)\, m_t]_+ \\
e_t &= (1-h)\, e_{t-1} + h\,[i_t + (\nu i_t + \mu)\, n_t]_+
\end{aligned} \tag{3}
$$

Tuning the time constants τ_{exc}, and τ_{inh} and choosing an appropriate Δt can often prove to be tedious and challenging. To alleviate this, we introduce a "learnable" integration step, where g and h are modeled as neural gates. These are computed as specified in Eq. 4. $\sigma(.)$ is the sigmoidal function, which squashes activities in the range $[0, 1]$. $\mathbf{W}_g$, $\mathbf{U}_g$, $\mathbf{W}_h$, and $\mathbf{U}_h$ are convolutional kernels of size $1 \times 1 \times 32 \times 32$.

$$
\begin{aligned}
G &= \left[g_{xy}^{(c)}\right] = \sigma(\mathbf{W}_g * I + \mathbf{U}_g * Z) \\
H &= \left[h_{xy}^{(c)}\right] = \sigma(\mathbf{W}_h * E + \mathbf{U}_h * I)
\end{aligned} \tag{4}
$$

E InT *PathTracker*

We visualize InT A attention units on *PathTracker* by simply binarizing the logits, where values greater than $mean(A[t]) + stddev(A[t])$ are set to 1 and units below that threshold are set to 0. When applying the same strategy to versions of the InT other than the complete circuit, we found attention that was far more diffuse. For this lesioned InT circuits, adjusting this threshold to be more conservative, choosing two or three or even four standard deviations above the mean, never yielded attention that looked like the complete model. For instance, the closest competitor to the complete InT is one in which its Softplus rectifications are changed to hyperbolic tangents, which remove model constraints for separate and competing forms of Inhibition and Excitation. This model's attention was subsequently diffuse and it also performed worse in generalization than the complete circuit (SI Fig. S4).

We also developed a version of the InT with attention that was biased against multi-object tracking. In the normal formulation, InT attention A is transformed with a sigmoid pointwise nonlinearity. This independently transforms every unit in A to be in $[0, 1]$, giving them the capacity to attend to multiple objects at once. In our version biased against multi-object tracking we replaced the sigmoid with a spatial softmax, which normalized the sum of units in each channel of A to 1. This model performed worse than the CNNs or TimeSformer on *Pathtracker* (SI Fig. S3)

F InT+TransT

We modify a state-of-the-art tracker, TransT, with our InT circuit, to promote alternative visual strategies for object tracking (Fig. S5). We note that our InT+TransT model beats almost every benchmark metric on the LaSOT, TrackingNet, and GOT-10K object tracking challenges (SI Table 1).

Method	Source	LaSOT			TrackingNet			GOT-10K		
		AUC	P_{Norm}	P	AUC	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$
InT+TransT	Ours	**65.0**	**74.0**	**69.3**	**81.94**	**87.48**	**80.94**	**72.2**	**82.2**	**68.2**
TransT	CVPR2021	**64.9**	**73.8**	**69.0**	**81.4**	**86.7**	**80.3**	**72.3**	**82.4**	**68.2**
TransT-GOT	CVPR2021	-	-	-	-	-	-	67.1	76.8	60.9
SiamR-CNN	CVPR2020	64.8	72.2	-	81.2	85.4	80.0	64.9	72.8	59.7
Ocean	ECCV2020	56.0	65.1	56.6	-	-	-	61.1	72.1	47.3
KYS	ECCV2020	55.4	63.3	-	74.0	80.0	68.8	63.6	75.1	51.5
DCFST	ECCV2020	-	-	-	75.2	80.9	70.0	63.8	75.3	49.8
SiamFC++	AAAI2020	54.4	62.3	54.7	75.4	80.0	70.5	59.5	69.5	47.9
PrDiMP	CVPR2020	59.8	68.8	60.8	75.8	81.6	70.4	63.4	73.8	54.3
CGACD	CVPR2020	51.8	62.6	-	71.1	80.0	69.3	-	-	-
SiamAttn	CVPR2020	56.0	64.8	-	75.2	81.7	-	-	-	-
MAML	CVPR2020	52.3	-	-	75.7	82.2	72.5	-	-	-
D3S	CVPR2020	-	-	-	72.8	76.8	66.4	59.7	67.6	46.2
SiamCAR	CVPR2020	50.7	60.0	51.0	-	-	-	56.9	67.0	41.5
SiamBAN	CVPR2020	51.4	59.8	52.1	-	-	-	-	-	-
DiMP	ICCV2019	56.9	65.0	56.7	74.0	80.1	68.7	61.1	71.7	49.2
SiamPRN++	CVPR2019	49.6	56.9	49.1	73.3	80.0	69.4	51.7	61.6	32.5
ATOM	CVPR2019	51.5	57.6	50.5	70.3	77.1	64.8	55.6	63.4	40.2
ECO	ICCV2017	32.4	33.8	30.1	55.4	61.8	49.2	31.6	30.9	11.1
MDNet	CVPR2016	39.7	46.0	37.3	60.6	70.5	56.5	29.9	30.3	9.9
SiamFC	ECCVW2016	33.6	42.0	33.9	57.1	66.3	53.3	34.8	35.3	9.8

Table S1: Object tracking results on the LaSOT [**?**], TrackingNet [7], and GOT-10K [**?**] benchmarks. First place is in red and second place is in blue. Our InT+TransT model beats all others except for two benchmark GOT-10K scores.

InT+TransT We add two InT modules (InT_1 and InT_2) to the TransT architecture (Fig. S5). The key difference between these modules and the ones used on *PathTracker* is that they used LayerNorm [52] instead of Batch Normalization. This was done because object tracking in natural images is memory intensive and forces smaller batch sizes than what we used for *PathTracker*, which can lead to poor results with Batch Normalization.

InT_1 (Fig. S5b) has the same dimensionality as the one described for *PathTracker* in the main text. ResNet50 features $y \in \mathbb{R}^{1024\times32\times32}$ are reduced to $z \in \mathbb{R}^{32\times32\times32}$ by virtue of convolution with a kernel $W_{in} \in \mathbb{R}^{1\times1\times1024\times32}$, i.e., $z = y * W_{in}$. As input, InT_1 received z. A binary mask $B \in \mathbb{R}^{1\times32\times32}$ that specified the location of the target object in the very first frame was used to initialize the recurrent excitatory/inhibitory units of InT_1. They took values $E_{t=0} = B * W_{E_1}$ and $I_{t=0} = B * W_{I_1}$ respectively, where kernels $W_{E_1}, W_{I_1} \in \mathbb{R}^{1\times1\times1\times32}$, and $E_t, I_t \in \mathbb{R}^{32\times32\times32}$. The subscript t for the recurrent population activities represent an arbitrary time point w.r.t. steps of processing.

To coregister the representations of InT_1 and $TransT$, we treat the excitatory units, E_t, of InT_1 by a transformation f_ϕ parameterized by three-layer convolutional neural network consisting of 1×1 kernels. f_ϕ essentially inflates dimensionality, i.e., $f_\phi(E_t) \in \mathbb{R}^{256\times32\times32}$. The network f_ϕ had softplus activation functions applied to the output of the first and second layers, and used kernels of dimensions $1 \times 1 \times 32 \times 256$, $1 \times 1 \times 256 \times 256$ and $1 \times 1 \times 256 \times 256$ in the three layers respectively. For notational convenience, we refer to $f_\phi(E_t)$ as X_t in this discussion subsequently.

The "search frame" query (Q_t) for the TransT cross-feature attention (CFA) component was computed as a function of X_t and Q_{t-1} as described here. Q_{t-1} was first subject to a transformation f_ψ, parameterized as another convnet, this time to register the query representation to the latent activities of the InT modules. $f_\psi(Q_{t-1}) \in \mathbb{R}^{32\times32\times32}$ is used to compute two quantities: (a) a measure of spatial certainty in Q_{t-1}, and (b) a measure of spatial agreement between Q_{t-1} and E_t. For spatial certainty we compute the channel wise L^2 norm of $f_\psi(Q_{t-1})$, yielding tensor $H^{(1)} \in \mathbb{R}^{1\times32\times32}$. For the spatial agreement measure, we compute the feature-wise outer product $H^{(2)} = f_\psi(Q_{t-1}) \otimes E_t \in \mathbb{R}^{1024\times32\times32}$. The mix-gate G_{mix} was then a convolution on $H = \left[H^{(1)} H^{(2)}\right]$, with a kernel $W_{mix} \in \mathbb{R}^{1\times1\times1025\times1}$, followed by a sigmoidal non-linearity. The final TransT query Q_t was then constructed as the sum of the original Q_t and $G_{mix} \odot X_t$. Functionally, this mix-gate helps the

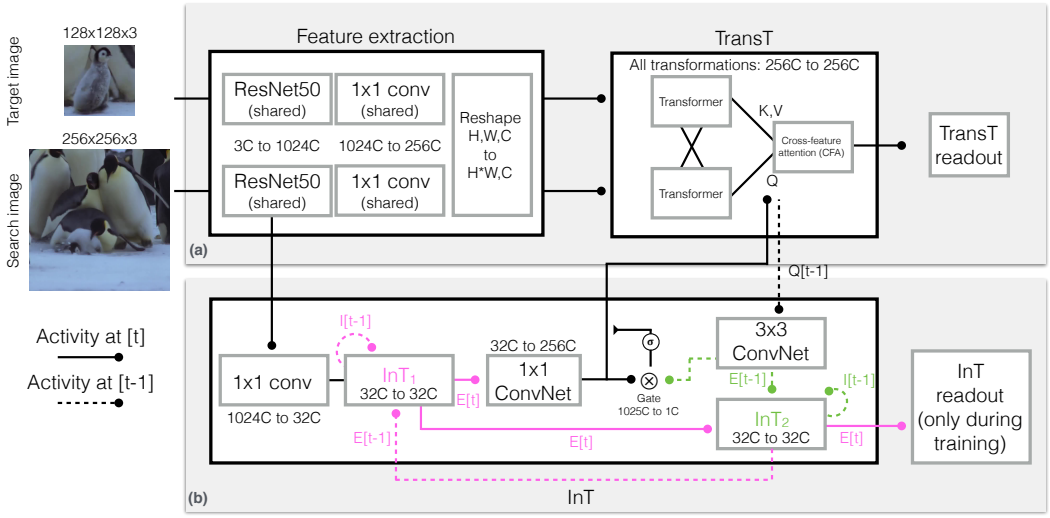

Figure S5: The (*a*) TransT and (*b*) InT addition to create our the InT+TransT. The InT additively modulate the TransT query (Q) in its CFA, which corresponds to its encoding of the search image which is compared to its encoding of the target. The InT activity is recurrent, and itself modulated by a "gate" which captures the similarity of InT activity and the TransT query from the prior step, along with the TransT query entropy. This gate shunts InT activity unless the TransT is low-confidence and the InT and TransT render different predictions, at which point the InT can adjust TransT queries. The InT is further supervised on each step of a video to predict target object bounding boxes.

InT+TransT compose a hybrid of appearance-free and appearance-based tracker query based on the intrinsic uncertainty of a video frame at a given moment in time. See SI Fig. S5 for a schematic.

The final step in the InT+TransT pipeline is "top-down" feedback from the TransT back to InT_1. This was done to encourage the two modules to align their object tracks and correct mistakes that emerged in one or the other resource [6]. $f_\psi(Q_{t=0})$, computed as described above, was used for initializing the excitatory units of InT_2 (InT$_2$ Fig. S5)b). The inhibitory units of InT_2 was initialized with $f_\psi(Q_{t=0}) * W_{I_2}$, where $W_{I_2} \in \mathbb{R}^{1,1,32,32}$. E_t from InT_1 served as the input drive to InT_2 at every time step t. To complete the loop, the recurrent excitatory state of InT_2 served as feedback for InT_1. We evaluated our InT+TransT on TrackingNet (published under the Apache License 2.0), LaSOT (published under the Apache License 2.0), and GOT-10K (published under CC BY-NC-SA 4.0). See Table F for a full comparison between our InT+TransT and other state-of-the-art models.

Object tracking training and evaluation The InT+TransT is trained with the same procedure as the original TransT, except that its InTs are given the intervening frames between the target and search images, as described in the main text. Otherwise, we refer the reader to training details in the TransT paper [4]. Evaluation was identical to the TransT, including the use of temporal smoothing for postprocessing ("Online Tracking"). As was the case for TransT, this involved interpolating the TransT bounding box predictions with a 32×32 Hanning window that penalized predictions on the current step t which greatly diverged from previous steps. See [4] for details.