LEARNING CORRESPONDENCE FROM THE CYCLE-CONSISTENCY OF TIME

Xiaolong Wang   Carnegie Mellon University

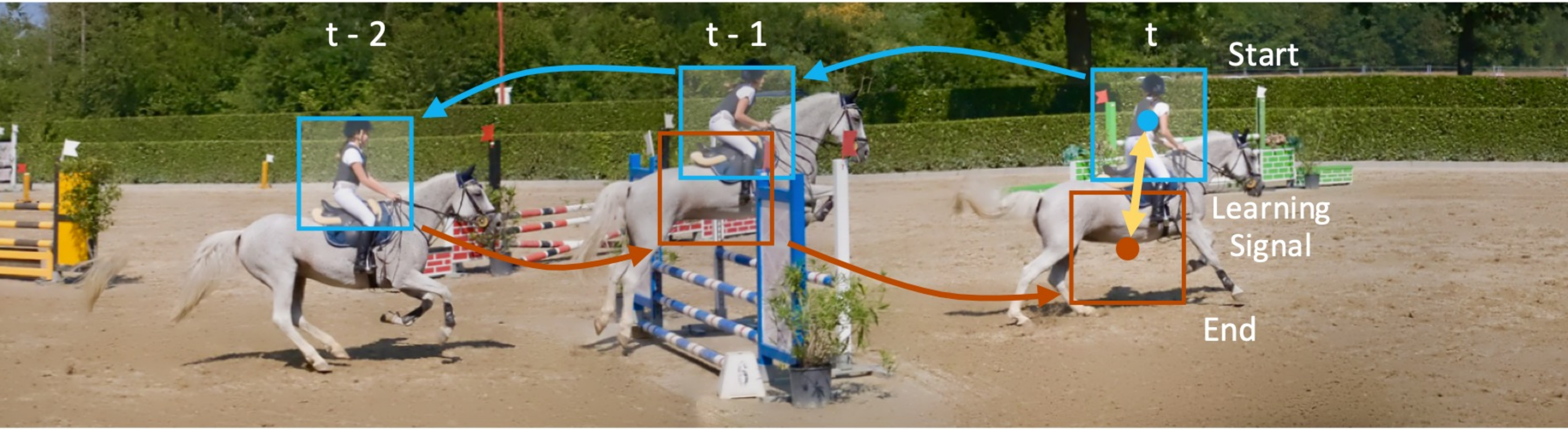
Allan Jabri*   UC Berkeley

Alexei A. Efros   UC Berkeley

https://arxiv.org/pdf/1903.07593.pdf

CORRESPONDENCE

- Pixel level: Optical Flow
  - Training data limitation: Synthetic datasets may not match real images
- Object level: Tracking
  - Training data limitation: human annotation of objects



CYCLE CONSISTENCY

- Concept: define a powerful feature descriptor network ϕ and a weak tracking operator $\mathcal{T}$ that together track a patch through an image
- Loss: how "cycle consistent" is ϕ when combined with $\mathcal{T}$

NOTATION:

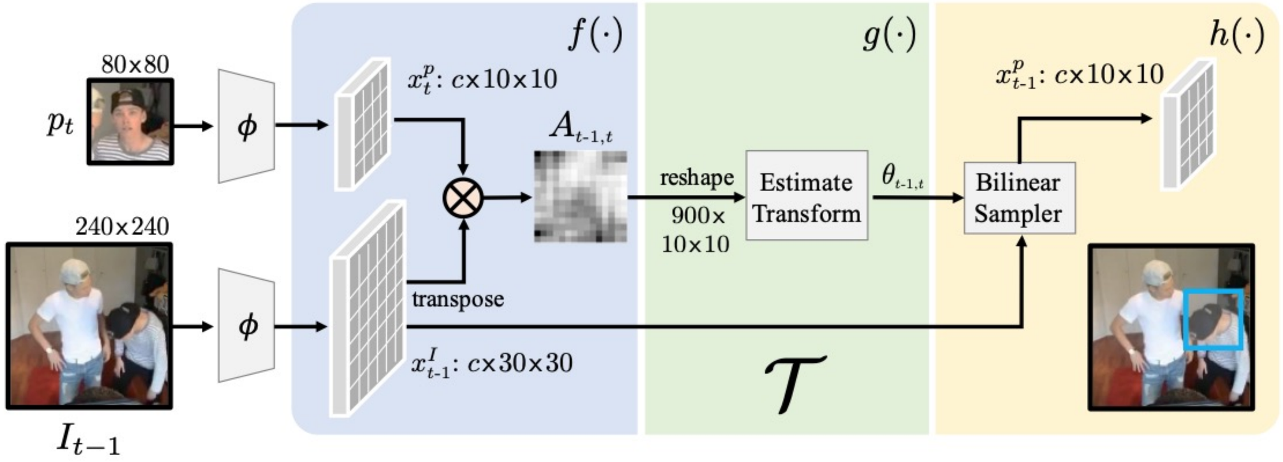
• $I_{t-k:t}$	A sequence of k+1 images from a video	
• p_t	A patch from image t	
• ϕ	An encoder that produces a grid of feature vectors	
• $x^I_{t-k:t}$	$\phi(I_{t-k:t})$	k+1 x c x 30 x 30
• x^p_t	$\phi(p_t)$	c x 10 x 10
• $\mathcal{T}$	$x^I_s \times x^p_t \rightarrow x^p_s$	

- $\mathcal{T}$ finds the patch in x^I_s that is most similar to x^p_t

TRAINING PROCESS

$\mathcal{T}$

- A: 900 x 100
- $A(i,j) = \frac{e^{x^I(j) \cdot x^p(i)}}{\sum_j e^{x^I(j) \cdot x^p(i)}}$



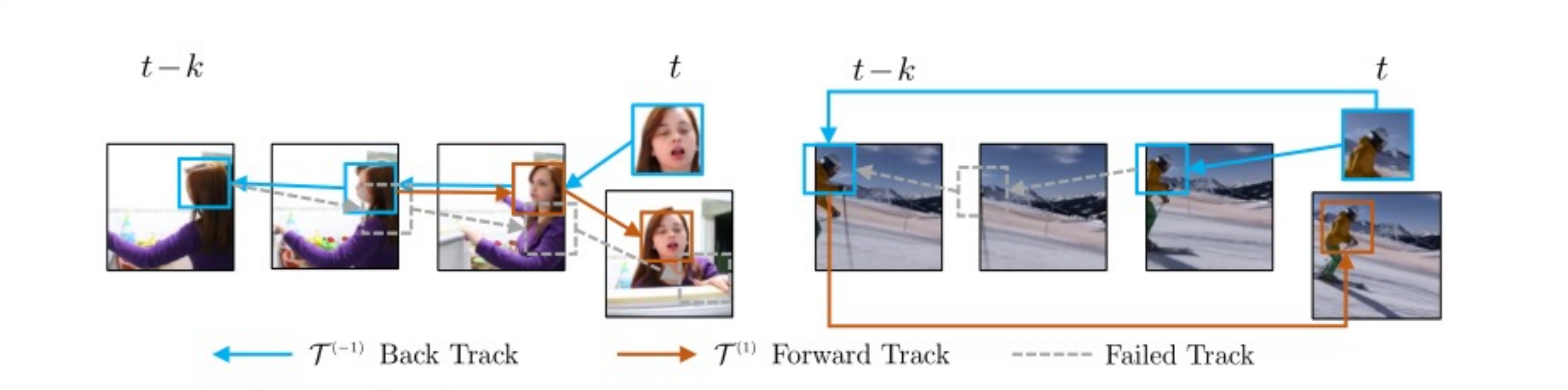
(b) Differentiable Tracking Operation $\mathcal{T}$

ITERATE OPERATOR $\mathcal{T}$

$$\mathcal{T}^{(i)}(x^I_{t-i}, x^p) = \mathcal{T}(x^I_{t-1}, \mathcal{T}(x^I_{t-2}, ...\mathcal{T}(x^I_{t-i}, x^p)))$$

$$\mathcal{T}^{(-i)}(x^I_{t-1}, x^p) = \mathcal{T}(x^I_{t-i}, \mathcal{T}(x^I_{t-i+1}, ...\mathcal{T}(x^I_{t-1}, x^p)))$$

FULL LOSS



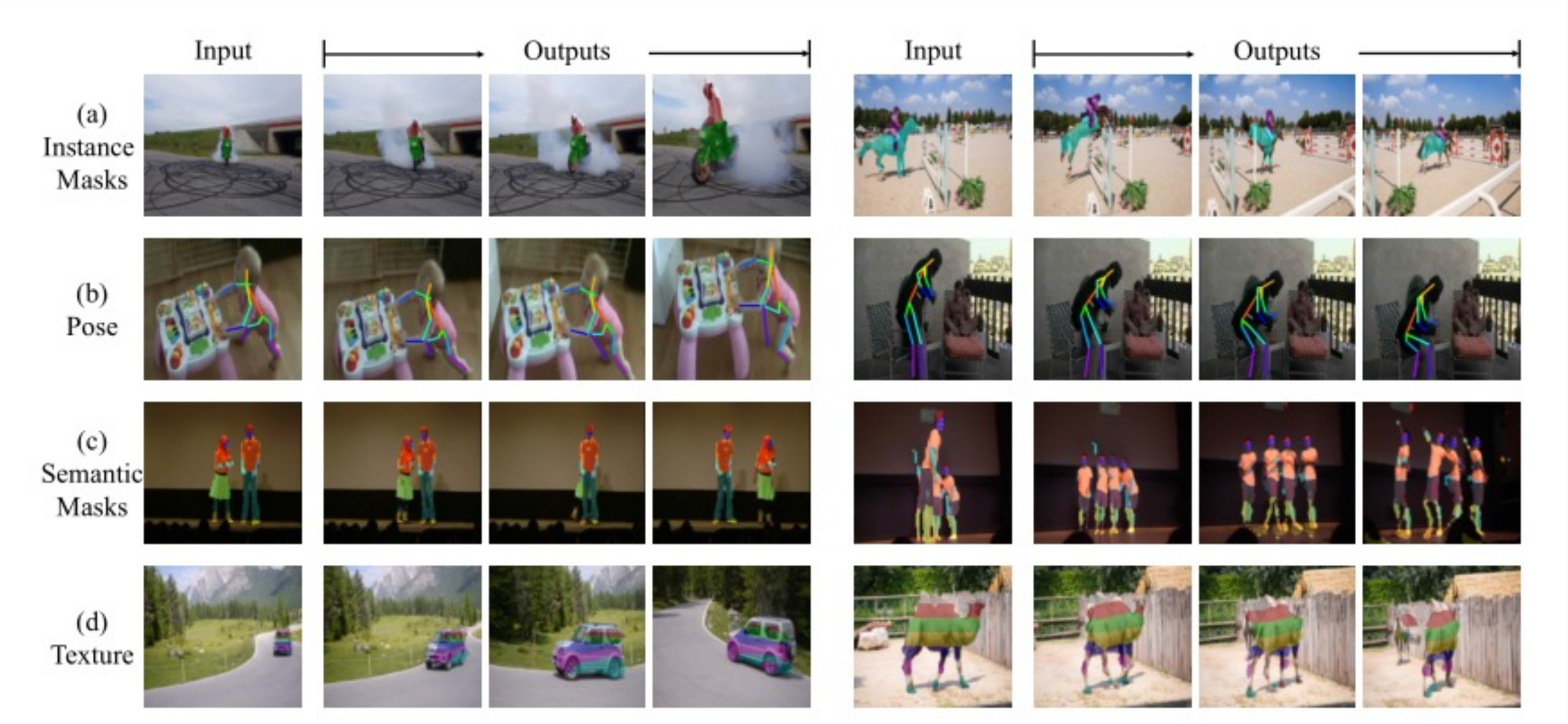
$\mathcal{L}_{long}^{3}$ $\mathcal{L}_{skip}^{3}$

- $\mathcal{L} = \sum_i \mathcal{L}_{sim}^{i} + \lambda \mathcal{L}_{skip}^{i} + \lambda \mathcal{L}_{long}^{i}$

INFERENCE PROCESS

- Drop $\mathcal{T}$ entirely! Just use features from ϕ
- Propagate labels:
  - Now A is over whole image instead of patch
  - $A(j,i) = \frac{e^{x_{t-1}^{I}(j)\cdot x_{t}^{I}(i)}}{\sum_j e^{x_{t-1}^{I}(j)\cdot x_{t}^{I}(i)}}$
  - $y_i = \sum_j A_{t-1,t}(j,i)\, y_j$
  - Finally, label map upsampled

VISUAL RESULTS



VISUAL RESULTS

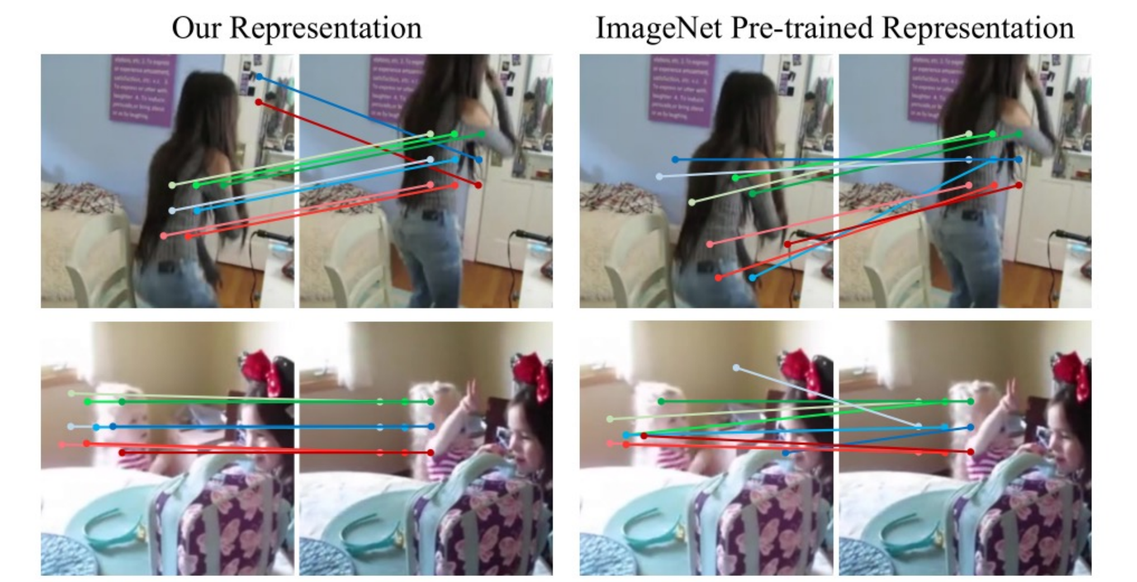


Figure 7: Visualizations of correspondence.

NUMERICAL RESULTS

- DAVIS mask propagation

model	Supervised	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)
Identity		22.1	23.6
Random Weights (ResNet-50)		12.4	12.5
Optical Flow (FlowNet2) [22]		26.7	25.2
SIFT Flow [39]		33.0	35.0
Transitive Inv. [74]		32.0	26.8
DeepCluster [8]		37.5	33.2
Video Colorization [69]		34.6	32.7
Ours (ResNet-18)		40.1	38.3
Ours (ResNet-50)		**41.9**	**39.4**
ImageNet (ResNet-50) [18]	✓	50.3	49.0
Fully Supervised [81, 7]	✓	55.1	62.1

JHMDB pose propagation

model	Supervised	PCK@.1	PCK@.2
Identity		43.1	64.5
Optical Flow (FlowNet2) [22]		45.2	62.9
SIFT Flow [39]		49.0	68.6
Transitive Inv. [74]		43.9	67.0
DeepCluster [8]		43.2	66.9
Video Colorization [69]		45.2	69.6
Ours (ResNet-18)		57.3	78.1
Ours (ResNet-50)		**57.7**	**78.5**
ImageNet (ResNet-50) [18]	✓	58.4	78.4
Fully Supervised [59]	✓	68.7	92.1

Table 2: Evaluation on pose propagation on JHMDB [26]. We report the PCK in different thresholds.

DISCUSSION QUESTIONS:

- Why do we use a neural network to interpret A(j, i) during training, but use it directly during test time?
  - Why do we learn rotation during training but not use it during test time?
- Do the authors make a convincing argument that their process is better than pretraining on ImageNet?