

1 **Videos with Tracks** The attached video file contains predicted tracks using Opt-CWM on both  
2 in-the-wild YouTube videos and a subset of TAP-Vid DAVIS. The examples showcase Opt-CWM’s  
3 capability to generate accurate predictions even in challenging real-world scenes with high levels of  
4 object and camera motion.

## 5 A Implementation Details

### 6 A.1 Architecture Details

#### 7 A.1.1 $\Psi^{\text{RGB}}$

8 The input video is first divided into non-overlapping spatiotemporal patches of size  $8 \times 8$ , with  
9 a subset of patches masked. Unlike MAE, we train with both revealed input patches and mask  
10 tokens provided to the encoder. We train with the ViT-B architecture [7] with each transformer block  
11 consisting of a multi-head self-attention block and an MLP block, both using LayerNorm (LN). The  
12 CWM decoder has the same architecture as the encoder. Each spatiotemporal patch has a learnable  
13 positional embedding which is added to both the encoder and decoder inputs. CWM does not use  
14 relative position or layer scaling [1, 7]. Please refer to [18, 2] for more details on the architecture.  
15 The 175M CWM model is based on ViT-B [4] but has twice the number of total layers. The 1B CWM  
16 model is similar to the 175M model, but has 48 layers with an embedding dimension of 2048 and 16  
17 heads.

18 **Default settings** We show the default pre-training settings in Table 1. CWM does not use color  
19 jittering, drop path, or gradient clip. Following ViT’s official code, Xavier uniform is used to initialize  
20 all transformer blocks. The learnable masked token is initialized as a zero tensor. Following MAE,  
21 we use the linear lr scaling rule:  $lr = \text{base\_lr} \times \text{batch\_size} / 256$  [7].

Table 1: Default pre-training setting of CWM

config	value
optimizer	AdamW [12]
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$ [3]
accumulative batch size	4096
learning rate schedule	cosine decay [11]
warmup epochs [5]	40
total epochs	800
flip augmentation	no
augmentation	MultiScaleCrop [19]

#### 22 A.1.2 $\Psi^{\text{flow}}$

23 The architecture of the flow-conditioned predictor,  $\Psi^{\text{flow}}$ , is a vision transformer with 16 layers and  
24 132M parameters. Input images are resized to 224x224, and the patch size is 8. Sinusoidal positional  
25 encodings are used. For the encoder, the embedding dimension is 768, and 12 attention heads are  
26 used. For the decoder, the embedding dimension is 384, and 6 attention heads are used.

27 This model has two parallel “streams”, the first of which takes RGB input and the second of which  
28 takes sparse flow, concatenated with RGB (which is masked to have the same sparsity as the flow), as  
29 input. All RGB inputs are from the first frame only; this requires the model to depend solely on flow  
30 to modify the RGB and predict the next frame.

31 The transformer architecture then applies self-attention to each stream and cross-attention between  
32 streams. The encoder has 12 layers, split into three groups of 4. In each group, there is one layer  
33 with self attention on each stream and cross attention from each stream to the other, followed by  
34 three layers with only self-attention on the first stream. The decoder has 4 layers; the first applies  
35 self-attention to each stream and cross-attention from each stream to the other; the second applies  
36 self-attention to the first stream and cross-attention from the second stream to the first; and the final  
37 two only apply self-attention to the first stream.

## 38 A.2 Training Details

### 39 A.2.1 $\Psi^{\text{RGB}}$

40 We train CWM at 256 resolution for 800 epochs and finetune at 512 resolution for 100 epochs by  
41 interpolating the positional embeddings. It takes approximately 4 days to train 800 epochs on a TPU  
42 v5-128 pod. We pre-train CWM on the Kinetics-400 dataset [10], without requiring any specialized  
43 temporal downsampling.

44 We train CWM 1B at 256 resolution for 200,000 iterations with a batch size of 512 on a custom video  
45 dataset called BVD (approximately equivalent to 400 kinetics epochs). We use the AdamW optimizer,  
46 with norm clipping 1.0 and weight decay 0.1. We warmup the learning rate over 2,000 steps to a peak  
47 of  $3\text{e-}4$ , then linearly decay to 0 over the next 198,000 steps. Training takes approximately 1 day on  
48 64 H100 GPUs.

### 49 A.2.2 $\Psi^{\text{flow}}$ and $\text{FLOW}_\theta$

50 We train  $\Psi^{\text{flow}}$  and  $\text{FLOW}_\theta$  jointly using an AdamW optimizer with weight decay of 0.05, betas of  
51  $(0.9, 0.95)$ , and a learning rate schedule with max learning rate  $1.875 \times 10^{-5}$ , 40 warmup epochs (10%  
52 of total training epochs), and cosine decay. We used a batch size of 32, training on the Kinetics-400  
53 dataset [10].

## 54 A.3 Training Data

55 We construct a training dataset called BVD (Bid Video Dataset) which consists of publicly available  
56 datasets such as kinetics400 [10] and SomethingSomethingV2 [6] along with other publicly available  
57 videos. We filter the videos based on CLIP [14] categories to remove thumbnails and videos with a  
58 lot of text in the frame. We additionally filter the videos based on optical flow to remove videos with  
59 little motion such as slide shows, or mostly static videos.

## 60 A.4 Inference Techniques

### 61 A.4.1 Multi-Mask

62 In the process of computing flows in  $\text{FLOW}_\theta$ , at inference time, we take an argmax over the difference  
63 between the predicted next frame with and without the counterfactual perturbation. This difference  
64 image,  $\Delta$ , depends on the choice of the random mask as this mask is used by  $\Psi^{\text{RGB}}$  for the next-frame  
65 reconstruction. As discussed in the main text, if a random mask reveals patches too close to where  
66 the perturbation should be reconstructed, the predictor  $\Psi^{\text{RGB}}$  may not reconstruct the perturbation  
67 properly, and the difference image will be noisy and diffuse, preventing the model from accurately  
68 predicting the next-frame location. Additionally, the reconstructed pixels will not necessarily be  
69 the same across different random samplings of visible patches, which may add random noise to the  
70 difference image. Both of these issues are ameliorated by our multi-mask technique, in which we  
71 compute difference images for a variety of sampled random masks (we found 10 to be a good number  
72 of masks for multi-masking), average over the difference images, and then take the argmax of this  
73 averaged  $\Delta_{\text{avg}}$  to compute next-frame location for determining flow.

### 74 A.4.2 Multiscale

75 Multiscale refinement of the original flow prediction improves Opt-CWM’s performance, as observed  
76 in Figure 3. Given an input frame pair and an initial flow prediction, we perform iterative multiscale  
77 through the following procedure. At each “zoom iteration”, we take a  $0.75H \times 0.75W$  crop of the  
78 input frames with original height  $H$  and width  $W$ . We center the second frame crop on the location  
79 predicted by the previous iteration.

80 The transformer-based architecture of the next frame predictor  $\Psi^{\text{RGB}}$  imposes a limit to the input  
81 resolution, which may occasionally prevent small objects or minute features of the input frame from  
82 being accurately reconstructed in great detail. Multiscale refinement of the initial flow prediction can  
83 be greatly beneficial under these circumstances. However, Figure 3 suggests that the improvement is  
84 not monotonic; indeed, excessive cropping may lead to the loss of global context that is necessary to

Table 2: **Quantitative results on TAP-Vid dataset (Constant five-Frame Gap (CFG)).** In the CFG protocol, point tracking is evaluated at fixed gaps of 5 frames, making it an easier setting that is more favorable to optical flow methods. “S” and “U” indicate supervised and unsupervised, respectively. Doduo is not strictly unsupervised as it uses segmentation labels. GMRW is trained on the Kubric dataset, (marked with ‡), making it a more favorable evaluation setting for that method because of the minimal domain gap. Best performing supervised models (shaded) are considered separately.

Method	DAVIS					Kinetics					Kubric				
	AJ ↑	AD ↓	$< \delta_{avg}^\circ$ ↑	OA ↑	OF1 ↑	AJ ↑	AD ↓	$< \delta_{avg}^\circ$ ↑	OA ↑	OF1 ↑	AJ ↑	AD ↓	$< \delta_{avg}^\circ$ ↑	OA ↑	OF1 ↑
S CoTracker-v3 [9]	74.49	1.21	86.59	90.74	73.08	79.45	0.82	87.86	95.65	74.12	78.15	1.02	89.07	92.59	79.92
RAFT [17]	69.69	1.43	83.83	81.98	46.08	79.01	0.86	87.59	92.73	49.49	73.38	1.24	83.73	91.00	63.17
SEA-RAFT [20]	69.89	1.44	84.82	82.00	47.52	75.12	1.07	85.82	88.90	39.42	77.53	1.00	87.02	92.50	68.65
DFlow [13]	78.09	0.99	87.86	90.19	68.57	80.07	0.82	87.62	95.86	75.09	87.19	0.77	93.60	93.12	79.18
U <sup>†</sup> Doduo [8]	25.61	1.61	72.56	37.49	22.59	35.26	1.19	77.62	43.00	11.63	56.57	1.74	68.63	87.26	55.01
U GMRW [15]	61.28	3.11	72.28	73.01	40.31	75.44	1.23	83.54	88.89	40.96	75.54 <sup>‡</sup>	1.61 <sup>‡</sup>	84.30 <sup>‡</sup>	83.92 <sup>‡</sup>	53.97 <sup>‡</sup>
SMURF [16]	65.75	2.40	79.45	82.26	42.65	78.76	0.97	87.16	93.13	47.69	69.05	1.59	82.38	90.84	53.49
CWM [2, 18]	27.56	4.65	38.55	88.90	5.41	34.00	3.93	43.37	95.17	5.95	30.72	4.05	42.33	88.44	4.27
Opt-CWM 1B (ours)	75.26	0.96	87.84	88.09	54.82	78.15	0.95	87.68	92.10	43.80	82.89	0.80	92.43	91.42	65.43

85 accurately reconstruct the scene. Opt-CWM is run on 4 zoom iterations, which we have empirically  
86 found to be optimal.

### 87 A.4.3 Occlusion Estimation

88 The difference image  $\Delta$  can also be used to predict whether a visible point becomes occluded in the  
89 next frame. Conceptually, as described in Section 3 in the main text, when a point becomes occluded,  
90 the counterfactual perturbation placed on the object should not be reconstructed in the second frame.  
91 Thus, while we take  $\arg\max \Delta$  to compute flow, we can instead use  $\max \Delta$  as a signal for occlusion.  
92 In particular, we compare  $\max \Delta$  to some threshold  $t_{occ}$  to predict occlusion (i.e., we consider the  
93 model to have predicted that a point becomes occluded if and only if  $\max \Delta < t_{occ}$ ).

94 In the multi-masking setting with 10 masking iterations, we have 10 difference images:  $\Delta_1, \Delta_2,$   
95 ...,  $\Delta_{10}$ . Instead of thresholding the average,  $\Delta_{avg}$ , we can get an improved signal by considering  
96  $\max \Delta_i$  for each  $i = 1, \dots, 10$ . In this setting, we found that checking  $\frac{1}{10} \sum_{i=1}^{10} \max \Delta_i < 0.05$   
97 provided a good signal for predicting occlusion, and this prediction criterion is what was evaluated in  
98 the OA and OF1 metrics of Table 1 in the main paper.

## 99 B Additional Quantitative Results

### 100 B.1 Constant Frame Gap Protocol

101 **TAP-Vid Constant Frame Gap (CFG).** For fair comparison with optical flow models, we also  
102 propose an additional protocol with fixed frame gaps that is more advantageous for these baselines  
103 (see supplementary for the effect of frame gap on flow baselines). In particular, a fixed 5-frame gap is  
104 used: metrics are computed on all frame-pairs that are 5 frames apart (and the point is visible in the  
105 first).

106 We show results for this protocol in Table 2. We observe that in this setting, which is favorable  
107 to optical-flow based models, Opt-CWM largely outperforms all unsupervised methods and is  
108 competitive with state-of-the-art fully supervised methods.

### 109 B.2 Precision Analysis

110 Figure 1 attempts to explain the high performance of Opt-CWM on TAP-Vid First through a similar  
111 analysis done in Section A.4.2. Our best-performing model (with optimal inference-time configura-  
112 tions) is able to predict the next frame location within 16 pixels of the ground truth for over 85% of  
113 the total number of visible points. Unlike baseline models, Opt-CWM is able to predict most points  
114 within a reasonable boundary. Further, Opt-CWM predictions are precise; it predicts the majority of  
115 the query points within 2 pixels of the ground truth. While SEA-RAFT, which is supervised, is also  
116 precise for lower thresholds, the magnitude of the error for wrong predictions is evidently higher, as  
117 its performance quickly plateaus for higher thresholds.

118 As discussed in Section 4 in the main paper, we further evaluate on a custom constant-frame gap  
119 protocol (CFG) for fairer comparison with optical flow baselines. As shown here in Figure 2, all

models improve significantly under this less challenging setup. In particular, optical flow baselines exhibit strong sub pixel precision. However, we see that in general, compared to self-supervised baselines, Opt-CWM make reasonable predictions of a point’s next frame location more often, at a rate comparable to the fully supervised SEA-RAFT.

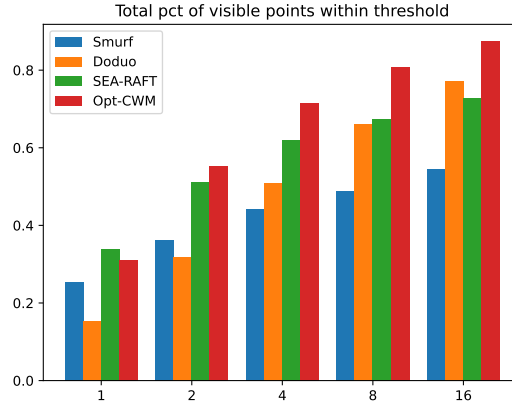


Figure 1: **TAP-Vid First: comparing baseline models on  $< \delta_{\text{avg}}$  broken down across thresholds ( $x$ -axis).** Fraction of points with error less than a fixed threshold, as a function of baseline model. Compared to baseline models, Opt-CWM maintains high performance on all thresholds even when making predictions across large frame gaps, as is necessary for TAP-Vid First.

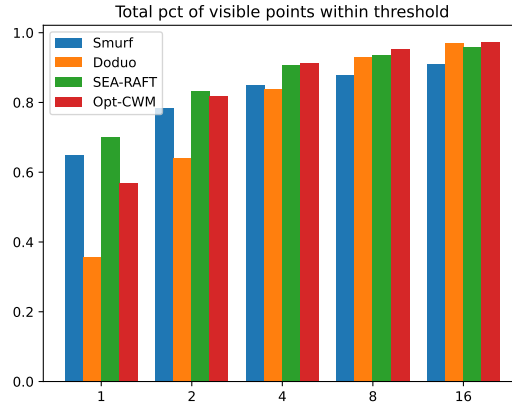


Figure 2: **TAP-Vid CFG: comparing baseline models on  $< \delta_{\text{avg}}$  broken down across thresholds ( $x$ -axis).** Fraction of points with error less than a fixed threshold, as a function of baseline model. For fair comparison, we also evaluate on a constant frame gap setting that is more favorable to optical flow baselines. While baseline methods show strong performance for very low thresholds ( $< 2$  pixels), we see that in general Opt-CWM outperforms self-supervised methods and is comparable with SEA-RAFT in predicting more points within a reasonable boundary.

### 124 B.3 Perturbation Across Epochs

125 The performance of  $\text{FLOW}_\theta$  is greatly dependent on the quality of its learned Gaussian perturbations. In  
 126 Figure 4, we see that the appearance of the perturbation evolves alongside the training of Opt-CWM.  
 127 As the perturbation converges into an optimal patch bespoke for the input frame, the quality of the  
 128 flow prediction improves in tandem.

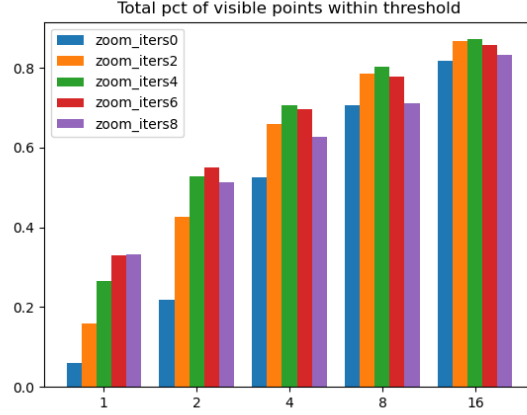


Figure 3:  $\langle \delta_{\text{avg}} \rangle$  broken down across thresholds ( $x$ -axis). Fraction of points with error less than a fixed threshold, as a function of number of multiscale (MS) iterations, for pixel thresholds 1, 2, 4, 8, and 16. We find that 4 zoom iterations tends to perform the best, especially for robustness on difficult examples (evidenced by better performance on higher thresholds).

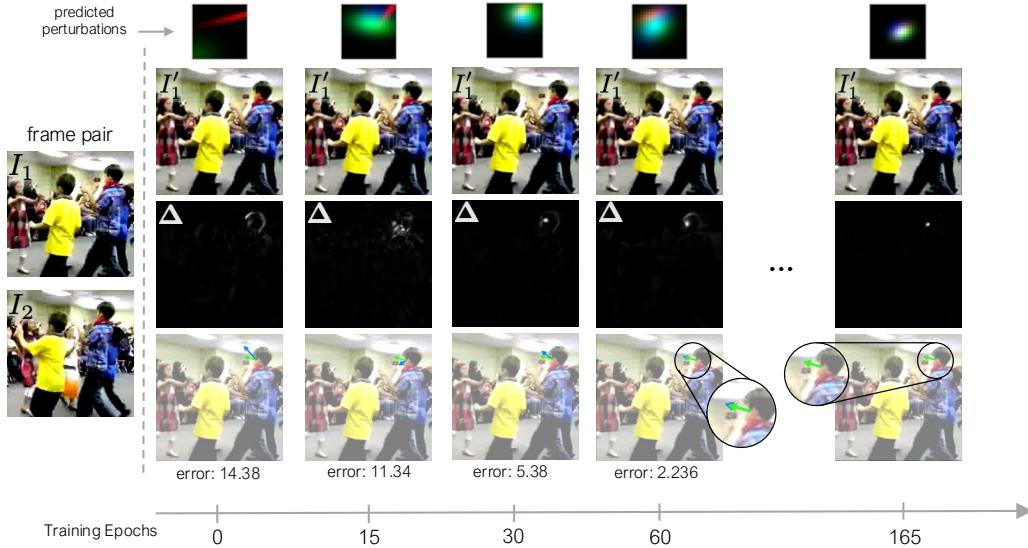


Figure 4: **Evolution of perturbations across training epochs:** We observe how the predicted perturbations change as the model trains. The perturbation starts as a disjoint streak of colors and converges to a localized peak. This in turn increasingly concentrates the difference image  $\Delta$  and leads to better flow prediction. Green is the ground truth flow obtained from the TAP-Vid dataset, and blue is our model’s prediction.

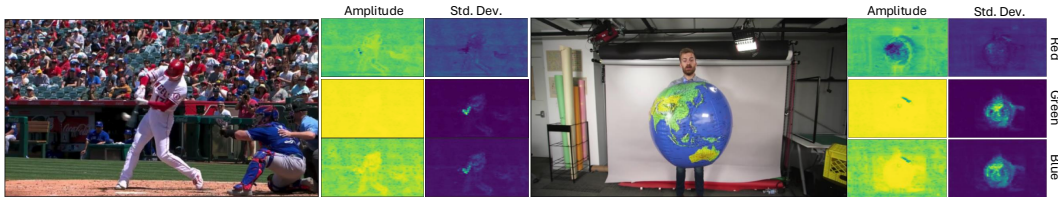


Figure 5: **Perturbation maps emergently reflect scene properties.** For two example frame pairs, we show the amplitudes and standard deviations, at each spatial position and for each color channel, of the optimal Gaussian perturbations predicted by  $\text{MLP}_{\theta}$ . These “perturbation maps” emergently reflect scene properties, with perturbation parameters varying in size and magnitude depending on where they are located in the image, corresponding to the presence of foreground objects and their parts.

## References

- [1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.
- [2] Daniel M Bear, Kevin Feigels, Honglin Chen, Wanhee Lee, Rahul Venkatesh, Klemen Kotar, Alex Durango, and Daniel LK Yamins. Unifying (machine) vision via counterfactual world modeling. *arXiv preprint arXiv:2306.01828*, 2023.
- [3] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.
- [4] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [5] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.
- [6] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz MuellerFreitag, Florian Hoppe, Christian Thureau, Ingo Bax, and Roland Memisevic. The “Something Something” Video Database for Learning and Evaluating Visual Common Sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5843–5851, 2017. URL <https://20bn.com/datasets/something-something/v2>. Dataset: 20BN–Something–Something V2.
- [7] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.
- [8] Zhenyu Jiang, Hanwen Jiang, and Yuke Zhu. Doduo: Learning dense visual correspondence from unsupervised semantic-aware flow. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12420–12427. IEEE, 2024.
- [9] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.
- [10] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017.
- [11] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
- [12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [13] Henrique Morimitsu, Xiaobin Zhu, Roberto M Cesar Jr, Xiangyang Ji, and Xu-Cheng Yin. Dpflow: Adaptive optical flow estimation with a dual-pyramid framework. *arXiv preprint arXiv:2503.14880*, 2025.
- [14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/radford21a.html>.
- [15] Ayush Shrivastava and Andrew Owens. Self-supervised any-point tracking by contrastive random walks. In *European Conference on Computer Vision (ECCV)*, 2024. URL <https://arxiv.org/abs/2409.16288>.

- 178 [16] Austin Stone, Daniel Maurer, Alper Ayyaci, Anelia Angelova, and Rico Jonschkowski. Smurf:  
179 Self-teaching multi-frame unsupervised raft with full-image warping. In *Proceedings of the*  
180 *IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2021.
- 181 [17] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In  
182 *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020,*  
183 *Proceedings, Part II 16*, pages 402–419. Springer, 2020.
- 184 [18] Rahul Venkatesh, Honglin Chen, Kevin Feiglis, Daniel M Bear, Khaled Jedoui, Klemen Kotar,  
185 Felix Binder, Wanhee Lee, Sherry Liu, Kevin A Smith, et al. Understanding physical dynamics  
186 with counterfactual world modeling. *arXiv preprint arXiv:2312.06721*, 2023.
- 187 [19] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool.  
188 Temporal segment networks for action recognition in videos. *IEEE transactions on pattern*  
189 *analysis and machine intelligence*, 41(11):2740–2755, 2018.
- 190 [20] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical  
191 flow. *arXiv preprint arXiv:2405.14793*, 2024.