- Supplementary Materials -
Representation Learning for Visual Object Tracking by Masked Appearance Transfer

A. Training Setting

A.1. MAT Pre-training

Table 1 shows the settings for the MAT pre-training. The template size and search region size are set to 2 and 4 times the size of the target according to the "**scale factor**". The "**jitter factor**" is used for the *PyTracking* [3] style data augmentation, denoting the *scale jitter factor* and the *center jitter factor* respectively.

Table 1. MAT pre-training setting.

Setting	Value
search region size	224×224
template size	112×112
scale factor of search region	4
scale factor of template	2
jitter factor of search region	$[0.5, 3.0]$
jitter factor of template	$[0.0, 0.0]$
optimizer	AdamW [9]
base learning rate	$1e^{-4}$
weight decay	$5e^{-2}$
learning rate schedule	MultiStep
MultiStep milestones	$[200]$
MultiStep gamma	0.1
batch size per GPU	32
gradients accumulation steps	2
samples per epoch	$64,000$
epochs	500

A.2. Tracker Training

Table 2 shows the settings for the tracker training. We keep most settings unchanged. The data augmentation method is the same as that in MAT pre-training. We use less epochs and smaller weight decay for our tracker. The ViT encoder (i.e., "backbone") is tuned with $0.1\times$ base learning rate, which is a common practice in many works, such as STARK [10], TransT [1], and MixFormer [2].

Table 2. Tracker training setting.

Setting	Value
search region size	224×224
template size	112×112
scale factor of search region	4
scale factor of template	2
jitter factor of search region	$[0.5, 3.0]$
jitter factor of template	$[0.0, 0.0]$
optimizer	AdamW [9]
base learning rate	$1e^{-4}$
weight decay	$1e^{-4}$
learning rate schedule	MultiStep
MultiStep milestones	$[240]$
MultiStep gamma	0.1
batch size per GPU	32
gradients accumulation steps	2
samples per epoch	$64,000$
epochs	300
learning rate of encoder	$0.1\times$base lr
loss weight for L_{GIoU}	2
loss weight for L_1	5

B. Tracker

The proposed tracker is illustrated in Figure 1.

Feature encoding. We use the MAT pre-trained ViT-B/16 encoder to encode the image tokens without using *masking out*. The search region is encoded to have the size of 196×768, where 196 is the number of tokens and 768 is the embedding dimensions. Similarly, the template is encoded to have the size of 49×768.

Matching. We use the *depth-wise correlation* matching operator [8] to fuse the encoded features. First, we use a linear layer to reduce the embedding dimensions from 768 to 256 for saving computational costs. Second, we resize these tokens to square feature maps, having the sizes of

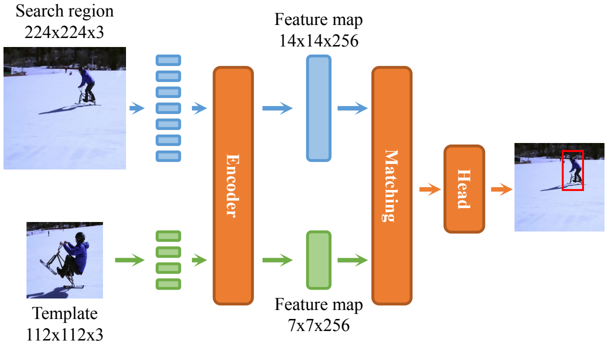

Figure 1. The ViT encoder encodes the images jointly. The depth-wise correlation operator fuses the encoded features. The anchor-free head predicts the top-left corner and bottom-right corner which are reformated into a bounding box.

$14 \times 14 \times 256$ and $7 \times 7 \times 256$, respectively. Third, we make a depth-wise correlation between the reshaped feature maps by using the PyTorch built-in function ‘conv2d’ and setting the padding to 3. Thus, we can get a response map in the size of $14 \times 14 \times 256$.

Head. The tracker head is the same as the MixFormer’s head [2], which is a CornerNet-based [7] head and can regress two corner probability distribution maps. For each map, it employs five 3×3 convolution layers, padding is set to 1 to keep the output resolution, and BatchNorm and ReLU are sandwiched between them.

Tracking pipeline. First, we crop the template in the initial frame according to the ground truth, and we do not update this template during tracking. Second, in each frame, we crop the search region according to the last predicted box by following the common practice in visual tracking. Both this search region and the template are jointly encoded. In this way, the target representations in both the template feature and search region feature are close to each other in the feature space. Then, we use the typical depth-wise correlation operator to match the template feature over the search region feature. Third, we use the head on the matched search region feature to predict the bounding box.

Computational cost during tracking. The tracking process consists of image processing, encoding forward pass, and prediction forward pass.

(1) The image processing, which includes cropping, resizing, and normalization, is consistent with the common practices of Siamese trackers, and it takes little time during tracking.

(2) We use the standard ViT-B/16 model as the encoder. The joint encoding has 21.04 GFLOPs, and its forward pass takes a significant amount of time during tracking.

(3) We use the depth-wise correlation operator and a lightweight prediction head. The total GFLOPs for the prediction forward pass is 0.43.

C. Visualization

C.1. Reconstruction

We show more reconstruction samples from the MAT pre-trained autoencoder with the default 25% masking ratio in Figure 2 and 3. We also show the reconstruction samples by using a 75% high masking ratio in Figure 4. For better visualization, we put the target at the center of the search region. We can see the low reconstruction quality with this high masking ratio.

C.2. Response Map

We show the response maps (in Figure 5) that were generated by using the *depth-wise correlation* operator to fuse the template feature and search region feature. For each quadruplet, the **3rd** response map is generated by using the original MAE [5] pre-trained ViT-B/16 [4] encoder to extract the features of the template and search region separately. The **4th** response map is generated by using our MAT pre-trained encoder to extract the features of the template and search region jointly.

For better visualization, we put the target at the center of the search region. We can see that the response maps in the 4th column always have a strong response at the center. But the maps in the 3rd column have worse responses in many cases. This observation suggests that our MAT method actually learned better representations for tracking.

References

[1] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8126–8135, 2021. 1

[2] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–12, 2022. 1, 2

[3] Martin Danelljan, Goutam Bhat, Christoph Mayer, and Felix Järemo-Lawin. Pytracking. https://github.com/visionml/pytracking, 2022. 1

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*, pages 1–12, 2021. 2

[5] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2022. 2

[6] Lianghua Huang, Xin Zhao, and Kaiqi Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1562–1577, 2021. 3, 4, 5, 6

[7] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Proceedings of the European Conference on Computer Vision*, pages 765–781, 2018. 2

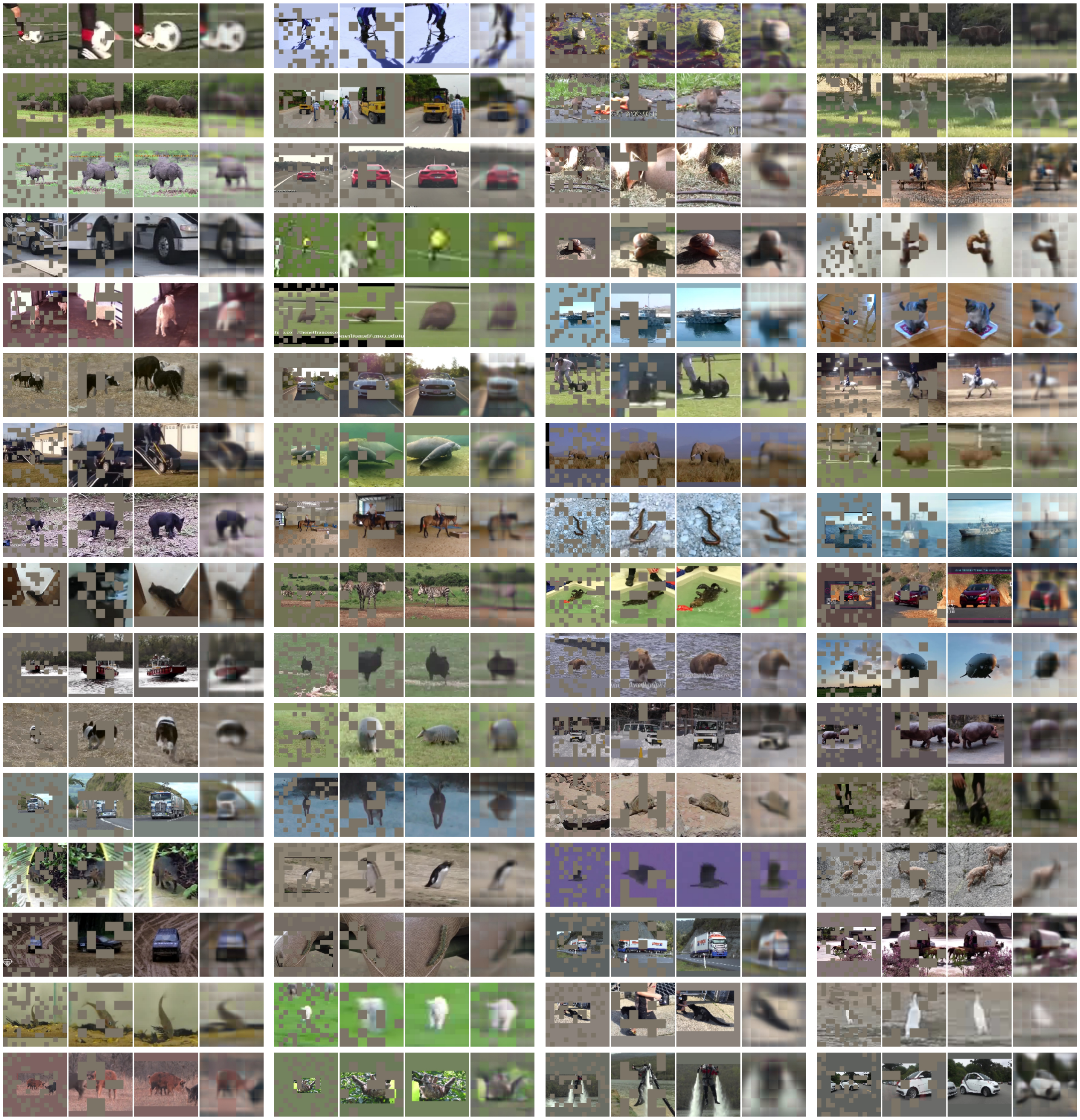

Figure 2. Random samples on the GOT10k [6] *val* set. For each quadruplet, we show the masked search regions (**1st**), the masked templates (**2nd**), the new templates from the search region (**3rd**), and the reconstructed new templates (**4th**), from left to right. The masking ratio is 25%.

[8] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019. 1

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, pages 1–11, 2019. 1

[10] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10448–10457, 2021. 1

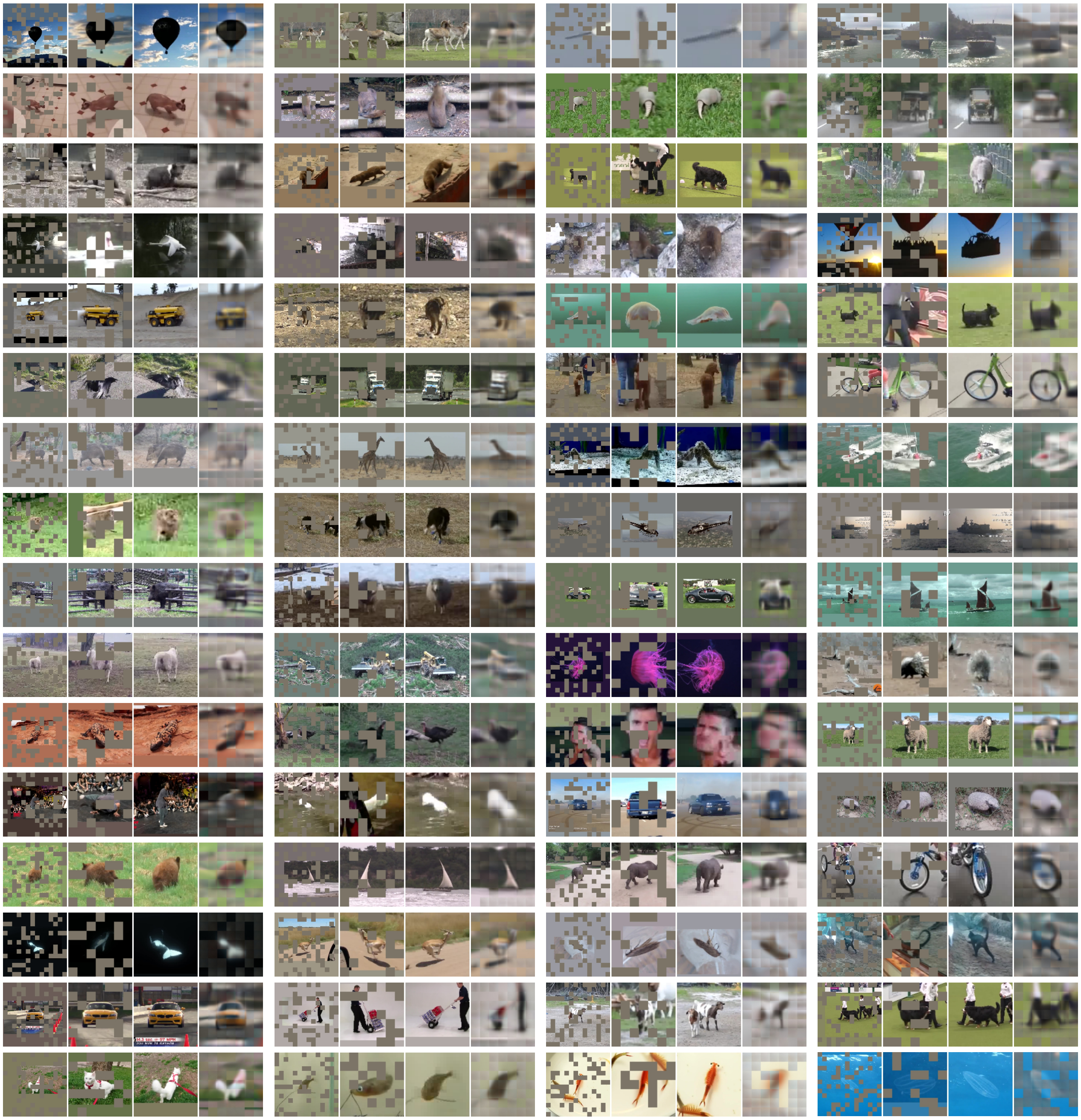

Figure 3. Random samples on the GOT10k [6] *val* set. For each quadruplet, we show the masked search regions (**1st**), the masked templates (**2nd**), the new templates from the search region (**3rd**), and the reconstructed new templates (**4th**), from left to right. The masking ratio is 25%.

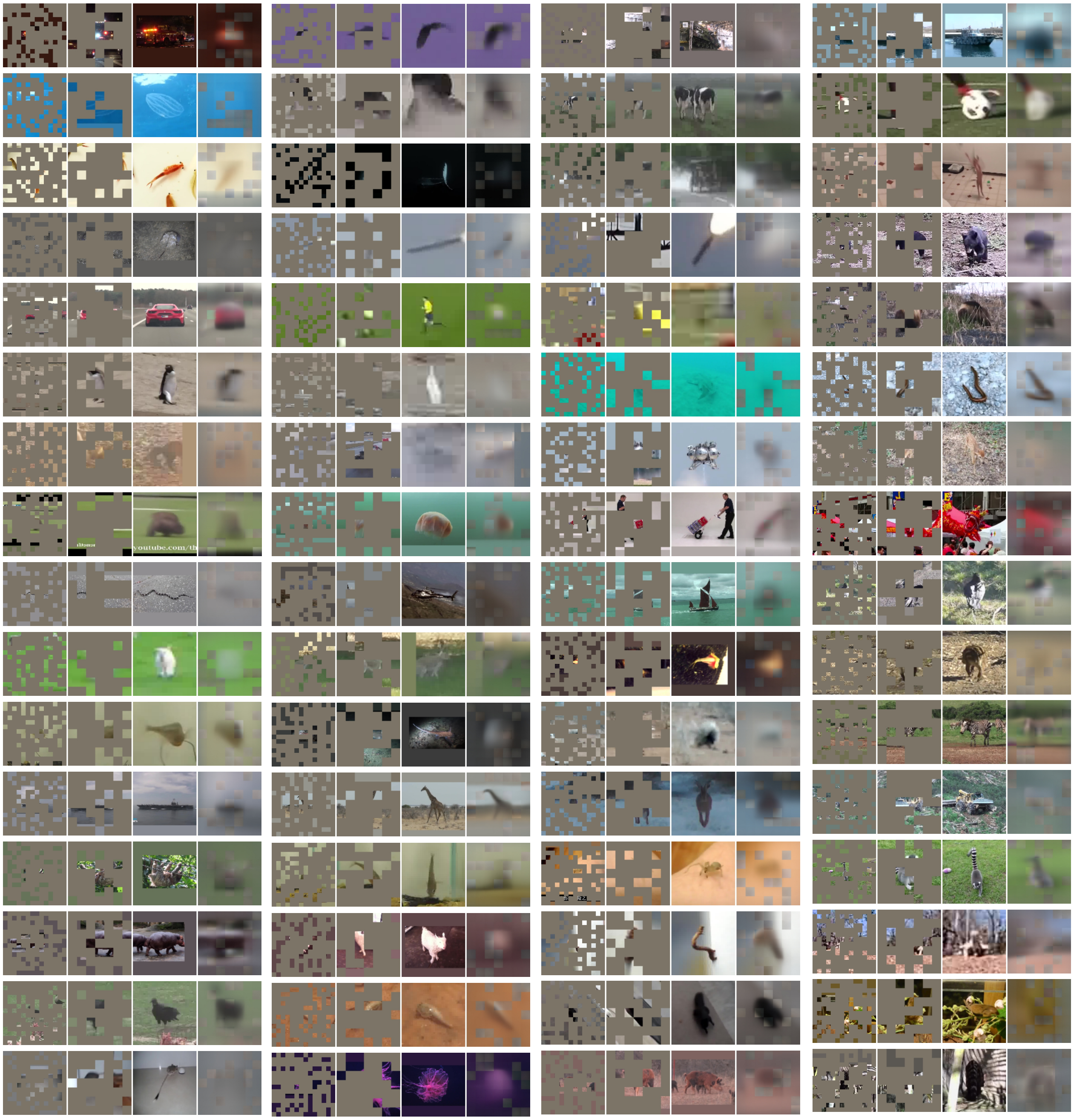

Figure 4. Random samples on the GOT10k [6] *val* set by using a 75% high masking ratio. For each quadruplet, we show the masked search regions (**1st**), the masked templates (**2nd**), the new templates from the search region (**3rd**), and the reconstructed new templates (**4th**), from left to right.

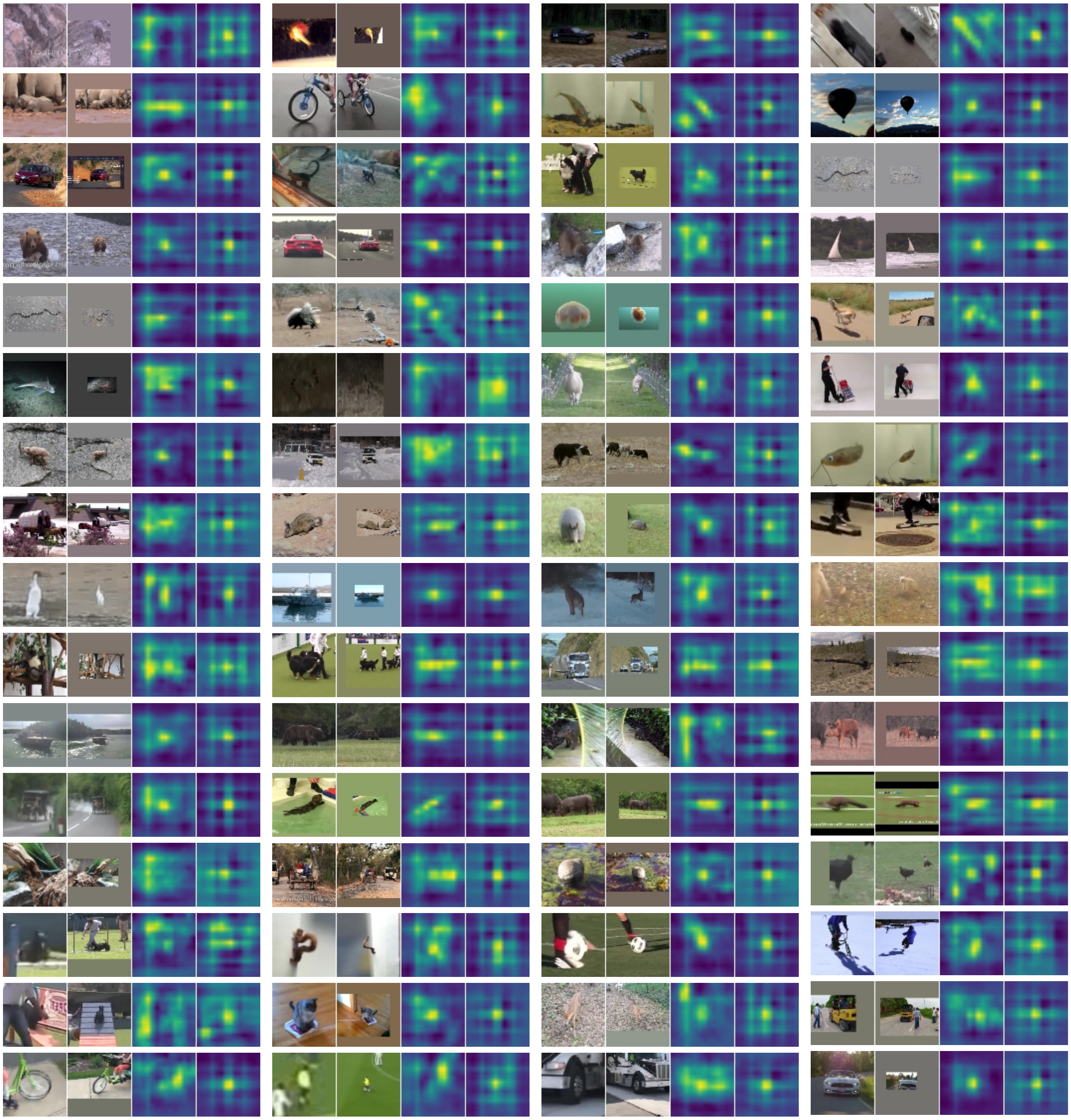

Figure 5. Random samples on the GOT10k [6] *val* set. For each quadruplet, we show the templates (**1st**), the search regions (**2nd**), the response maps w.r.t. the original MAE encoder (**3rd**), and the response maps w.r.t. our MAT pre-trained encoder (**4th**), from left to right.