FAST VIDEO OBJECT SEGMENTATION VIA DYNAMIC YOLACT

Tianfang Meng *Wenqiang Zhang*[✉*]

Shanghai Key Laboratory of Intelligent Information Processing
School of Computer Science, Fudan University, Shanghai, China

ABSTRACT

Video Object Segmentation (VOS) is a fundamental task in video recognition with many practical applications. It aims at predicting segmentation masks of multiple objects in an entire video. Recent video object segmentation(VOS) researches have achieved remarkable performance. However, as a video processing task, the inference speed of the VOS method is also essential. VOS can be considered an extension of semantic segmentation from a static image to a dynamic image sequence. Following this idea, we propose a fast VOS framework based on YOLACT, a real-time static image segmentation framework. We employ a fast online training technique to make YOLACT grow wings to handle dynamic video sequences and achieve competitive performance(77.2 $J\&F$ and 30.9 FPS on DAVIS17) among fast VOS methods. Moreover, by linearly combining mask bases to generate masks for arbitrary objects, our method can process multi-object videos with minimal extra computations.

Index Terms— Video Object Segmentation, Conjugate Gradient Optimization, Online Training

1. INTRODUCTION

Video Object Segmentation (VOS) is a fundamental task in video recognition with many applications scenarios. The goal of the VOS task is to predict the segmentation mask for each object in the frames of a video. Due to the movement and deformation of the object across the video frames, VOS becomes a challenging problem. In the semi-supervised VOS task, which we focus on in this work, the ground-truth segmentation mask of the first frame in the video is given, we are to segment object instances in the rest of the video frames.

Recent VOS methods[1, 2] have achieved remarkable accuracy. The current state-of-the-art VOS methods are mainly based on pixel matching. In such frameworks, the object mask of the current frame is predicted by matching its feature with the reference frame(usually the first frame or the previous frame) feature. However, due to the high computation cost, these methods are not available for many application scenarios with real-time requirements. Generally, video processing methods with 30 or higher FPS(frames per second) are considered real-time methods. Some general model compression methods[3, 4, 5] can be adopted to speed up inference, but researchers are also exploring specific faster VOS frameworks[6, 7, 8, 9]. Furthermore, multi-object segmentation is also a great challenge in VOS. Many of the current methods are designed for single-object scenarios and perform poorly in the inference speed when dealing with multi-target videos.

Considering the above issues, we propose a fast VOS framework based on a online training mechanism to obtain competitive segmentation results with a real-time inference speed. The existing online training methods include first-frame fine-tuning methods[10, 7] and tracking methods[9], which use more segmented frames to continuously optimize the model. The advantage of these methods is that they can be improved based on mature image segmentation methods, which leads to higher accuracy and portability. However, due to online backpropagation and optimization steps, these methods tend to have low inference speeds. To deal with this defect, we apply the online training process on only a tiny part of an image segmentation model and adopt the GN-CG online optimizer[9, 11] to achieve a real-time inference speed. Furthermore, a mask base predicting technique based on YOLACT[12] is adopted in our model, which brings competitive performances on multi-object videos.

The main contributions of this paper can be summarized as follows: (1) We propose a real-time VOS framework based on YOLACT[12], which predicts the segmentation mask for each instance by linearly combining the mask bases. With the online training mechanism, the dynamic YOLACT framework achieves fast and robust VOS performances. (2) Taking advantage of a fast optimization technique to train the lightweight module for each object online, our model can maintain real-time inference speed for multi-object videos.

2. PROPOSED METHOD

Fig.1 illustrates the overview of our framework. The model process the frame feature with two subnetworks. In ProtoNet,

*This work was supported in part by National Key R&D Program of China (No.2020AAA0108301) and National Natural Science Foundation of China (No.62072112). This work was also supported in part by scientific and technological innovation action plan of Shanghai Science and Technology Committee (No.205111031020).

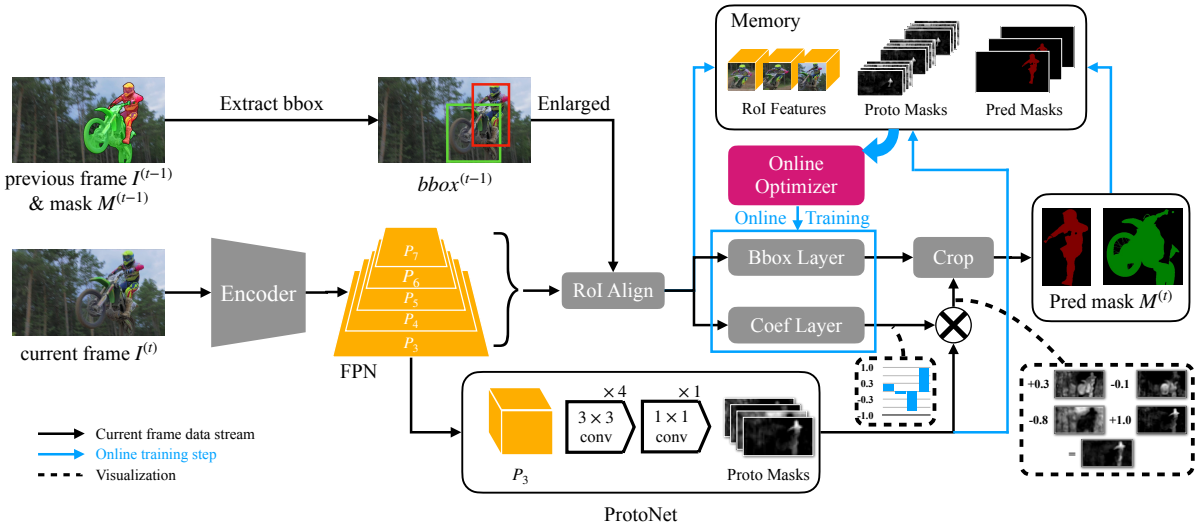

Fig. 1: An overview of the proposed dynamic YOLACT framework for real-time VOS.

one of the subnetworks, based on the largest-scale feature P_3 from the bottom layer of FPN, several proto masks are predicted. The proto masks are the object-agnostic mask bases, by which we can obtain the segmentation mask for any objects via linear combinations of them. In the other subnetwork, the model adaptively selects the feature with a proper scale from FPN according to the bounding box of the object in the previous frame. Then an RoI align[13] operation is adopted to obtain the feature of the target object with a fixed size. This feature is object-specific and is exploited to regress the precise bounding box in the Bbox Layer and predict the combination coefficients of the proto masks in the Coef Layer. Finally, the segmentation masks of the target objects are generated by the linear combinations of the proto masks and their coefficients and then cropped by the predicted bounding box.

The light-colored arrows in Fig.1 indicate the online training process which makes YOLACT grow wings to perform dynamic video object segmentation and tracking. The memory, which consists of RoI(Region of Interest) features, proto masks, and predicted masks of the past frames, is used to train the subnetworks of the Bbox and Coef Layer with an online optimizer.

2.1. Fast VOS framework via mask bases combination

Given a video $V = \{I^{(1)}, I^{(2)}, ..., I^{(T)}\}$ with T frames and the reference mask $\boldsymbol{y}_m^{(1)} = \{y_m^{(1,1)}, y_m^{(1,2)}, ..., y_m^{(1,n)}\}$ of n objects in the first frame, we are to predict the object masks $\{\hat{y}_m^{(t)}\}_{t=2}^{T}$ for the rest frames. We employ ResNet-50[14] as the backbone encoder and use FPN[15] to extract multi-scale features. Here, we denote the five-layer features from FPN as $P_i, (i \in \{3, 4, 5, 6, 7\})$, where P_i represents the feature map whose height and width are $1/2^i$ of the original image size. The number of feature map channels for each layer is 256, that is, $P_i \in \mathbb{R}^{256 \times H/2^i \times W/2^i}$, where H and W denote the height and width of the original image. Then the feature is processed by two subnetworks: the Bbox & Coef prediction net and the ProtoNet.

As for the first subnetwork, according to the bounding box data from the previous frame, one feature map from P_3 to P_7 with a proper scale is selected and cropped by RoI align[13]. Our model select the feature layer with a similar rule in FPN[15], that is,

$$i = \lfloor i_0 + \log_2 \sqrt{h_b w_b / HW} \rfloor \tag{1}$$

In Eq.1, H and W are the height and width of the feature map, and h_b and w_b are the height and width of the bounding box. According to the formula, we select feature P_i for further prediction, and i_0 is set to 7. The height and width of the bounding box data of the previous frame are multiplied by a factor α to deal with the situation where the object appears to be shifted or zoomed. Then the feature maps $F_R^{(i)}, i \in [1, n]$ for all objects with a fixed shape of $256 \times d \times d$ is obtained for each target object.

For the feature $F_R^{(i)}$ of each of the n objects, we use them to predict the bounding box $\hat{y}_b$ and k coefficients $\hat{y}_c$. The bounding box and coefficient prediction layers are the only two layers in the network that need to be trained online. We refer to the convolution-based bounding box regression structure in [16] to build our Bbox and Coef Layer, which requires less number of online training parameters.

As for the second subnetwork, the ProtoNet, we use four 3×3 and one 1×1 convolutional layers to constitute it. ProtoNet takes P_3 from FPN as input and outputs k proto masks with the size $k \times H/8 \times W/8$, whose linear combination is the final segmentation mask. The proto masks are denoted by $B \in \mathbb{R}^{k \times H/8 \times W/8}$ in the following. The ProtoNet is object-agnostic and exclusively trained offline.

The final object masks are generated by the linear combination of k proto masks $\boldsymbol{B} = B_1, B_2, ..., B_k$. The k coefficients from the Coef Layer correspond to the weights of the proto masks.

2.2. Fast Online Training

In the inference step, the RoI features and segmentation masks of each frame are stored in memory to train the Bbox and Coef Layers online. Each object has an independent memory with a maximum capacity of 60 most recent frames.

For faster online optimization, we employ the GN-CG optimizer in [9], which uses the Gauss-Newton(GN) optimization as its parameter optimization strategy, and uses the Conjugate Gradient(CG) descent to estimate the optimization distance in the GN steps. This method requires more computations in a single iteration(related to the number of parameters) but can optimize the parameters to expected ranges with minimal iterations, which is in line with our model.

Our loss function consists of two parts: bounding box loss L_b and coefficient lossL_c. In the offline training phase, our L_b and L_c are the same as in [12]. In the online training phase, due to light computation requirements, we use the following MSE(mean square error) loss,

$$L_b(\boldsymbol{w}; M) = \sum_{t=1}^{T-1} \left\| \tilde{y}_b^{(t)} - \hat{y}_b^{(t)} \right\|^2 \tag{2}$$

$$L_c(\boldsymbol{w}; M) = \sum_{t=1}^{T-1} \left\| \tilde{y}_m^{(t)} - U(\hat{y}_m^{(t)}) \right\|^2 \tag{3}$$

Here, $M = \{B^{(t)}, F_R^{(t)}, \tilde{y}_m^{(t)}\}_{t=0}^{T-1}$ is the memory that consists of the proto masks B, RoI features F_R and predicted masks $\tilde{y}_m$ obtained in the inference step from frame $t = 1$ to $T - 1$. $\tilde{y}_b$ is the predicted bounding box extracted from $\tilde{y}_m$, which is used as the online training labels of the bounding box. $\hat{y}_b$ and $\hat{y}_m$ are the prediction results at the online training step. U represents the upsampling operation with a bilinear interpolation that resizes the predicted masks $\hat{y}_m$ to the same size as the labels.

To minimize loss L_b and L_c, we adopt the GN-CG optimizer in the online training step. In each iteration of the GN-CG optimization method, the quadratic approximations of the loss L_b and L_c are calculated to get the optimal increment $\Delta \boldsymbol{w}_b$ and $\Delta \boldsymbol{w}_c$. Taking L_b as an example, the following formula is the approximate form of it, and L_c has similar form.

$$L_b(\boldsymbol{w}_b + \Delta \boldsymbol{w}_b) \approx \Delta \boldsymbol{w}_b^T J_{\boldsymbol{w}_b}^T J_{\boldsymbol{w}_b} \Delta \boldsymbol{w}_b + 2 \Delta \boldsymbol{w}_b^T J_{\boldsymbol{w}_b}^T r_{\boldsymbol{w}_b} + r_{\boldsymbol{w}_b}^T r_{\boldsymbol{w}_b} \tag{4}$$

In Eq.4, $r_{\boldsymbol{w}_b} = \tilde{y}_b - \hat{y}_b$ is the residual, which is the unsquared form of L_b, and $J_{\boldsymbol{w}_b}$ is the Jacobian matrix of $r_{\boldsymbol{w}_b}$ at $\boldsymbol{w}_b$. Then Eq.4 becomes a positive definite quadratic problem. We use the Conjugate Gradient(CG) descent[17] to minimize it over $\Delta \boldsymbol{w}_b$, and then perform next GN iteration after updating $\boldsymbol{w}_b \leftarrow \boldsymbol{w}_b + \Delta \boldsymbol{w}_b$.

3. EXPERIMENTS

We evaluated our method on the two VOS benchmarks, DAVIS and YouTube-VOS, and achieved competitive results. With minimal extra computations on multiple target objects, our method maintains a high inference speed on multi-object videos.

3.1. Datasets and Evaluation Metrics

We evaluated our method on DAVIS and YouTube-VOS datasets, which are two large-scale VOS benchmarks. DAVIS has two versions, DAVIS16 (single object) and DAVIS17 (multiple object). The latter is an extension of the former, which contains 60 training videos and 30 validation videos. The validation set of the YouTube-VOS dataset includes 65 categories that have appeared in the training set and 26 categories that have not, and their results are distinguished by *Seen* and *Unseen* in the table.

$J\&F$ is used to evaluate the performance of the methods. The regional accuracy J measures the IoU(Intersection over Union) of all pixels, and the boundary accuracy F measures the accuracy of boundary pixels of the object. FPS(Frames Per Second) is used to evaluate the computational speed of the methods.

3.2. Implementation Details

We pre-trained our model on the MS-COCO dataset[20], which contains a large amount of static image data and annotations, and then trained it on the benchmark datasets to get a robust ProtoNet to generate proto masks with any objects.

In the offline training step the model is optimized by AdamW[21]. The learning rate is set to 5×10^{-4} and decay to 1×10^{-5} with a polynomial learning rate policy. The proposed model is implemented with PyTorch[22] and runs on a single NVIDIA v100 GPU.

3.3. Results on Benchmarks

We compare the performance of our method with other fast VOS methods on the benchmark datasets in Table 1. The performances of some accuracy-focused methods are also listed. As is shown in the table, on the single-object dataset DAVIS16, our method has a competitive performance on accuracy($J\&F = 83.8\%$) and the highest inference speed, with an FPS of 34.6. On the multi-object benchmark DAVIS17, our approach achieves $J\&F = 77.2\%$ and FPS$= 30.9$, which leads in accuracy among fast (FPS ≥ 10.0) VOS methods. On the YouTube-VOS benchmark, our method achieves the highest accuracy ($J\&F = 75.8\%$) among the fast VOS methods. Meanwhile, for the objects of the *Unseen* categories, our model maintains high regional accuracy J (70.1%) and edge accuracy F (77.5%), which reflects the high generalization ability of the ProtoNet.

Table 1: State-of-the-art comparison on benchmarks. The method compares the accuracy separately according to whether the FPS is greater than 10.

Method	YouTube-VOS					DAVIS16				DAVIS17			
	Overall	Seen		Unseen		$J\&F$	J	F	FPS	$J\&F$	J	F	FPS
		J	F	J	F								
OSVOS[10]	58.8	59.8	60.5	54.2	60.7	-	-	-	-	60.3	56.6	63.9	0.22
PReMVOS[18]	66.9	71.4	75.9	56.5	63.7	86.8	84.9	88.6	0.03	77.9	73.9	81.8	0.03
STM[2]	79.4	79.7	84.2	72.8	80.9	**89.4**	**88.7**	90.1	**9.27**	81.8	**79.2**	84.3	**7.97**
AFB-URR[19]	79.6	78.8	83.1	74.1	79.6	-	-	-	-	74.6	73.0	76.1	4.00
CFBI[1]	**81.0**	**80.6**	**85.1**	**75.2**	**83.0**	**89.4**	88.3	**90.5**	8.31	**81.9**	79.1	**84.6**	7.61
AGAME[8]	66.0	66.9	-	61.2	-	81.9	81.5	82.2	14.3	71.1	68.5	73.6	14.3
TVOS[6]	67.8	67.1	69.4	63.0	71.6	-	-	-	-	72.3	69.9	74.7	**37.0**
FRTM[9]	72.1	72.3	76.2	65.9	74.1	81.7	-	-	21.9	76.7	-	-	15.3
Ours	**75.8**	**75.9**	**79.7**	**70.1**	**77.5**	**83.8**	**83.4**	**84.1**	**34.6**	**77.2**	**76.3**	**78.1**	30.9

Table 2: Ablation studies on DAVIS17 benchmark.

Variants	$J\&F$	J	F	FPS
w/o Bbox Layer	49.8	48.3	51.3	-
w/o ProtoNet	72.4	71.0	73.9	13.4
SGD online	77.0	**76.5**	77.4	0.82
SGD online(less iters)	59.1	58.7	59.5	8.09
Full model & GN-CG	**77.2**	76.3	**78.1**	**30.9**

In addition, many methods encounter an evident speed reduction when processing multi-object videos. In contrast, our method maintains a high inference speed in multi-object scenarios with the minimal extra computations to predict the bounding boxes and coefficients of different objects.

Some qualitative results on the DAVIS17 benchmark are displayed in Fig.2. As shown in the figure, the prediction of the instance-level bounding boxes improves the accuracy of the results of videos with similar target objects (e.g., multiple dogs or people). Furthermore, the online training mechanism helps the model generate accurate segmentation masks when encountering complex situations with shifting and deformation of the target object.

3.4. Ablation studies

The results of ablation studies in Table 2 demonstrate the effectiveness of the key modules we proposed. As shown in the table, our Bbox Layer effectively improves the $J\&F$, and ProtoNet brings better FPS for multi-object video. As for the online optimizer, if the commonly used stochastic gradient descent (SGD) optimizer is adopted, the large requirement for the number of iterations leads to low FPS. In turn, the accuracy drops dramatically if the number of iterations is limited. The GN-CG optimizer we employ has a better trade-off between speed and accuracy.

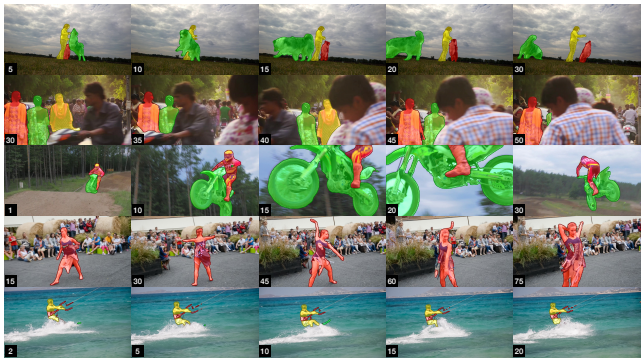

Fig. 2: Qualitative results of our approach on the DAVIS17 dataset. The numbers in the lower-left corner are the frame numbers.

4. CONCLUSION

We propose a fast and high-accuracy video object segmentation method by applying a fast online training technique to the image segmentation method. Since only tiny modules need to be trained online for each object, our method maintains a high processing speed in multi-object scenarios. The competitive accuracy and speed on benchmark datasets demonstrate the effectiveness of the proposed method.

5. REFERENCES

[1] Zongxin Yang, Yunchao Wei, and Yi Yang, "Collaborative video object segmentation by foreground-background integration," in *European Conference on Computer Vision*. Springer, 2020, pp. 332–348.

[2] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, 2019, pp. 9226–9235.

[3] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han, "Amc: Automl for model compression and acceleration on mobile devices," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 784–800.

[4] Jinyang Guo, Wanli Ouyang, and Dong Xu, "Channel pruning guided by classification loss and feature importance," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 10885–10892.

[5] Jinyang Guo, Wanli Ouyang, and Dong Xu, "Multi-dimensional pruning: A unified framework for model compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1508–1517.

[6] Yizhuo Zhang, Zhirong Wu, Houwen Peng, and Stephen Lin, "A transductive approach for video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6949–6958.

[7] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao, "Ranet: Ranking attention network for fast video object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3978–3987.

[8] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg, "A generative appearance model for end-to-end video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8953–8962.

[9] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg, "Learning fast and robust target models for video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7406–7415.

[10] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool, "One-shot video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 221–230.

[11] Henning Tjaden, Ulrich Schwanecke, Elmar Schömer, and Daniel Cremers, "A region-based gauss-newton approach to real-time monocular multiple object tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1797–1812, 2018.

[12] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9157–9166.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[17] Magnus Rudolph Hestenes, Eduard Stiefel, et al., *Methods of conjugate gradients for solving linear systems*, vol. 49, NBS Washington, DC, 1952.

[18] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe, "Premvos: Proposal-generation, refinement and merging for video object segmentation," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 565–580.

[19] Yongqing Liang, Xin Li, Navid Jafari, and Qin Chen, "Video object segmentation with adaptive feature bank and uncertain-region refinement," *arXiv preprint arXiv:2010.07958*, 2020.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[21] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.