REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
29-08-2019	Final Report	29-May-2018 - 24-May-2019

4. TITLE AND SUBTITLE
Final Report: DURIP: Building a GPU Computational Infrastructure Platform for Heterogeneous Big Data Analysis and Understanding

5a. CONTRACT NUMBER
W911NF-18-1-0209

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611103

6. AUTHORS

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
North Carolina State University
2701 Sullivan Drive
Admin Srvcs III, Box 7514
Raleigh, NC 27695 -7514

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
72148-CS-RIP.13

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Tianfu Wu
UU	UU	UU	UU		19b. TELEPHONE NUMBER 919-515-4361

RPPR Final Report
as of 29-Aug-2019

Agency Code:

Proposal Number: 72148CSRIP **Agreement Number: W911NF-18-1-0209**
INVESTIGATOR(S):

Name: Hamid D. Krim
Email: hamid.krim.civ@mail.mil
Phone Number: 9195132270
Principal: N

Name: Tianfu Wu
Email: twu19@ncsu.edu
Phone Number: 9195154361
Principal: Y

Organization: **North Carolina State University**
Address: 2701 Sullivan Drive, Raleigh, NC 276957514
Country: USA
DUNS Number: 042092122 EIN: 566000756

Report Date: 24-Aug-2019 Date Received: 29-Aug-2019
Final Report for Period Beginning 29-May-2018 and Ending 24-May-2019
Title: DURIP: Building a GPU Computational Infrastructure Platform for Heterogeneous Big Data Analysis and Understanding
Begin Performance Period: 29-May-2018 **End Performance Period:** 24-May-2019
Report Term: 0-Other

Submitted By: Hamid Krim Email: hamid.krim.civ@mail.mil
Phone: (919) 513-2270

Distribution Statement: 1-Approved for public release; distribution is unlimited.

STEM Degrees: **STEM Participants:**

Major Goals: In this proposed effort, we plan on building a powerful and flexible GPU computational infrastructure platform which consists of (i) a GPU cluster with more than 79,000 GPU cores and around 80 CPU cores including three requested cutting-edge GPU supercomputers and ten existing desktops, and (ii) a mobile base with on-board computer for robots. The platform will be used by both the Vision, Information, and Statistical Signal Theories and Applications (VISSTA) lab directed by the co-PI Krim, and the Interpretable Visual Modeling and Computing Lab (iVMCL) currently created by the PI Wu in the department of Electrical and Computer Engineering at NC State University (NCSU). The proposed computational platform will enable the two to address ongoing research projects on heterogeneous Big Data
analysis and deep understanding, as well as to smoothly prepare for future ones.

This new platform will complement and match the input modalities already present at the two labs, and expand current capabilities in aggregating, parsing, fusing and ultimately analyzing and understanding heterogeneous Big Data in the following applications relevant to the Department of Defense (DoD): (i) Sensor Networks of various modalities (static or mobile) such as deep understanding of scene and events of a camera network; (ii) Social Networks with information stored locally, and (iii) Bioinformatics data, specifically related to the brain connectome, both based on topological data analysis theory; (iv) Robot autonomy by learning from situated dialogue (i.e., verbal instructions grounded on visual demonstration) and physiological sensing. Not only will the proposed GPU cluster enable PI Wu and co-PI Krim to investigate and pursue a genuine parallel implementation of many already successful centralized models and algorithms, but also be beneficial to students (undergraduates and graduates) in the classes regularly taught by PI Wu and co-PI Krim, as well as other research groups at NCSU since the proposed cluster will be seamlessly integrated, and bring new feature, into the university's computing platform.

Accomplishments: With the support of this grant, our team has been developing a series of work on heterogeneous big data analyses and understanding in three domain areas:

- Domain I: Discriminative learning (image classification, line segment detection, object detection, tracking and segmentation and parsing); and

- Domain II: Generative learning (unconditional and conditional image synthesis).
- Domain III: Reinforcement learning

This report summarizes the developments in 6 tasks.
- Task 1: Developing deep grammar networks for deep learning.
- Task 2: Developing attentive normalization methods for deep learning.
- Task 3: Developing interpretable learning-to-learn methods for handling catastrophic forgetting
- Task 4: Developing attraction field representations for robust line segment detection.
- Task 5: Developing attentive pooling and reconfigurable normalization methods for image synthesis, image completion and image style transfer.
- Task 6: Developing novel experience replay methods for deep reinforcement learning.

We report a total of 11 papers in the pipeline of publication (8 published and 3 under review). These papers cover a broad range of topics in vision, learning, robotics and AI.

Training Opportunities: Nothing to Report

Results Dissemination: 8 research paper published, and source codes for some paper are released at the PI's lab's Github, https://github.com/iVMCL

Honors and Awards: Nothing to Report

Protocol Activity Status:

Technology Transfer: Nothing to Report

PARTICIPANTS:

Participant Type: PD/PI
Participant: Tianfu Wu
Person Months Worked: 1.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Co PD/PI
Participant: Hamid Krim
Person Months Worked: 1.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

ARTICLES:

RPPR Final Report
as of 29-Aug-2019

Publication Type: Journal Article Peer Reviewed: Y **Publication Status:** 4-Under Review
Journal: IEEE Transaction on Multimedia
Publication Identifier Type: Publication Identifier:
Volume: Issue: First Page #:
Date Submitted: 4/12/19 12:00AM Date Published:
Publication Location:
Article Title: A Bottom-Up and Top-Down Integrated Framework for Online Object Tracking
Authors: Meihui Li, Lingbing Peng, Tianfu Wu, Zhenming Peng
Keywords: Online object tracking, Bottom-up and Top- down, Graph regularized sparse coding, Alternating direction method of multipliers.
Abstract: Robust online object tracking entails integrating short-term trackers and long-term trackers in an elegant frame- work to handle structural and appearance variations of unknown objects in an online manner. The integration and synergy between short-term and long-term trackers have yet studied well in the literature, especially in pre-training free settings. To address this issue, this paper presents a bottom-up and top-down integrated framework. The bottom-up component realizes a data-driven approach for particle generation. It exploits a short-term tracker to generate bounding box proposals in a new frame based on current tracking results. In the top-down component, this paper proposes a graph regularized sparse coding scheme as the long-term tracker. A particle graph is computed whose nodes are the bottom-up discriminative particles and edges are formed on- the-fly in terms of appearance and spatial-temporal similarities between particles.
Distribution Statement: 1-Approved for public release; distribution is unlimited.
Acknowledged Federal Support: **Y**

Publication Type: Journal Article Peer Reviewed: Y **Publication Status:** 4-Under Review
Journal: IEEE Trans. On Pattern Analysis and Machine Intelligence
Publication Identifier Type: DOI Publication Identifier:
Volume: Issue: First Page #:
Date Submitted: 8/29/19 12:00AM Date Published:
Publication Location:
Article Title: Joint Concept Matching based Learning for Zero-Shot Recognition
Authors: 11) Wen Tang, Ashkan Panahi, Hamid Krim
Keywords: Common Distinct Latent Space, Class-specific Information, Reconstruction of Features, Inductive Zero-shot Learning
Abstract: Zero-shot learning (ZSL) which aims to recognize unseen object classes by only training on seen object classes, has increasingly been of great interest in Machine Learning, and has registered with some successes. Most existing ZSL methods typically learn a projection map between the visual feature space and the semantic space and mainly suffer which is prone to a projection domain shift primarily due to a large domain gap between seen and unseen classes. In this paper, we propose a novel inductive ZSL model based on projecting both visual and semantic features into a common distinct latent space with class-specific knowledge, and on reconstructing both visual and semantic features by such a distinct common space to narrow the domain shift gap. We show that all these constraints on the latent space, class-specific knowledge, reconstruction of features and their combinations enhance the robustness against the projection domain shift problem, and improve the generalization ability
Distribution Statement: 1-Approved for public release; distribution is unlimited.
Acknowledged Federal Support: **Y**

CONFERENCE PAPERS:

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: Asian Conference on Computer Vision (ACCV)
Date Received: 12-Apr-2019 Conference Date: 04-Dec-2018 Date Published: 04-Dec-2018
Conference Location: Perth, Australia
Paper Title: Neural Abstract Style Transfer for Chinese Traditional Painting
Authors: Bo Li, Caiming Xiong, Tianfu Wu, Yu Zhou, Lun Zhang, and Rufeng Chu
Acknowledged Federal Support: **Y**

RPPR Final Report
as of 29-Aug-2019

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: IEEE Conference on Computer Vision and Pattern Recognition
Date Received: 29-Aug-2019 Conference Date: 18-Jun-2019 Date Published: 18-Jun-2019
Conference Location: Long Beach, CA
Paper Title: AOGNets: Compositional Grammatical Architectures for Deep Learning
Authors: Xilai Li, Xi Song, Tianfu Wu
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: IEEE Conference on Computer Vision and Patter Recognition
Date Received: 29-Aug-2019 Conference Date: 18-Jun-2019 Date Published: 18-Jun-2019
Conference Location: Long Beach, CA
Paper Title: Learning Attraction Field Representation for Robust Line Segment Detection
Authors: Nan Xue, Song Bai, Fudong Wang, Gui-Song Xia, Tianfu Wu, Liangpei Zhang
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: International Conference on Machine Learning
Date Received: 29-Aug-2019 Conference Date: 11-Jun-2019 Date Published: 11-Jun-2019
Conference Location: Long Beach, CA
Paper Title: Learn to Grow: A Continual Structure Learning Framework for Overcoming Catastrophic Forgetting
Authors: 3) Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, Caiming Xiong
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: International Conference on Computer Vision
Date Received: 29-Aug-2019 Conference Date: 28-Oct-2019 Date Published:
Conference Location: Soeul, South Koera
Paper Title: Image Synthesis from Reconfigurable Layout and Style
Authors: Wei Sun and Tianfu Wu
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: International Conference on Computer Vision
Date Received: 29-Aug-2019 Conference Date: 28-Oct-2019 Date Published:
Conference Location: Soeul, South Korea
Paper Title: Towards Interpretable Object Detection by Unfolding Latent Structures
Authors: Tianfu Wu and Xi Song
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 5-Submitted
Conference Name: Thirty-Fourth AAAI Conference on Artificial Intelligence
Date Received: 29-Aug-2019 Conference Date: 07-Mar-2020 Date Published:
Conference Location: New York, USA
Paper Title: Attentive Normalization
Authors: Xilai Li, Wei Sun, Tianfu Wu
Acknowledged Federal Support: **Y**

RPPR Final Report
as of 29-Aug-2019

Publication Type: Conference Paper or Presentation **Publication Status:** 0-Other
Conference Name: Thirty-Fourth AAAI Conference on Artificial Intelligence
Date Received: 29-Aug-2019 Conference Date: 08-Feb-2020 Date Published:
Conference Location: New York, USA
Paper Title: Learning Attentive Atrous Convolution in Generative Image Modeling
Authors: Wei Sun, Tianfu Wu
Acknowledged Federal Support: **Y**

WEBSITES:

URL: https://github.com/iVMCL/AOGNets
Date Received: 12-Apr-2019
Title: AOGNets source code
Description: Github code repository
URL: https://github.com/iVMCL/afm_cvpr2019
Date Received: 12-Apr-2019
Title: Robust LSD via AFM source code
Description: Github code respository

ARO DURIP Project
Award No. W911NF1810209

Building a GPU Computational Infrastructure Platform for Heterogeneous Big Data Analysis and Understanding

Final report

PI: Dr. Tianfu Wu, NC State University
Contact: twu19@ncsu.edu, Tel: 919-515-4361

Summary

With the support of this grant, our team has been developing a series of work on heterogeneous big data analyses and understanding in three domain areas:

- Domain I: Discriminative learning (image classification, line segment detection, object detection, tracking and segmentation and parsing); and
- Domain II: Generative learning (unconditional and conditional image synthesis).
- Domain III: Reinforcement learning

This report summarizes the developments in 6 tasks.

- Task 1: Developing deep grammar networks for deep learning.
- Task 2: Developing attentive normalization methods for deep learning.
- Task 3: Developing interpretable learning-to-learn methods for handling catastrophic forgetting
- Task 4: Developing attraction field representations for robust line segment detection.
- Task 5: Developing attentive pooling and reconfigurable normalization methods for image synthesis, image completion and image style transfer.
- Task 6: Developing novel experience replay methods for deep reinforcement learning.

We report a total of 11 papers in the pipeline of publication (8 published and 3 under review). These papers cover a broad range of topics in vision, learning, robotics and AI.

Contents

1. Task 1: Developing deep grammar networks for deep learning

Objectives: Neural architectures are the foundation for improving performance of deep neural networks (DNNs). The objective of this task is to develop deep compositional grammatical architectures which harness the best of two worlds: grammar models and DNNs. The proposed architectures integrate compositionality and reconfigurability of the former and the capability of learning rich features of the latter in a principled way.

Accomplishments: During this report period, we have made significant technical progress: We utilize AND-OR Grammars (AOG) as network generators and call the resulting networks, AOGNets[1]. Our proposed AOGNet is *the first work in the literature* which deeply integrate grammar models and DNNs for better feature exploration and exploitation in a compositional, reconfigurable and adaptive way. In our current experiments, we showed that AOGNets outperform all state-of-the-art neural architectures including Google's InceptionNets, Microsfot's ResNets, and Facebook's ResNeXts etc. in visual recognition tasks. *A provisional patent application was filed on AOGNets in November 2018*.

1.1. Method Overview

Technically speaking, neural architectures have been explored in very restricted space in the literature. Learning the optimal network topology automatically in a domain-agnostic way remains an open problem since the seminal work of K. Fukushima's Neocognitron. Recently, Professor Geoffrey Hinton (who is well-known for being called the Godfather of AI) also criticized the insufficiency of current mostly feed-forward network architectures. As Figure 1.a illustrates, network architecture design and search can be posed as a combinatorial search problem in a product space of two sub-spaces:

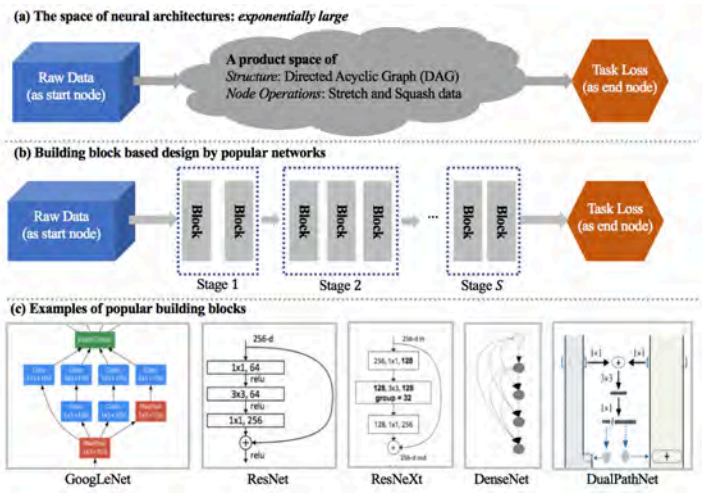


Figure 1. Illustration of the space of neural architectures in (a), the stage-wise building block based schema that popular networks have explored the space in (b), and examples of popular building blocks in convolutional neural networks in (c).

- The structure space which consists of all directed acyclic graphs (DAGs) with the start node representing input raw data and the end node representing task loss functions. DAGs are entailed for feasible computation in implementation.

- The node operation space which consists of all possible transformation functions for implementing nodes in a DAG, such as Convolution+BatchNorm+ReLU in computer vision.

The structure space is almost unbounded, and the node operation space for a given structure is also combinatorial. Neural architecture design and search is a challenging problem due to the exponentially large space and the highly non- convex non-linear objective function to be optimized in the search. As illustrated in Figure 1.b, to mitigate the difficulty, neural architecture design and search have been simplified to design or search a building block structure. Then, a DNN consists of a predefined number of stages each of which has a small number of building blocks. This stage-wise building-block based design is also supported by the theoretical study under some assumptions. Figure 1.c shows examples of some popular building blocks with different structures.

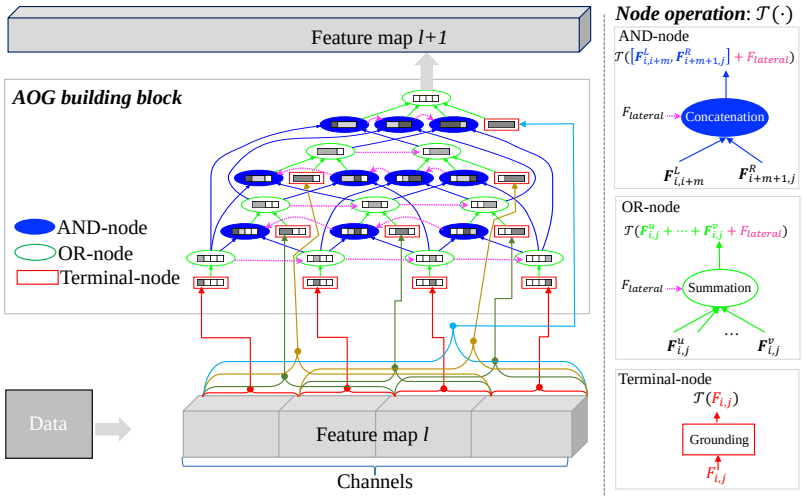


Figure 2. Illustration of our AOG building block for grammar- guided network generator. The resulting networks, AOGNets obtain 80.18% top-1 accuracy with 40.3M parameters in ImageNet, significantly outperforming ResNet-152 (77.0%, 60.2M), ResNeXt-101 (79.6%, 83.9M), DenseNet-Cosine-264 (79.6%, ~73M) and DualPathNet-98 (79.85%, 61.6M). See [1] for details.

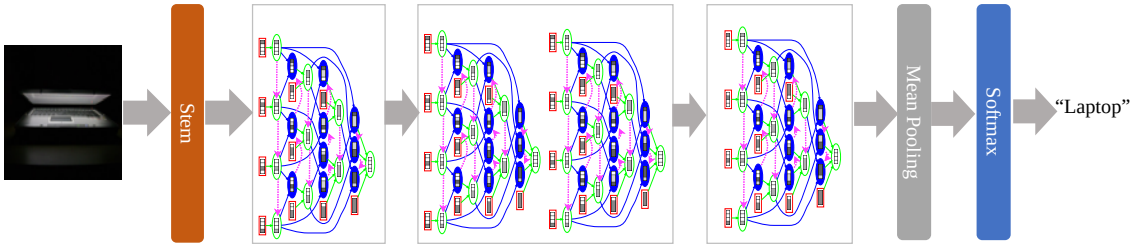


Figure 3. Illustration of an AOGNet consisting of 3 stages with 1 AOG building block in stage 1 and 3 and 2 blocks in stage 2. The number of stages and building blocks per stage are hyper-parameters tuned for different applications.

We proposed **grammar-guided network generators** which can generate "high-quality" DNNs by exploiting *compositionality, reconfigurability and lateral connectivity* which are well-known

principles in cognitive science, neuroscience and pattern theory. They are fundamental for the remarkable capabilities possessed by humans, of learning rich knowledge and adapting to different environments, especially in vision and language. They have not been, however, fully and explicitly integrated in existing DNNs. We presented **compositional grammatic architectures** that realize compositionality, reconfigurability and lateral connectivity for building block design in a principled way. We utilize AND-OR Grammars (AOG) and propose AOG building blocks that unify the best practices developed in existing popular building blocks. Our method deeply integrates hierarchical and compositional grammars and DNNs for harnessing the best of both worlds in deep representation learning. Figure 2 illustrates the proposed AND-OR Grammar (AOG) building block, and Figure 3 shows an example of AOGNets.

1.2. Result Summary

In experiments, AOGNet is tested in two golden testbeds: the ImageNet-1K classification benchmark and the MS-COCO object detection and segmentation benchmark. In ImageNet-1K, AOGNet obtains better performance than ResNet and most of its variants, ResNeXt and its attention based variants such as SENet, DenseNet and DualPathNet. AOGNet also obtains the best model interpretability score using network dissection. AOGNet further shows better potential in adversarial defense. In MS-COCO, AOGNet obtains better performance than the ResNet and ResNeXt backbones in Mask R-CNN.

Method	#Params	FLOPS	top-1	top-5
ResNet-101 [21]	44.5M	8G	23.6	7.1
ResNet-152 [21]	60.2M	11G	23.0	6.7
ResNeXt-50 [63]	25.03M	4.2G	22.2	5.6
ResNeXt-101 (32×4*d*) [63]	44M	8.0G	21.2	5.6
ResNeXt-101 (64×4*d*) [63]	83.9M	16.0G	20.4	5.3
ResNeXt-101 + BAM [46]	44.6M	8.05G	20.67	-
ResNeXt-101 + CBAM [61]	49.2M	8.0G	20.60	-
ResNeXt-50+SE [24]	27.7M	4.3G	21.1	5.49
ResNeXt-101+SE [24]	48.9M	8.46G	20.58	5.01
DensetNet-161 [26]	27.9M	7.7G	22.2	-
DensetNet-169 [26]	$\sim 13.5M$	$\sim 4G$	23.8	6.85
DensetNet-264 [26]	$\sim 33.4M$	-	22.2	6.1
DensetNet-cosine-264 [47]	$\sim 73M$	$\sim 26G$	20.4	-
DPN-68 [6]	12.8M	2.5G	23.57	6.93
DPN-92 [6]	38.0M	6.5G	20.73	5.37
DPN-98 [6]	61.6M	11.7G	20.15	5.15
AOGNet-12M	11.9M	2.36G	22.28	6.14
AOGNet-40M	40.3M	8.86G	19.82	4.88
AOGNet-60M	60.7M	14.36G	**19.34**	**4.78**

Table 1. The top-1 and top-5 error rates (%) on the ImageNet-1K validation set using single model and single-crop testing. Our AOGNets obtain the best accuracy. Please refer to [1] for details of the references in the table.

Table 1 shows the results in ImageNet-1K. Our AOGNets are the best among the models with comparable model sizes in comparison in terms of top-1 and top-5 accuracy. Our small AOGNet-12M outperforms ResNets (44.5M and 60.2M) by 1.32% and 0.72% respectively. *We note that our AOGNets use the same bottleneck operation function as ResNets, so the improvement must be contributed by the AOG building block structure.* Our AOGNet-40M obtains better performance than all other methods in comparison, including ResNeXt-101+SE (48.9M) which represents the most powerful and widely used combination in practice. AOGNet-40M also obtains better performance than the runner-up, DPN-98 (61.6M), which indicates that the hierarchical and

compositional integration of information flow in our AOG building block is more effective than the cascade-based integration in the DPN. Our AOGNet-60M achieves the best results.

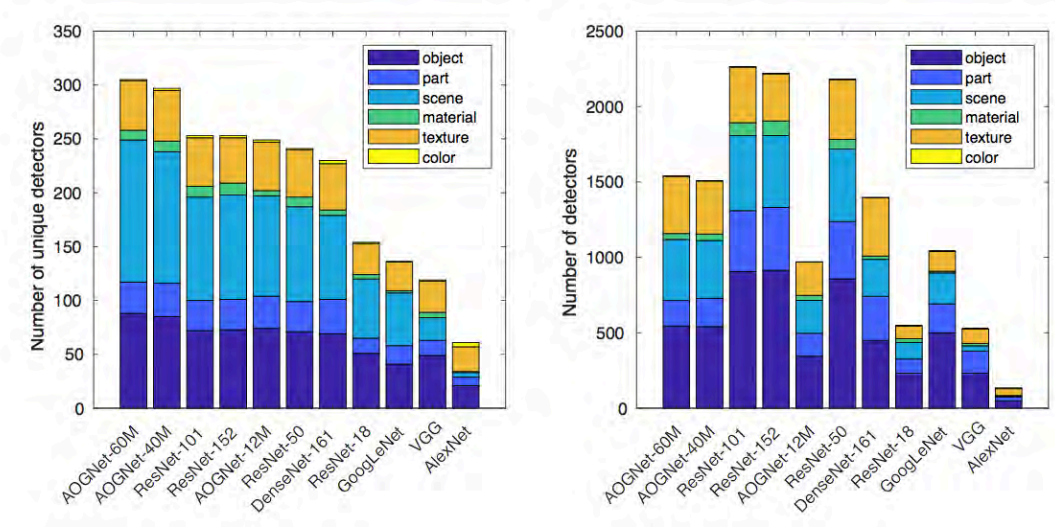


Figure 4. Comparisons of model interpretability using the widely used network dissection method on ImageNet pretrained networks. Our AOGNets obtain the highest interpretability score in terms of the protocol based on the number of unique detectors (left), although they use smaller number of detectors (right, i.e., less complicated models in terms of model parameters).

Model Interpretability has been recognized as a critical concern in developing deep learning based AI systems. We use the network dissection metric which compares the number of unique "detectors" (i.e., filter kernels) in the last convolution layer. Our AOGNet obtains the best score in comparison (Figure 4), which indicates the AOG building block has great potential to induce model interpretabilty by design, while achieving the best accuracy performance.

Method	#Params	$\epsilon = 0.1$	$\epsilon = 0.3$	clean
ResNet-101	44.5M	12.3	0.40	77.37
ResNet-152	60.2M	16.3	0.85	78.31
DenseNet-161	28.7M	13.0	2.1	77.65
AOGNet-12M	12.0M	18.1	1.4	77.72
AOGNet-40M	40.3M	28.3	2.2	80.18
AOGNet-60M	60.1M	30.2	2.6	80.66

Table 2. Top-1 accuracy comparisons under white-box adversarial attack using 1-step FGSM with the Foolbox toolkit. Our AOGNets show great potential in adversarial defense.

Adversarial robustness is another crucial issue faced by many DNNs. We conduct a simple experiment to com- pare the out-of-the-box adversarial robustness of different DNNs. Table 2 shows the results. Under the vanilla settings, our AOGNets show better potential in adversarial defense, especially when the perturbation energy is controlled relatively low (i.e. $\varepsilon = 0.1$). We investigate this with different attacks and adversarial training in our on-going work.

Method	#Params	FLOPS	top-1	top-5
MobileNetV1 [23]	4.2M	575M	29.4	10.5
SqueezeNext [14]	4.4M	-	30.92	10.6
ShuffleNet (1.5) [69]	3.4M	292M	28.5	-
ShuffleNet (x2) [69]	5.4M	524M	26.3	-
CondenseNet (G=C=4) [25]	4.8M	529M	26.2	8.3
MobileNetV2 [51]	3.4M	300M	28.0	9.0
MobileNetV2 (1.4) [51]	6.9M	585M	25.3	7.5
NASNet-C (N=3) [72]	4.9M	558M	27.5	9.0
AOGNet-4M	4.2M	557M	25.6	7.91

Table 3. The top-1 and top-5 error rates (%) on the ImageNet-1Kvalidation set under mobile settings.

Mobile settings. We train an AOGNet-4M under the typical mobile settings (mode size < 5M parameters, and FLOPs < 600M). Table 3 shows the comparison results. We obtain performance

on par to or better than the popular networks specifically designed for mobile platforms such as the MobileNets and ShuffleNets. Our AOGNet also outperforms the auto-searched network, NASNet (which used around 800 GPUs in search). We note that we use the same AOGNet structure, thus showing promising device-agnostic capability of our AOGNets. This is potentially important and useful for deploying DNNs to different platforms in practice since no extra efforts of hand-crafting or searching neural architectures are entailed.

Method	#Params	t (s/img)	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{m}	AP^{m}_{50}	AP^{m}_{75}
ResNet-50-C4	35.9M	0.130	35.6	56.1	38.3	31.5	52.7	33.4
ResNet-101-C4	54.9M	0.180	39.2	59.3	42.2	33.8	55.6	36.0
AOGNet-12M-C4	14.6M	0.092	36.8	56.3	39.8	32.0	52.9	33.7
AOGNet-40M-C4	48.1M	0.184	**41.4**	**61.4**	**45.2**	**35.5**	**57.8**	**37.7**
ResNet-50-FPN	44.3M	0.125	37.8	59.2	41.1	34.2	56.0	36.3
ResNet-101-FPN	63.3M	0.145	40.1	61.7	44.0	36.1	58.1	38.3
ResNeXt-101-FPN	107.4M	0.202	42.2	63.9	46.1	37.8	60.5	40.2
AOGNet-12M-FPN	31.2M	0.122	38.0	59.8	41.3	34.6	56.6	36.4
AOGNet-40M-FPN	59.4M	0.147	41.8	63.9	45.7	37.6	60.3	40.1
AOGNet-60M-FPN	78.9M	0.171	**42.5**	**64.4**	**46.7**	**37.9**	**60.9**	**40.3**

Table 4. Mask-RCNN results on coco val2017 using the 1x training schedule. Results of ResNets and ResNeXts are reported by the state-of-the-art maskrcnn-benchmark.

In MS-COCO, Table 4 shows the comparison results. Our AOGNets obtain better results than the ResNet and ResNeXt backbones with smaller model sizes and similar or slightly better inference time. The results show the effectiveness of our AOGNets learning better features in object detection and segmentation tasks.

2. Task 2: Developing attentive normalization methods for deep learning

Objectives: Batch Normalization (BN) is a vital pillar in the development of deep learning with many recent variations such as Group Normalization (GN) and Switchable Normalization (SN). Channel-wise feature attention methods such as the squeeze-and-excitation (SE) unit have also shown impressive performance improvement. Feature normalization and feature attention have been studied separately, however. The objective of this task is to *develop a novel and lightweight integration of feature normalization and feature channel- wise attention.*

Accomplishments: We developed **Attentive Normalization (AN)** [2] which is a lightweight integration of feature normalization and attention. AN is complementary and applicable to existing variants of BN. In experiments, we test AN in the ImageNet-1K classification dataset and the MS-COCO object detection and instance segmentation dataset with significantly better performance obtained than the vanilla BN. Our AN also outperforms two state-of-the-art variants of BN, GN and SN.

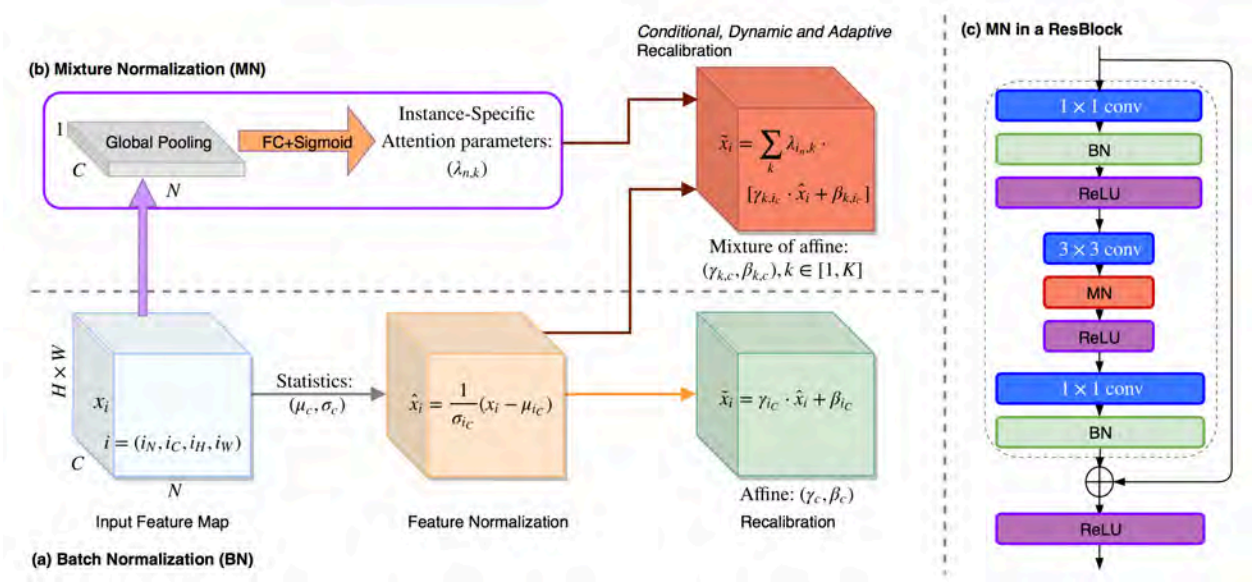


Figure 5. Illustration of the proposed Mixture Normalization (MN) in (b) using the vanilla Batch Normalization (BN) as backbone (a). MN shares the feature normalization component with BN, and differs in how the affine transformation is done. (c) shows our lightweight deployment of MN in the bottleneck of a ResNet building block (ResBlock) which follows the 3 × 3 convolution unit to potentially jointly integrate spatial attention in learning the instance-specific attention parameters. MN can also use other variants of BN as backbones. The input feature map is represented using the convention (N, C, H, W) for the batch axis, channel axis, spatial height and width axes respectively. xi represents a feature response in the input feature map with position index $i = (i_N, i_C, i_H, i_W)$. $\hat{x}_i$ represents the normalized response using the pooled channel-wise mean and variance. $\tilde{x}_i$ is the response after affine transformation with learned scale and offset parameters. See [2] for details.

2.1. Method Overview

BN and its variants take into account different ways of computing the mean and variance within a min-batch for feature normalization, followed by a learnable channel-wise affine transformation. SE explicitly learns how to adaptively recalibrate channel-wise feature responses. AN absorbs SE into the affine transformation of BN. As illustrated in Figure 5, unlike BN which only learns one affine transformation for each channel, AN learns a small number K of affine transformation components per channel (e.g., $K = 5$ is a hyperparameter). The scale and offset parameters for the final instance-specific channel-wise affine transformation is computed as weighted sum of the mixture of affine transformation components. The instance-specific weights are learned from the input feature map. For example, we utilize the squeeze module in the SE unit to learn the weights. It consists of a global average pooling layer, a fully-connected layer and the sigmoid activation function. It first utilizes the mean of each filter to represent its "importance" and then learns the

interdependencies between the filters from the eye of their means to capture channel-wise attention. We can also learn weights for the scale and offset parameters separately.When deploying our AN, e.g., into the Bottleneck building block of ResNets, to control the extra parameters introduced by our AN, we use a lightweight deployment for it. It follows the 3×3 convolution unit since it has the least number of channels. Potentially, this will jointly integrate local spatial attention in learning the instance-specific attention parameters. We keep the other two feature normalization (BN) units. By mixing AN and BN, we also obtain a new type of Bottleneck operations.

2.2. Result Summary

We tested AN using ResNet50 as backbone, which is also used by other types of variants of BN (so we can compare results). We test AN in the ImageNet-1K classification dataset and the MS-COCO object detection and instance segmentation dataset with significantly better performance obtained than the vanilla BN. Our AN also outperforms two state-of-the-art variants of BN, GN and SN.

Method	#Params	FLOPS	top-1	top-5
ResNet-50-BN	25.56M	4.09	23.01	6.68
ResNet-50-GN	25.56M	4.09	23.52	6.85
ResNet-50-SN (8,32)	25.56M	-	22.43	6.35
ResNet-50-SE	28.09 M	-	22.37	6.36
ResNet-50-AN	25.69M	4.09	**22.00**	**6.06**
ResNet-101-BN	44.57M	8.12	20.71	5.43
ResNet-101-AN	44.71M	8.12	**20.06**	**5.12**
DenseNet-161-BN	28.73M	8.50	22.35	6.20
DenseNet-161-AN	30.28M	8.50	**20.13**	**4.94**
MobileNet-v2-BN	3.50M	0.335	28.69	9.33
MobileNet-v2-AN	3.56M	0.335	**26.67**	**8.56**

Table 5. The top-1 and top-5 error rates (%) on the ImageNet-1K validation set using single model and single-crop testing.

Table 5 shows the comparison results. Our AN obtains the best top-1 and top-5 accuracy results with negligible extra parameters at almost no extra computational cost. Our AN improves BN by almost 0.6% on top-1, and outperforms SN by 0.2%, which shows the effectiveness of our lightweight integration of feature normalization and attention.

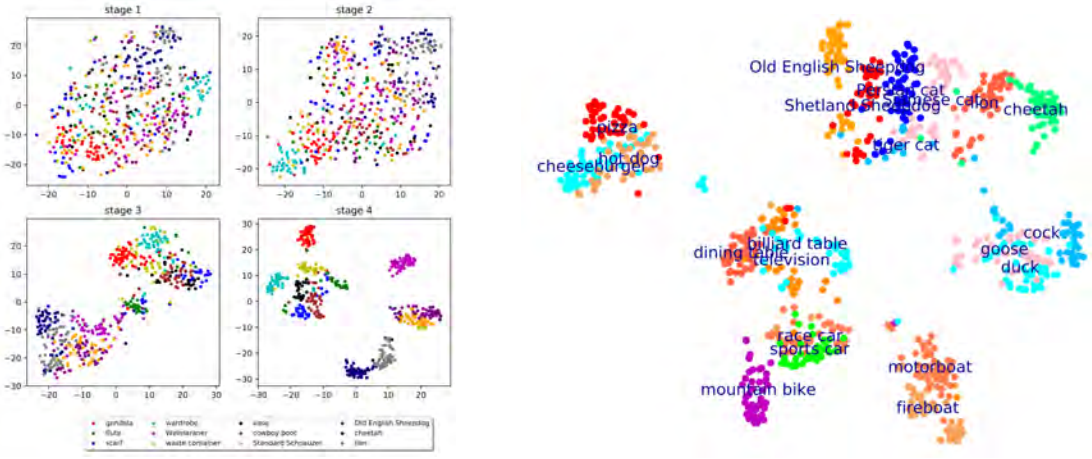


Figure 6. *Left*: t-SNE plot comparison of learned weights in the mixture for the four stages of ResNet-50 with randomly selected 12 ImageNet classes. In each stage, the learned y vectors of all

the units are concatenated for visualization. *Right*: More semantically or visually similar classes tend to have closer embeddings.

Figure 6 (Left) shows the t-SNE plots for the four stages in ResNet50. In each stage, we concatenate the learned weights in all Bottleneck units as the clustering features for images. We randomly select 12 categories (gondola, vase, lion, etc.) in the validation dataset. We observe that the learned weights become stronger and stronger for clustering the images for deeper stages. In the final stage (stage4), we clearly see the clusters are formed. This effect shows that the learned weights are indeed informative and meaningful for the classification task. The sub-network used to learn the weights (Eqn. 6 in the paper [2]) shares the similar settings with the classification head classifier (consisting of a global average pooling, a FC layer and softmax). So, we can treat the learned weights as some latent classification codes. The final affine transformation is then guided by this latent classification codes for recalibrating the normalized feature responses, which may be the underlying driving force introduced by our AN. We further investigate if the learned weights can pre- serve the semantic similarities, that is visually or semantically similar categories should be also closer to each other in the t-SNE plots. Figure 6 (Right) verifies the hypothesis. For example, we can see “pizza”, “hot dog” and “cheeseburger” are very close, as well as “race car” and “sports car”. Our AN shows strong capability in *preserving semantic similarities which is one of the most important criteria for representation learning*.

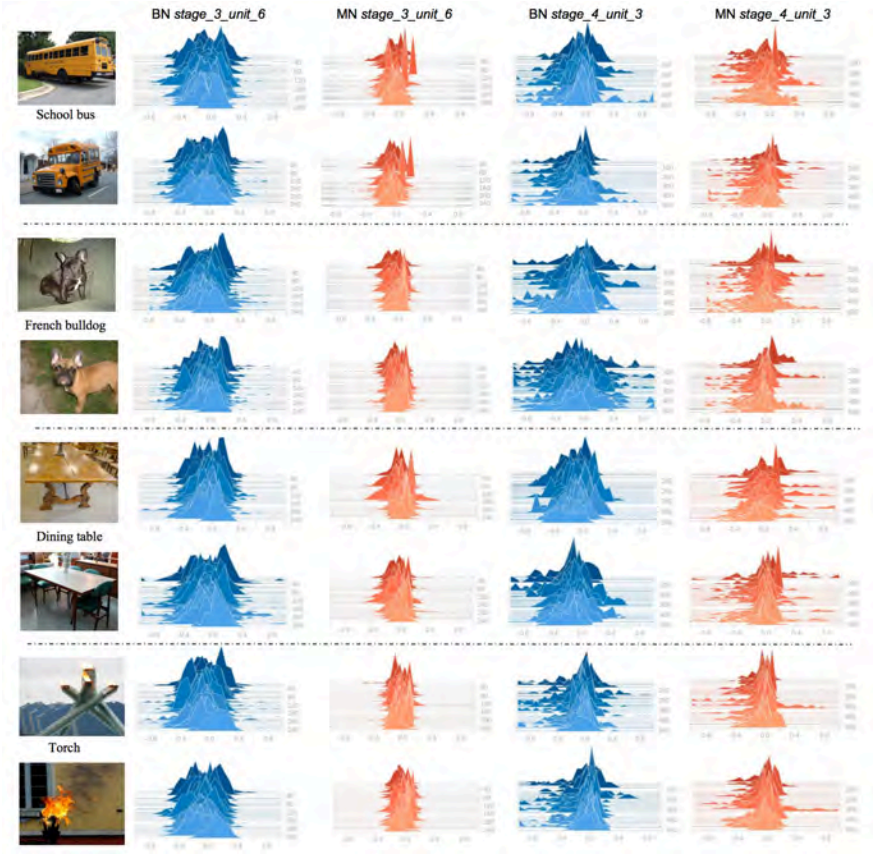


Figure 7. Illustration of the effects of MN and BN on filter responses. We show the filter response histograms (marginal distributions) for different images in different categories. Here we show results of a 4-stage ResNet50. stage i unit j means the histograms are plot for the output feature

map of the j-th ResBlock in the i-th stage. From the histograms, we observe that for images from the same class (e.g., school bus), the histograms of our MN show higher similarities with smaller variance.

Figure 7 shows empirical comparisons between our AN and the vanilla BN. This empirically shows that a channel-wise attention guided mixture of affine transformation helps recalibrate the normalized responses in a more meaningful way.

	Backbone	Head	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{m}	AP^{m}_{50}	AP^{m}_{75}
ResNet-50	BN*	-	38.6	59.8	42.1	34.5	56.4	36.3
	GN	GN	40.3	61.0	44.0	35.7	57.9	37.7
	SN	SN†	41.0	62.3	45.1	36.5	58.9	38.7
	AN (w/ BN)	-	40.5	62.4	44.1	36.5	59.2	38.5
	AN (w/ BN)	AN (w/ GN)	**41.5**	62.2	45.3	**37.1**	59.6	39.5
ResNet-101	BN*	-	40.3	61.5	44.1	36.5	58.1	39.1
	GN	GN	41.8	62.5	45.4	36.8	59.2	39.0
	AN (w/ BN)	-	43.5	64.5	47.4	38.3	60.9	41.0
	AN (w/ BN)	AN (w/ GN)	**43.7**	64.7	48.1	**39.3**	61.6	42.7

Table 6. Detection and segmentation results in COCO, using Mask R- CNN with ResNet-50-FPN. All models use 2x lr scheduling (180k iterations). BN∗ means BN is frozen in fine-tuning for object detection. SN† means that only Layer Norm and Instance Norm are used in the SN. In our implementation of the MN head, MN (w/ GN) means that we use the mixture version of GN in the head.

In MS-COCO, when fine-tuning the ImageNet pretrained ResNet50+AN and ResNet101+AN on COCO for object detection and segmentation, we freeze all the gamma and beta parameters and the tracked running mean and variance, but allow the FC layers to continue learn except for the FC layer in the first stage. As Table 6 shows, with only AN in the backbone, our AN obtains comparable performance to the model with GN in both backbone and the head classifiers although GN stands right in its sweet pot. This shows that our AN does not suffer too much from the small batch settings in fine-tuning. We conjecture that the FC layers can compensate the small batch issue by learning instance-specific feature channel-wise attention. Compared with the vanilla BN, we significantly improve the performance, which shows the effectiveness of integrating feature normalization and attention in transferring models between different tasks.

3. Task 3: Developing interpretable learning-to-learn methods for handling catastrophic forgetting

Objectives: Learning different tasks continuously is a common and practical scenario that happens all through the course of human learning. The learning of new skills from new tasks usually does not have negative impact on the previously learned tasks. Furthermore, with learning multiple tasks that are highly related, it often helps to advance all related skills. However, this is commonly not the case in current deep learning models. When presented a sequence of learning tasks, the model experiences so called “catastrophic forgetting” problem where the model “forgets” the previous learned task while learning the new task. Addressing catastrophic forgetting is one of the key challenges in continual learning where machine learning systems are trained with *sequential or streaming tasks*. Despite recent remarkable progress in state-of-the-art deep learning, deep neural networks (DNNs) are still plagued with the catastrophic forgetting problem. The objective

of this task is to *develop a conceptually simple yet general and effective framework for handling catastrophic forgetting in continual learning with DNNs.* Then, we can study continual learning in long-term and large-scale object tracking-by-detection-and-parsing.

Accomplishments: We propose a *learn-to-grow* framework[3] based on differentiable neural architecture search (NAS). The proposed method consists of two components: a neural structure optimization component and a parameter learning and/or fine-tuning component. By separating the explicit neural structure learning and the parameter estimation, not only is the proposed method capable of evolving neural structures in an intuitively meaningful way, but also shows strong capabilities of alleviating catastrophic forgetting in experiments. Furthermore, the proposed method outperforms all other baselines on the permuted MNIST dataset, the split CIFAR100 dataset and the Visual Domain Decathlon dataset in continual learning setting.

3.1. Method Overview

In our learn-to-grow framework, the first neural structure optimization component learns the best neural structure for the current task on top of the current DNN trained with previous tasks. It learns whether to reuse or adapt building blocks in the current DNN, or to create new ones if needed under the differentiable neural architecture search framework. The parameter estimation/fine-tuning component estimates parameters for newly introduced structures, and fine-tunes the old ones if preferred. Figure 8 illustrates the proposed framework.

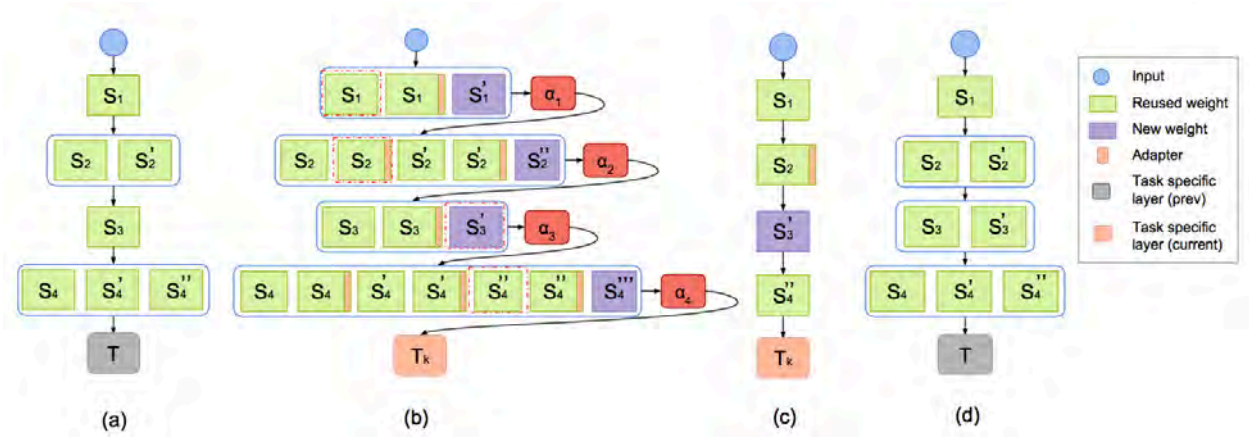


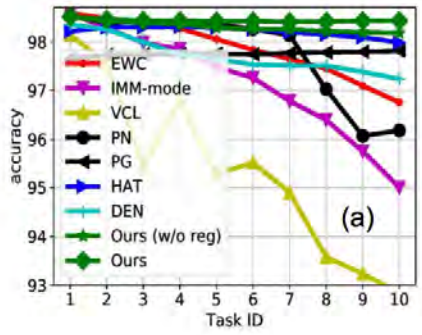
Figure 8. Illustration of the proposed *learn-to-grow* framework. a) Current state of super model. In this example, the 1st and 3rd layers have single copy of weight, while the 2nd and 4th has two and three respectively. b) During search, each copy of weight for each layer will have a "reuse" and an "adaptation" options plus a "new" option, thus totally $2|S^l| + 1$ choices. α is the weight parameters for the architecture. c) Parameter optimization with selected architecture on the current task k. d) Update super model to add the newly created $S_3{}'$.

In *neural structure optimization*, we utilize differentiable NAS. We assume that one already has in mind a global structure that may work for all tasks (i.e., super-net), and we are only selecting connectivity pattern between layers and their corresponding operator. It is straight forward to adapt this to more complicated cases, we make the simplification because: 1) it is common in a multi-task continual learning scenario that one has some rough clue regarding the overall model structure; 2) this simplifies the optimization problem significantly. Let's define a certain network with L shareable layers and one task-specific layer (i.e. last layer) for each task. A super network S is

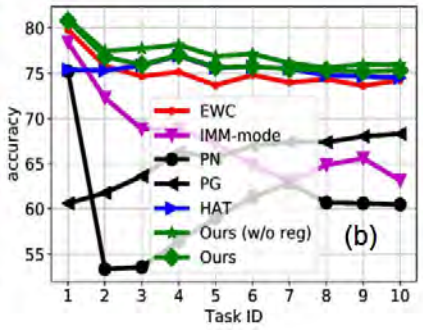
maintained so that all the new task-specific layers and new shareable layers will be stored into S. The goal of search is trying to find out the optimal choice for each of the L layers, given the current task data D_i and all the shareable layer's weights stored in S. The candidate choices for each layer could be "reuse", "adaptation" and "new". The *reuse* choice will make new task use the same parameter as the previous task. The *adaptation* option adds a small parameter overhead that trains an additive function to the original layer output. The *new* operator will spawn new parameters of exactly the size of the current layer parameters.

In *parameter estimation and/or fine-tuning*, after we get the optimal choices for each layer from the search procedure, we retrain the optimal architecture on the current task. There are two strategies to deal with "reuse", we can either fix it unchanged during retraining just as in search, or we can tune it with some regularization – simple l_2 regularization or more sophisticated regularizations like elastic weight consolidation. We tested both in experiments.

3.2. Result Summary



(a)



(b)

Method	Acc (%)	#params
SGD	72.51	3.35M
EWC	96.75	3.35M
IMM	95.00	3.35M
VCL	95.32	3.35M
HAT	97.98	3.46M
PN	96.18	3.96M
PG	97.81	3.05M
DEN	97.71	3.53M
Ours (fix)	98.46	2.56M
Ours (tune)	98.29	2.56M
Ours (tune+L2Reg)	98.48	2.56M

(c)

Method	Acc (%)	#params
SGD	21.02	6.55M
EWC	74.23	6.55M
IMM	63.13	6.55M
HAT	74.52	6.82M
PN	60.48	7.04M
PG	68.32	6.80M
Ours (fix)	75.31	6.86M
Ours (tune)	75.05	6.86M
Ours (tune+L2Reg)	76.21	6.86M

(d)

Figure 9. Comparative performance on a) permuted MNIST and b) split CIFAR-100 dataset. Methods include Kirkpatrick et al. (2017, EWC), Lee et al. (2017b, IMM), Fernando et al. (2017, PathNet (PN)), Rusu et al. (2016, Progressive Net (PG)), Serra` et al. (2018, HAT), Lee et al. (2017b, DEN), Nguyen et al. (2018, VCL), ours (w/o reg) denotes the case where finetuning for current tasks is done without using any regularization to prevent forgetting, and ours represents the case where the l2 regularization is used. c) Results of different continual learning approaches on 10 permutated MNIST datasets. The averaged accuracy after all 10 tasks are learned and total number of parameters are compared. d) Results of different continual learning approaches on split CIFAR100 dataset. The averaged accuracy after all 10 tasks are learned and total number of parameters are compared.

Comparison with Other Methods. As shown in Figure 9, we compare the performance of various methods on the permuted MNIST dataset with ten different permutations, and the CIFAR-100 dataset where we randomly partition the classes of CIFAR-100 into 10 disjoint sets, and regard learning each of the 10-class classification as one task. It is clear that our method (either tuned with or without regularization) performs competitive or better than other methods on these tasks. This result suggests that although theoretically, structure can be learned along with parameter, in practice, the current optimization have a hard time achieving this. This in turn indicates the importance of explicit taking structure learning into account when learning tasks continuously.

We also conduct comprehensive ablation studies on different aspects of our learn-to-grow framework in the paper [3] which show the significance of the proposed framework.

4. Task 4: Developing bottom-up/top-down integrated framework for online object tracking

Objective: Robust online object tracking entails integrating short-term trackers and long-term trackers in an elegant framework to handle structural and appearance variations of unknown objects in an online manner. The integration and synergy between short-term and long-term trackers have yet studied well in the literature, especially in pre-training free settings. To address this issue, *this objective of this task* is to develop a bottom-up and top-down integrated framework. The bottom-up component realizes a data-driven approach for particle generation. It exploits a short-term tracker to generate bounding box proposals in a new frame based on current tracking results. In the top-down component, a graph regularized sparse coding scheme is proposed as the long- term tracker.

Accomplishment: We are interested in model-free settings in online object tracking. The proposed bottom-up/Top-down integration method is tested on the widely used OTB-100 benchmark and the VOT2016 benchmark with better performance obtained than baselines including deep learning based trackers. In addition, the outputs from the top-down sparse coding are potentially useful for downstream tasks such as action recognition, multiple-object tracking, and object re-identification.

4.1. Method Overview

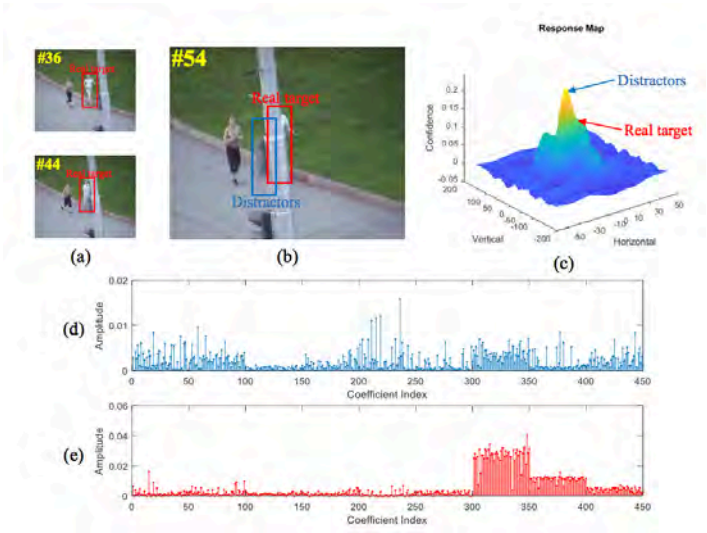


Figure 10. Approximation of the discriminative particles by the top-down component. (a): The real target areas of sequence Jogging-1. (b): Real target and distractors in the 54thframe. (c):

Response map obtained from the bottom-up component. (d): Distractor approximated by the top-down component. (e): Real target approximated by the top-down component.

This paper proposes a bottom-up and top-down integrated framework to make the short-term tracker and long-term tracker work cooperatively. The bottom-up component realizes a data-driven approach which guides the gaze to the visual attention areas according to the object feature analysis. We exploit the short-term tracker as the bottom-up component to generate bounding box proposals that are to be carried forward to the top-down component. The areas corresponding to the response peaks and sidelobes are presented as discriminative particles in the new frame, in which both of the real target and other distractors are included, as shown in Figure 10 (b). The top-down component is driven by the high-level cognitive knowledge and aims to approximate the proposals obtained from the bottom-up component by the long-term memory of the target status. A novel graph regularized sparse coding scheme is presented as the representation model of long-term tracking. We first compute a particle graph whose nodes are the discriminative particles and edges are formed in terms of appearance and spatial-temporal similarities between bottom-up particles. And the constraint mode of the sparse coefficients is induced by the particle graph. Moreover, part-based representations are exploited to model the particles, which aims to deal with target partial variations. The sparse coding results of the distractor and the real target are shown in Figure 10 (d) and (e) respectively. For the distractor with the maximum response score in the bottom-up component, the energy of the coefficients is scattered in different dictionary entries. By comparison, in the representation of the real target, the non-zero elements of the coefficients are mainly distributed on the corresponding sub-dictionary.

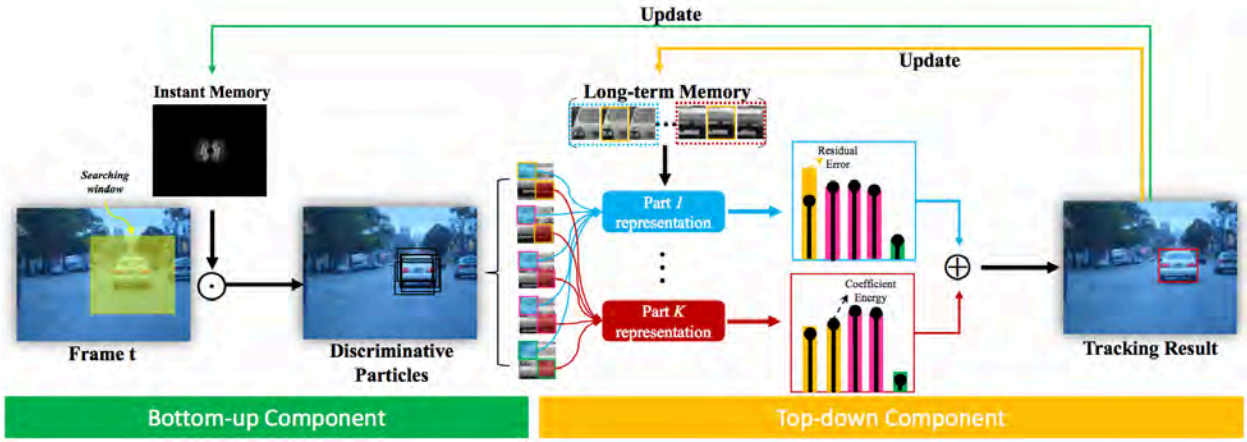


Figure 11. Overview of the proposed integrated tracking framework. Our method contains a bottom-up component, which applies the instant memory (a classifier) to generate bounding box proposals, and a top-down component to model each particle part individually with an over-complete dictionary that contains long- term memory of the tracking objects. The representation model is based on a novel graph regularized sparse coding scheme, in which the underlying relationship of bottom-up particles is utilized as high-level cognitive information in the representation process. And the final tracking result is inferred based on the sparse coding coefficients energies and probabilities of all parts.

The overview of the proposed tracking framework is illustrated in Figure 11. When a new frame arrives, a classifier with short-term memory of the tracking object is used to generate bounding box proposals (i.e., discriminative particles) within the searching window. Next, each particle is

divided into several local image blocks, which are then represented individually by a stored template set of long-term object appearance. The representation results of all parts are combined together to determine the final tracking result. Figure 12 shows some examples of the learned top-down sparse coding dictionary.

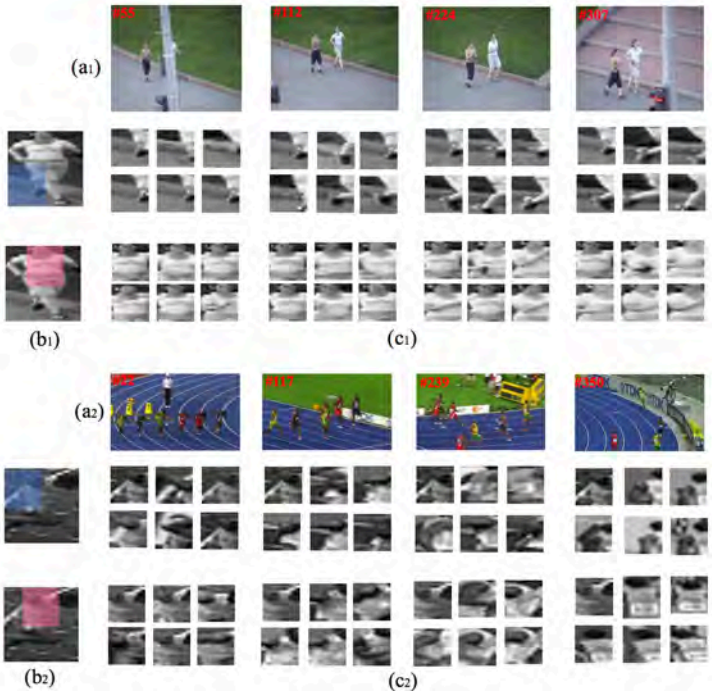


Figure 12. Illustration of dictionary entries of the long-term tracker. Two sequence videos (Jogging-1 and Bolt) are shown as valid test examples. (a): Sequence frames of different tracking period. (b): The initial target to be tracked, the blue and red masks illustrate two selected local parts to be represented. (c): The dictionary entries for target representation.

4.2. Result Summary

In the first frame, 50 positive samples are selected to initialize the dictionary. Each candidate image is resized to 32 × 32 pixels and is divided into 16 × 16 local image blocks with 8-pixel patch step size. Our experiment is implemented in MATLAB on a laptop with an Intel Core i7-6700HQ 2.60GHz CPU and 16G RAM.

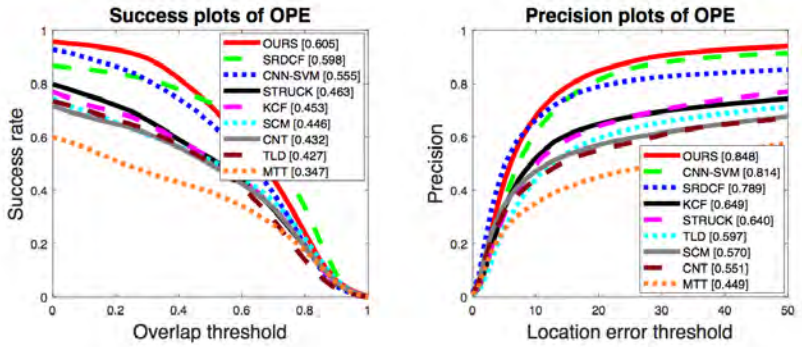


Figure 13. Plots of OPE on the OTB-100 benchmark. The performance score for each tracker is illustrated in the legend. We evaluate the proposed algorithm on OTB-100 with comparisons to 8

state-of-the-art trackers, including 2 correlation filter based trackers: SRDCF and KCF; 2 sparse representation based trackers: SCM and MTT; 2 trackers with CNN features: CNN-SVM and CNT and 2 tracking-by-detection based trackers: STRUCK and TLD. From the results, we can see that the proposed tracker outperforms other state-of-the-art trackers.

In *the OTB-100 benchmark* which consists of 100 test sequences. Figure 13 shows the comparison results. In the tracking process, for each sequence, only the target location of the first frame is manually labeled. Two basic metrics, center error and overlap rate are employed to evaluate the performance of each tracker. Based on these two metrics, the benchmark result is reported as the precision plots and success plots respectively. The precision plots show the position error between the predicting bounding boxes and the ground truth bounding boxes. Its performance score is the distance precision at a threshold of 20 pixels. The success plots take the position and scale variation into account. Its performance score is the area under curve value. Figure 14 shows some qualitative results.

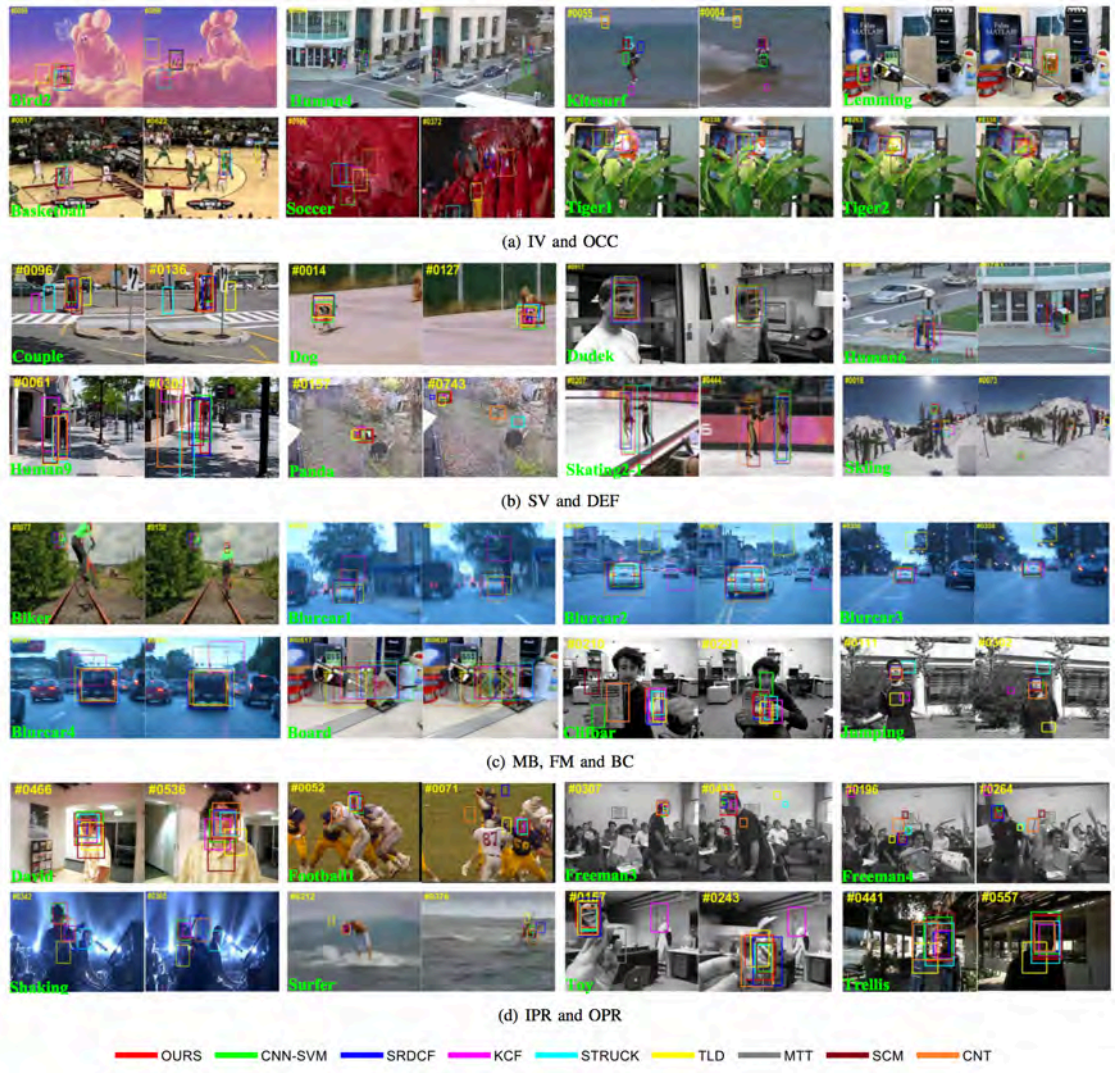


Figure 14. Qualitative comparisons in OTB-100.

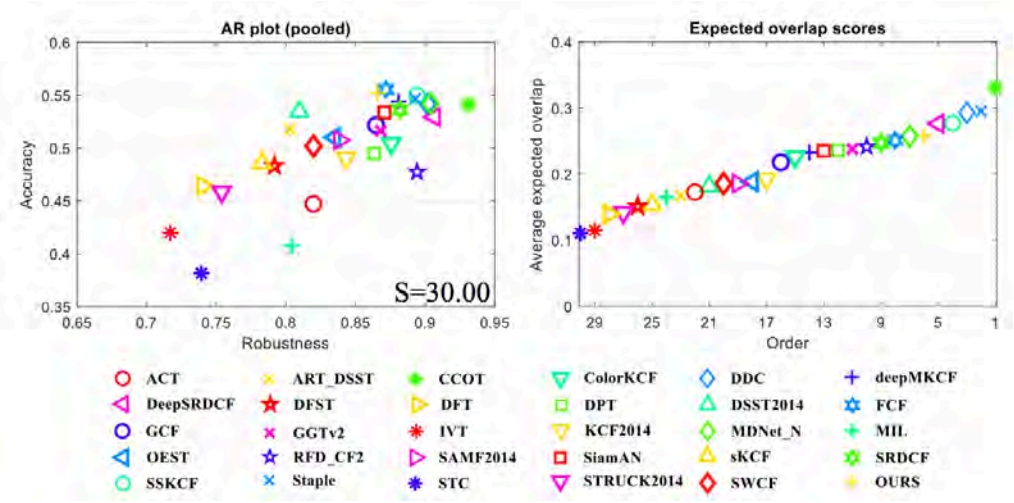


Figure 15. Pooled AR plot and expected average overlap ranks. The sensitivity parameter for calculating robustness is defined as 30.

In *the VOT2016 benchmark* which consists of 60 video sequences. Figure 15 shows the comparison results. Unlike OTB where a tracker is initialized at the beginning of a sequence and left to track until the end, the VOT challenges apply a reset-based methodology, in which trackers are re-initialized five frames after failure. Three measures are used to analysis tracking behavior in the reset-based experiment: 1) accuracy (A), 2) robustness (R) and 3) expected average overlap (EAO). The accuracy is the average overlap value of the predicted and ground truth bounding boxes during success tracking periods. The robustness measures the number of tracking failures. The third measure, EAO, is a combination of the raw values of per-frame accuracies and failures in a principled manner. We compare the proposed tracker with 29 related and state- of-the-art tracking methods in the VOT2016 challenge. Sixteen trackers are based on the correlation filter framework with hand-crafted features. Five trackers apply CNN features into correlation filter, CCOT, DDC, deepMKCF, deepSRDCF, and RFD-CF2. The remaining eight trackers are selected from other tracking frameworks, MDNet-N and SiamAN are based on convolutional neural networks architecture, MIL and STRUCK2014 are in the tracking-by-detection framework, DPT and GGTv2, are two part-based trackers, DFT is based on distributed fields, IVT is based on sub-space learning. The MDNet-N tracking method is an extension of MDNet, which is the winner of VOT2015. The CCOT method is the winner of VOT2016.

5. Task 5: Developing attraction field representations for robust line segment detection

Objective: Line segment detection (LSD) is an important yet challenging low-level task in computer vision. LSD usually consists of two steps: line heat map generation and line segment model fitting. The former can be computed either simply by the gradient magnitude map (mainly used before the recent resurgence of deep learning), or by a learned convolutional neural network (ConvNet) in state-of-the-art methods. The latter needs to address the challenging issue of handling un- known multi-scale discretization nuisance factors (e.g., the classic zig-zag artifacts of line segments in digital images) when aligning pixels or linelets to form line segments in the line heat map. The main drawbacks of existing two-stage methods are in two-fold: lacking elegant solutions to solve the local ambiguity and/or class imbalance in line heat map generation, and requiring extra carefully designed heuristics or supervisedly learned contextual information in inferring line segments in the line heat map. The *objective of this task* is to develop methods which focus on

learning based LSD framework and utilize a single-stage method which rigorously addresses the drawbacks of existing LSD approaches.

Accomplishment: In experiments, the proposed method is tested on the WireFrame dataset and the YorkUrban dataset with state-of-the-art performance obtained. In particular, we improve the performance by large margin 4.5% on the WireFrame dataset against state-of-the-art methods. Our method is also fast with 6.6 ~ 10.4 FPS, outperforming most of line segment detectors.

5.1. Method Overview

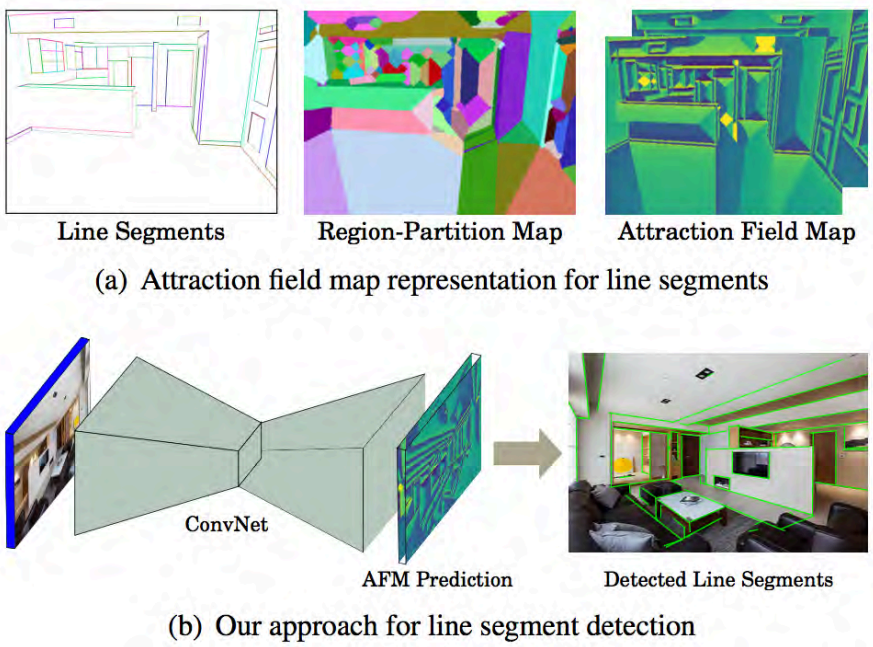


Figure 16. Illustration of the proposed method. (a) The proposed attraction field dual representation for line segment maps. A line segment map can be almost perfectly recovered from its attraction filed map (AFM), by using a simple squeeze algorithm. (b) The proposed formulation of posing the LSD problem as the region coloring problem. The latter is addressed by learning ConvNets.

The proposed method for LSD is motivated by two observations: a) The duality between region representation and boundary contour representation of objects or surfaces, which is a well-known fact in computer vision; b) The recent remarkable progresses for image semantic segmentation by deep ConvNet based methods such as U-Net and DeepLab. *The intuitive idea of this task* is that if we can bridge line segment maps and their dual region representations, we will pose the problem of LSD as the problem of region coloring, and thus open the door to leveraging the best practices developed in state-of-the-art deep ConvNet based image semantic segmentation methods to improve perfor- mance for LSD. By dual region representations, it means they are capable of recovering the input line segment maps in a nearly perfect way via a simple algorithm. We present an efficient and straightforward method for computing the dual region representation. By re-formulating LSD as the equivalent region coloring problem, we address the afore- mentioned challenges of handling local ambiguity and class imbalance in a principled way.

Figure 16 illustrates the proposed method. Given a 2D line segment map, we represent each line segment by its geometry model using the two end-points. In computing the dual region representation, there are three components:

- A region-partition map. It is computed by assigning every pixel to one and only one line segment based on a proposed point to line segmentation distance function. The pixels associated with one line segment form a region. All regions represent a partition of the image lattice (*i.e.*, mutually exclusive and the union occupies the entire image lattice).
- An attraction field map. Each pixel in a partition region has one and only one corresponding projection point on the geometry line segment (but the reverse is often a one-to-many mapping). In the attraction field map, every pixel in a partition region is then represented by its attraction/projection vector between the pixel and its projection point on the geometry line segment.
- A light-weight squeeze module. It follows the attraction field to squash partition regions in an attraction field map to line segments that almost perfectly recovers the input ones, thus bridging the duality between region-partition based attraction field maps and line segment maps

The proposed method can also be viewed as an intuitive expansion-and-contraction operation between 1D line segments and 2D regions in a simple projection vector field: The region-partition map generation jointly expands all line segments into partition regions, and the squeeze module degenerates regions into line segments.

5.2. Result Summary

We test our method on two widely used benchmarks, the WireFrame dataset and YorkUrban dataset. All methods are evaluated quantitatively by the precision and recall protocol. The precision rate indicates the proportion of positive detection among all of the detected line segments whereas recall reflects the fraction of detected line segments among all in the scene. The detected and ground-truth line segments are digitized to image domain and we define the "positive detection" pixel-wised. The line segment pixels within 0.01 of the image diagonal is regarded as positive. After getting the precision (P) and recall (R), we compare the performance of algorithms with F-measure $F = 2 \times \frac{P \cdot R}{P+R}$. Figure 17 and Table 7 summarize the comparison results. Figure 18 show qualitative comparisons.

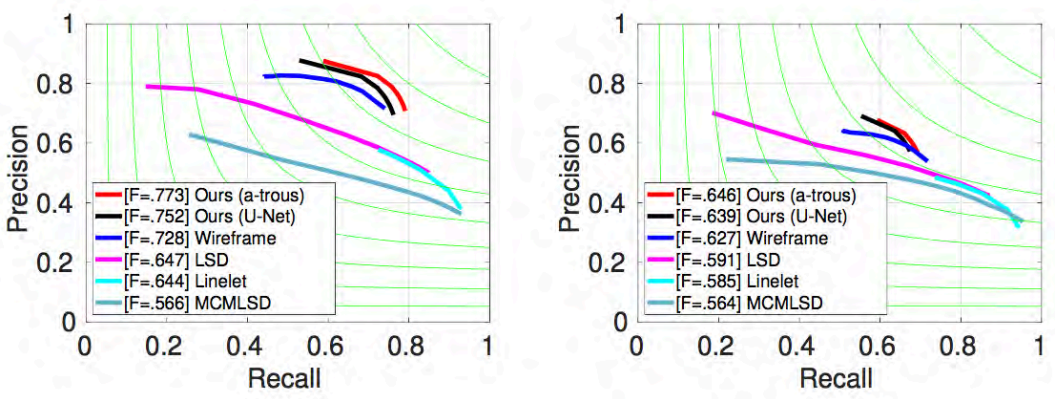


Figure 17. The PR curves of different line segment detection methods on the WireFrame (left) and YorkUrban (right) datasets.

Methods	Wireframe dataset	York Urban dataset	FPS
LSD [23]	0.647	0.591	19.6
MCMLSD [1]	0.566	0.564	0.2
Linelet [5]	0.644	0.585	0.14
Wireframe parser [12]	0.728	0.627	2.24
Ours (U-Net)	**0.752**	**0.639**	**10.3**
Ours (*a-trous*)	**0.773**	**0.646**	**6.6**

Table 7. F-measure evaluation with state-of-the-art approaches on the WireFrame dataset and York Urban dataset. The last column reports the average speed of different methods in frames per second (FPS) on the WireFrame dataset.

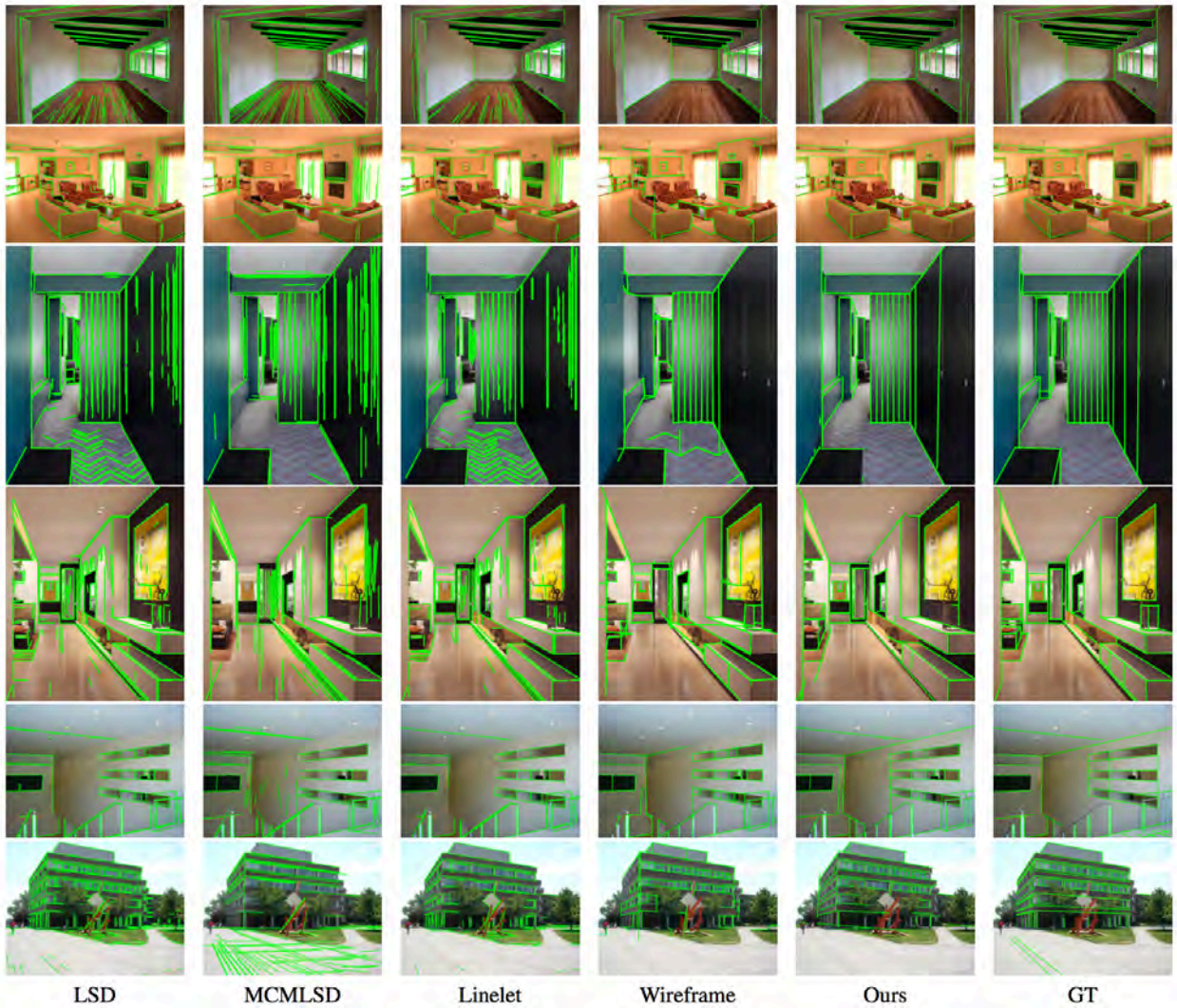


Figure 18. Some Results of line segment detection on Wireframe and YorkUrban datasets with different approaches LSD, MCMLSD, Linelet, Deep Wireframe Parser and ours with the a-trous Residual U-Net are shown from left to right. The ground truths are listed in last column as reference.

6. Task 6: Developing attentive pooling and reconfigurable normalization methods for image synthesis and image style transfer

Objective: Generative learning is one of the most important and challenging tasks in computer vision and machine learning. Image synthesis is an important generative image modeling task in computer vision which aims at synthesizing realis- tic and novel images by learning high-dimensional data distributions. Image-to-Image translation is usually built on image synthesis with two different settings, paired and un- paired translations. Generative adversarial networks (GANs) have recently become the most popular framework for generative learning. The *objective of this task* is two-fold: i) Develop attentive pooling modules for generators in GANs which can simulate intuitive "drawing" process (e.g., coarse-to-fine); ii) Develop conditional feature normalization schema for generators in GANs which can leverage the provided information in conditional GANs (recall that we developed conditional feature normalization methods for discriminative tasks in Task 2).

Accomplishments: We explored two directions in generative learning with GANs. We develop a method of learning Attentive Atrous Convolution (*AAC*) which is a novel architectural unit and can be easily integrated into generators of GANs. The proposed AAC integrates the widely used Atrous Spatial Pyramid Pooling (ASPP) in discriminative learning tasks, a proposed cascade attention mechanism and residual connections. In experiments, the proposed AAC is integrated in GANs for image synthesis and tested on the Celeba-HQ-128 dataset. It is also integrated in CycleGANs for unpaired image-to-image translation task and tested on the Cityscape dataset, the Facade and Aerial Maps dataset. The proposed AAC significantly improves performance of the baseline GANs and CycleGANs. It also obtains comparable or better performance than some state-of-the-art variants of GANs and CycleGANs. Coarse-to-fine and fine-to-coarse AAC are studied and intriguing attention maps are observed in both tasks. On the other hand, we develop methods which are capable of synthesizing realistic and sharp images from reconfigurable spatial layout (i.e., bounding boxes + class labels in an image lattice) and style (i.e., structural and appearance variations encoded by latent vectors), especially at high resolution. We present a layout- and style-based architecture for generative adversarial networks (termed *LostGANs*) that can be trained end-to-end to generate images from reconfigurable layout and style. In experiments, the proposed method is tested on the COCO-Stuff dataset and the Visual Genome dataset with state-of-the-art performance obtained.

6.1. Method Overview

In GANs, less attention has been paid to neural architecture design, especially for the generators. The intuitive idea of this study is that exploring new architectures could improve performance of generative learning in a way complementary to existing efforts. The goal of this study is then to design a generic and light-weight architectural unit that can be easily integrated into generators of GANs and CycleGANs. *For the proposed AAC module* (Figure 19), it leverages advantages of the three components to facilitate effective end- to-end generative learning: (i) the capability of fusing multi-scale information by ASPP; (ii) the capability of capturing relative importance between both spatial locations (especially multi-scale con- text) or feature channels by attention; (iii) the capability of preserving information and enhancing optimization feasibility by residual connections. The proposed AAC building block harnesses advantages of the three components in generative learning tasks.

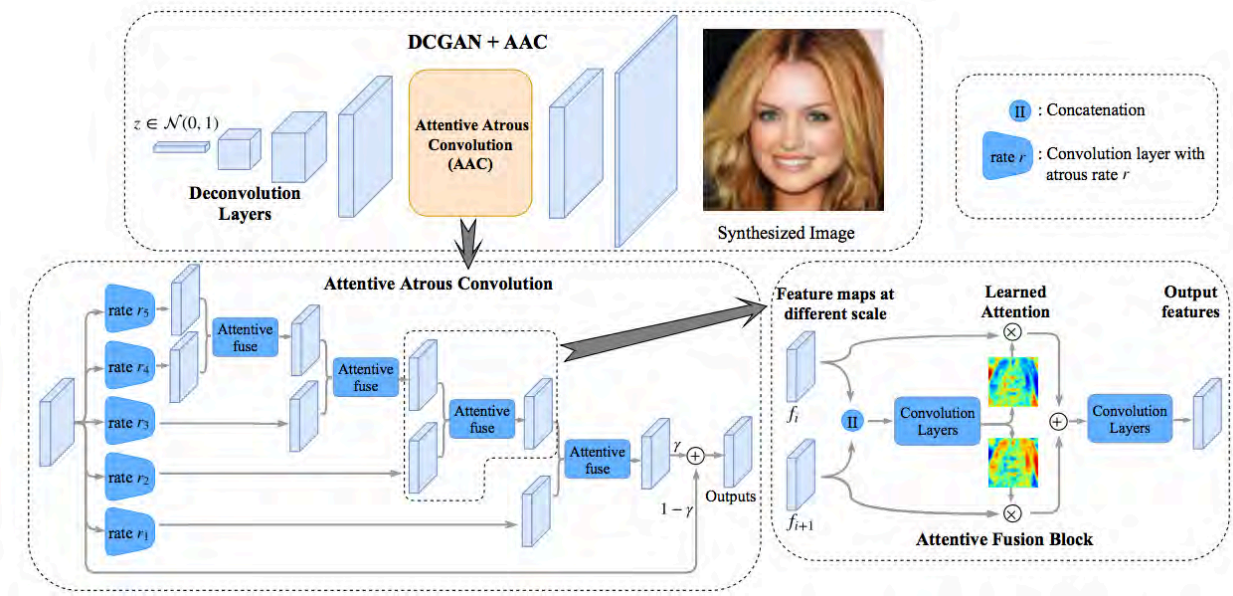


Figure 19. The proposed Attentive Atrous Convolution (AAC) building block. Top: Illustration of the integration of the proposed AAC in the generator of an unconditional GAN for image synthesis. Bottom-left: The detailed neural architecture of the proposed AAC. Bottom-right: The operation of the attentive fuse component between two consecutive levels in the pyramid.

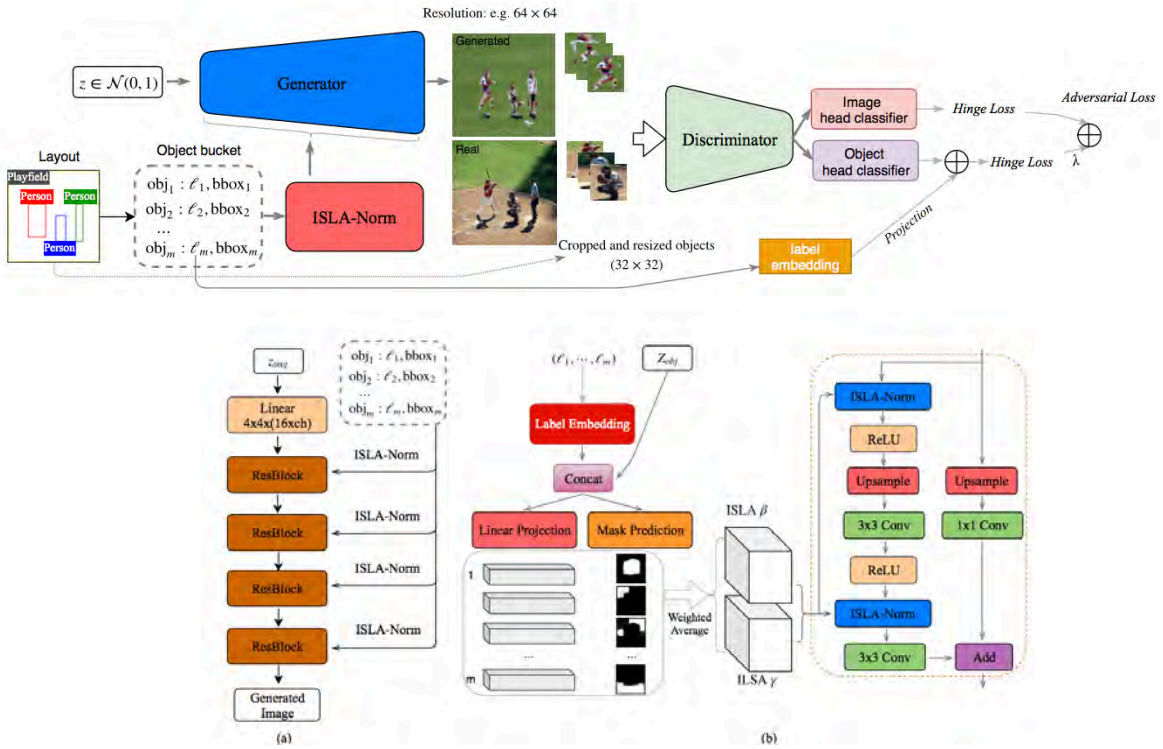


Figure 20. *Top*: Illustration of the proposed layout- and style-based GANs (LostGANs) for image synthesis from reconfigurable layout and style. Both the generator and discriminator use ResNets as backbones. *Bottom*: Illustration of the generator (a) and the ISLA-Norm (b) in our LostGAN.

For the proposed LostGANs (Figure 20), to enable image synthesis from reconfigurable layouts and sytles, our method consists of the following three aspects: *First*, since layout-to-image entails highly expressive neural architectures handling multi-object generation and their diverse occurrence and configurations in layouts. We utilize ResNet for both the generator and discriminator in the proposed LostGAN. We are studying AOGNet in LostGANs in our on-going work. *Second*, to account for the gap between bounding boxes in a layout and underlying object shapes, we introduce an encoder for layout to predict masks for each bounding box. As we will show in experiments, our LostGAN can predict reasonably good masks in a weakly-supervised manner. The masks help place objects in the generated images with fine-grained geometric properties. So, we address layout-to-image by computing layout-to-mask-to-image. *Third*, to achieve instance-sensitive and layout-aware style control, we extend the Adaptive Instance Normalization (AdaIN) used in the StyleGAN to *object instance-specific and layout-aware feature normalization (ISLA-Norm)* for the generator for fine-grained spatially distributed multi-object style control. ISLA-Norm computes the mean and variance as done in BatchNorm, but computes object instance-specific and layout-aware affine transformations (i.e., gamma and beta parameters) separately for each sample in a min-batch as done in AdaIN. We utilize the projection-based approach. From the layout encoder, we compute object instance-specific style latent codes (gamma and beta parameters) via simple linear projection. Then, we place the projection-based latent codes in the corresponding predicted masks, and thus induce layout-aware affine transformations for recalibrating normalized feature responses.

6.2. Result Summary

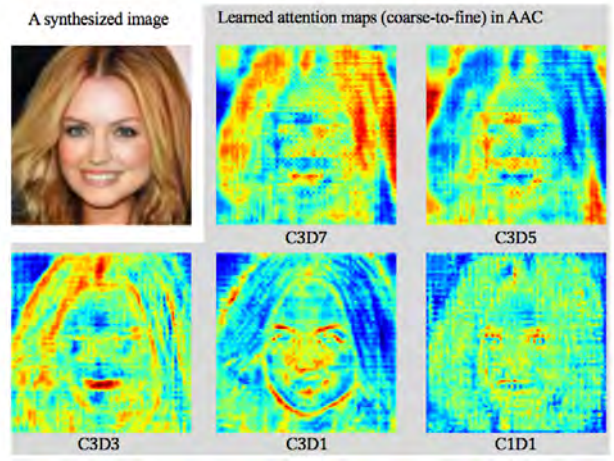


Figure 21. Illustration of attention maps learned by the proposed Attentive Atrous Convolution (AAC) when integrated in GANs for image synthesis on the CelebA-HQ-128 dataset. The attention maps are visualized using heat maps where $CkDr$ represents convolution with kernel $k \times k$ and Atrous rate r.

Evaluation of the proposed ACC in conditional and unconditional image synthesis. In experiments, the proposed method is tested in both image synthesis tasks using the state-of-the-art SNDC-GANs, and unpaired image-to-image translation tasks using the popular CycleGANs. We obtain significantly better performance than the vanilla SNDCGANs and CycleGANs and the baseline ASPP module. Although our models are much smaller, we obtain comparable performance to a recent variant of CycleGANs, the SCAN which use stacked CycleGANs in the progressive training protocol. Figure 21 shows a face synthesis example and the learned coarse-to-fine attention heat

maps. Figure 22 show more examples in comparison with vanilla SNDCGANs and SNDCGANs+ASSP. Figure 23 and 24 show examples in image-to-image translation tasks.

Figure 22. Examples of generated images by the proposed model trained on CelebA-HQ and the FID distance (smaller is better).

Figure 23. Comparisons on Cityscapes dataset of 256x256 resolution.

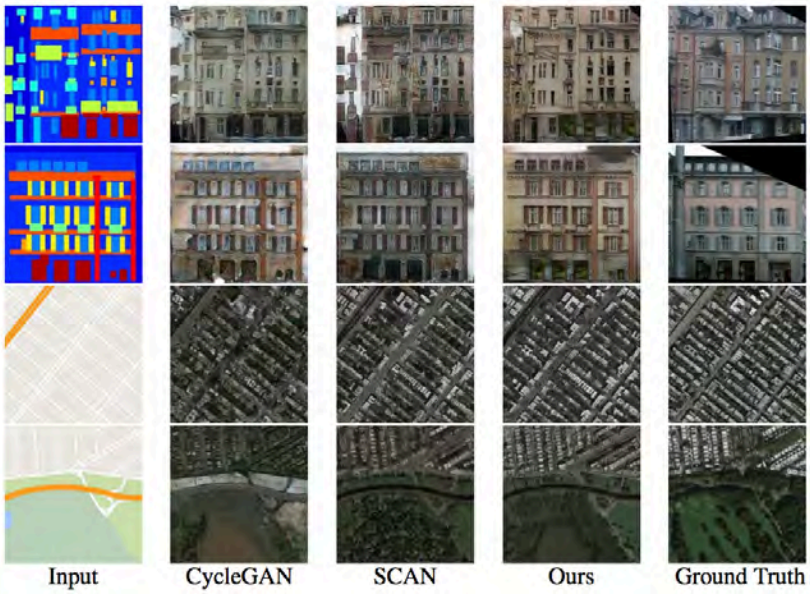


Figure 24. Results on Labels ⇒ Facades and Labels ⇒ Maps.

Evaluation of the proposed LostGANs. In experiments, the proposed method is tested on the COCO-Stuff dataset and the Visual Genome dataset with state-of-the-art performance obtained. Figure 25, 26 and 27 show examples of our method handling reconfigurable layouts and styles in image synthesis. Table 8 shows the quantitative comparison. Figure 28 shows qualitative comparison.

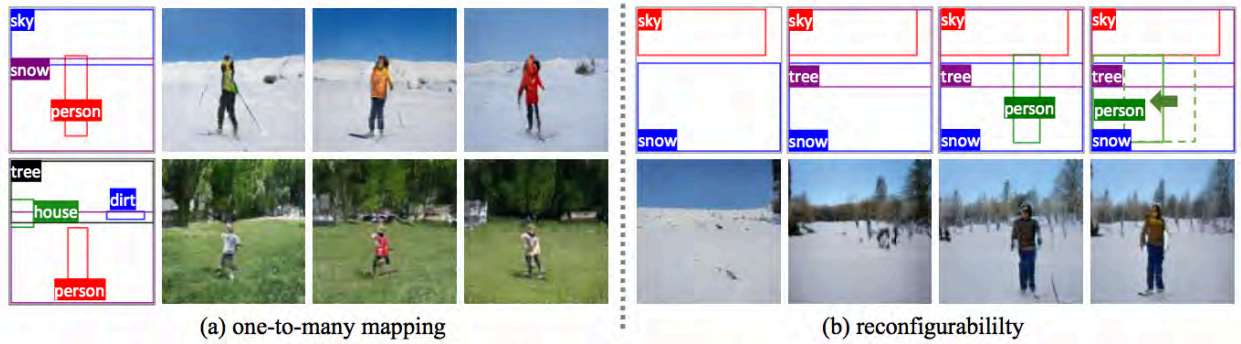


Figure 25. Left: Our model preserves one-to-many mapping for image synthesis from layout and style. Three samples are generated for each input layout by sampling the style latent codes. Right: Our model is also adaptive w.r.t. reconfigurations of layouts (by adding new object bounding boxes or changing the location of a bounding box). The results are generated at resolution 128 × 128.

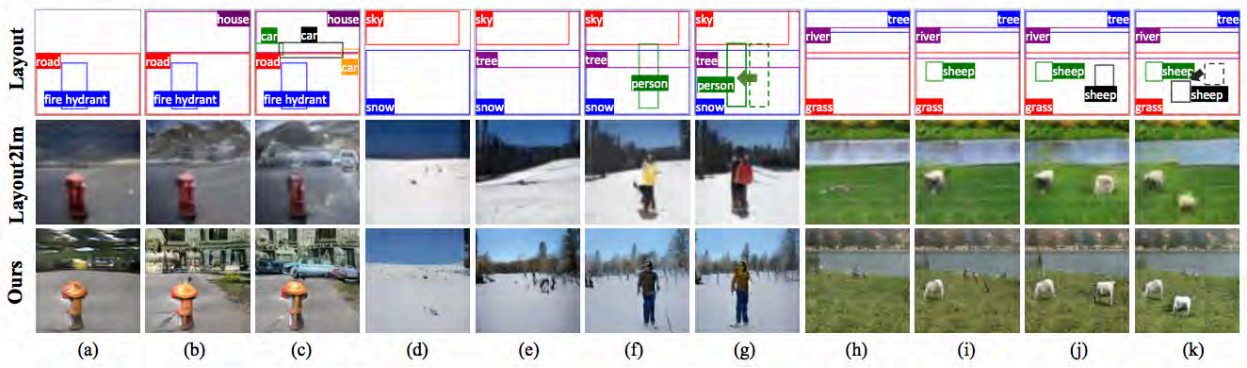


Figure 26. Generation results by adding new objects or change spatial position of objects.

Figure 27. Multiple samples generated from same layout. Synthesized images have various visual appearance while preserving objects at desired location.

Methods	Inception Score		Diversity Score	
	COCO	VG	COCO	VG
Real Images (64×64)	16.3 ± 0.4	13.9 ± 0.5	-	-
Real Images (128×128)	22.3 ± 0.5	20.5 ± 1.5	-	-
pix2pix	3.5 ± 0.1	2.7 ± 0.02	0	0
sg2im(GT Layout)	7.3 ± 0.1	6.3 ± 0.2	0.02 ± 0.01	0.15 ± 0.12
Layout2Im	9.1 ± 0.1	8.1 ± 0.1	0.15 ± 0.06	0.17 ± 0.09
Ours 64×64	$\mathbf{9.3 \pm 0.2}$	$\mathbf{8.7 \pm 0.3}$	$\mathbf{0.25 \pm 0.08}$	$\mathbf{0.26 \pm 0.09}$
Ours 128×128	$\mathbf{13.1 \pm 0.2}$	$\mathbf{11.2 \pm 0.5}$	$\mathbf{0.38 \pm 0.08}$	$\mathbf{0.38 \pm 0.10}$

Table 8. Quantitative comparisons using Inception Score (higher is better) and Diversity Score (higher is better) evaluation on COCO-Stuff and VG dataset.

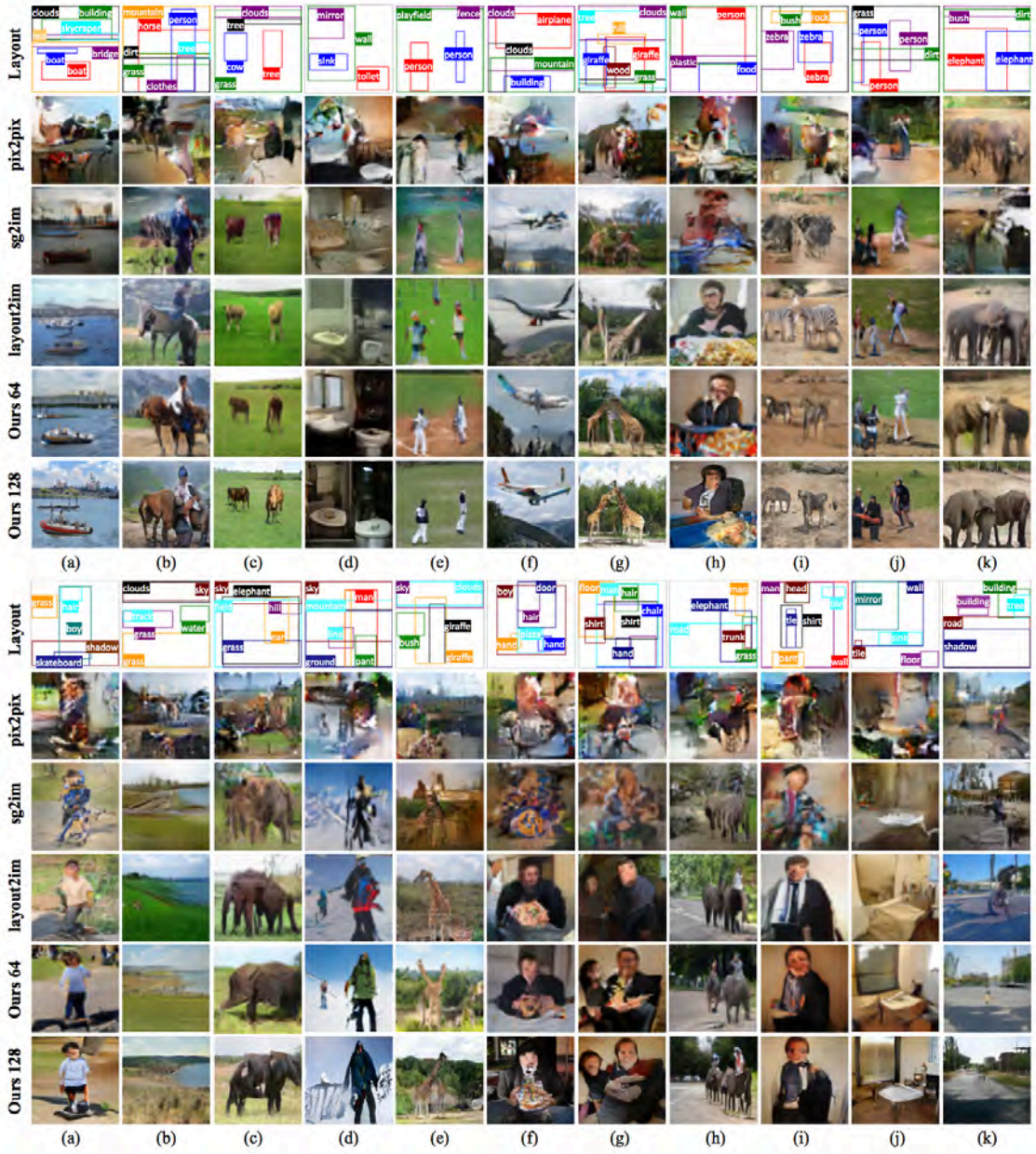


Figure 28. Generated samples from given layouts on COCO-Stuff (top) and Visual Genome (bottom).

7. Task 7: Developing novel experience replay methods for deep reinforcement learning

Objective: Hindsight Experience Replay (HER) was recently proposed to tackle the sample-inefficiency of standard experience replay in reinforcement learning from sparse rewards due to disproportionately few successful episodes observed by an agent. In its operation, HER introduces an optimistic bias in the hindsight experiences and therefore achieves only a suboptimal improvement in sample-efficiency. Motivated by counterfactual reasoning, the objective of this task is to develop a weighted reward mechanism which extends HER by assigning a proportionally larger influence to rewards collected during hindsight replay and a smaller influence to rewards collected during the real episode, and the proposed method is titled *Aggressive Rewards to Counter bias in Hindsight Experience Replay (ARCHER)*.

Accomplishment: In experiments, we validate our algorithm on two continuous control environments from DeepMind Control Suite in combination with various reward functions, task complexities and goal-sampling strategies. Our experiments demonstrate that ARCHER consistently attains a higher success rate in less time, thus establishing its benefit in achieving good sample-efficiency. A few interesting directions emerge for further exploration. Some of our experiments reveal that ARCHER enjoys higher sample-efficiency only until a context-dependent number of samples, after which vanilla HER catches up to ARCHER. This result makes intuitive sense as the high performance of ARCHER leads to the fast convergence of real and hindsight experiences, and diminished hindsight bias. Hence, a scheduled annealing of ARCHER remains of interest. Also, we specifically constructed a simple linear relation to derive an informative hindsight reward function, however there may exist a more complex mapping between real and hindsight rewards and hence it may be advantageous to introduce a generative model to learn this latent mapping. Furthermore, measuring the performance of ARCHER on a real-world robot presents an important future direction.

7.1. Method Overview

We first examine the source of bias in HER, and then present our algorithm ARCHER which uses more aggressive rewards for hindsight experiences to combat the bias, and thus achieving greater sample-efficiency.

In the vanilla HER, compare the real experience tuple $(s_t||g, a_t, s_{t+1}||g, r_t)$ to the artificially constructed hindsight experience tuple $(s_t||g^h, a_t, s_{t+1}||\ g\text{^}h\ \ , r_t^h)$. This conversion of the a real experience to its corresponding hindsight experience makes the following unjustified assumption - Given different inputs $s_t||g$ and $s_t||g^h$, the policy network returns the same action, a_t. This assumption overestimates the probability assigned by the policy network to a_t, given the input $s_t||g^h$. If we actually execute the policy network with $s_t||g^h$ as input, it is unlikely to output a_t, making s_{t+1} also unlikely. Hence, we observe a chain of compounding uncertainty along the hinsight episode. Therefore, to more effectively use HER, we require to correct the hindsight bias induced by this overestimated probability. The intuitive check would be to generate hindsight experiences by using models capable of counterfactual reasoning, i.e. by asking the network what if g^h was the actual goal, instead of mere substitution of real experiences. However, this a critical limitation of deductive learning models and remains a challenge for the future.

We propose a simple solution to offset this bias. We make that case that a hindsight experience and a real experience cannot be treated in the same manner as real experiences are authentically generated by interacting with the environment, and hence their probability is unbiased. In contrast, to overcome hindsight bias, we need to match the true probability of the hindsight experiences to

their biased probability. To do so, we nudge the current policy to be more consistent with the hindsight data in the replay buffer. Hence, to meet the overestimated hindsight likelihood of at for $s_t||g^h$, we utilize more aggressive hindsight rewards, so that a large positive reward given to a successful hindsight transition greatly increases the Q-value of the hindsight state-action pair, which indirectly drives an aggressive policy update towards choosing this maximizing action for the given hindsight state. Figure 29 shows the algorithm.

Algorithm 1 ARCHER

Given:
- an off-policy RL algorithm $\mathbb{A}$, ▷ e.g. DDPG
- a strategy $\mathbb{S}$ for sampling goals for replay
- a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \rightarrow \mathbb{R}$
- real reward weight λ_r, hindsight reward weight λ_h

Initialize $\mathbb{A}$
Initialize replay buffer R
for episode = 1, M **do**
  Sample a goal g and initial state s_0
  for $t = 0, T-1$ **do**
    Sample an action a_t using the behavior policy from $\mathbb{A}$:
      $a_t \leftarrow \mu(s_t||g\,;\theta^\mu) + \mathcal{N}$
    Execute the action a_t and observe a new state s_{t+1}
  for $t = 0, T-1$ **do**
    $r_t = \lambda_r \times r(s_t, a_t, g)$
    Store the transition $(s_t||g, a_t, r_t, s_{t+1}||g)$ in R
    Sample a set of additional goals for replay, $\mathcal{G} = \mathbb{S}$(current episode)
    for $g^h \in \mathcal{G}$ **do**
      $r_t^h = \lambda_h \times r(s_t, a_t, g^h)$
      Store the transition $(s_t||g^h, a_t, r_t^h, s_{t+1}||g^h)$ in R ▷ HER with weighted rewards
  for $t = 1, N$ **do**
    Sample a minibatch B from the replay buffer R
    Perform one step of optimization using $\mathbb{A}$ and the minibatch B

Figure 29. The proposed ARCHER algorithm.

7.2. Result Summary

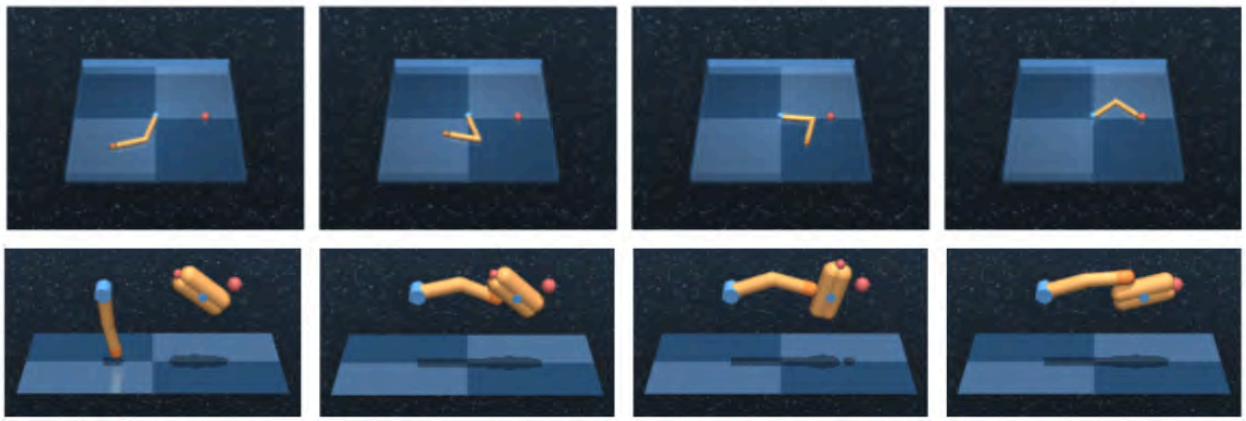


Figure 30. Illustration of the two environments: Reacher (Top) and Finger (Bottom).

We evaluate our method on the DeepMind (DM) Control Suite simulation software (Figure 30). This library consists of a set of continuous control environments in Python, built on top of the MuJoCo physics engine. Each environment in the suite provides a physics task along with a well-defined continuous action space A, continuous state/observation space S, and intrinsic transition dynamics based on the physics engine. For our experiments, we program our own reward functions to conduct ablation studies on ARCHER and verify its robustness, as detailed in the following

sections. Figure 31 shows the superior performance of the proposed ARCHER compared with the vanilla HER.

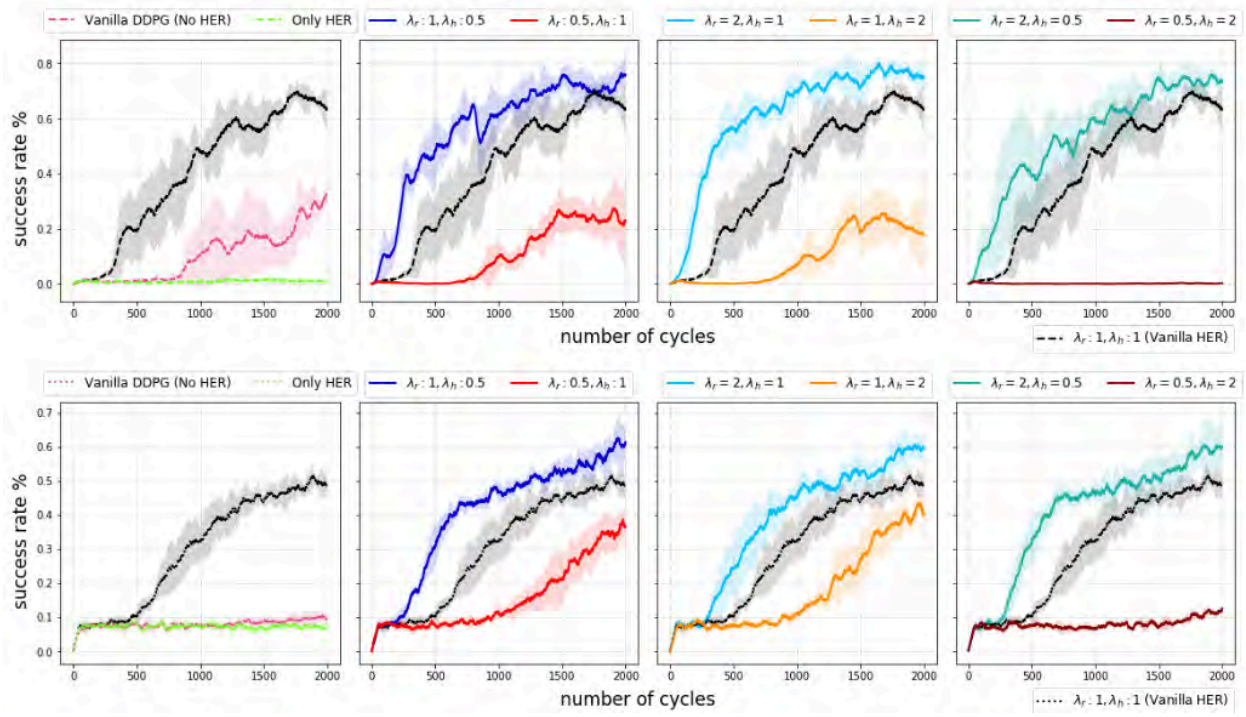


Figure 31. Policy performance in the Reacher (Top) and the Finger (Bottom) environments with sparse binary negative rewards and the final sampling strategy for hindsight goals.

8. Publications

We report 11 papers (8 published and 3 under reviewer).

1) Xilai Li, Xi Song and Tianfu Wu. AOGNets: Compositional Grammatical Architectures for Deep Learning. In IEEE Conference on Computer Vision and Patter Recognition (CVPR), Long Beach, CA, 2019. [Published] (patent pending)
2) Xilai Li, Wei Sun and Tianfu Wu. Attentive Normalization. In the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2019. [Submitted]
3) Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher and Caiming Xiong. Learn to Grow: A Continual Structure Learning Framework for Overcoming Catastrophic Forgetting. In International Conference on Machine Learning (ICML), 2019. [Published]
4) Meihui Li, Lingbing Peng, Tianfu Wu and Zhenming Peng. A Bottom-Up and Top-Down Integrated Framework for Online Object Tracking. IEEE Trans. On Multimedia, 2019. [Submitted]
5) Nan Xue, Song Bai, Fudong Wang, Gui-Song Xia, Tianfu Wu and Liangpei Zhang, Learning Attraction Field Representation for Robust Line Segment Detection. In IEEE Conference on Computer Vision and Patter Recognition (CVPR), Long Beach, CA, 2019. [Published]

6) Wei Sun and Tianfu Wu. Learning Attentive Atrous Convolution in Generative Image Modeling. In the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2019. [Submitted]

7) Wei Sun and Tianfu Wu. Image Synthesis from Reconfigurable Layout and Style. In International Conference on Computer Vision (ICCV), Seoul, Korea, 2019. [Published]

8) Tianfu Wu and Xi Song. Towards Interpretable Object Detection by Unfolding Latent Structures. In International Conference on Computer Vision (ICCV), Seoul, Korea, 2019. [Published]

9) Sameera Lanka and Tianfu Wu. ARCHER: Aggressive Rewards to Counter bias in Hindsight Experience Replay. In 32nd Conference on Neural Information Processing Systems (NeurIPS 2018) Deep Reinforcement Learning Workshop, Montreal, Canada, 2018. [Published]

10) Bo Li, Caiming Xiong, Tianfu Wu, Yu Zhou, Lun Zhang and Rufeng Chu. Neural Abstract Style Transfer for Chinese Traditional Painting. In Asian Conference of Computer Vision (ACCV), 2018. [Published]

11) Wen Tang, Ashkan Panahi and Hamid Krim, Joint Concept Matching based Learning for Zero-Shot Recognition, IEEE Trans. On Pattern Analysis and Machine Intelligence. [Submitted]