Improving the Efficiency of Object Grasp Detection on Embedded Platforms Using the AOGNet Neural Network Architecture

Clive A. A. Simpson[1]* and Paul Gaynor[1]

[1] University of the West Indies, Mona, Jamaica W.I.

clivesimpson@myuwimona.edu.jm

Abstract. Robot grasp detection, commonly performed using Deep Neural Networks (DNNs), has proven to be a memory and power-intensive task that is required in resource-constrained environments. This paper proposes the use of And-Or-Grammar Networks (AOGNets) to reduce the constraints on embedded platforms. The experiments compare the accuracy, memory usage, space requirement, processing time, and power consumption of an AOGNet that is tuned to image recognition with implementations of Resnet, ResNeXt and Squeezenet on an Nvidia Jetson Nano. This paper also proposes using the AOGNet architecture for object grasp detection, as its performance on image classification tasks demonstrate that it is more tuned to the stringent operational requirements of embedded platforms.

Keywords: And-Or Grammar Networks, Computer Vision, Neural Networks, Object Grasp Detection

1 Introduction

1.1 Overview

Object grasp detection is an initial step in the grasping process where graspable regions on an object are identified on the input image [1]. Forming a grasp on an object using visual information involves the stages of object grasp detection, grasp planning, motion planning, and motion execution shown in Figure 1. An optimal grasp is selected in the grasp planning step from the set of predicted grasps created in the grasp detection steps. The motion planning and control steps will use the optimal grasp to plan and execute the motion of the robotic arm to form the grasp. Deep neural network models have been developed and trained on datasets to identify graspable regions, giving accurate results as summarized in Table 1 below. Computer vision tasks using deep neural network (DNN) models are resource intensive and prove inefficient on embedded platforms. Strategies used to improve the efficiency of DNN models such as employing hardware accelerators like Graphical Processing Units (GPUs) to improve inference speed [7], model compression to improve memory efficiency [8], and improving the architectural

C. Kiran Mai et al. (eds.), *Proceedings of the Fourth International Conference on Advances in Computer Engineering and Communication Systems (ICACECS 2023)*, Atlantis Highlights in Computer Sciences 18,
https://doi.org/10.2991/978-94-6463-314-6_8

design of DNN models to improve both inference speed and storage requirements [9][10]. Several advances in neural network architectures have been made, where architectural features have been introduced to improve the accuracy of the model. Architectural modifications also allow some DNN models to perform accurately using a small number of parameters for embedded platforms. Some of these models include mobileNet [11], ResNet-18, ResNet-34, ResNet-50 [12], ResNeXt-50 [13], ShuffleNet [14] and SqueezeNet [15].

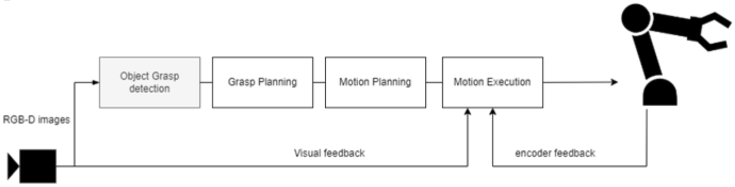

Fig. 1. General stages of the Robot Grasping Process

This paper proposes to adopt a neural network architecture, the And-Or Grammar Network (AOGNet) architecture, for the grasp detection step based on its performance on other computer vision tasks. The AOGNet model combines architectural features from top-performing models within the framework of an And-Or Graph [10]. The results presented for AOGNet from image classification and object detection showed comparable performance with existing architectures while using a smaller number of parameters. This paper evaluates the potential application of the AOGNet model as an efficient model for object grasp detection to be deployed on embedded platforms. The contribution of this paper is the corroboration of the performance results of the AOGNet model to initially reported results and the proposal of its application for grasp detection tasks in future work.

1.2 Paper Organization

This paper has five sections. An overview of the AOGNet architecture in section 2. Section 3 outlines the methodology used to evaluate the AOGNet for object detection on the CIFAR-10 dataset and proposes a methodology for evaluating the model for grasp detection on the Cornell Grasp dataset. Section 4 presents the results for object detection on the CIFAR-10 datasets and section 5 discusses the results presented in section 4.

Table 1. Sample object grasp detection datasets and reported accuracy results.

Dataset	Models	5 fold cross validation
Cornell Grasp Dataset [2]	Genrative Residual Convolutional Neural Network [5]	97.7
	Resnet-50 Multi-grasp predictor [6]	96
Jaquard Dataset [4]	Genrative Residual Convolutional Neural Network [5]	94.6

2 Background

2.1 AND-OR Image Grammar

Image Grammars are also effective tools for computer vision tasks due to their noise resistance [16]. The application of Image grammar to image processing seeks to find a scalable and recursive way of representing images so that they can be interpreted and used for reasoning tasks [17]. The AND-OR image grammar is the most common type of image grammar as they are useful in representing a wide variation of configurations in images and have been shown to be applicable in object detection [18], and human pose estimation [19]. AND-OR Grammars as a Grammar framework that allows for modeling of image structures and events being depicted within these images. AND-OR Graphs consist of three types of nodes [17]:

1. **Terminal nodes** – represent input primitives, which in the case of images would be pixel values from a sketch graph which was derived from the original image.
2. **AND nodes** – these represent compositional configurations.
3. **OR nodes** – these represent alternative configurations.

2.2 AOGNet Architecture

The AOGNet architecture consists of AND-OR graphs that are arranged sequentially in groups called AOG blocks (Figure 2). The AND-OR graphs in the AOG block are like the previously described AND-OR graphs in section 2.1 but incorporate additional features adopted from existing deep neural network (DNN) architectures that performed successfully on image classification tasks. These DNN architectures include ResNet, ResNeXt, DenseNets, Deep Pyramidal ResNets, and Deep Layer Aggregation Networks.

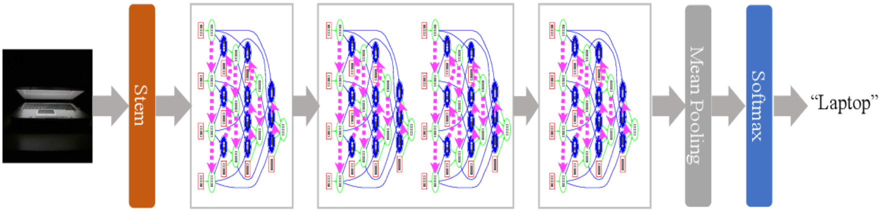

Fig. 2. AOGNet [10] showing AOG Block layers.

The OR nodes perform an aggregation function where inputs are combined to form a single value like what is done in the ResNet architecture. AND nodes perform concatenation of input values like what is performed in the DenseNet architecture. The AND-OR graph also uses lateral pathways to connect "AND" or "OR" nodes in the same level which allows information to flow within the same layer in addition to being fed forward into the next layer. Lateral connections were borrowed from the Deep Layer Aggregation architecture. Skip connections between layers of AND nodes are also incorporated allowing some information to flow from one layer of AND nodes to another and by passing the OR node layer between them. This feature was adopted from the ResNet architecture. The reduction in the feature map from the input (terminal nodes) to the output (root node) was borrowed from the Deep Pyramidal ResNet architecture. Figure 3 shows the structure of the AND-OR graphs within the AOG block, with the features incorporated from the DNN models.

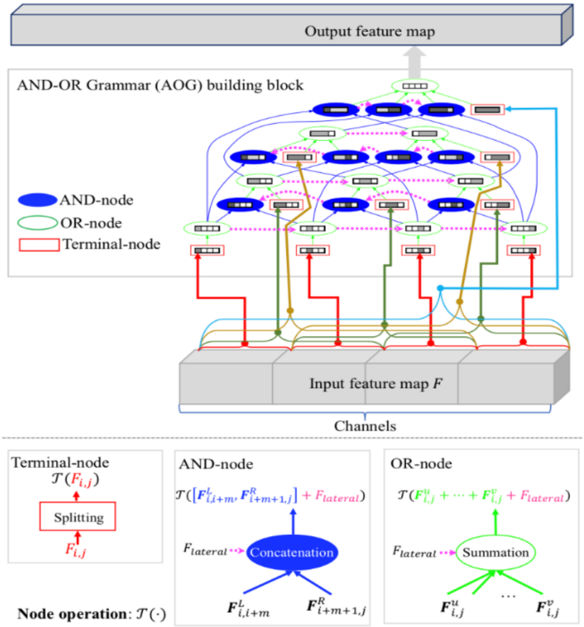

Fig. 3. AOG Block [10] showing AND, OR and Terminal Node Operations

3 Methodology

The AOGNet model was trained and evaluated on performing Image classification tasks on the CIFAR-10 dataset and the performance was compared with that of ResNet, ShuffleNet, and ResNeXt. This section outlines the training and evaluation of these models as well as proposes further steps towards modifying and adopting the AOGNet model for object grasp detection on the Cornel Grasp Dataset for future work.

3.1 Datasets

CIFAR-10 dataset. The CIFAR 10 Dataset is a small object classification dataset with 50,000 training images and 10,000 test images of 10 classes. Figure 4 shows 32 sample images from the CIFAR 10 training set.

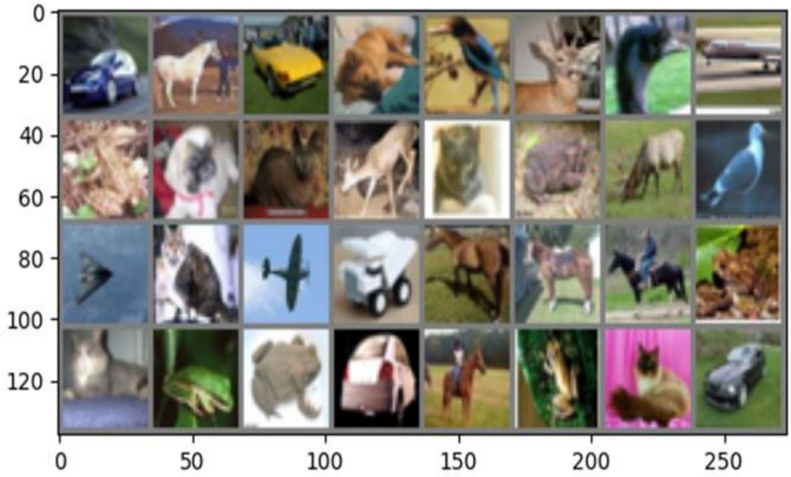

Fig. 4. Grid of 32 sample images from the CIFAR-10 dataset

Cornell Grasping Dataset. The Cornell Grasping Dataset consists of 1035 images of 280 objects in various spatial orientations, along with annotation files used to create grasp rectangles on each image. The dataset also consists of point cloud datafiles for each image. Figure 5 shows a sample image from the dataset with positive and negative grasp rectangle plotted on the object. The positive grasp annotations show best possible grasping areas where the negative annotation show areas to avoid forming grasps.

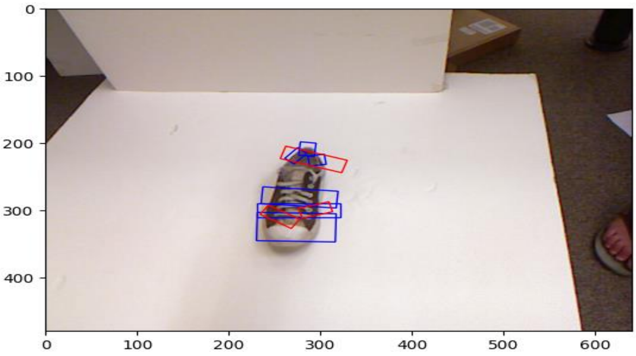

Fig. 5. Showing the sample image of a shoe from the Cornell grasp dataset

3.2 Training and Evaluation

Image Classification and Detection Tasks. The models will be trained on datasets that are currently used as the benchmark for image classification and object recognition which are the CIFAR-10, CIFAR-100, ImageNet-1K, and COCO datasets. The performance measures to be used for classification tasks will be the Top-1 and Top-5 accuracy. The performance measure for object recognition will be average precision, which is also a common measure used in the literature that allows for this model to be compared with those from the literature for object recognition tasks. For both image classification and object recognition tasks the training time and running time for models of different parameter sizes will also be used as a performance measure. The results for the CIFAR-10 data set are presented in this paper and results for the other image classification and object detection datasets will be presented in future work.

Object Grasp Detection. The AOGNET models will be adapted for object grasp detection. The resulting architectures will be trained on the Cornell Grasping Dataset and Google Grasp Dataset, and compared with the GraspNet [3] neural network architecture. The performance measures to be used for Grasp detection tasks are affordance, grasp accuracy, Model size, power consumption, and runtime. The Grasp detection AOGNET shall also undergo an ablation study, as well as parameter sensitivity analysis.

Modifying the AOGNet Architecture. The AOGNet will be created and trained initially using the source code provided by its original creators to create AOGNet Models of roughly 1 Million and 2 Million parameters. The AOGNet architecture will be created and evaluated to provide a baseline for the comparison of object detection and image classification tasks on the embedded platform with other mobile architectures. Other variants of the AOGNet will then be created to introduce the following architectural features:

1. **Dense lateral connections** – making every node at a given level of the AND-OR tree be connected to every other node at the same level, as shown in Figure 6.

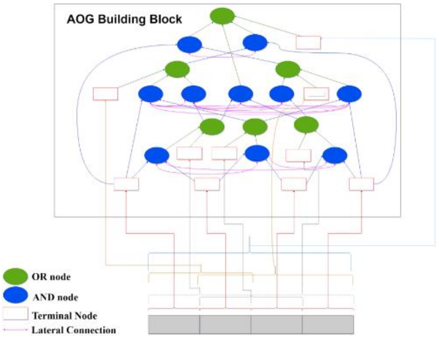

Fig. 6. Showing AOGNET building block with dense lateral connections

2. **Increased Branch splitting rule for k > 2** - making every node in a given layer branch into more than 2 nodes in the following layer.

3. **Top-down connections** - Use of hour-glass topology AOGNET.

4. **Introduction of other node operations** – use other architectural features such as fire module and channel shuffle that are used in SqueezeNet and shufflenet, to exploit their advantages to determine whether this modification would improve accuracy with a reduced number of parameters for the AOGNet architecture.

The Modified AOGNet Architecture shall also be trained for object grasp detection on the Cornell Grasping Dataset.

4 Preliminary Results

Two AOGNet models were generated; Aognet-1M consisting of 3 layers of 1 AOGNet Block in each layer, and AOGNet-2M consisting of two AOGNet Blocks in each of the three layers. These were trained on the CIFAR-10 Dataset along with Squeezenet, ResNet-18, ResNet-50, and ResNeXt-50. Table 2 summarizes the training hardware and parameters used for each model. Though AOGNet models have a smaller number of parameters it was observed that its training speed was slower than other models even on the faster NVIDIA A100 hardware (Figure 7).

Table 2. Training parameters and runtime for test models

Model	**GPU Hardware**	**Algorithm**	**Learning rate (lr)**	**Momentum**	**Epochs**	**Batch size**	**Training time (mins)**
Squeezenet	A100	SGD	0.001	0.9	2000	32	499.065
ResNet 18	Tesla T4	SGD	0.001	0.9	1000	32	328.785
ResNet 34	Tesla T4	SGD	0.001	0.9	1000	32	558.291
ResNet 50	Tesla T4	SGD	0.001	0.9	300	32	208.681
ResNeXt 50	Tesla T4	SGD	0.001	0.9	300	32	230.870
Aognet-1M	A100	SGD	0.001	0.9	300	32	139.706
Aognet-2M	A100	SGD	0.001	0.9	200	32	1342.955

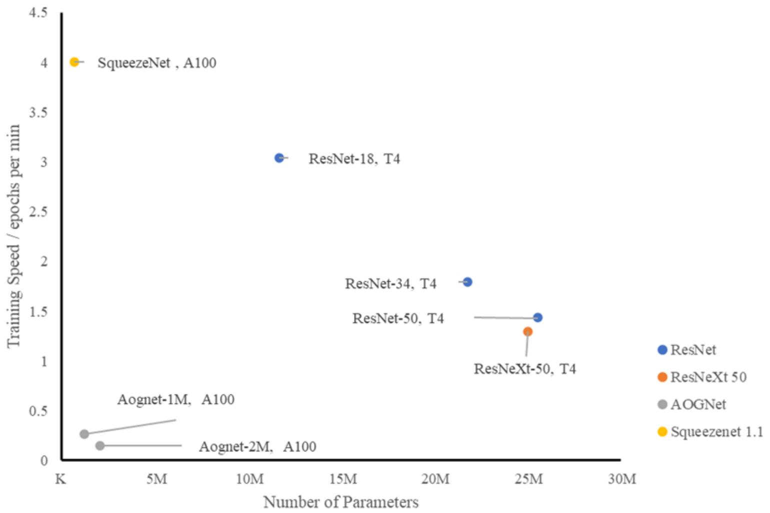

Fig. 7. Showing the comparison of training speeds for models

All models were evaluated on the NVIDA Jetson Nano. The AOGNet models demonstrated higher accuracy for classifying images with a smaller parameter size than other models as shown in table 3 and figure 8. This supports the reported findings of the improvement AOGNet makes in accuracy for image classification on the ImageNet-1K dataset.

Table 3. top-1 and top-5 accuracy, and runtime for neural network architectures being evaluated on the NVIDA Jetson Nano (CPU)

Method	Number of Parameters	Top-1 Accuracy (%)	Top-5 Accuracy (%)	Avg. Runtime (ms)
SqueezeNetV1.1	727,626	71.11	93.48	13.577
ResNet-18	11,689,512	74.36	97.66	26.026
ResNet-34	21,797,672	74.67	97.15	41.626
ResNet-50	25,557,032	74.93	97.68	48.684
ResNeXt-50	25,028,904	75.44	97.91	55.748
Aognet-1M	1,248,282	77.76	98.15	-
Aognet- 2M	2,063,023	74.99	97.87	-

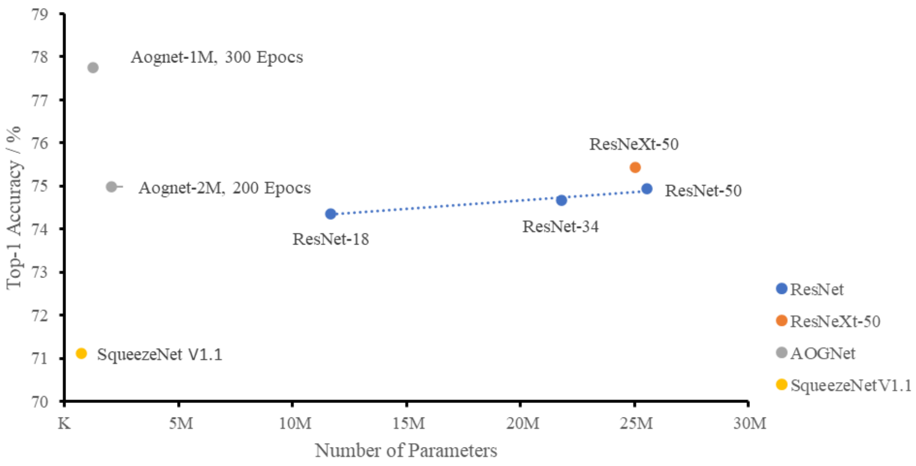

Fig. 8. showing the comparison of top-1 accuracies for test models on the CIFAR-10 dataset

5 Conclusion and Future work

This paper demonstrated the improved accuracy of AOGNets over other common mobile deep neural network models. The drawback however is the slower training rate of the AOGNet model. Future work shall involve modifying the AOGNet model by introducing dense lateral connections, squeeze Expand and channel shuffle architectural features and increase the branching at each layer from 2 to 3 to see its impact on model performance. This will be followed by adapting the AOGNet for object grasp detection on the Cornell Grasp Dataset.

Acknowledgement

This work is supported by the Universal Service Fund, Jamaica.

References

1. Chu, F.-J., Xu, R., & Vela, P. A. (2018). Real-world Multi-object, Multi-grasp Detection. IEEE Robotics and Automation Letters. doi:DOI 10.1109/LRA.2018.2852777
2. Li, Y., Lei, Q., Cheng, C., Zhang, G., Wang, W., & Xu, Z. (2019). A review: Machine Learning on Robotic Grasping. Proceedings of SPIE 11041, Eleventh International Conference on Machine Vision (ICMV 2018). 10442U. Munich, Germany: SPIE.
3. Fang, H.-S., Wang, C., Gou, M., & Lu, C. (2020). GraspNet-1Billion: A Large-Scale Benchmark for General Object Grasping. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 11444-11453). IEEE

4. Depierre, A., Dellandrea, E., & Emmanuel, L. (2018). Jacquard: A Large Scale Dataset for Robotic Grasp Detection. IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3511-3516). Madrid, Spain: Institute of Electrical and Electronics Engineers (IEEE).
5. Kumra, S., Joshi, S., & Sahin, F. (2020). Antipodal Robotic Grasping using Generative Residual Convolutional Neural Network. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 9626–9633). Las Vagas, USA: Institute of Electrical and Electronics Engineers (IEEE)
6. Chu, F. & Xu, R. & Vela, P. (2018). Real-World Multiobject, Multigrasp Detection. IEEE Robotics and Automation Letters. PP. 1-1. 10.1109/LRA.2018.2852777.
7. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020, March 23). A Comprehensive Survey on Graph Neural Networks. IEEE transactions on neural networks and learning systems.
8. Song, H., Mao, H., & Dally, W. J. (2016). Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. 4th International Conference on Learning Representations, ICLR 2016 (pp. 1-14). San Juan, Puerto Rico: ArXiV
9. Sun, Y., Xue, B., Zhang, M., Yen, G. G., & Lv, J. (2020, April 20). Automatically Designing CNN Architectures Using Genetic Algorithm for Image Classification. IEEE Transactions on Cybernetics, 50(9), 3840-3854.
10. Li, X., Song, X., & Wu, T. (2019). Aognets: Compositional grammatical architectures for deep learning. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 6220-6230). Long Beach, CA, USA: IEEE.
11. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017, April 17). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.* Retrieved 06 10, 2022, from ArXiv: https://arxiv.org/pdf/1704.04861.pdf
12. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770-778). Las Vegas, NV, USA: Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/CVPR.2016.90
13. Xie, S., Girshick, R., Dollar, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA: Institute of Electrical and Electronics Engineers (IEEE). doi:10.1109/CVPR.2017.634
14. Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *2018 IEEE/CVF Conference on*
15. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016, November 4). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. Retrieved June 10, 2022, from arxiv: https://arxiv.org/pdf/1602.07360v4.pdf
16. Chanda, G., & Dellaert, F. (2004). Gramatical Methods in Computer Vision: An Overview. Technical Report, Georgia Institute of Technology, College of Computing, Atlanta, GA.
17. Zhu, S.-C., & Mumford, D. (2007). A Stochastic Grammar of Images. Foundations and Trends in Computer Graphics and Vision, 2(4), 259-362.
18. Song, X., Wu, T., Jia, Y., & Zhu, S. (2013). Discriminatively Trained And-Or Tree Models for Object Detection. *2013 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3278-3285). Portland, OR, USA: IEEE Computer Society.
19. Girshick, R. B., Felzenszwalb, P. F., & McAllester, D. (2011). Object Detection with Grammar Models. *NIPS'11: Proceedings of the 24th International Conference on Neural Information Processing System* (pp. 442–450). Granada, Spain: Curran Associates Inc.