Efficient and Robust Training of Dense Object Nets for Multi-Object Robot Manipulation

David B. Adrian\textsuperscript{1,2}, Andras Gabor Kupcsik\textsuperscript{1}, Markus Spies\textsuperscript{1}, Heiko Neumann\textsuperscript{2}

Abstract—We propose a framework for robust and efficient training of Dense Object Nets (DON) \cite{adrian2022dense} with a focus on industrial multi-object robot manipulation scenarios. DON is a popular approach to obtain dense, view-invariant object descriptors, which can be used for a multitude of downstream tasks in robot manipulation, such as, pose estimation, state representation for control, etc. However, the original work \cite{adrian2022dense} focused training on singulated objects, with limited results on instance-specific, multi-object applications. Additionally, a complex data collection pipeline, including 3D reconstruction and mask annotation of each object, is required for training. In this paper, we further improve the efficacy of DON with a simplified data collection and training regime, that consistently yields higher precision and enables robust tracking of keypoints with less data requirements. In particular, we focus on training with multi-object data instead of singulated objects, combined with a well-chosen augmentation scheme. We additionally propose an alternative loss formulation to the original pixelwise formulation that offers better results and is less sensitive to hyperparameters. Finally, we demonstrate the robustness and accuracy of our proposed framework on a real-world robotic grasping task.

I. INTRODUCTION

A well-chosen object representation is an integral part of any robotic manipulation pipeline and is the basis for downstream tasks, such as grasping or assembly. With object CAD models and annotated training datasets, direct pose estimation methods are readily available \cite{leutenegger2015keypoint}. However, such annotations are expensive and not feasible for many industrial manipulation scenarios, where a fast and inexpensive teach-in process for new objects is of essence. Without annotated data, proposed object representations range from hand-designed features such as SIFT \cite{lowe2004distinctive,lowe1999object}, to implicit representations obtained through end-to-end learning of the task \cite{doersch2015unsupervised,adams2018learning}, or self-supervised approaches such as Dense Object Nets (DON) \cite{adrian2022dense} or grasp2vec \cite{dornhege2020grasp2vec}.

DON \cite{adrian2022dense} is a recently proposed self-supervised contrastive learning framework for view-invariant object representations. The DON model provides a dense local descriptor representation by projecting a query RGB image $I \in \mathbb{R}^{H \times W \times 3}$ to a dense descriptor image $I_D \in \mathbb{R}^{H \times W \times D}$, with $D \in \mathbb{N}^+$, the dimension of the descriptor space. DON exploit the geometric prior of a sequence of registered RGBD frames to sample pixel correspondences between alternative views of the same object. The correspondence information is then used with contrastive learning to train a model that uniquely embeds the visual appearance of objects in descriptor space. Offering intra-class generalization, consistent descriptors for deformable objects, and more, DON have been adapted to several use-cases, such as rope manipulation \cite{dornhege2020grasp2vec}, block stacking \cite{florence2021dense}, control \cite{park2022learning}, and 6D grasp pose estimation \cite{park2023learning}.

The original formulation of DON mostly focuses on singulated objects, both during training, as well as in its demonstrated use-cases. However, in industrial applications, e.g., grasping specific objects in a heap, often multiple objects in a potentially cluttered environment need to be detected and manipulated. Here objects are commonly coming from a fixed and known set. For these applications, we require distinct and easily trackable descriptors that robustly differentiate between every object of the manipulation task, instead of intra-class generalization. Also reported by Florence et al. \cite{florence2021dense}, the original DON training approach is prone to fail in such multi-object scenarios, even with their proposed cross-object loss. While \cite{adrian2022dense} offers an early discussion on instance-specific training for multi-object scenarios, no quantitative results are provided. Furthermore, the self-supervised training approach of DON requires the generation of object masks, which is cumbersome to automate and error prone in practice. This also prevents training on data with naturally occurring occlusion or the inclusion of unknown, distracting objects in the scene, which are important to create realistic training data.

In this work, we improve the training pipeline of DON...
in different aspects. First, we present a simplified data collection process for training by recording scenes of multiple objects as opposed to only one object per scene. This allows us to record more views per object, including occlusions. Obtaining as many diverse perspectives per object is important to obtain view- and scale-invariant features, thus increasing our data efficiency by a large factor. As scenes can now be densely packed, we no longer require object masks to ensure a good ratio between object and background correspondences. This greatly reduces the technical challenges and makes the training pipeline easier to fully automate.

Second, to further improve data-efficiency, we propose a data augmentation scheme as more commonly seen in state-of-the-art visual contrastive learning [12], [13], [14], [15], that helps to prevent overfitting while training, but also increases accuracy and provides robustness in practice.

Finally, as loss function, we present an adaptation of InfoNCE [16], also popularized as normalized temperature-scaled cross entropy (NT-Xent) in the visual domain by SimCLR [12]. As opposed to the pixelwise contrastive loss of [1], we found it to be less sensitive to its hyperparameters, while offering better performance at the same time.

II. RELATED WORK

In case of known CAD models and available annotated training datasets, pose estimation for manipulation from RGB-D images have recently shown great results [2]. In our use-case, we focus on scenarios where such data is not available and review related work in that respect.

Here, correspondence estimation is at the core of many computer vision problems such as 3D reconstruction, simultaneous localization and mapping (SLAM) and structure-from-motion (SfM), or object recognition and localization. This is often achieved by (sparse) local visual feature detection and description, with methods ranging from hand-crafted, generic features such as SIFT [3] or ORB [17], to learned feature-pipelines like LIFT [18], L2-Net [19], Universal Correspondence Networks (UCN) [20], SuperPoint [21], or KP3D [22]. We also find hybrid approaches, e.g., in the domain of object pose estimation Pitteri et al. [23] present a hand-crafted transition- and orientation-invariant surface feature embedding that describes local geometries combined with a network trained to predict these features from RGB images.

In this work, we focus on contrastive learning of object-centric dense local visual descriptors. Extending the dense pixelwise correspondence estimation framework in [24] and UCN, Florence et al. [1] proposed DON, a fully automatable approach with a focus on object manipulation. It was applied to various tasks, including block stacking [9], 6D grasp pose estimation [11], control [10], [25], or rope manipulation [8].

Global visual descriptor learning is a closely related topic. Here, typically a Siamese network is used to process augmentations of the same image and find underlying features that are invariant to these mappings. SimCLR [12], SwAV [15], or MoCo [26] and MoCo-v2 [27] all utilize some form of contrastive(-like) learning. In contrast, BYOL [13] only uses positive pairs and Barlow Twins [14] optimizes the empirical cross-correlation. Similar ideas were recently also adapted for general local descriptor learning, e.g., in DenseCL [28] or combined global-local descriptor learning [29].

Over the last years many impressive object manipulation solutions emerged in the robot learning community, typically relying on inexpensive RGB-D cameras. For example, DexNet-based algorithms were proposed for grasping using depth images [30]. Zeng et al. use vision-guided learning to grasp unknown objects for a throwing task [31]. Pinto and Gupta investigate large-scale self-supervision for grasp learning [32], while Kalashnikov et al. [33] employ reinforcement learning at large-scale to grasp from a heap. One of the closest works to DON for object manipulation is proposed by Vecerik et al. [34]. They show how task-relevant object keypoints can be learned by human- and self-supervision, e.g., to learn cable plugging.

III. METHOD

Similar to [1], we record static scenes including objects we wish to learn using a robot and a wrist-mounted RGBD camera. Every recorded scene consists of thousands of registered RGBD images. As in the original DON approach, we randomly sample image pairs \((I^A, I^B)\) from unique scenes for training. The sampled image pairs present two distinct, but partially overlapping views of the scene.

In our work, we record scenes that contain multiple objects we wish to learn (Fig. 2b), as opposed to the single object scenes (Fig. 2a) in [1]. We also skip the object mask generation step, which is, in practice, error-prone and difficult to automate. More details about the data collection are presented in Sec. IV-B. In what follows we describe our training approach relying on image pairs sampled from multi-object scenes. Specifically, we focus on image augmentation, pixel correspondence sampling, and the loss function. We provide an overview of the general data flow in Fig. 3.

A. Augmentations

Augmentations have been shown in self-supervised and contrastive learning to not only be beneficial, but integral to learning discriminative global feature representations, cf. [12], [15], [14]. These methods typically start with a single
image and create two corresponding views using two randomly sampled sequences of augmentations. In our work, the two different views are readily sampled from the dataset, however, image augmentations can still diversify our training dataset to improve data-efficiency and mitigate overfitting. The idea of augmentations for DON was also used in [1] with background randomization and vertical/horizontal flipping. For this study, we focus on a broader set of common augmentations: resize and crop, perspective and affine distortions, horizontal and vertical flips, rotations, blur, color jitter, and grayscale conversion.

In particular, transformations such as perspective distortions, naturally appear in robotic manipulation scenarios. Similarly, blurring and color distortions frequently occur in practice due to changing light conditions or motion blur. Thus, utilizing this set of augmentations should not only help to prevent over-fitting due to limited training dataset size, but also provides additional cases for improving robustness.

For both images we randomly sample a subset of augmentations and apply them (Fig. 3). In practice, however, we found that applying the transformations to only one of the images to perform better. As DON relies heavily of color cues, using color affecting augmentations on both images impacted training heavily. An ablation of the individual augmentations and further discussion is summarized in Sec. IV.

We utilize the readily available implementations by the Torchvision library of the Pytorch project [35].

B. Correspondence Sampling

Correspondence sampling for an image pair is straightforward given the respective camera poses, intrinsic matrix, and depth information using knowledge from multiple view geometry [36]. However, as we rely on densely packed scenes and image pairs with potentially large perspective changes, we face occlusion and partially overlapping fields-of-view. Hence, instead of sampling individual pixels directly and checking for their validity afterwards, we propose the following direct approach: map every pixel of \( I^A \) via their world-coordinate representation to the other view, then determine which pixels are actually inside the other view and not occluded. This yields a Boolean mask \( M \), determining which pixels in \( I^A \) have a true correspondence in \( I^B \). We can now randomly sample valid pixel correspondences according to \( M \) given the previously computed mapping. We refer to these as matches, or positive pairs, in the following.

C. Loss

Although the pixelwise formulation in [1] shows impressive results, we found it to be sensitive to its hyper-parameters. To this end, we propose an adaptation of InfoNCE [16], also called NT-Xent in recent work such as SimCLR [12], for the purpose of training DON.

Instead of the original pixelwise formulation in [1], and its cross-object variant, we require only a single loss function, that works for both singulated and multi-object settings. Here, we randomly sample \( N \) correspondences from an image pair \((I^A, I^B)\), with each correspondence yielding two descriptors \( d^A_i \) and \( d^B_j \), for a total of \( N \) pairs, or \( 2N \) descriptors. For a given descriptor, we treat all other \( 2(N - 1) \) descriptors respectively as negative examples. Hence, unlike the vanilla pixelwise contrastive formulation in [1], we directly optimize all descriptors with respect to each other.

For one descriptor pair \((d_i, d_j)\), we define the loss as

\[
l_{i,j} = -\log \frac{\exp(D(d_i, d_j) / \tau)}{\sum_{k=1; k \neq i}^{2N} \exp(D(d_i, d_k) / \tau)},
\]

where \( \tau \) is a temperature scaling factor and \( D(\cdot, \cdot) \) denotes a distance or a similarity measure. The complete loss is given by the sum over all pairwise losses \( l_{i,j} \). This loss is comparable to the one in [12], where each image pair in a mini-batch is considered as negatives. In our work instead, we compute the loss directly on pixel-level descriptors. Finally, we coalesce all correspondences across the image pairs in the batch into a single mini-batch dimension. In the following, we use the cosine similarity, defined as

\[
D(d_i, d_j) = \frac{(d_i \cdot d_j)}{\|d_i\|_2 \|d_j\|_2},
\]

that is the dot product between vectors that have been normalized to unit length.

IV. EVALUATION

In this section we compare our proposed training pipeline to that of vanilla DON. We first describe how data collection and training are performed. Then, we explain our metrics for two different evaluation scenarios: i) we show how accurately the different models predict correspondences in image plane, and ii) for downstream tasks we measure keypoint tracking accuracy in world coordinates. Finally, we perform ablation studies and show results on a real world robot grasp experiment.

A. Hardware Setup

We used a Franka Emika Panda 7-DoF arm, equipped with a wrist-mounted Intel RealSense D435 camera for recording of RGB-D images, see Fig. 1. All scenes were recorded using the same robot end-effector trajectory, at 640 × 480px and 30Hz. We sub-sampled recordings by enforcing at least 5° rotation and 2cm translation between consecutive frames.
yielding around 450 frames per scene. A parallel gripper was employed for the grasping tasks.

B. Data Collection

We selected a set of 8 objects for training, including objects with repeating patterns, similar visual features, and one deformable object. We collected separate sets of training data for the multi-object and single-object setting, see Fig. 2. For single-object training with vanilla and cross-object DON [1] we recorded multiple scenes for each object isolated in the workspace, for a total of 20 scenes. For multi-object data, we captured 15 scenes in total, each containing a subset of the known objects. For validation we recorded one multi-object and 5 single-object scenes, each with some unknown objects in the vicinity. A total of 8 multi-object scenes were recorded for the test set. In each test scene, a subset of the 8 objects were presented in different configurations, with some additional unseen objects that serve as distractions. We manually provided object-specific masks for each of the known objects, allowing us to sample query points specifically for each objects for validation and testing.

C. Training

Following [1], and for comparison with existing literature, we utilized the same ImageNet-pretrained ResNet-34 with output stride of 8, that is upsampled to the original input dimension; cf. the public reference implementation of [1]. We note that, unlike various state-of-the-art models [13], [12], [15], we do not employ a projection layer, or other advanced mechanisms like gradient stopping, after extracting the representation, but directly apply the loss as in [1].

We trained using ADAM [37] optimizer, with a learning rate of $3e-5$, momentum of 0.9, and weight decay of $1e-4$. All models were trained for 500 epochs, with each epoch consisting of 1000 image pairs. Our approach using NT-Xent loss was trained with a batch size of 2, while the pixelwise contrastive losses using a batch size of 1 (as in [1]), as we found this to perform best, respectively, see Sec. IV-E.2. We found the closest matching descriptor for each query point in some other view $l$ via its descriptor image $I^k_D$ as prediction $p^k_i = (u^*_i, v^*_i) = \arg \min_{u_i, v_i} D(I^l_D(u_i, v_i), d^k_i), \forall i \in [1, N]$. Here, $D$ denotes the cosine similarity for NT-Xent variants, or the $\ell_2$ distance for pixelwise losses.

D. Evaluation Metrics

To measure performance we adopted the PCK@$k$ metric, as in preceding works [1], [9], defined for a set of pixel pairs $\mathcal{T} = \{p_i, q_i\}_{i=1}^N$ as

$$\text{PCK}@k(\mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{(p_i, q_i) \in \mathcal{T}} \left[ \frac{\|p_i - q_i\|_2 \leq k}{k} \right],$$ (3)

where $k$ is the maximum allowed error, measured in pixels, for a prediction to still be counted as correct. $\mathcal{T}$ is a set of predicted $p_i$ and ground-truth $q_i$ pixels, and $\lfloor \cdot \rfloor$ denotes the Iverson bracket, which evaluates to 1 if the logical proposition inside is satisfied and 0 otherwise.

We evaluated the function many times over a range, that is $\forall k \in [1, 100]$, and report the area under the curve (AUC). We do not consider $k > 100$ as such a mismatch in image plane, with a diagonal of 800 pixels, leads to failure in downstream tasks. Additionally, in Sec. IV-E.2, we evaluated the model accuracy via a keypoint tracking task, measuring world-coordinate prediction errors in millimeters, which is relevant for downstream tasks.

In both cases, we selected $N$ query pixel locations $\{q^k_i\}_{i=1}^N$ on an object in a given image $I^k$. Subsequently, the descriptor representation at these pixel locations $\{f_0(I^k)_{q^k_i}\}_{i=1}^N = \{d^k_i\}_{i=1}^N$ are extracted. We found the closest matching descriptor for each query point in some other view $l$ via its descriptor image $I^l_D$ as prediction $p^k_i = (u^*_i, v^*_i) = \arg \min_{u_i, v_i} D(I^l_D(u_i, v_i), d^k_i), \forall i \in [1, N]$. Here, $D$ denotes the cosine similarity for NT-Xent variants, or the $\ell_2$ distance for pixelwise losses.

E. Results

1) Correspondence Accuracy: As first evaluation, we selected 500 image pairs per object type across all scenes, utilizing the object-specific masks from our multi-object test data. We randomly sampled given the constraints of a minimum translation of 20cm and minimum angle of $\pi/12$ between poses, ensuring enough variation in perspective to test for view-invariance. Per image pair 100 query points are randomly selected on the object in one of the images. Given these query descriptors, we found their respective closest matching descriptor in the second image as described above and calculated the PCK@$k$ errors. The sampled image pairs and correspondences were evaluated for each model configuration.

Table I summarizes the performances of each loss variation. Clearly, multi-object trained models outperform both vanilla and cross-object loss trained DON. For vanilla pixelwise, and its cross-object variant, augmentations are detrimental. We partially attribute this to augmentations, such as perspective distortion, that reduced the already small area even more. Consequently, masked sampling starts to become inadequate for training efficiently, requiring more epochs.
2) Keypoint Tracking: Using dense descriptor representations for downstream tasks requires detecting keypoints relevant for the task. The keypoints can be defined manually, or by a selection algorithm. Thus, in the following we evaluated how accurately we can track keypoints in world coordinates using the different models.

We treated our recorded scenes as video-sequences with length $M$. We manually annotated query points $\{q_i\}_{i=1}^N$ on a single frame, which serve as reference keypoints to track. Selecting between 2 – 5 query points per object and sequence, we computed predictions for each query point in the remaining $M-1$ images. After detecting the closest pixels to the queries in every frame, as described in Sec. IV, we projected the corresponding depth value to world-coordinates. Note that we have access to the camera intrinsic and extrinsic parameters, therefore this computation is consistent within a sequence. Finally, the error is calculated as the Euclidean distance between the respective world-coordinate representations $\|\tilde{p}_i^k - \tilde{q}_k\|_2$, $\tilde{p}, \tilde{q} \in \mathbb{R}^3$. The prediction error histograms averaged over all queries are shown in Fig. 4.

### TABLE I: Correspondence accuracy over randomly sampled image pairs per loss and training configuration.

| Model Representation | AUC±σ for PCK@k, \(\forall k \in [0, 100]\) |
|----------------------|-----------------------------------------------|
| Pixelwise            | 0.313 ± 0.011 |
| Pixelwise + Augmentations | 0.279 ± 0.019 |
| Pixelwise + Cross-Object | 0.387 ± 0.017 |
| Pixelwise + Cross-Object + Augmentations | 0.290 ± 0.036 |
| Pixelwise + Multi-Object + Augmentations | 0.565 ± 0.008 |
| NT-Xent + Multi-Object | 0.514 ± 0.027 |
| NT-Xent + Multi-Object + Augmentations | 0.621 ± 0.015 |

1) Data Efficiency: We evaluated the impact of available data with respect to the overall performance in comparison to training on singulated objects. We found a much higher data efficiency, see Fig. 5 as with just 3 multi-object scenes and augmentations, we obtained the same performance as the pixelwise cross-object loss with 20 scenes. Adding further scenes increases performance, with augmentations adding a larger benefit when not much data is available.

2) Augmentations: In order to evaluate which augmentations are beneficial, and how much they respectively contribute to the overall performance, we trained multiple configurations of the NT-Xent multi-object model starting with no augmentations and each time incrementally adding new augmentations, see Fig. 6.

All models exhibit clear long tail distributions. However, our training pipeline with NT-Xent and augmented multi-object scenes led to more accurate and robust tracking performance.

### F. Ablation

All following results are reported with respect to the correspondence accuracy task in Sec. IV-E. Models are with default parameters, with one aspect varied as described. Each configuration is trained 5 times, and averaged results are reported.
4) **Temperature**: We found NT-Xent stable with values in the range $0.01 - 0.3$ and optimal value around 0.1. The performance dropped for larger values around 0.7 – 1.0. This is in line with results reported for SimCLR [12].

5) **Batch Size**: NT-Xent loss is mostly invariant to the batch size, see Table III, in contrast, the pixelwise formulation exhibits strong decrease in performance for batch-sizes larger than 1. We hypothesize, that if each scene can only contain a small subset of objects, training with higher batch sizes could optimize more objects representations simultaneously, but this requires further investigation.

6) **Number of correspondences**: Both losses appeared robust to the number of correspondences, see Table IV but pixelwise loss performance deteriorated below 20,000 correspondences. We note that the large number of correspondences for the pixelwise loss are only possible because we do not sample exclusively within masks, which would typically only occupy around 4,000-10,000 pixels.

### G. Robotic Experiment: Grasp prediction

Instead of the single point grasping shown in [1], we evaluated the models on a 6D grasp pose prediction task, similar to [11]. To this end, we took a single RGB image of each object, we manually selected two pixels to denote an axis, and extracted the respective descriptors $d_1$, $d_2$. In the test scenes, we find the respective predictions for each descriptor and calculate their world-coordinate representations $\tilde{p}_1$, $\tilde{p}_2$.

### V. CONCLUSION

We presented a training framework for Dense Object Nets with a particular focus on multi-object tasks. Our proposal reduced the required data amount and drastically simplified the data pre-processing. We removed the need for mask generation, while improving performance for multi-object scenarios. We investigated which image augmentations are required to improve robustness and demonstrated that our combined proposal leads to significant improvements, both on benchmark tests, as well as for oriented grasping requiring high precision. Despite the improvements, we found that all trained DONs suffer from becoming unstable in very densely packed scenes. We believe this to be due to network backbone used, that uses a simple bilinear upsampling to produce the dense descriptor image. Hence, further research should investigate and focus on improving the stability of descriptors.
[30] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-Net 2.0: Deep learning to plan Robust grasps with synthetic point clouds and analytic grasp metrics,” *Robotics: Science and Systems*, vol. 13, 2017.

[31] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, “Tossingbot: Learning to throw throwable objects with residual physics,” 2019.

[32] L. Pinto and A. Gupta, “Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours,” in *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, pp. 3406–3413. [Online]. Available: https://doi.org/10.1109/ICRA.2016.7487517

[33] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, “Scalable deep reinforcement learning for vision-based robotic manipulation,” in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., vol. 87. PMLR, 29–31 Oct 2018, pp. 651–673. [Online]. Available: https://proceedings.mlr.press/v87/kalashnikov18a.html

[34] M. Vecerik, J.-B. Regli, O. Sushkov, D. Barker, R. Peceviciute, T. Rothörl, C. Schuster, R. Hadsell, L. Agapito, and J. Scholz, “S3K: Self-Supervised Semantic Keypoints for Robotic Manipulation via Multi-View Consistency,” no. CoRL, 2020. [Online]. Available: http://arxiv.org/abs/2009.14711

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 8024–8035.

[36] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press.

[37] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds.